CALIFORNIA CENTER FOR INNOVATIVE TRANSPORTATION
INSTITUTE OF TRANSPORTATION STUDIES
UNIVERSITY OF CALIFORNIA, BERKELEY

Mobile Millennium Final Report

Alexandre M. Bayen, Ph.D., Principal Investigator
Joe Butler, Project Manager
Anthony D. Patire, Ph.D., Postdoctoral Researcher

CCIT Research Report
UCB-ITS-CWP-2011-6

ISSN 1557-2269

The California Center for Innovative Transportation works with researchers, practitioners, and industry to implement transportation research and innovation, including products and services that improve the efficiency, safety, and security of the transportation system.

CALIFORNIA CENTER FOR INNOVATIVE TRANSPORTATION
INSTITUTE OF TRANSPORTATION STUDIES
UNIVERSITY OF CALIFORNIA, BERKELEY

Mobile Millennium Final Report

Alexandre M. Bayen, Ph.D., Principal Investigator
Joe Butler, Project Manager
Anthony D. Patire, Ph.D., Postdoctoral Researcher

CCIT Research Report
UCB-ITS-CWP-2011-6

This work was performed by the California Center for Innovative Transportation, a research group at the University of California, Berkeley, in cooperation with the State of California Business, Transportation, and Housing Agency's Department of Transportation, and the United States Department of Transportation's Federal Highway Administration.

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the State of California. This report does not constitute a standard, specification, or regulation.

September 2011

Acknowledgements

The Mobile Millennium team would like to acknowledge the financial contribution of the California Department of Transportation and the US Department of Transportation to this work. Without the strong involvement of these partners, it would never have been possible to build a successful research organization in such a short amount of time, and to achieve the goals of Mobile Millennium so quickly. We would also like to acknowledge the contributions of our industry partners Nokia/NAVTEQ, who have supported this work both financially and with generous in-kind support, leading to the use of thousands of Nokia phones over the duration of the project, and corresponding data plans. Without the strong support of such an industry partnership, the project would never have received as much attention and become a flagship of ITS research at UC Berkeley. The authors want to thank the National Science Foundation for providing funding to support fundamental research, without which this project could not have achieved the level of success and depth it has. This work was supported by grants #0615299 and #0845076. The funds from the National Science Foundation have greatly helped UC Berkeley to ally practitioner oriented goals with scientific research goals in a way which makes this partnership unique, and are aligned with the mission of public service, education and research of the University. Finally, numerous other organizations have contributed to the start, or the development of Mobile Millennium, and are mentioned here alphabetically: BAE Systems, the Center for Information Technology and Research in the Interest of Society (CITRIS), the Defense Advanced Research Projects Agency (DARPA), the Department of Civil and Environmental Engineering at UC Berkeley, the Ministère des Transports, France, (now MEDDAAT), the University of California Transportation Center (UCTC), Tekes (the Finnish Funding Agency for Technology and Innovation), Telenav Inc., VIMADES, the Volvo Foundation and the VTT Labs. We are grateful to all of these funders for their interest in our project and the support they have provided us in these years.

Numerous individuals in their respective organizations have contributed to the success of Mobile

Millennium. We are grateful for their support, and their trust in our capabilities. We acknowledge them below by organization.

Contributors and partners, institutional support

California Department of Transportation

Mobile Millennium has benefited from the very strong support of the California Department of Transportation, which has provided a solid anchor for us in California. We are extremely grateful to Director Randell Iwasaki and to Larry Orcutt, Director of the Divison of Research and Innovation, who helped us set up the project at the Federal level, and who provided local support from California. We are extremely grateful for their faith in our work and our ability to deliver. In addition, we further acknowledge Larry Orcutt for his confidence to provide the initial Caltrans funding to this project. We are thankful for the ongoing guidance of Greg Larson who helped with the ongoing decisions and issues to be made as part of the project. We are also thankful for the help of Hassan Aboukhadijeh, Asfand Siddiqui, and Gurprit Hansra.

US Department of Transportation

We want to express our gratitude to RITA Administrator Paul Brubaker, for his vision for Safe Trip 21, and for putting his trust in our project. Safe Trip 21 made it possible for the project to acquire a scale of national importance. We want to express our sincere gratitude to Gary Ritter, for his tenacity and energy at making the project happen, and for his support throughout the project, in particular for the public events organized as part of the project.

CCIT team

Mobile Millennium found its roots in Mobile Century, and the two projects shared numerous participants. We are extremely grateful to CCIT Director Tom West for his undefeated efforts to bring the project to the spotlight, and for his help with setting up the project with both the US and California DOTs. Without Tom West's help, this project would not have been able to take off at the scale it did. The Mobile Millennium would never have been constructed without the involvement of Joe Butler, who brought years of experience from Industry, necessary to build a system such as Mobile Millennium. Most of the system used during the Mobile Millennium and available today was developed under Joe Butler's leadership. We wish to thank J.D. Margulici, the Associate Director of CCIT, who had the initial vision of mobile probes using phones, and who contributed to the success of Mobile Millennium by launching Mobile Century and setting up the institutional basis for Mobile Millennium to succeed.

The staff of CCIT contributed to many of the aspects involved in leading a project such as Mobile Millennium. This involves logistical planning, financing, tasking, managing and successful implementation. In particular, we thank Steve Andrews, Coralie Claudel, Marika Benko, Osama Elhamshary, Tia Dodson, Chris Flens-Batina, Manju Kumar, Jed Arnold, Benson Chiou, Lori Luddington, Xiaohong Pan, Erica Sherlock-Thomas, and Arthur Wiedmer.

Nokia / NAVTEQ team

The success of Mobile Millennium is due to the work, tenacity and visionary experience of Quinn Jacobson, Ken Tracton and their team: Toch Iwuchukwu, Baik Hoh, Cynthia Kuo, Carl Snellmann. Working with such an incredible team of engineers has been not only a privilege but a true opportunity for CCIT to learn the world of the mobile internet. We are extremely grateful for the institutional support given by Nokia's leadership throughout the project, in particular from Bob Ianucci, Henry Tirri and John Paul Shen. We want to express our deepest gratitude to Lisa Waits for her ongoing support and energy to keep our project on track. We are also grateful for numerous other Nokia staff members who have helped us succeed, in particular Karen Lachtanski, Dave Sutter, and John Loughney.

We want to express our gratitude to NAVTEQ, in particular to the engineering team who helped us set up our system, get maps and understand the data we were sharing: Matt Lindsay, Brian Smyth, Drew Bittenbender, Max Peysakhov and Candace Sleeman. We are grateful to Aaron Crane for helping us to institutionalize the collaboration between NAVTEQ and UC Berkeley in the post launch phase of Mobile Millennium. We want to thank NAVTEQ for their institutional support for our interactions with the DOTs, in particular Harry Voccola, Jeremy Wolstan, Howard Hayes and John McLeod.

UC Berkeley

Our success was made possible by an extremely strong support from the College of Engineering, and the personal intervention of Dean Shankar Sastry when we needed his help for fundraising and showing the commitment of the College to the success of the project. With the help of CITRIS, the Center for Information Technology Research in the Interest of Society, and the personal involvement of Director Paul Wright, our project reached unprecedented visibility at the level of the UC Berkeley campus, and the UC system. Few projects have benefited from such support of the College and CITRIS, we are grateful for this trust in our capabilities. We are also grateful for the support of the Civil and Environmental Engineering Department, and thank two successive Chairs, Professor Greg Fenves and Professor Lisa Alvarez Cohen, as well as the support of ITS, its Director Professor Samer Madanat, its Acting Director Professor Mike Cassidy and its Assistant Director, Steven Campbell. The staff from these units have provided invaluable support, which we are grateful for: Gary Baldwin, Lorie Mariano, Yvette Subramananian, Anthony Saint-George, Masoud Nikravesh, Jean Paul Jacob, Aaron Walburg, Khossrov Taherian, Margaret Chang, Norine Shima, Jillene Bohr, John Li, Derek Johnson, Ann Guy, Barbara Blackford, Sylvia Bierhuis and Bill Oman. Finally, the ongoing support of the media office enabled us to gain significant visibility outside of campus, and we want to express our gratitude to Sarah Yang and her team for her close work with Ann Guy. Together with the PR office, Ann Guy has brought very valuable help to the project by raising its visibility and exposure.

Academic partners

Numerous academic partners have contributed to the success of Mobile Millennium, and we are

grateful for their help. In particular, we want to acknowledge the unlimited scientific generosity and the vision of Professor Jean-Pierre Aubin, and the help of Professor Halina Frankowska and Professor Patrick Saint-Pierre. Their vision has shaped our vision. We are also grateful to numerous colleagues who have helped us when facing mathematical, scientific or technical difficulties so we could bring our contributions to the next level, in particular Professor Benedetto Piccoli (Rutgers University), Professors Michael Franklin, Carlos Daganzo, Laurent El Ghaoui (UC Berkeley) and Dr. Xavier Litrico (CEMAGREF). We are particularly grateful to Deborah Estrin for her vision of participatory sensing, her support, and her pioneering work that inspired us. We are also grateful to Marco Gruteser from Rutgers University for the collaborative work on privacy.

The Team

The authors would like to apologize in advance to anyone who contributed to develop, build, and deploy the traffic monitoring system implemented as part of Mobile Millennium but whose name we neglected to mention through our own oversight. You have our gratitude.

During the compilation of this report, we thank the many people who provided crucial subject matter through personal interviews, Skype interviews, original project documents, briefing materials, personal notes, emails, and other sources that made it possible to provide the rich narrative herein. We also thank John Nguyen for his painstaking work to edit the bibliography and many of the source materials. Finally, we thank Ethan Xuan, Elie Daou, and Boris Prodhomme for their work to help edit and format this document.

Management

Alex Bayen (Principal Investigator), Tom West (Partnerships & Outreach), Joe Butler (Project Manager), Steve Andrews (Project Coordinator), Coralie Claudel, Rich Kleinman (Project Administrator and Finances)

Team Members

Post doctoral researchers

Jeff Ban, Ryan Herring, Anthony Patire, Olli-Pekka Tossavainen.

Ph.D. students

Saurabh Amin, Sebastien Blandin, Christian Claudel, Juan Carlos Herrera, Ryan Herring, Aude Hofleitner, Tim Hunter, Samitha Samaranayake, Issam Strub, Daniel Work.

M.S. students

Paul Borokhov, Pierre Emmanuel Mazare, Matthieu Nahoum, Samy Merzgui, Julie Percelay, Arthur Wiedmer.

CCIT Staff, Alumni, and Visitors

Daniel Edwards (Associate Development Engineer), Sanessh Apte (Associate Development Engineer), Jonathan Felder (Systems Administrator), John Nguyen (Editorial Assistant), Alfred Tran (Undergraduate Research Apprentice), Andre Lockhart (Project Manager), Besen Chiou (Senior Development Engineer), Bill Vogel (Software Engineer), Douglas Putnam (Undergraduate Research Apprentice), Elena Agapie (Visiting Student), Elliot Schatmeier (Undergraduate Research Apprentice), Ernan Anguiano (Undergraduate Research Apprentice), Kayvan Nowrouzi (Masters Student), Joseph Curtis (Undergraduate Research Apprentice), Kevin Luang (Undergraduate Research Apprentice), Maha Haji (Millennium Webmaster), Marcella Gomez (Undergraduate Research Apprentice), Michelle Papilla (Undergraduate Research Apprentice), Miguel De Gracia (Undergraduate Research Apprentice), Morgan Smith (Software Engineer), Nina Harvey (Graduate Student Researcher), Omar El Bizri (Visiting Undergraduate Research Apprentice), Tyler Smith (Undergraduate Research Apprentice), Xiaohong Pan (Assistant Development Engineer), Pierre Emmanuel Mazare (Visiting Graduate Student Researcher), Matthieu Nahoum (France Visiting Scholar), Sarah Stern (Undergraduate Research Apprentice), Yanli Li (Finland Visiting Scholar).

Executive Summary

A follow up to Mobile Century

The phenomenal success of *Mobile Century* demonstrated the feasibility of traffic monitoring by using data from GPS-enabled cell phones in a controlled environment. The next logical step was to demonstrate similar capabilities of a system in an environment in which users were representative of the general public. *Mobile Millennium* was developed at a large scale to establish the technical feasibility of such a system, and to demonstrate that under the right incentivization mechanisms, a proper recruitment strategy could be built, and would lead to successful operations of a pilot through a field operational test. A secondary objective was to demonstrate that with the amount of probe data available from the system (or more generally from industry), one could complement existing infrastructure of the DOTs to provide improved travel information services.

Traffic congestion mitigation

Traffic congestion in the USA alone causes a $78 billion drain on its economy annually. This figure doubled during the period from 1997 to 2007, and addressing this challenge is a high priority. To improve transportation operations, governmental agencies need an *integrated* view of the full transportation network. Likewise, the traveling public needs *location based* information, available in real-time, to make more efficient decisions. The emergence of the *mobile internet* on cellular devices and the rapid proliferation of location based services provide opportunities to improve the transportation experience of the traveling public. However one missing component, key to addressing these issues, is an exhaustive *trafficinformation system* that could be used at global scale.

Building a large-scale prototype

Mobile Millennium was a first attempt to address these challenges by rethinking what it means to build a traffic information system. Building the prototype traffic monitoring system included a smartphone application to gather data and a back-end system capable of collecting the data and processing them in real-time. This prototype served as the backbone for operations during the

field operational test.

Partnership with industry

Another accomplishment was the creation of a proper intellectual property (IP) framework to enable academia, industry and government to work together. This IP agreement served as a model for a master agreement currently in place at UC Berkeley to handle Nokia based funding. The success of these endeavors demonstrated the ability of academia and industry to work together at a scale rarely achieved before.

Research achievements

New traffic models (both highways and arterials) were developed, as well as algorithms, to integrate streaming data into the models (static and mobile data). Numerous mathematical and algorithmic contributions generated as part of this work resulted in more than 40 publications in scholarly journals and conferences.

Addressing sustained engagement

One difficulty encountered during the project was that of sustaining the engagement of the users. Before the dawn of the app store only a few apps were available for smartphones, so user behavior with respect to apps was relatively unknown. This project shed light on user behavior, i.e. the conditions under which people continue using apps.

Data dissemination, user experience, user interface

This work also pioneered the era of user interface design, in that it was one of the first mobile phone based mapping apps deployed in the US. In this respect, *Mobile Millennium* also represents an early instantiation of location based services, geolocalized mapping services, and mobile apps. *Mobile Millennium* was developed before the app store and at a time when very few apps were available.

Pioneering the era of crowdsourcing, participatory sensing, data brokering and data fusion

Mobile Millennium was a pioneer project in the era of crowdsourcing and participatory sensing and was one of the first systems in history to tackle crowdsourcing in the context of traffic at large scale. History has since shown that crowdsourced data created a very fragmented market (not only for traffic but in general). Due to the fragmentation of the market, no single source of data possesses all four properties required for a successful traffic information system. These properites are: ubiquity, timeliness, accuracy, and reliability. As a result of this situation, any future success will depend on data brokering to enable data fusion at a global scale. Because the market is so fragmented, any entity (academia, industry or government) will most likely be forced to acquire data, or create data deals with other entities, to collect data with the necessary features to enable traffic information systems that are able to address modern challenges. The era of data feed fusion has just begun, and numerous tasks still need to be completed before this can converge to practical solutions applicable to DOTs, industry and academia.

Pioneering control theory and machine learning applications to traffic engineering

As part of the contributions which were necessary for this work, the team contributed to advancing several scientific fields which led to breakthroughs in transportation engineering. In particular, using the framwork of viability theory, the team created new descriptions of traffic flow capable of integrating mobile data. Using machine learning techniques, the team created new methods for estimation of travel time at low penetration rates in the arterial network. Using stochastic dynamic programming, the team created new routing algorithms capable of handling uncertainty in planning based on traffic forecasts. The project overall enabled fundamental breakthroughs in science necessary to advance the state of practice.

Guiding the California DOT and the US DOT through the infancy of data fusion

With a potentially new source of ubiquitous data (mobile phone data), how would one operate data fusion at a global scale in an efficient manner and how would one manage to use this data efficiently in order to modernize traffic information systems. This question is far from being answered today (also because sources of data keep changing). However *Mobile Millennium* provided a first step in this direction and substantial breakthroughs in this area. Today, data fusion efforts are ongoing everywherein the US. Data sources have exploded, and large scale data analytics is on the verge of becoming an academic topic, which is extremely valuable to leading industry entities such as Google, Facebook, LinkedIn, Twitter, Microsoft, IBM and many others.

Legacy

The *Mobile Millennium* project was a groundbreaking pilot deployment that pushed forward the state of the art in crowdsourcing and participatory sensoring. The *Mobile Millennium* system, now in place at PATH, in and of itself constitutes a significant technological contribution that will enable continued advancements to the state of the art and the state of practice. In addition, the *Mobile Millennium* system unlocks new avenues of future research as new sources of data become available.

Table of Contents

Acknowledgements i
Contributors and partners, institutional support ii
California Department of Transportation ii
US Department of Transportation ii
CCIT team ii
Nokia / NAVTEQ team iii
UC Berkeley iii
Academic partners iii
The Team iv
Management iv
Team Members iv
Executive Summary vi
1. Introduction 2
1.1. Problem statement 2
1.1.1. Motivation 2
1.1.2. Problem 3
1.2. Institutional partnerships 4
1.2.1. California DOT and the US DOT 4
1.2.2. Nokia / NAVTEQ team 5
1.2.3. UC Berkeley team 5
1.2.4. Other contributors 7
1.3. Scope 8
1.3.1. Review of funding sources from the DOTs 8
1.3.2. Categorization of deliverables 8
1.4. Research objectives 9
1.4.1. Definition of Mobile Millennium's initial goals 9
1.4.2. Summary of research achievements and findings 11
1.4.3. Significance of the research findings to overall operations of Caltrans 15
1.5. Organization of report 17
2. Background 19
2.1. Context 19
2.1.1. The rise of the mobile internet 19
2.1.2. Large scale cyber-physical infrastructure systems 19

2.1.3. Historical context in 2007: probes as a contributor to congestion alleviation?.....20
2.2. An early instantiation of crowdsourcing and participatory sensing.....21
2.2.1. Historical context 21
2.2.2. Issues and trends.....21
2.2.3. State of the art.....22
2.2.4. Assessment of available information 23
2.3. Grand challenges of the data and the technology for traffic information systems 24
2.3.1. Ubiquity.....24
2.3.2. Timeliness 24
2.3.3. Accuracy.....24
2.3.4. Reliability 25
2.3.5. Assessment: challenges for the creation of traffic information systems 25
2.4. Relation to existing work 25
2.4.1. Vehicle Infrastructure Integration (VII) 26
2.4.2. V2V networks.....26
2.4.3. The I95 corridor coalition.....27
2.4.4. Cellular tower information 27
2.4.5. Mobile Century.....28
3. A rethinking of traffic information systems 30
3.1. Taxonomy of Traffic Sensor Types 31
3.1.1. Loop Detectors 31
3.1.2. Radar 31
3.1.3. Video 32
3.1.4. License Plate Readers.....32
3.1.5. RFID Transponders 33
3.1.6. Bluetooth 33
3.1.7. Wireless Sensors.....33
3.1.8. Virtual Trip Lines (VTL) 34
3.1.9. Sparsely-sampled GPS 35
3.1.10. High-frequency GPS 36
3.2.1. Driver Privacy 38
3.2.2. Raw Data Accuracy and Filtering 38
3.2.3. Scalability.....39
3.2.4. Network Abstraction 39

3.2.5. Map Matching and Path Inference 43
3.2.6. Visualization 44
3.2.7. Mobile Client 46
3.2.8. Sensor Deployment 47
3.3. A Berkeley Prototype: intro to the Mobile Millennium System 47
3.3.1. History of the Project 47
3.3.2. System Architecture 49
3.3.3. Database Design 51
3.3.4. System Modules 52
3.3.5. Field Experiments 54
4. Outreach 56
4.1. Introduction 56
4.1.1. Partnerships and Funding 57
4.1.2. Awards and Special Ceremonies 59
4.2. Marketing – Message and Medium 61
4.3. Users, Phones and Functions 87
4.3.1. User Downloads 87
4.3.2. Phones and Carriers supported 91
4.3.3. System Functionality 92
4.4. Legal Items 93
4.4.1. Human Subjects Review (UC) 94
Privacy Statement for University of California, Berkeley Websites 95
4.4.2. Privacy Policy - Nokia 97
4.4.3. Terms of Service - UC 102
4.4.4. Terms of Service - Nokia 105
4.4.5. End User Software Agreement (Nokia) 110
4.5. Web Interfaces 114
4.5.1. The Web Site 114
4.5.2. Support Forum 118
4.5.3. Newsletter 121
4.5.4. Visualization 123
4.6. Survey Process 125
4.6.1. Summary 125
4.6.2. Survey Related Emails 130

4.7. Survey Results 134
4.7.1. First Survey 134
4.7.2. Second Survey 140
4.8. Conclusion 151
5. Demos and Field Experiments 152
5.1. Introduction 152
5.1.1. Moving from concept to field operational test 152
5.1.2. Purpose of the field operational field (FOT) 153
5.1.3. Chronology 154
5.2. Field test components 156
5.2.1 System operation 157
5.2.2 System operation testing: Berkeley-San Francisco, CA 157
5.2.3 Berkeley-San Francisco, CA road tests 160
5.2.4. System operation testing: New York City, NY 173
5.2.5. New York City, NY arterial testing 174
5.2.6. Consumer adoption 178
5.2.7. System evaluation 185
6. Core Systems 182
6.1. System Overview 182
6.1.1. Experience and background of the personnel developing the system 185
6.1.2. Funding levels and paradigms 186
6.1.3. System Description – languages, Infrastructure, Etc 188
6.1.4. System Architecture – Data Flow and Cross Cutting Layers 190
6.2. Important terms and definitions 193
6.3. Conventions used in this document 194
6.4. System requirements 195
6.5. System architecture overview 195
6.5.1. Database 196
6.5.2. Core system 196
Input Feeds 196
Filters 197
Models 197
Output Feeds 197
Analysis 197

6.6. System architecture detailed module descriptions 198
6.6.1. Database 198
6.6.2. Core system 199
6.6.3. Input Feeds 202
6.6.4. Filters 202
6.6.5. Models 203
6.6.6. Output Feeds 203
6.6.7. Analysis 203
6.7. Model Graph Specification 205
6.8. Hardware 206
6.8.1. Introduction 206
6.8.2. Live System 207
6.8.3. Development System 207
6.8.4. Data Warehouse 208
6.8.5. Experiment Machines 208
6.8.6. Mobile Millennium Wiki Machine 209
6.8.7. IBM PeMS Machine 209
6.8.8. traffic.calccit.org Webserver 210
6.8.9. Mobile Millennium Gateway Machine 210
6.8.10. Mobile Millennium Monitor Machine 210
6.8.11. CITRIS Visualizer Machine 211
6.8.12. Conclusion 211
6.9. Mobile Millennium Run Scripts 212
6.9.1. Introduction 212
6.9.2. Use Instructions 212
6.9.3. Configuration 213
6.10. Mobile Millennium Monitoring System 214
6.10.1. Introduction 214
6.10.2. Dashboard 214
6.10.3. Monitoring Database 217
6.10.4. Back-end 217
6.10.5. Conclusion 218
7. System Modules 219
7.1. Introduction 219

7.2. ALARM 219
7.3. ARTERIAL 219
7.4. CABSPOTTING 220
7.5. CORE 220
7.5.1. Database 220
7.5.2. Exceptions 221
7.5.3. Geometry 221
7.5.4. Monitor 221
7.5.5. Time 222
7.6. DASHBOARD 222
7.7. DEV_ENV 222
7.7.1. geo 222
7.7.2. process 222
7.7.3. tcphelper 223
7.7.4. util.encrypt 223
7.8. DIVA 223
7.9. The Highway Model 224
7.9.1. HIGHWAY 224
7.9.2. HIGHWAYCOMMON 224
7.9.3. HIGHWAYFLOWMODEL 224
7.10. MACHINE_STAT_TRACKER 224
7.11. MM_MANAGER 225
7.12. NETCONFIG 226
7.13. PATH_INFERENCE 226
7.14. PEMS 227
7.14.1. rawFeed 227
7.14.2. filterParameters 228
7.14.3. filter 228
8. Visualization 230
8.1. Introduction 230
8.2. VIZ: The First Mobile Millennium Visualizer 230
8.2.1. Design Strategy 230
8.2.2. Server Code 235
8.2.3. Client 238

8.2.4. Results 239
8.3. CITRIS VIZ 241
8.4. Special Applications 245
8.4.1. *Mobile Millennium* Website 245
8.4.2. AASHTO 246
8.4.3. Routing Research 248
8.4.4. Drifter Project 249
8.4.5. ClearSky Project 251
8.5. DIVA: The Second Mobile Millennium Visualizer 252
8.5.1. Server 253
8.5.2. Client 254
8.6. Conclusion 255
9. IBM Traffic Prediction Tools 256
9.1. Introduction 256
9.2. Filtered PeMS Feed 258
9.3. CHP Data 261
9.4. TPT Input Feed 263
9.5. Results 266
9.5.1. Total error 266
9.5.2. Error over time, all VDS 267
9.5.3. Individual VDS error over time 268
9.6. Conclusion 269
10. High Performance Computing 270
10.1. Background 270
10.1.1. Path Inference Algorithm 270
10.1.2. Link Travel Time Estimation 271
10.2. High Performance Computing Systems (HPC) 272
10.2.1. NERSC Carver Cluster 273
10.2.2. UCB CITRIS Cluster 274
10.2.3. Amazon EC2 274
10.2.4. Cluster Specifications 274
10.3. Parallelization Approach 275
10.4. First Parallelization Effort 283
10.4.1. Performance Testing 283

10.4.2. Results 284
10.4.3. Analysis of the tests 286
10.4.4. Summary of the tests 289
10.5. Second Parallelization Effort 289
10.5.1. In-Memory Computation 289
10.5.2. Broadcast of Large Parameter Vectors 290
10.5.3. Access to On-Site Storage System 292
10.5.4. Performance Evaluation 293
10.6. Related Work 295
11. The Mobile Millennium Highway Model 313
11.1. Formulations of Distributed Parameters Systems 314
11.1.1. The forward problem 314
11.1.2. The estimation problem 315
11.2. Velocity estimation from GPS-equipped mobile devices 316
11.3. Related work 317
11.3.1. Traffic flow theory 317
11.3.2. Traffic estimation 318
11.4. Outline of Highway Model Description 319
12. LWR PDE 321
12.1. Derivation of a mass conservation law for traffic 321
12.2. Weak solutions and the entropy condition 326
12.3. The Riemann problem 329
12.3.1. Riemann solver 330
12.3.2. Weak boundary conditions revisited 336
12.4. Numerical discretization 337
12.4.1. Godunov Scheme 337
12.4.2. A note on linearization 339
13. Derivation of a velocity evolution equation 342
13.1. Velocity functions 342
13.2. Derivation of a velocity PDE in conservative form for the Greenshields flux function 344
13.3. Numerical approximation of the velocity evolution equation 348
13.4. Extension of the model to networks 350
13.4.1. Network model and edge boundary conditions at junctions 350
13.4.2. Discrete CTM-v network algorithm 357

14. Velocity estimation....360
14.1. Development of a recursive velocity estimation algorithm....360
14.1.1. State–space model....360
14.1.2. Extended Kalman filtering for nonlinear systems....365
14.1.3. Ensemble Kalman filter....366
14.1.4. Large scale real–time implementation....367
14.2. Experimental Results....369
14.2.1. Experimental setup....369
14.2.2. Numerical implementation....370
14.2.3. Comparison with inductive loop detectors....371
15. Literature review and background material....376
15.1. Machine Learning/Probabilistic Graphical Models....376
15.1.1. Probabilistic Graphical Models....377
15.1.2. Statistical Filtering....377
15.1.3. Expectation Maximization....380
15.2. Previous Arterial Traffic Estimation Techniques....381
15.2.1. Flow Models....382
15.2.2. Statistical Models....383
16. Travel time delay patterns through signalized intersections....389
16.1. Notation and Assumptions....389
16.1.1. Notation....389
16.1.2. Traffic Flow Modeling Assumptions....391
16.2. Travel Time Patterns Through Signalized Intersections....392
17. An extension of traffic fundamentals: A hydrodynamic theory based statistical model of arterial traffic....395
17.1. Modeling arterial traffic....396
17.1.1. Traffic model....396
17.1.2. Traffic flow modeling assumptions....397
17.1.3. Arterial traffic dynamics....398
17.1.4. Notation....401
17.2. Modeling the spatial distribution of vehicles on an arterial link....402
17.2.1. General case....402
17.2.2. Undersaturated regime....403
17.2.3. Congested regime....405
17.3. Modeling the probability distribution of delay among the vehicles entering the link in a

cycle 406
17.3.1. *Total delay* and *measured delay* between locations $x1$ and $x2$ 407
17.3.2. Probability distribution of the total and measured delay between $x1$ and $x2$ in the undersaturated regime 408
17.3.3. Probability distribution of the measured delay between $x1$ and $x2$ in the congested regime 413
17.4. Probability distributions of travel times 416
17.4.1. Travel time distributions 418
17.4.2. Quasi-concavity properties of the probability distributions of link travel times 420
17.4.3. Log-concavity properties of the different components of the mixture model 425
17.5. Conclusion 426
18. Traffic Models for Arterial Estimation and Prediction 428
18.1. Regression Models 428
18.1.1. Assumptions 428
18.1.2. Graph Model of the Road Network 429
18.1.3. Traffic Level of Service Indicators 430
18.1.4. Estimating Level of Service Indicators 431
18.2. Historic modeling using Bayesian inference for real-time estimation 433
18.2.1. Historic Model of Traffic 433
18.2.2. Real-Time Estimation 434
18.3. Probabilistic Graphical Model 434
18.3.1. Assumptions 435
18.3.2. Graphical Model 435
19. Learning and Inference Algorithms for Arterial Traffic Estimation 438
19.1. Solution Methods for Regression Models 438
19.1.1. Logistic Regression 439
19.1.2. STARMA 440
19.1.3. Results 442
19.2. Solution Methods for Bayesian Model 448
19.2.1. Learning Historic Model Parameters 450
18.2.2. Bayesian Real-time Traffic Estimation 455
18.3. Solution Methods for Probabilistic Graphical Model 456
18.3.1. State Estimation 457
18.3.2. Parameter Estimation: the EM Algorithm 459
Results 461

20. Enhancing Privacy and Accuracy in Probe Vehicle Based Traffic Monitoring.....465
20.1. Traffic Monitoring Challenges in Probe Vehicle Systems 466
20.1.1. Privacy Risks.....466
20.1.2. Lack of Guaranteed Accuracy of Sensor Data 467
20.2. Virtual Trip Lines 468
20.2.1. Strengths.....469
20.2.2. Virtual Trip Line Measurements 469
20.3. Architecture Designs.....473
20.3.1. Achieving Authenticated but Anonymous Data Collection 473
20.3.2. Guaranteeing K-Anonymity at Low Density Using Temporal Cloaking.....475
20.3.3. Balancing Privacy and Accuracy Requirements 478
20.4. Experimental Evaluation.....478
20.4.1. Trip Line Placement 479
20.5. Results.....481
20.5.1. VTL Measurement Accuracy 481
20.5.2. Guaranteed Privacy via VTL-based Temporal Cloaking 483
20.5.3. Reconstructing VTLID-Location Mapping.....484
20.5.4. Accuracy-Centric Architecture.....485
20.6. Discussion.....487
20.6.1. Security.....487
20.6.2. Involvement of Cellular Networks Operators 489
20.6.3. Challenges in Arterial Roads Traffic Estimation 489
20.6.4. The Mobile Millennium Field Operational Test 490
20.7. Conclusions.....490
21. Kernel regression for travel-time estimation via convex optimization 491
21.1. Background.....491
21.2. Problem Statement.....492
21.2.1. Travel-time estimation 492
21.2.2. Regularization 493
21.2.3. Kernel methods.....493
21.2.4. Learning setting.....494
21.3. Analysis.....494
21.3.1. Convex formulation.....494
21.3.2. Cross-validation.....495

21.3.3. Kernel regression 496
21.3.4. Rank-one kernel optimization 497
21.3.5. Choice of kernels 498
21.4. Simulation results 499
21.4.1. Dataset description 499
21.4.2. Analysis method 499
21.4.3. Results and discussion 500
21.5. Conclusions and future work 501
22. Reliable Routing in Stochastic Networks 503
22.1. Introduction 503
22.2. The Stochastic On-time Arrival (SOTA) Problem 505
22.3. Continuous time exact formulation of the SOTA problem with single iteration convergence algorithm 508
22.3.1. Solution algorithm for single iteration convergence 508
22.3.2. Extended algorithm for time-varying link travel-times 509
22.3.3. Generalized algorithm for correlated link travel-times 513
22.4. Discrete formulation of the SOTA algorithm with a Fast Fourier Transform solution 514
22.4.1. Complexity analysis 515
22.4.2. Acceleration of Algorithm 29 with localization 518
22.4.3. Search space pruning by elimination of infeasible paths 522
22.5. Implementation of the algorithm in the Mobile Millennium system 523
22.5.1. Numerical results 525
22.6. Conclusions 531
23. A general phase transition model for vehicular traffic 532
23.1. Background 532
23.2. The Colombo phase transition model 534
23.3. Extension of the Colombo phase transition model 536
23.3.1. Analysis of the standard state 537
23.3.2. Analysis of the perturbation 538
23.3.3. Definition of parameters 541
23.3.4. Cauchy problem 541
23.3.5. Model properties 542
23.3.6. Numerics 544
23.4. The Newell-Daganzo phase transition model 546
23.4.1. Analysis 546

23.4.2. Solution to the Riemann problem 547
23.4.3. Model properties 548
23.4.4. Benchmark test 549
23.5. The Greenshields phase transition model 551
23.5.1. Analysis 551
23.5.2. Solution to the Riemann problem 552
23.5.3. Model properties 553
23.5.4. Benchmark test 554
23.6. Conclusion 555
24. Hamilton-Jacobi PDEs: A new framework 558
24.1. Partial differential equation models of large scale infrastructure systems 558
24.2. Control and estimation of partial differential equations 559
24.2.1. Filtering based methods 559
24.2.2. Other methods 560
24.3. Hamilton-Jacobi equations 560
24.4. Numerical analysis for Hamilton-Jacobi equations 561
24.5. Scientific Contributions 562
25. Hamilton-Jacobi PDEs: Fast and exact semi-analytic schemes 564
25.1. Macroscopic highway traffic modeling 564
25.1.1. State of the art 564
25.1.2. First order scalar conservation laws 565
25.1.3. Hamilton Jacobi equations with concave Hamiltonians 566
25.1.4. Hamiltonian 567
25.2. Value conditions 569
25.2.1. General definition 569
25.2.2. Initial, boundary and internal conditions 569
25.3. Viability formulation of the solution 571
25.3.1. Barron-Jensen/Frankowska solutions 571
25.3.2. Viability characterization of Barron-Jensen/Frankowska solutions 571
25.3.3. The Lax-Hopf formula 577
25.4. Properties of the Barron-Jensen/Frankowska solutions to Hamilton-Jacobi equations 580
25.4.1. Domain of definition 580
25.4.2. The inf-morphism property 581
25.4.3. Convexity property of the solutions associated with convex value conditions 583

25.5. Analytic solutions associated with affine initial, boundary and internal conditions 584
25.5.1. Analytic Lax-Hopf formula associated with an affine initial condition.....584
25.5.2. Analytic Lax-Hopf formula associated with an affine upstream boundary condition588
25.5.3. Analytic Lax-Hopf formula associated with an affine downstream boundary condition 593
25.5.4. Analytic Lax-Hopf formula associated with an affine internal condition.....598
25.6. Extension to piecewise affine initial, boundary and internal conditions 606
25.6.1. Semi-analytic solutions 606
25.6.2. Lax Hopf algorithm 607
25.7. Extension to scalar conservation laws 608
25.7.1. Spatial derivatives of the solutions to affine initial, boundary and internal conditions 608
25.7.2. Computation of the density function 610
25.7.3. Extension of the Lax-Hopf algorithm for scalar conservation laws.....611
25.8. Numerical examples.....612
25.8.1. Integration of internal conditions into Hamilton-Jacobi equations 612
25.8.2. Numerical validation of the Lax-Hopf algorithm (density function) 614
25.8.3. Comparison with standard numerical schemes 616
26. Hamilton-Jacobi PDEs: Convex formulations of the model constraints.....618
26.1. Model constraints for well-posedness.....618
26.1.1. Compatibility conditions 621
26.1.2. Sufficient conditions on the Hamiltonian for compatibility of true value conditions622
26.2. Properties of the model compatibility constraints 628
26.2.1. Concavity property of the solutions with respect to their coefficients.....628
26.2.2. Convex formulation of the model compatibility constraints.....629
26.2.3. Monotonicity property of the model compatibility conditions 630
27. Hamilton-Jacobi PDEs: Applications of the new framework 632
27.1. Traffic flow measurement data and value conditions.....632
27.1.1. Fixed detector data 633
27.1.2. Mobile sensor data.....633
27.1.3. Experimental setup 633
27.1.4. Link between measurement data and value conditions 634
27.2. Explicit instantiation of the model compatibility conditions for triangular Hamiltonians 637
27.3. Data constraints.....643

27.4. Compatibility and consistency problems 644
27.4.1. Data and model compatibility problem 645
27.4.2. Data consistency problem 645
27.5. Estimation problems 646
27.5.1. Definition for general functions of traffic-related coefficients 646
27.5.2. Lower and upper bounds on traffic coefficients 646
27.5.3. Guaranteed ranges for traffic coefficients estimation 651
27.6. Data assimilation and data reconciliation problems 651
27.6.1. Problem definition 651
27.6.2. Numerical example 652
27.7. Cybersecurity, sensor fault detection and privacy analysis problems 654
27.7.1. Consistency problems applied to sensor failure detection 654
27.7.2. Consistency problems applied to cybersecurity 656
27.7.3. Privacy analysis problems 659
28. Recommendations 662
28.1. Summary of the project 662
28.2. Relevance of the research approach 663
28.2.1. Research agenda in the context of 2008 663
28.2.2. Mobile Millennium, a step towards nextGen ATIS 664
28.3. Immediate conclusions from the experience 665
28.3.1. Industry perspective 665
28.3.2. Government perspective 667
28.3.3. University perspective 667
28.4. Advances to research, technology and practice 668
28.4.1. Post facto context: closing the project in 2011 668
28.4.2. Research 670
28.4.3. Technology 673
28.4.4. Practice 675
28.4.5. General outreach 677
28.5. Lessons Learned 678
28.6. Plans for deploying the research findings 680
28.6.1. Immediate practical application of the research findings 680
28.6.2. Institutional context for deployment of the research findings 681
28.6.3. Potential benefits 682

28.7. Recommendations for the future 682
28.7.1. Traffic app developments / industry 682
28.7.2. Data fusion, data hybridization, data procurement 684
28.7.3. Beyond highway and arterial traffic 684
References 687
A. Technical Appendix 713
A.1. Derivation of the probability distribution of total delay between arbitrary locations in the congested regime 713
Case 1: $x1$ is upstream of the total queue and $x2$ is in the remaining queue (Figure A.1) 714
Case 2: $x1$ and $x2$ are upstream of the remaining queue (Figure A.2) 715
Case 3: $x1$ is in the remaining queue, and thus so is $x2$ (Figure A.3) 715
Case 4: $x1$ is in the triangular queue, $x2$ is in the remaining queue 716

Mobile Millennium

Final Report

Prepared by:

Alexandre Bayen	Joe Butler	Anthony Patire	Steve Andrews	Murali Annavaram
Saneesh Apte	Jeff Ban	Sebastien Blandin	Christian Claudel	Daniel Edwards
Laurent El Ghaoui	Jonathan Felder	P. Goatin	Marco Gruteser	Juan C. Herrera
Ryan Herring	Aude Hofleitner	Baik Hoh	Tim Hunter	Toch Iwuchukwu
Quinn Jacobson	Samy Merzgui	Benedetto Piccoli	Samitha Samaranayake	Daniel Work

For:
California Department of Transportation

Division of Research and Innovation

California Center for Innovative Transportation
University of California, Berkeley
2105 Bancroft Way, Suite 300, Berkeley, CA 94720-3830
Phone: (510) 642-4522 Fax: (510) 642-0910 http://www.calccit.org

Part I

Overview

1. Introduction

The Mobile Millennium project was a groundbreaking pilot deployment that pushed forward the state of the art in crowdsourcing and participatory sensing, and demonstrated the technical feasibility of creating a traffic information system relying principally on probe data. This final report for the Mobile Millennium project closes the contract for task order P-6615 Connected Traveler Safe Trip 21, as well as the RTA C812 and RTA C903 agreements.

This chapter begins by describing the main thrust of the research effort. Next, the nature of the institutional partnerships (upon which the success of this project is based) are discussed as a model for stakeholder cooperation. A brief review of the funding sources and their relation to the completed work are then listed. In the following section, key research achievements and findings are summarized, along with remarks on the significance of these findings and their potential impact on the future operations of Caltrans. This chapter concludes by describing the organization of the remainder of the report.

1.1. Problem statement

1.1.1. Motivation

Fighting economic losses due to inefficiency in the transportation network

Traffic congestion in the USA alone causes a $78 billion drain on its economy annually, a figure that doubled during the period from 1997 to 2007. The current recession has only marginally improved the state of traffic congestion due to reduced activities in some regions of the nation, but long term projections in 2009 by the *Energy Information Administration* (EIA) at the US Department of Energy show a flattening - but not a significant decline - in this trend.

Prerequisites for transformational results

In 2007, it appeared unlikely that traditional physically-centered mitigation strategies, such as building more highways, widening existing roads, and expanding or starting new transit or rail routes, would be successful by themselves. These approaches are simply not sustainable in the

current economic and environmental climate. Rather, innovative paradigms are needed to marshal breakthroughs for operations of the transportation network, to transform the manner by which the traffic management issues are addressed. One missing component, key to addressing these issues, is an exhaustive *traffic information system* that could be used at global scale. With partial information only, it is very unlikely that management of the transportation system could be improved significantly, and that the adoption of new transportation paradigms by the public could shift from its current state.

A societal challenge

At the heart of this problem are significant barriers that prevent operations and system use from being addressed at a national scale. In 2007, we proposed to tackle the *technical* high-risk high-reward issues associated with these challenges, which we believe to be centered around information gathering, integration, and distribution. To better operate the system, agencies need an *integrated* view of the full transportation system. To participate in a change in the way the transportation system is used, the public needs *location based* information, available in real time, to make more efficient decisions, which might include increased use of the transit system. With the emergence of the *mobile internet* on cellular devices and the rapid proliferation of location based services, it was the right time to launch initiatives that would result in paradigm shifts in the way the public interacts with the transportation network. In particular, we were specifically interested in finding ways in which to give incentives to the public to use new technology to improve its transportation experience.

1.1.2. Problem

Mobile Millennium enabled the research community, industry partners, and government to answer several questions of interest to transportation engineering at the time the project was started. The motivation for Mobile Millennium came from the difficulties of public agencies to instrument infrastructure systems at a global scale. While California had been very fortunate to have significant instrumentation (in particular the loop detector systems deployed on the major freeways of California), it was clear from the start of the project (and it is still clear in the current financial situation of California) that the secondary network (arterial roads) would not be instrumented in the near future. Furthermore, the aging loop detector system suffered significant performance issues, and the natural question arose of knowing what could be done using probe data for traffic. Traffic problems can be sorted in three main categories:

- Traffic information systems
- Operation systems (traffic control)
- Planning

The data required for these three distinct problems is different in nature. Mobile Millennium focuses principally on the first of these three categories: traffic information systems. The fundamental question of interest was: “would it be possible to create traffic information systems based principally on probe data.” This question is not only of interest to the government, but also to industry, as demonstrated by the large amount of traffic applications that have emerged from the private sector since the start of the project.

Solving the problem of determining if such a traffic information system can be built relies on numerous other questions that had to be investigated as well as part of this work. In particular, how to create the proper infrastructure to collect data appropriate for building a next generation traffic information system. One of the challenges not anticipated at the time was the fragmentation of the market, i.e. the fact that even if successful applications take off, one needs to figure out a way (technological, or institutional) to make sure that data is available at a global scale, and in sufficient quantity.

Mobile Millennium was principally a technology driven project. While it investigated some of the policy questions pertaining to the collection of probe data, in particular privacy, it did not investigate the implication of shifting from dedicated infrastructure data to crowdsourced / participatory sensing data[1]. The goal was mainly to establish the technical feasibility of such a system, and to demonstrate that under the right incentivization mechanisms, a proper recruitment strategy could be built, and would lead to successful operations of a pilot through a field operational test. Solving the technical challenges leading to such a test required numerous breakthroughs in research, which are part of the achievements of this work (summarized in this report), as well as significant development of systems capabilities, which were done at UC Berkeley and Nokia.

1.2. Institutional partnerships

The Mobile Millennium project is one of the most successful examples of a tripartite collaboration and joint effort between Government, Academia and Industry in the field of transportation. The nature of the partnership is explained here and the origin of the collaboration is presented as well. The following narrative describes the California Partners for Advanced Transit and Highways (PATH) and the California Center for Innovative Transportation (CCIT) as they were before the recent merge.

1.2.1. California DOT and the US DOT

The U.S. DOT SafeTrip-21 (Safe and Efficient Travel through Innovation and Partnerships for the 21st Century) Initiative was established to test and evaluate integrated, intermodal, Intelligent Transportation Systems (ITS) applications. The overall goals of the SafeTrip-21 Initiative were to (1) Expand and accelerate the U.S. DOT's research in vehicle connectivity with the wireless communications environment, (2) Build upon Intelligent Transportation Systems (ITS) research in advanced-technology applications, (3) Explore and validate the benefits of deployment-ready applications that provide travelers, drivers, and transit and commercial motor vehicle operators with enhanced safety, real-time information, and navigation assistance. The Volpe Center made awards to establish two test beds: the California Connected Traveler (CACT) Test Bed, which involved integrated locations in the San Francisco Bay Area, and the I-95 Corridor Test Bed, which involved locations along the I-95 Corridor from North Carolina to New Jersey. Two independent applications, although not formally part of the test beds, were also tested in California and North Carolina. The California Connected Traveler testbed was led by the

[1]The terms *crowdsource* and *participatory sensing* are described in their proper context in Chapter 2.

California Department of Transportation in partnership with the University of California at Berkeley and two organizations administered by its Institute of Transportation Studies. These were California Partners for Advanced Transit and Highways (PATH), with a mission to develop solutions to the problems of California's surface transportation systems through cutting edge research, and California Center for Innovative Transportation (CCIT), with a focus on deployment and a mission to accelerate the implementation of research results. Mobile Millennium was hosted by CCIT, described below. Mobile Millennium was funded by a consortium of funders, the most important of which were the US Department of Transportation and the California Department of Transportation.

1.2.2. Nokia / NAVTEQ team

The Mobile Millennium project was a partnership established with the Nokia Research Center (NRC) in Palo Alto, one of the research branches of Nokia in the US. Before this project had started, Nokia Research Center was building momentum to grow its mobile traffic probe program in scope and scale. The center had identified the Mobile Millennium field operational test as the next step on the roadmap toward commercialization. This effort followed a very fruitful collaboration between Nokia and UC Berkeley leading to the Mobile Century project. Nokia provided hundreds of highly-subsidized GPS phones to enable that deployment, equipped its employees with thousands of these internally (who ran the pilot), which represented a considerable investment of resources. In collaboration with the Berkeley team, Nokia Research Center set the technical agenda of the deployment. The NRC staff continued to design, develop and operate client-side and server-side software that served as the engine for traffic data collection and dissemination during the Mobile Millennium field operational test.

At the time the project started, Nokia was in the process of acquiring NAVTEQ. Being the industry leader in the production and distribution of digital maps and real-time traffic information, NAVTEQ brought technical know-how and operational expertise to the program. They also brought maps which were used in the project, as well as NAVTEQ traffic patterns, a NAVTEQ product that contains historical traffic data. NAVTEQ also led the dissemination of the Mobile Millennium technology and information at the 2008 World Congress in New-York City. The Mobile Millennium project team worked with NAVTEQ's subsidiary Traffic.com to fuse probe data with current traffic information and to leverage existing distribution channels.

1.2.3. UC Berkeley team

Traffic and transportation engineering at UC Berkeley

The University of California at Berkeley has led the field of transportation research in the US for several decades. At UC Berkeley, transportation research is mainly performed under the Institute of Transportation Studies (ITS), which hosts numerous famous programs with different foci on transportation problems, in particular the California Center for Innovative Transportation (CCIT), the Partners for Advanced Transit and Highways (PATH), the UC Transportation Center (UCTC), the Transportation Sustainability Research Center (TSRC), and several others. One of the assets of ITS is the quality and world leadership of the programs, faculty, and staff that support it. In particular, the Civil and Environmental Engineering Department has been ranked

number one in the US for years, and its transportation engineering program is also the first in the US. This comes in addition to stellar Computer Science, Electrical Engineering, Mechanical Engineering and Operations Research Departments, which also provide significant contingents to ITS. UC Berkeley has a tradition of bridging academic research with the world of practitioners, which is instantiated through several of the aforementioned centers, in particular, PATH for prototyping of new transportation research for the field, and CCIT for the deployment of technology and accelerating the adoption by industry. The proximity of Silicon Valley and its major leaders has made it possible for ITS to be at the forefront of transportation, by constantly keeping its eyes on developing technology and making its adoption by transportation practitioners possible.

The California Center for Innovative Transportation (CCIT)

CCIT's mission is to accelerate the implementation of research results in order to enable a safer, cleaner and more efficient transportation system. Oftentimes, it is necessary to conduct specialized research and development to achieve desired implementation goals. CCIT has typically held an ongoing portfolio of 15-20 projects in areas such as traffic monitoring technologies, highway operations improvements, traveler information systems, public transportation service enhancements, and transportation safety and security. Below is a list of examples of CCIT's recent successes:

- A 3-year demonstration of the use of microscopic traffic simulation tools to model highway corridors instructed the implementation of the statewide Corridor Mobility Improvement Account (CMIA) program, which has generalized this practice;
- CCIT developed software used by Caltrans' Transportation Management Centers (TMC) to configure and automate the display of traveler information on highway Changeable Message Signs (CMS). This software is successfully serving the San Francisco Bay Area, and further deployment is underway in the Sacramento Valley and Sierra Nevada regions;
- A statewide training program enabled over 500 planners, engineers, and managers at Caltrans and regional agencies to receive hands-on training on the use of the freeway Performance Measurement System (PeMS), thus advancing the state of highway operations practice.

Such focused efforts come in addition to over 40 academic publications per year aimed at building support for selected innovations among practitioners, regular workshops on emerging transportation technologies and techniques, the maintenance and operations of technology testing infrastructure at the Berkeley Highway Lab, as well as the facilitation of collaborative agreements involving the public and the private sector at state, national, and international levels.

CCIT was the natural host for this project within UC Berkeley because of its emphasis on deployment. It provided the right emphasis on practice and theory for this project. It was therefore the most appropriate place to locate the project, and to provide the support for the project to succeed.

The Lagrangian Sensor Systems Laboratory (LSSL)

Headed by the program's Principal Investigator, Professor Alexandre Bayen, the Lagrangian Sensing Laboratory was at the core of the traffic engineering work package set forth in the Mobile Millennium technical concept. Drawing from multiple programs in the area of infrastructure monitoring, including waterways and civilian air traffic in addition to highways, Professor Bayen's strong research group possessed a diverse expertise, key to the success of a multidisciplinary, integrated project. The name Lagrangian Sensor Systems Laboratory (LSSL) disappeared through the project, for two reasons. First, from an organizational perspective, students were reorganized into projects, as needed for administrative purposes (hence the name of the lab disappeared, to leave room for the project visibility). Second, after some time, it became clear that Mobile Millennium had gained such visibility that it would make more sense for the effort to be known under a project name than a lab name. The students composing the Mobile Millennium team have diverse backgrounds: Electrical Engineering and Computer Science, Civil and Environmental Engineering, Industrial Engineering and Operations Research, and Mechanical Engineering. These diverse backgrounds reflect the nature of the project, which is multidisciplinary. The emphasis of the laboratory at the start of the project was on mobile sensing, and was significantly strengthened by the success of the project. Today, the Mobile Millennium group (and other projects attached to it) is the leading group on mobile sensing and smartphone based sensing on campus.

1.2.4. Other contributors

In addition to the partners mentioned above, the Mobile Millennium project was funded by several other sources that either provided seed funding, or direct funding, in their various capacities. These are listed below, and show the level of interest generated by the project over the years. Some of these sources of funding, or in kind support, still fund outgrowths of the project today, and are thus mentioned as part of the effort.

- BAE Systems
- Center for Information Technology and Research in the Interest of Society (CITRIS)
- Defense Advanced Research Projects Agency (DARPA)
- Department of Civil and Environmental Engineering, UC Berkeley
- Ministere des Transports, France, (now MEDDAAT)
- National Science Foundation (NSF)
- University of California Transportation Center (UCTC)
- Tekes, Finnish Funding Agency for Technology and Innovation
- Telenav Inc.
- VIMADES
- Volvo Foundation
- VTT

1.3. Scope

1.3.1. Review of funding sources from the DOTs

Mobile Millennium has been funded from several task orders, beginning with funds originally allocated to Mobile Century. In addition, there have been research technical agreements, and amendments that have been added over time. These funding sources are:

- Task Order 1021: Mobile Century
- Task Order 1029: Mobile Century
- P-6615 Connected Traveler Safe Trip 21
- RTA C812 Mobile Millennium
- RTA C903 Mobile Millennium

1.3.2. Categorization of deliverables

Most of the deliverables can be classified into three main categories:

- (1) Systems,
- (2) Highways,
- (3) Arterials.

The remaining tasks are classified as follows:

- (4) Field experiments and demonstrations
- (5) Media and outreach
- (6) IBM subcontract

The core hardware and software systems that made Mobile Millennium possible were supported by all the contracts listed above. Initial funding for traffic server development came from TO 1029 and amendments to TO 1021. Under the real-time and data storage work packages, P-6615 provided additional funding for the design and development of hardware and software systems to support the project. This effort included building interfaces and feeds, database and storage infrastructure, and adapting the available digital maps for traffic modeling. These funds were supplemented with C812 and C903 to complete the system architecture, and to support continued operation, maintenance, refinements, and hardware and software upgrades. Additional visualization and system monitoring capabilities were also implemented. All of this is discussed in detail in Part II: Mobile Millennium System.

The traffic engineering work packages in each of the contracts listed above supported core scientific activities. Amendments to TO 1021 provided funds for exploratory work in arterial modeling. For both highway and arterial modeling efforts, P-6615 supported work in traffic modeling, algorithm development and implementation, as well as data fusion. The main thrust of this work is discussed in Part III: Mobile Millennium Highway Model, and in Part IV: Mobile Millennium Arterial Models.

C812 provided additional support for data assimilation, model and algorithm refinements for prediction, and second-order models for highways. C903 supported additional work on travel time prediction and validation. Much of this work appears in Part V: Mathematical and Algorithmic Contributions, and Part VI: Framework and Applications for Hamilton-Jacobi PDEs.

Two amendments were made to C812 in order to support bluetooth testing, and validation for arterial models. The main thrust of the arterial research is discussed in Part IV: Mobile Millennium Arterial Models.

The millennium and fielding work packages in C812 along with the evaluation work package in C903 supported a number of field experiments and demonstrations. Key demonstrations include AASHTO, and the ITS World Congress. Test runs involving 10 to 20 vehicles were performed for debugging, and to provide data for systems improvements. The logistics and resources for these efforts are described in Part I: Overview.

The millennium work package in C812 along with the outreach work package in C903 supported the efforts necessary to deploy a meaningful customer service. This included logistics, business, and legal preparation, as well as outreach, advertising, customer support, and polling. A public facing website was created to engage potential users, provide client downloads, and customer service. All of this is discussed in Part I: Overview.

Support for the deployment and technology transfer envisioned in C812 were provided by an amendment to C903. This amendment included a subcontract to IBM for collaboration on multi-sourced traffic information. All of this is described in Part II: Mobile Millennium System.

1.4. Research objectives

1.4.1. Definition of Mobile Millennium's initial goals

Mobile Millennium, as a project, was tailored to answer numerous questions for academia, industry and government. By the unique nature of the partnership (UC Berkeley, Nokia, California and US DOTs), and the unique opportunity in history to pioneer a new era of technology, it was initially driven by practitioners goals, which are outlined below.

A follow up to Mobile Century

After the phenomenal success of Mobile Century, which demonstrated the feasibility of traffic monitoring by cell phone information in a controlled environment, it became clear that in order to push the concept to an operational level, it would be necessary to demonstrate similar capabilities of a system in an environment in which users were now representative of the general public. Of course such a setting comes with significant challenges, some of which known at the time, some of which appearing later during the field operational test. Mobile Millennium was just a logical follow up to Mobile Century, and thus was developed at a large scale. The goal of a field operational test was to demonstrate the feasibility of the concept in an unstructured

environment, which would be a prerequisite for adoption both by industry and government practitioners.

Creating a proper IP framework to work in partnership with industry

One of the goals of the project was to create a proper intellectual property (IP) framework, to enable academia, industry and government to work together. The challenge of this goal was to find a proper way for both research teams (UC Berkeley and Nokia) to be able to work jointly, and preserve each of the player's abilities to protect their IP. The process of creating an IP agreement was a preliminary to starting the joint work, and was a major achievement of the project. In fact the IP agreement carved between Nokia and UC Berkeley later on served as a model for a master agreement currently in place at UC Berkeley to handle Nokia based funding.

Building a prototype

The most important technological goal of the work was to build a prototype traffic monitoring system, which would serve as the backbone for operations during the field operational test. Nokia and UC Berkeley chose an extremely challenging path, which was to have a part of the system reside at UC Berkeley, and another reside at Nokia (the two being interfaced). This was successfully achieved and demonstrated the ability of academia and industry to work together at a scale rarely achieved before. The goal of the prototype was to test the new technology (procedures and algorithms) and tailor it so it could be moved to production later (at the end of the project). Building the prototype included building a smartphone client and a backend system capable of collecting the smartphone data and capable of processing the data in real-time.

Launching a system (with demo)

One of the important requirements at the launch of the system was to bootstrap it and to have operational traffic maps displayed on the phone from day one. This included working together with NAVTEQ, who contributed by giving their maps and their traffic pattern product that contained historical data for the secondary network (arterial). Launching the system included numerous tests and numerous steps which were completed as part of the project.

Create a user recruitment campaign (participatory experiment)

Mobile Millennium was the first traffic application launched in North America by Nokia at the scale of the nation. While it was only available to the driving public in California, the system was designed to cover the entire US and was beta tested for the entire US. In the pre-app store era, launching a traffic application of this nature was very challenging. It included doing media outreach, building a website where people could download the application, creating versions of the application for several platforms (Symbian Nokia phones and Blackberry phones), and having a specific customer support service (web-based and forum based).

Addressing sustained engagement

One of the most difficult goals of the project was to create sustained engagement of the users. At the time, a relatively small number of apps were available for smartphones, so user behavior with respect to apps was relatively unknown. The project enabled us to understand user behavior, i.e. the conditions under which people continue using apps. This was one of the goals of the field operational test.

1.4.2. Summary of research achievements and findings

Research on traffic flow modeling and related topics

A significant portion of the work of Mobile Millennium included developing new models to model traffic (both highways and arterials), as well as algorithms to integrate streaming data into the models (static and mobile data). The report outlines the numerous contributions performed, which are listed below:

- *Mathematical research contributions*

  - Proof of existence and uniqueness of a Barron-Jensen Frankowska solution to a *Hamilton-Jacobi* PDE for a Cauchy problem. Extension of this proof for the case of internal boundary conditions. Derivation of an analytical solution to the problem in the case of piecewise affine conditions. This equation serves as a model for Lagrangian descriptions of traffic flow (i.e. from a mobile particle perspective).
  - Derivation of a new PDE model for highway velocity evolution, called *Lighthill-Whitham-Richards-v* (LWR-v) model. Proof of equivalence of the new model with the classical LWR PDE, in the case of parabolic flux functions.
  - Derivation of a nonlinear discrete time dynamical system model for velocity that is compatible with the entropy solution of the LWR PDE, called *Cell Transmission Model-v* (CTM-v). Extension to general networks using linear programming.
  - Second order traffic models based on the Colombo model, and development of the corresponding numerical schemes (Godunov scheme). This is a generalization of first order models, which enables us to characterize the variability of traffic in the congested regime.
  - Derivation of convex optimization based frameworks for data assimilation using the Hamilton-Jacobi equation, to integrate mobile data into robust estimation problems. Applications to problems of filtering (sensor placement problems), and computation of travel time (robust travel time estimation problems).
  - Design of a controller for exponential stability of switched linear hyperbolic PDEs.
  - Derivation of new statistical models of traffic in arterial networks, based on hydrodynamic theory. Integration of signal timing into the models.
  - Derivation of new measurement models for Lagrangian data into arterial traffic models, using hydrodynamic flow theory.

- *Algorithmic contributions.*

    - *Ensemble Kalman Filtering* (EnKF) algorithms for real-time estimation for the CTM-v model. EnKF algorithms for one-dimensional and two-dimensional model equations using Lagrangian sensing.
    - *Newtonian relaxation based algorithms* for integration of Lagrangian data into flow models.
    - *Optimization based algorithms for machine learning approaches to estimation*. These machine learning methods enable us to learn from probe data the characteristics of the network as well as the flow patterns in the arterial network.
    - *Hidden Markov Models and Coupled Hidden Markov Model based algorithms*. In order to perform inference on the arterial traffic flow, we use several standard machine learning techniques that enable us to infer traffic in real-time from these models.
    - *Kernel based algorithms* for traffic inference. These models enable us to perform traffic estimation based on a characterization of traffic using kernel theory (which relies on optimization and linear matrix inequalities).
    - *Stochastic routing algorithms* for routing in arterial networks. This algorithm takes traffic forecast into account in routing vehicles in the network. It is adaptive in that it takes a policy instead of a fixed route, and adapts the policy as the vehicle progresses through the network.

All of these contributions are specifically presented in following chapters of this report. The corresponding journal and conference publications are also listed later in the report. All of these constitute a significant advancement of science, research and technology.

Implementation: real-time system, production systems, data filtering, databases, cloud computing

A second significant contribution of this project is the creation of the Mobile Millennium system, which in itself also constitutes a significant technological contribution that enabled us to advance the state of the art and the state of practice. These contributions also generated publications, which are in a different domain as the ones presented in the previous sections, and are also worth mentioning here (and are listed further in the report). In particular, implementation contributions include:

- *Prototype smartphone client for Nokia and Blackberry phones.* This task was principally achieved by the Nokia team (in NRC Palo Alto), which was the front end of the system. This application was the first traffic application deployed by Nokia in North America. This includes:

    - *Mapping system* to display live traffic on the phones
    - *Data collection system* to collect GPS data from the participating phones and send it to the system

- *Feedback system* to collect user feedback on the quality of traffic information (directly through the phones)
- *Voice interface* to provide the user with voice based traffic information (in particular traffic accidents)

- *Prototype backend data aggregation system.* The back end system constructed by UC Berkeley was also one of the first prototypes in the world capable of integrating dozens of streams of data in real time to feed the system with traffic information. Basic components of the system include:

  - *Feed systems* capable of integrating any source of live data into a real-time system.
  - *Filters* capable of cleaning the data in real-time so the data from the feeds can be efficiently used by the rest of the systems.
  - *Model implementations* that constitute the live version of the algorithms developed and summarized in the earlier section
  - *Database* capable of integrating all the data processed in real-time so the data can be archived and queried later on if necessary.
  - *Data assimilation algorithms* that constitute the implementation of the algorithms described in the section above, and constitute the algorithmic core of Mobile Millennium.
  - *Information generating routines* that transform the results of the algorithms into meaningful information for practitioners.
  - *Archival procedures* that feed the database with the processed information created through the processes above
  - *Visualization procedures and tool*. The UC Berkeley Mobile Millennium visualizer was successfully used by the team to produce demonstrations of the tool, at numerous public events.
  - *Process monitoring tools* to enable us to monitor in real time the performance (and failures of the system).

Data dissemination, user experience, user interface

Data dissemination was at the heart of the work. In particular, the group spent significant time cleaning and studying the Mobile Century data to understand better how to use the data for Mobile Millennium. The data was subsequently posted on the Mobile Millennium website so it could be used by the research community at large for traffic data analysis. Later on, other contributions were posted as well, in particular the Hamilton Jacobi toolbox used to reconstruct traffic flow from Lagrangian data. Part of the work of Mobile Millennium consisted in making this information accessible to the research community at large, so it could benefit from the experience accumulated during the effort.

User experience was partially captured during this field operational test, and is summarized in this report in Chapter 4. User recruitment was a major challenge, and user experience was also a

new topic key to the success of the project. This project took off at the very early ages of the smartphone era, at a time when the iPhone app store was just starting. As of the time this project concluded (in 2011), the iPhone app store contains more than 200,000 apps, and so does the Android app market. When the Mobile Millennium project was started, the supremacy of the iPhone had not been established yet, and the Android platform did not exist. Thus, the user experience collected as part of this work represents the early instantiation of the smartphone app era, which is still in development today.

This work also pioneered the era of user interface, in that it was one of the first mobile phone based mapping apps deployed in the US. Several years later, Google maps established a supremacy and flexibility that enabled numerous other mapping and traffic information apps to leverage its rich infrastructure. At the time when this project was started, Google maps had very few functionalities, and was also at its infancy. Google arterial traffic data started to appear in 2009 on the web based version of Google maps (in February, soon after the launch of Mobile Millennium arterial traffic during the CITRIS opening in February 2009), and a few months later on the mobile Google maps. This was later followed by most of the companies working in traffic today, in particular INRIX, BeatTheTraffic, and several others. In this respect, Mobile Millennium also represents an early instantiation of the era of location based services, geolocalized mapping services, and mobile apps.

Pioneering the era of crowdsourcing, participatory sensing, data brokering and data fusion

One of the other contributions from Mobile Millennium was the pioneering of the era of crowdsourcing and participatory sensing (see Section 2 for definitions). Mobile Millennium was one of the first systems in history to tackle crowdsourcing in the context of traffic at large scale. This paradigm has been followed by numerous companies, which shows that it was a very appropriate approach for building next generation traffic information systems. What was not clear at the time when this project started was the scale at which one can hope to achieve success in building these systems. Several years after starting the project, history has shown that crowdsourced data created a very fragmented market (not only for traffic but in general). In the specific context of traffic, it is now clear that at least for the coming years, no single entity will have enough data to sustain reliable traffic information at a global scale. Even companies like Google have not achieved the penetration of drivers required to have efforts solely relying on probes to provide traffic information.

As a result of this situation, one of the likely approaches to succeed in the future is data brokering as a prerequisite for data fusion at a global scale. Because the market is so fragmented, any entity (academia, industry or government) will most likely be forced to acquire data, or create data deals with other entities, to achieve a sufficient volume capable of creating the appropriate amounts of information necessary for operating traffic information systems. This was one of the major findings of this work. Our team realized this very early on, and at the time of completion of the report, the Mobile Millennium system integrates dozen of feeds including industry feeds (NAVTEQ, Telenav), public feeds (CHP, cabspotting, PeMS) and our own feeds (VTL / Nokia).

The era of data feed fusion has just begun, and numerous tasks still need to be completed before this can converge to practical solutions applicable to DOTs, industry and academia. These issues are summarized in the conclusion chapter of this report, and offer various promising directions for future work.

1.4.3. Significance of the research findings to overall operations of Caltrans

Throughout the Mobile Millennium project, the Mobile Millennium team had numerous meetings with US DOT staff, and California DOT staff (principally Division of Operations) to constantly align the objectives of this program with goals of the DOTs. The significance of the research findings (and the findings of operational deployments) is summarized in this section (and detailed through the different chapters of this report).

Context of use within the US DOT and the California DOT

If one divides the needs and organization of the US DOT and the California DOT into three categories, traveler information, operations, and planning, the Mobile Millennium effort falls mainly in the first category, traveler information. This was by design when the project was conceived and confirmed at each of the meetings between the Division of Research and Innovation, UC Berkeley and Nokia/NAVTEQ. The main contribution in the project was to demonstrate that with this new technology, it was possible to create an information system capable of relying principally on probe data. Because this was an open problem at the time, the secondary objective of the project was to demonstrate that with the amount of probe data available from the system (or more generally from industry), one could complement the already existing infrastructure of the DOTs to provide improved travel information services, which would otherwise be of poor quality or non-existing, using static infrastructure only. The project thus focused on establishing such evidence, by building a prototype system, and tackling both problems, i.e. generating traffic information from probes only, and generating traffic information from probes and loops (and other sources of data as well), to demonstrate that it was realistic for the technology to be used and operated by practitioners and in particular by the US DOT and the California DOT.

Technology exploration

One of the other important objectives of this research and work was to perform technology exploration. One of the missions of the University is to assist the California DOT (and to a certain extent US DOT when under the proper contracting mechanisms) in understanding how technology can impact the field of transportation in significant ways. This project, having roots traceable back to 2007 was started at a time when the mobile internet, and smartphone technology was at its infancy. At the time, it was unclear how this technology could benefit to the California DOT and the US DOT. In fact, several failed attempts to use cellular phone information to provide travel information had raised doubts in the practitioner community that cell phones were ever going to be a viable source of information for traveler information. Part of our work for the California DOT and the US DOT was specifically to demonstrate that we could

prototype a system giving evidence of such claims. Thus, the work can also be viewed as a form of active technology scouting in which the scouting activity consists in constructing a prototype system to make such an assessment.

Guiding the California DOT and the US DOT through the infancy of data fusion

One of the questions which arose immediately when working on the Mobile Millennium project was the problem of data fusion. At the infancy of the mobile internet, one question was in everybody's mind: with a potentially new source of ubiquitous data (mobile phone data), how would one operate data fusion at a global scale in an efficient manner and how would one manage to use this data efficiently in order to modernize traffic information systems. This question is far from being answered today (also because sources of data keep changing). However Mobile Millennium provided a first step in this direction which was considered to be a significant breakthrough by the community. Today, data fusion efforts are ongoing everywhere in the US. Data sources have exploded, and large scale data analytics is on the verge of becoming an academic topic, which is extremely valuable to leading industry entities such as Google, Facebook, LinkedIn, Twitter, Microsoft, IBM and many others. This work performed for the California DOT and the US DOT was a pioneering effort that showed how such fusion tasks can be achieved with a new data type, probe data, and how such fusion can be used by the government.

Rethinking traffic information systems

This project started at a time when traffic information systems had just undergone a first mutation, i.e. moving from operational center owned and operated by the government and available at best by phone, from being web based. The 511 website was an example of such a system, soon followed in the last decade by the emergence of numerous web based traffic information systems (for example traffic.com, mapquest, Google Traffic, Yahoo traffic and several other such services). In 2008, traffic information systems underwent a second mutation: they started to become part of the suite of mobile applications available through connected networks. They also started to integrate numerous additional and new sources of data, the most predominant one being probe data. These sets of information raised all kinds of questions when designing next generation traffic information systems. In particular, how to integrate new sources of data, by which process does one fuse heterogeneous sources of data, how to diversify the broadcast mechanisms for traffic information (phone, data based, sound based, etc.), how to handle compatibility with recent driver distraction laws, etc. The work performed in the Mobile Millennium project provided the government with numerous answers to these questions, which now move the California DOT and the US DOT one level further: with sufficient understanding of these new technologies to be able to design such a system inside the California DOT and the US DOT. Of course, several policy questions need to be answered before this can be achieved, which offers several opportunities for future work to migrate this technology to the government. These will be investigated in the conclusion chapter.

1.5. Organization of report

The chapters of this report are grouped into seven main parts. Each of these parts describes one facet of the Mobile Millennium project.

The first five chapters (including this introduction) of Part I provide a high level overview. Chapter 2 describes the historical context of the project, and the recent societal and technological trends that shaped its evolution. Chapter 3 paints a picture of what a traffic information system of the future should be. The ingredients for a successful system are described; current and near term sources of traffic data are surveyed, and practical considerations are explored. In this context, a summary description of the Berkeley Prototype Mobile Millennium System is given.

One unique feature of this project is the involvement of the traveling public in the pilot deployment of Mobile Millennium. To facilitate this, substantial efforts of public outreach were necessary. These efforts are described in Chapter 4.

The overview concludes with Chapter 5. This chapter tells a narrative of the logistical support that insured success of the various demonstrations and field experiments. Lessons learned from each deployment in the series informed the planning efforts for the successive deployments.

Part II of this report describes in detail the Mobile Millennium system that serves as a foundation for all the research presented here. Chapter 6 begins with a description of the data flow through an early version of the Mobile Millennium system, and explains from a high level how the system evolved. This chapter also presents the major components of the system, including the hardware and software systems. A description of each of the software modules and the heavy duty computing they perform is provided in Chapter 7. The module that transforms the output into a human-understandable map is the visualization module, and Chapter 8 is completely devoted to its description.

The last two chapters of Part II explain parallel efforts that extend the system to new frontiers. Chapter 9 showcases the adaptability of the Mobile Millennium system and its capabilities to support a research endeavor with IBM's Traffic Prediction Tool. Chapter 10 describes two successive efforts to utilize the power of cloud computing to yield crucial improvements in computation time, thus making possible new algorithms for real-time deployment.

One of the most successful and visible components of Mobile Millennium is its highway model, and the four chapters in Part III present it in detail. A simplistic version of this model was explained in the Final Report for Mobile Century. However, substantial improvements were made over the course of the Mobile Millennium project. Chapter 11 provides an overview of the model in the context of distributed parameters systems. Chapter 12 explains the mathematical preliminaries. Chapter 13 extends the model to networks (a crucial addition not present during Mobile Century). Chapter 14 explains the data assimilation algorithm, and revisits the Mobile Century dataset.

Another set of major advancements of Mobile Millennium over Mobile Century includes the arterial models. These arterial models are presented in Part IV over the course of five chapters.

Chapter 15 presents the background for the arterial research in a context that fuses machine learning and traffic flow theory. Chapter 16 characterizes the patterns of travel time though signalized intersections in the form of a theoretical model. Chapter 17 provides a probabilistic context for this model, and derives parametric travel time probability distributions for the possible states of a link. Using these distributions, traffic conditions can be inferred from measured travel times. To do this, three estimation models are presented in Chapter 18. Algorithms to solve these models, and results obtained from field experiments, are presented in Chapter 19.

Several key mathematical and algorithmic contributions of Mobile Millennium are collected in Part V. These advancements address improvements over Mobile Century, extensions of Mobile Millennium, or alternative approaches or models for either highways or arterials. Chapter 20 extends the VTL paradigm (beyond that described in the Mobile Century final report [269]) to achieve guaranteed privacy via temporal cloaking. In addition, it investigates the trade-offs between improved privacy and travel time estimation accuracy. Chapter 21 presents an alternative approach to estimating travel times along a signalized arterial using a convex optimization framework. Chapter 22 considers the challenge to maximize the probability of arriving on time at a destination given a departure time and a time budget. The described theoretical framework enables an advancement over common routing algorithms that do not consider travel time variability, and are unable to take advantage of real-time information. The proposed algorithms are targeted toward real-time mobile phone applications. Chapter 23 describes a second-order vehicular traffic flow model that allows the density-flux relation to be set-valued. In addition, this second-order model is potentially well suited as an alternative method for assimilation of velocity based and density based data.

Part VI consists of four chapters that together describe a new framework for the solution of Hamilton-Jacobi partial differential equations (HJ PDEs), and applies the framework to a number of traffic applications. Chapter 24 introduces a framework that can integrate trajectory-based as well as loop-detector based data. Chapter 25 develops a semi-analytic numerical scheme for solving the HJ PDE exactly and without requiring a computational grid. Chapter 26 presents the derivation of model constraints as convex inequalities on systems modeled by HJ PDEs. Important properties of these constraints are proven. The next chapter exploits these properties. Three main application areas for this framework are selected to illustrate its utility in Chapter 27: (1) finding bounds on vehicle travel times; (2) detecting errors caused by incorrectly mapped loop detectors; and, (3) privacy analysis.

Part VII concludes this report in Chapter 28 with an evaluation of Mobile Millennium. This chapter reiterates the key advances, describes the lessons learned, presents plans for deploying the research findings, and proposes recommendations for the future.

2. Background

2.1. Context

2.1.1. The rise of the mobile internet

The convergence of communication and sensing on multi-media platforms such as smartphones provides the engineering community with unprecedented monitoring capabilities. At the time of the start of the project, smartphones such as the Nokia N95 included a video camera, numerous sensors (accelerometers, light sensors, GPS), communication outlets (wireless, radios, bluetooth, infrared, USB, video-output / microphone), computational power and memory. The rapid penetration of GPS in phones enabled geolocalization and context awareness, leading to the explosion of *Location Based Services* (heavily relying on mapping) using phones. For example, *Nokia Maps* display theaters and museums near the phone, *Google Mobile* provides driving directions from the phone location, and the *iPhone Travelocity* shows hotels near the phone. Their low cost, portability and computational capabilities make smartphones useful for numerous sensing applications in which they act as sensors moving with humans embedded in the built infrastructure. Large scale applications include traffic flow estimation, physical activity monitoring for assisted living at home, geotagging, and population migration tracking [168].

2.1.2. Large scale cyber-physical infrastructure systems

The rapid multiplication of smartphones has contributed to the emergence of a particular class of *Cyber-Physical Systems* (CPS) specific to the built environment: *large scale cyber-physical infrastructure systems*. CPS integrate computational and physical processes, using embedded computers and networks to monitor and control the physical processes, usually with feedback loops where physical processes affect computations and vice versa. Using the motion of humans and goods in the built environment, the notion of cyber-physical systems can be extended to infrastructure such as transportation networks, water distribution networks, the power grid, instrumented bridges, etc. Large scale cyber-physical infrastructure systems thus encompass physical processes related to the infrastructure, such as the modeling of motion of people (“physical”) and the corresponding information gathering, communication and computing system

("cyber"). The *Mobile Millennium* project deals with the specific case of mobile sensing in large scale systems that are spatially distributed.

2.1.3. Historical context in 2007: probes as a contributor to congestion alleviation?

The demand for mobility has dramatically increased, leading to a $78 billion annual drain on the U.S. economy in the form of 4.2 billion lost hours in commute time and 2.9 billion gallons of wasted fuel, which amounts to 58 fully-loaded supertankers. While there is almost no space for additional roads or freeways in the vast majority of urban and suburban environments, there is an enormous potential gain from real-time knowledge of traffic. This information has the potential to enable system solutions such as *ramp metering*, *dynamic speed limits*, and individual level solutions in which drivers can access traffic in real-time, and obtain customized itineraries from their location to avoid major congestion. When this project started, only major freeways were instrumented in the US. The fundamental missing piece of information was that of secondary itineraries including expressways and arterial roads. At the start of this project, traffic information came from fixed (Eulerian) loop detectors in the pavement [203], RFID transponders, radars or cameras [180]. While this information can easily be accessed on the internet [1,10,11,28] and on phones using cell phone versions of these websites, these services only provide information in locations where there are fixed detectors. To provide a global solution to this traffic information gathering problem, one needs traffic information everywhere within the transportation network. Given the high costs of deploying a traffic monitoring system and the lack of public infrastructure, mobile probes provide a feasible alternative. With the notable exception of the data from dedicated fleets [251] such as the police force, taxis, FedEx, UPS (all of which have very limited coverage), such traffic data simply does not exist on a *global scale*.

Starting in 2007, there had been increased levels of competition between cell phone manufacturers, network providers, internet service providers, computer and software manufacturers, and mapping companies. A few recent events are noteworthy in the context of the *Mobile Millennium* project. In particular, Google made a move towards the phone industry with the launch of the open Linux-based Android platform (to become the Google Phone). Because of the pressure to use open platforms, Nokia, who manufactures 40% of the cell phones in the world, bought Symbian (which runs on high end Nokia phones) with the intention of making it open-source (Apple also partially opened its iPhone OS to software developers with the release of a development kit). In the context of geolocalization, Nokia bought Navteq, which is the largest mapping company in the world (and equips Google Maps). Navteq owns Traffic.com, one of the leading internet traffic broadcast companies. Its competitors include Inrix, which is closely linked to Microsoft. In parallel, dedicated car GPS infrastructure companies such as Tom-Tom and Dash also started developing probe vehicle data collection infrastructure from the car aftermarket device angle.

All these companies envisioned a system that could gather *enough probe data* from mobile devices to reconstruct traffic in real-time, while preserving privacy of the users and minimizing GPS use (because of its energy consumption), and to broadcast the data back to the phones and the Internet. The intense competition indicated that companies believed that this technology

would have high value for the public.

Mobile Millennium began in this context, which was the early era of the mobile internet in the US.

2.2. An early instantiation of crowdsourcing and participatory sensing

Crowdsourcing is a recent term, which in its current context is traced back to Jeff Howe [197] in a June 2006 *Wired magazine* article "The Rise of Crowdsourcing." It was recently introduced in the dictionary as "the practice of obtaining needed services, ideas, or content by soliciting contributions from a large group of people and especially from the online community rather than from traditional employees or suppliers" [39]. While it is related to participatory sensing, which does not have a formal dictionary definition, it shares the same principles: collecting data from a set of users working collaboratively [138].

2.2.1. Historical context

Early instantiations of crowdsourcing and participatory sensing include the *Longitude Prize*, a reward offered by the government of Britain for a method for precise determination of a ship's longitude [307]. The prize was created in 1714 by the *Board of Longitude*. Another instantiation in the 19th Century is the *Oxford English Dictionary*, which made an open call to the public for volunteer contributions to index all words in English and provide example quotations for them [337]. In recent years, with the emergence of the Internet and social networking, these initiatives have taken a larger scale and more rapid pace, with numerous high visibility efforts, such as the *DARPA Network Challenge* [31] to collaboratively localize marker balloons deployed in the US, or the *Netflix Prize*, a competition for collaborative filtering algorithms to forecast user ratings for films, based on past ratings [32]. In the era of the mobile internet and with social networking, crowdsourcing efforts now include: location, locally sensed data (speed, acceleration, orientation), opinions, geolocalized posting (pictures, comments, ratings), remotely performed work, personal information, maps (geographic, wireless, activity, crime), among many others.

2.2.2. Issues and trends

There are three current trends that were already visible when the project started, and that are important to consider in order to understand how the Mobile Millennium project was conceived and led.

- <u>Information rich maps.</u> With the convergence of sensing, communication and computation on single cellular platforms, and the ubiquity of the internet and mobile web, the current trends of crowdsourcing and participatory sensing have led to the emergence of information rich maps. Early applications included traffic information collected from

smartphones [94,63], present today in numerous industry led products (Google, INRIX, NAVTEQ, Waze, BeatTheTraffic.com). The concept was soon extended to enriching maps with other user generated content either through location based services or posting from public records, for example crime [41], geolocalized real estate data [45], photographic geolocalized postings [34], pedestrian and sports GPS traces [40], earthquake information [325], and many other types of information.

- Geolocalized human activity. The explosion of location based services has led to the emergence of a new type of user generated content that includes sharing personal information [33], professional information [37], geolocalized social network activity (placing Facebook activity on maps) [38], checked in activities (presence in a restaurant, at a landmark location, etc.) [35]. This new information complements traditional cell tower information (which was already used in an operational context in the case of tracking of al-Zarqawi by the US military [85]), by enriching available feeds by attributes disclosed, knowingly or not, willingly or not, by the user.
- Human work. Finally, the concept of crowdsourcing and participatory sensing were pushed further and finally reached collaborative work. Wikipedia [43] was probably the first such joint effort of considerable size to create a completely crowdsourced encyclopedia on a voluntary basis. It was followed by numerous voluntary "services", in which users volunteer their time, for example Facebook translation [195] or Yahoo! Answers [44], in which users provide answers to other user's questions on a joint online forum. This concept was also made for profit, in the famous Mechanical Turk [3], which created a new work marketplace, in which workers remotely perform tasks at a distributed and large scale for money. This innovation represents a new trend in crowdsourcing data in which the crowdsourced workers are now active and follow directions. Success stories for Mechanical Turk type activities include: tagging, identification, labeling, parsing, clustering, and recognition.

2.2.3. State of the art

Crowdsourcing and participatory sensing are not academic fields (yet). They are a combination of a wide range of disciplines that can be clustered in various categories: (1) remote sensing and hardware (GPS, sensor networks, vision); (2) communication, signal processing and user interfaces (cellular technology, web technologies, geolocalization, mapping); (3) cloud computing (web support, databases, architecture); (4) large scale data analytics (machine learning, data mining, data assimilation, filtering). One of the challenges in the context of geospatial intelligence is that most of the new knowledge required for this field lies *outside* the field of traditional "geospatial" domains.

For these reasons, the training for this field is distributed among Departments and academic programs, which include: Engineering (Aerospace, Civil, Computer Science, Electrical, Environmental, Mechanical), Geography, Urban Planning, and Architecture. Each of these Departments has programs that contribute to the four aforementioned areas. A few research institutes within academia merit to be mentioned, for their span of activities aligned with training in the field. •

- *CENS: Center for Embedded Networked Sensing* at UCLA, one of the first centers

to study participatory sensing and to generate explicit academic contributions in the field.

- *CSAIL: Computer Science and Artificial Intelligence Laboratory* at MIT, by the diversity of its faculty spans most of the field required for crowdsourcing.
- *WINLAB: Wireless Information Network Laboratory* at Rutgers University, focused on numerous aspects of crowdsourcing relevant to this field, in particular privacy. At the time the *Mobile Millennium* project was started, there was no institutionalized effort at UC Berkeley for crowdsourcing or participatory sensing. Thus, our team started working together with WINLAB at Rutgers, who helped us with the privacyaspects of this work. We also worked very closely to Deborah Estrin at UCLA, a visionary figure, who is one of the founders of the concept of participatory sensing, on which her group is currently working, and for which her group established clear leadership in the field. A few years after the start of Mobile Millennium, several new initiatives have started at UC Berkeley to bring these concepts into an academic field, the most notorious one being the AMP Lab (Algorithms, Machines, People), which now collaborates with the Mobile Millennium team on an ongoing basis, and partially funds outgrowths of this effort.

The Mobile Millennium effort has advanced the state of the art in participatory sensing, which is still a field in major expansion, both academically, from a practitioner's perspective, and from an industry standpoint.

2.2.4. Assessment of available information

This very brief section summarizes some of the key data usually available from participatory sensing (in the context of Mobile Millennium and also from a broader perspective). It is by no means complete, for further information, the work of Deborah Estrinand her group [280,297,253,282,281,299,298] is the reference in the field. The aforementioned information available from crowdsourcing today results mainly from data that includes:

- *Sensing data.* From smartphones: GPS, accelerometer, magnetometer, photos, videos, sound. From sensors: any other information connectable to a smartphone (all car sensors through OBD-II, CANBus, biomedical sensors, cellscope, etc.).
- *Other user generated content.* Postings, text, activity, check-ins, tags, email activity, transactions (credit cards, RFID tags, badge-ing, transit schedules), internet activity, social activity, network use (cell, wireless). As will appear through this report, participatory sensing data has numerous issues that make its use challenging. The corresponding information comes with numerous issues resulting from the nature of the data. These include:
- Accuracy, sparsity, reliability, quality, verifiability, spoofability.
- Proprietary / nonproprietary.
- Privacy: disclosed willingly, unwillingly, disclosed knowingly, not knowingly.
- Availability and useability: public or private, obtainable (or not) by subpena.
- Incentivization mechanisms for data collection (volunteer, prizes, salary)

All of these issues were considered in using the data, and are covered in the different chapters

that follow. Many of these issues still constitute open problems, both from an academic perspective, from an industry standpoint, and also for policy.

2.3. Grand challenges of the data and the technology for traffic information systems

The main challenge in using traffic information was (and still is) that no single source of data possesses the four following features: ubiquity, timeliness, accuracy, and reliability. These features, however, were (and are) a prerequisite of efficient use of information for future transportation operation systems. These properties are also required to ensure the value of the information being provided to the public.

2.3.1. Ubiquity

A major enabling technology to offer motorists alternatives to highways is the possibility of using the secondary (arterial) network. This ability relies on information about the state of the arterial network, which is not available in most US cities. Agencies are thus unable to provide recommendations through channels like the radio, changeable message signs, internet, etc. Motorists are therefore reluctant to leave the highways without certainty of finding uncongested alternate routes. Because of the cost of dedicated infrastructure, equipping the secondary network with monitoring infrastructure is not an economically feasible option. Ubiquity also has a temporal meaning; high availability is a necessary feature of a real world system.

2.3.2. Timeliness

The timescales at which traffic congestion evolves are rapid enough that traffic information not received in a timely manner could prove to be useless, and even detrimental, to operations and individual choices in the transportation network. One reason why information currently gathered and available for operations is not always timely is the different layers of filtering, processing, aggregation and broadcast the data go through. In the absence of integrated systems, data gathered from sensors travels from proprietary system to proprietary system, aging a few minutes in each step, which reduces its value when transmitted to the user. Currently measured delays in the process commonly exceed 30 minutes on some traffic information systems, which makes the data simply obsolete.

2.3.3. Accuracy

Accuracy of collected data is critical to proper operations of the transportation system. Accuracy refers to the pointwise error between the system's estimate of congestion and the actual congestion. Misplacement of an accident on the network because of inaccuracy in the measurements can lead to wrong estimates of traffic, which in turn lead to inefficient operations, sometimes with dramatic consequences for users (for example, a single wrong choice in routing through the Bay Area can lead to 200% in commute time difference, based on the bridge used). Misplacement of the edge of extent of congestion can lead to significant differences in the estimate for *total time traveled* (TTT) because of saturation of the mainline of freeways when off

ramps would have provided a better alternative.

2.3.4. Reliability

One of the main issues in the adoption of transit systems, or even in use of travel information systems for more efficient planning, is the reliability of the information. In addition to the accuracy of the estimate, one also needs a confidence interval for the estimate. Ideally, one would like to be able to use the data to make statements of the type "travel time using this mode of transportation along this route is within 10% of this value with 90% confidence". This type of question is critical. For example, schedule adherence in the San Francisco area is estimated to be 70% as reported by the San Francisco Municipal Transportation Authority in 2009. For the past two years, official *Metropolitan Transportation Authority* (MTA) numbers for New York show a system reliability of around 80% for subways and 66% for buses. From a user's perspective, incorrect information from the static transit schedule database can lead to a commute plan that appears feasible, yet is impossible due to missed connections. Similarly, low quality highway traffic information systems are not giving the traveling public incentives to use them.

2.3.5. Assessment: challenges for the creation of traffic information systems

Unless these four challenges are overcome, the creation of efficient transportation information systems for agency operations and users is problematic. Yet, without such information systems, the negative impacts of congestion on economic efficiency cannot be ameliorated by increased operation efficiency. Finally, it does not seem possible to induce paradigm changes in the way the public travels without interfacing traffic information systems with transit information systems properly. In order to provide alternate choices to established patterns, it is key not only to integrate transit more deeply into the transportation network, but also to enable the public to make informed decisions about its use.

2.4. Relation to existing work

The project started at a time when smartphone penetration was increasing at a very rapid pace in the US, and thus focused this aspect of technology principally. It is related to several other efforts ongoing at the same time, which we list next.

This section summarizes several initiatives or projects that were already underway at the time Mobile Millennium was started. This information is available from the web or public sources and is reproduced as such. Mobile Millennium was a natural follow up of the Mobile Century experiment (described last in this section). While the concept of monitoring traffic from smartphones is similar to some of the concepts inherent to VII, V2V, and other efforts, Mobile Millennium, by its nature, was unique in that it tried to capture the rise of a new technology (smartphones) and thus leapfrog several efforts that were based on other types of technology. These efforts are summarized below for completeness of the context in which this work was undertaken.

2.4.1. Vehicle Infrastructure Integration (VII)

Vehicle Infrastructure Integration (VII) is an initiative fostering research and applications development for a series of technologies directly linking road vehicles to their physical surroundings, first and foremost in order to improve road safety. The technology draws on several disciplines, including transport engineering, electrical engineering, automotive engineering, and computer science. VII specifically covers road transport although similar technologies are in place or under development for other modes of transport. Planes, for example, use ground-based beacons for automated guidance, allowing the autopilot to fly the plane without human intervention. In highway engineering, improving the safety of a roadway can enhance overall efficiency. VII targets improvements in both safety and efficiency. Vehicle infrastructure integration is a branch of engineering that deals with the study and application of a series of techniques directly linking road vehicles to their physical surroundings in order to improve road safety.

UC Berkeley already had significant experience with VII, through the PATH program. Although the topic of VII is related to cellular based sensing and traffic monitoring, early on in the project it was decided to focus on cellular technology only. The effort was kept separated from other VII related efforts on the UC Berkeley campus, in particular from the networked traveler project, which built directly on the VII concept.

2.4.2. V2V networks

Vehicular Communication Systems are an emerging type of network in which vehicles and roadside units are the communicating nodes–providing each other with information, such as safety warnings and traffic information. Vehicular communication systems can be more effective in avoiding accidents and traffic congestion when pursued as a cooperative approach, rather than if each vehicle individually attempts to solve these problems.

Vehicular networks are generally considered to contain two types of nodes: vehicles and roadside stations. Both use Dedicated Short Range Communications (DSRC) devices. DSRC works in the 5.9 GHz band with a bandwidth of 75 MHz and an approximate range of 1000m. The network should support both private data communications and public (mainly safety) communications but higher priority is given to public communications. Vehicular communications is usually developed as a part of Intelligent Transport Systems (ITS). ITS seeks to achieve safety and productivity through intelligent transportation that integrates communication between mobile and fixed nodes. Toward this end ITS relies heavily on wired and wireless communications.

For the same reason as for the VII effort, it was decided early on that Mobile Millennium would not rely on DSRC radios, and thus the effort was separated from efforts involving wireless communications also happening at Berkeley, so it could focus solely on cellular device technology.

2.4.3. The I95 corridor coalition

The I-95 Corridor Coalition is an alliance of transportation agencies, toll authorities, and related organizations, including public safety, from the State of Maine to the State of Florida, with affiliate members in Canada. The Coalition provides a forum for key decision and policy makers to address transportation management and operations issues of common interest. This volunteer, consensus-driven organization enables its myriad state, local and regional member agencies to work together to improve transportation system performance far more than they could working individually. The Coalition has successfully served as a model for multi-state/jurisdictional interagency cooperation and coordination for over a decade.

The Coalition began in the early 1990's as an informal group of transportation professionals working together to more effectively manage major highway incidents that impacted travel across jurisdictional boundaries. In 1993, the Coalition was formally established to enhance transportation mobility, safety, and efficiency in the region.

During the 1990's, the focus of the Coalition's program evolved from studying and testing intelligent transportation systems (ITS) technologies to a broader perspective that embraced integrated deployments and coordinated operations. The Coalition's perspective evolved from a concentration on highways to one that encompasses all modes of travel and focuses on the efficient transfer of people and goods between modes. Facilitation of regional incident management in areas such as pre-planning, coordination and communication among transportation and public safety agencies in the corridor remains a key part of the Coalition's focus. Today, the Coalition emphasizes information management as the underpinning of seamless operations across jurisdictions and modes.

At the time of the start of Mobile Millennium, the work of the I95 coalition was inspirational to our team, since it was heading towards providing early instantiations of traffic information systems, based on partnerships, including a partnership with INRIX. The approach followed by Mobile Millennium was related to the approach followed by the I95 coalition, with significant differences, in particular the initial focus on cell phone data, and the need to create arterial models.

2.4.4. Cellular tower information

Several companies had claimed prior to 2007 that traffic information could be reconstructed with cell tower information (i.e. using information from the towers as opposed to GPS information directly readable from cellular devices). The effectiveness of such a method had been (and probably still is) the subject of ongoing debate. At the time of the start of Mobile Millennium, one of the goals of the project was specifically to demonstrate the differences in reconstructing traffic from GPS data, vs. from cell tower data. Among the most prominent companies working in this area at the time was Airsage. AirSage, Inc. is a nationwide provider of traffic, location, and movement data. AirSage calculates real-time traffic speeds and locations using mobile phone signaling data from wireless service carriers including Sprint Nextel and Verizon Wireless. The company is headquartered in Atlanta, Georgia. AirSage collects and analyzes mobile phone signaling data to determine phone locations and calculate traffic flow for over 220,000 centerline

miles of roadway in the United States including interstates, highways, and arterials. Cellular networks use geographically dispersed cell towers to provide wide-area radio coverage. Signaling data contains information about the network connection including data about signal strength between the transceiver and the phone, as well as round trip delay times. With this data, AirSage uses a combination of positioning techniques such as signal strength multilateration and triangulation, to generate real-time phone location probabilities. They associate those changing locations over time with Geographic Information System data to generate route segment speed estimates.

Several years after the start of the Mobile Millennium, the industry still has not seen any global solution for traffic emerge from Airsage, so the debate of the validity of the cell tower data (vs. GPS) still continues.

2.4.5. Mobile Century

On February 8, 2008, CCIT, Caltrans, Nokia, and UC Berkeley's Department of Civil and Environmental Engineering collaborated to conduct an unprecedented experiment in the area of traffic monitoring. Mobile Century was intended as a proof of concept. The event was enormously successful, both technically and logistically.

The goal of this controlled field experiment was to test traffic data collection from GPS-equipped cell phones driving on a stretch of a highway located in the San Francisco Bay Area. One hundred vehicles carrying the GPS-enabled Nokia N95 drove along a 10-mile stretch of I-880 from 9:30am to 6:30pm.

As described in the final report for Mobile Century [269], the principal objectives for this experiment were to feature online, real-time data processing; privacy-preservation; and data efficiency, i.e. not requiring excessive cellular network load. The sheer scale of the experiment required significant logistical effort. A base station was erected at Union Landing, to house a temporary control center. Over one hundred graduate students from UC Berkeley were employed to circulate in loops along Interstate 880 between Hayward and Fremont, California, for an entire day. During the experimental deployment, an average penetration rate of probe vehicles was sustained near 2% (a significant logistical feat), which is viewed as realistic in the near future considering the increasing penetration of GPS-enabled cellular devices.

Classical methods of traffic modeling operate in the density domain, and use data such as occupancies and flows from inductive loop detectors. Understanding how to use velocity measurements instead was a significant technical contribution. In this work, the classical model was converted to the velocity domain, and GPS-based measurements were directly fed into the model.

Mobile Century proved that data from GPS-enabled mobile phones alone were sufficient to infer traffic features, i.e., to construct an accurate velocity map over time and space. The methods employed were able to function properly during both congested and free flow traffic conditions, and to detect correctly a traffic incident that occurred during the deployment.

3. A rethinking of traffic information systems

Traffic information systems have numerous challenging requirements for them to be useful on a large scale. Among these requirements, traffic information must be accurate and available in real-time, while also easily interpreted by the user. This implies that the system needs to be built using redundant servers to handle the "always available" requirement and also requires substantial computing power to estimate traffic conditions across the entire road network (which can be hundreds of thousands of links within a geographic area). Furthermore, the system must be able to visualize traffic conditions in a way that users can understand the information they need quickly. This invariably includes color-coded maps indicating congestion, but also must include travel time information between arbitrary points on the network, vehicle density estimates for traffic management centers (and potentially air pollution estimation models), and in-vehicle navigation systems with real-time routing. All of these components need to interact through a common infrastructure and be able to provide information to any other part of the system quickly.

These requirements lead to a number of practical design issues that need to be examined in detail to understand how the entire traffic information system works as a whole. System design was a major part of the research done and provides an essential basis for the more theoretical traffic estimation theory found later chapters. This chapter presents many of the core features of a prototypical traffic information system and details the design decisions that must be made to satisfy the global requirements as specified above. These features include privacy, data accuracy assessment, scalability, road network representation, map matching, visualization, and sensor deployment (all included in section 2).

The first part of this chapter (section 1) presents the numerous sources of traffic data that exist in the world today as well as how ubiquitous and reliable these sources of data are for performing real-time traffic estimation on the arterial network. All of these sources of data come with their own particularities in terms of coverage, accuracy, and timeliness. A good traffic information system is capable of using all available data sources, determining their relative merits and then estimating traffic conditions. The use of many (but not all) of these data sources will explicitly appear in later chapters.

3.1. Taxonomy of Traffic Sensor Types

A number of sensors have been developed in the past 50 years designed to collect various types of traffic data. In general, traffic data includes flows (number of vehicles per time unit), density (number of vehicles per distance unit), occupancy (percentage of time a vehicle is over of specific location, which is directly related to density), velocity (distance per unit time), and travel time (time to travel between two locations). One additional data type possible are vehicle trajectories, which are always represented by a sequence of discrete time/location pairs for each vehicle. From vehicle trajectory data with a location-reporting frequency of several seconds or less, travel times and short distance velocities can be directly computed. When the location-reporting frequency is more than 10 seconds, directly measuring travel times and velocities becomes non-trivial. The mathematical details of these data types will be discussed in more detail at the beginning of the literature review in chapter 15.

The remainder of this section lists the most ubiquitous traffic sensors and describes the data types that each of them provides. This includes a discussion of the accuracy, timeliness, and spatial resolution of the data provided by each sensor type. Also presented are typical placement strategies and common road types that are covered by each sensor.

3.1.1. Loop Detectors

Inductive loop detectors are built into the roadway so that they can detect each vehicle that passes over them. They work by detecting the metal of a vehicle as it passes over the detector. Properly calibrated, a loop detector is capable of providing high-accuracy flow and occupancy data [93], the latter of which can be used to infer density [202]. When two loop detectors are placed close together, velocity can be measured by looking at consecutive crossing times. While the quality of the measurements from loop detectors is often good, filtering is still required from producing quality input data to highway estimation models [106]. Loop detectors are not capable of directly measuring travel times.

Loop detectors are commonly found on most major highways throughout the United States and Europe. Many of these locations have loop detectors connected to an internet connection that can be used to transmit the data to a central server in real-time (that can subsequently be used in traffic information systems). Many locations throughout the United States and Europe also have loop detectors placed on arterial roads. However, for arterial roads, it is very rare for the loop detector to be connected to the internet for easy transmission of the data to a central server. For this reason, arterial estimation algorithms cannot rely on loop detector data as there is not enough of it to estimate conditions on the whole network.

3.1.2. Radar

Radar detectors can be placed on poles along the side of the road enabling them to collect flow, occupancy and velocity data. In general, radar detectors provide lower accuracy data than loop

detectors [242].

As of this writing, dedicated radar detectors that are connected to the internet and providing data in real-time are still relatively rare in the United States and Europe. Where these are available, they are placed almost exclusively on highways. Radars are generally not well suited to mass data collection on arterials due to the fact that accuracy decreases in arterial environments [242]. For this reason and the fact that almost no radar data exists on arterials, they are not considered viable inputs for arterial estimation algorithms.

3.1.3. Video

Video recording can be used to collect traffic data in two ways. The first way is to use high resolution cameras placed high above the roadway to track all vehicles within the view of the camera. The second way is to use the video cameras to record license plate numbers at specified locations, which is equivalent to using video as a license plate reader (see section 4 for further description of this kind of data collection).

Using high-resolution cameras to track vehicle trajectories does not provide data in real-time due to the large amount of post-processing work that needs to be done on the images to turn them into actual vehicle trajectory data [242]. When properly processed, video can provided very high-resolution vehicle trajectories (vehicle positions every tenth of a second). However, this technology is expensive to deploy and can only cover a relatively small portion of the roadway (generally less than a mile). The NGSIM project [18] is an example of the use of this kind of technology, which to date has mostly been used to provide researchers with high-accuracy vehicle trajectories over a small spatio-temporal domain (less than a mile for less than an hour). This kind of data is valuable to arterial traffic estimation research, but given that the data does not come in real-time, it cannot be used in real-time traffic information systems.

3.1.4. License Plate Readers

When placed directly above a lane of traffic, license plate readers are capable of automatically extracting the numbers and letters from passing vehicles. When multiple readers are setup at two points along the road, it is possible to extract travel time information for vehicles passing both locations.

License plate readers suffer from the logistical problem of finding good locations to place them so that they can be used effectively. When properly positioned and calibrated, these devices are capable of providing high-accuracy travel times [6]. However, even when the devices correctly measure individual vehicle travel times, one still needs to filter the travel times to account for vehicles that stop in the middle of the route between the two sensors. In fact, this need to filter travel times arises whenever collecting travel times by placing two sensors capable of re-identifying vehicles (but that do not track the vehicle in between). One particular filtering strategy is the *Median Absolute Deviation* filter, described in [66].

Due to the difficulty in placing these devices, they are not common throughout the roadway. As of this writing, they remain a data collection tool for specific studies, but not for large-scale

traffic data collection. This makes them unusable for traffic information systems.

3.1.5. RFID Transponders

Radio-Frequency Identification (RFID) is a ubiquitous technology in many industries. Transit agencies make use of RFID in several ways. One of the original uses of this technology was for collecting tolls from drivers when crossing a bridge or entering/exiting a toll road. The vehicle has a RFID transponder which is detected by a reader placed at the entrance/exit of the toll road [66].

This same technology can be used for traffic data collection by placing readers at various points along the roadway. Travel times can be collected between pairs of points and processed in the same way that travel times from license plate readers can be processed (see section 4). The accuracy of RFID transponders varies depending on the strength of the signal. It is generally accurate enough to provide long distance travel time estimates, but may not provide high-accuracy travel times over short distances. RFID readers are generally placed far apart from each other in current deployments, making them useful for collecting long distance travel time information, but not for providing input data to detailed traffic estimation algorithms. They are placed almost exclusively on highways, making it uncommon to find this technology on arterial roads.

3.1.6. Bluetooth

Bluetooth readers have been developed in recent years [334], which are capable of scanning the surrounding airwaves for Bluetooth enabled devices. If readers are placed at points along the roadway, travel times can then be measured between consecutive readers for all vehicles carrying a Bluetooth device. In addition to the filtering challenges associated with these travel time measurements described in section 4, Bluetooth readers also suffer from the problem of having a relatively high detection range. This is a good thing in the sense that the readers rarely miss detecting a vehicle, but bad in the sense that it is difficult to determine the precise time that a vehicle passed the reader as it might be detected continuously for more than a minute.

Bluetooth readers are an emerging technology and are therefore not available in most areas. If they were available in large quantities on arterials, they could potentially be used for traffic estimation, but due to the general lack of sensors of this type, they are not considered viable data providers for traffic information systems at the current time. Also, it is likely that enhancement algorithms will be developed that will correct for the inaccuracies in the data provided by these devices.

3.1.7. Wireless Sensors

Wireless sensors are small devices embedded into the roadway (similar to loop detectors), but the detection mechanism in these sensors allows for re-identifying vehicles at subsequent sensor locations with up to 80% accuracy for one particular system [217]. Thus, these sensors provide travel times for a large percentage of the flow of traffic (and the travel times must still be filtered

as discussed in section 4). The primary advantage of these sensors over other travel time measurement sensors is that they are much cheaper to produce, potentially allowing for large-scale deployment on arterial roads. However, at the current time, they are only available in a small number of locations. Sensys Networks [26] is currently one of the leading providers of these sensors. In general, these devices have been used on arterials and not on highways.

3.1.8. Virtual Trip Lines (VTL)

Virtual trip lines (VTL) comprise the basis of a "participatory sensing" system that allows individuals to download an application onto their GPS-enabled smartphone that both sends traffic data as well as receives traffic information and alerts. A VTL is a virtual line drawn on the road. The basic idea is that the phone monitors its own GPS position every few seconds and has downloaded a list of VTLs in the general region that the phone resides in. When the phone crosses one of the VTLs, it sends an update to the central VTL server indicating its velocity and time of crossing as well as the travel time from the previous VTL it crossed. The accuracy of the velocity data generated by frequent GPS sampling varies greatly with the type of GPS chip in the phone and can be very good in some cases and very bad in others. It is generally accurate enough for highway traffic estimation when properly filtered [339]. For arterials, the velocity measurements are not reliable, so the travel time measurements are the only data that is suitable for arterial traffic estimation. The travel time measurements also need to be filtered as described in section 4. For highways, the velocity measurements are used.

Nokia [20] originally developed the first VTL-based traffic data collection system in 2007. This system was first tested as part of the *Mobile Century* experiment in February, 2008. The results of this initial experiment can be found in [181].

VTLs are capable of providing high-quality traffic data while also helping to preserve the privacy of individuals by only disclosing data at pre-specified locations [190]. One challenge is determining where the VTLs should be placed throughout the network so as to collect relevant traffic data while not infringing individual privacy or placing VTLs so densely that an unnecessary amount of data is transmitted through the communication network. No studies have been conducted to date on proper VTL placement for addressing these issues. An experimental deployment of VTLs was tested as part of the *Mobile Millennium* project, covering all of California as seen in figure 1.1 for some parts of the Bay Area.

Figure 1.1. Example of a Bay Area VTL deployment as part of the *Mobile Millennium* system.

3.1.9. Sparsely-sampled GPS

Sparsely-sampled probe GPS data refers to the case where probe vehicles send their current GPS location at a fixed frequency, which is not frequent enough to directly measure velocities or link travel times (i.e. sampling frequency is more than about 10 seconds). There are several challenges associated with this type of data. First, GPS measurements must be mapped to the road network representation used by the traffic information system, which means that the correct position on the road as well as the path in between successive measurements must be determined. This process is known as map matching and path inference, which is described in more detail in section 5. Second, probe vehicles can often travel multiple links between measurements when the sampling frequency is low, which means that one must infer what the likely travel times on each link of the path were.

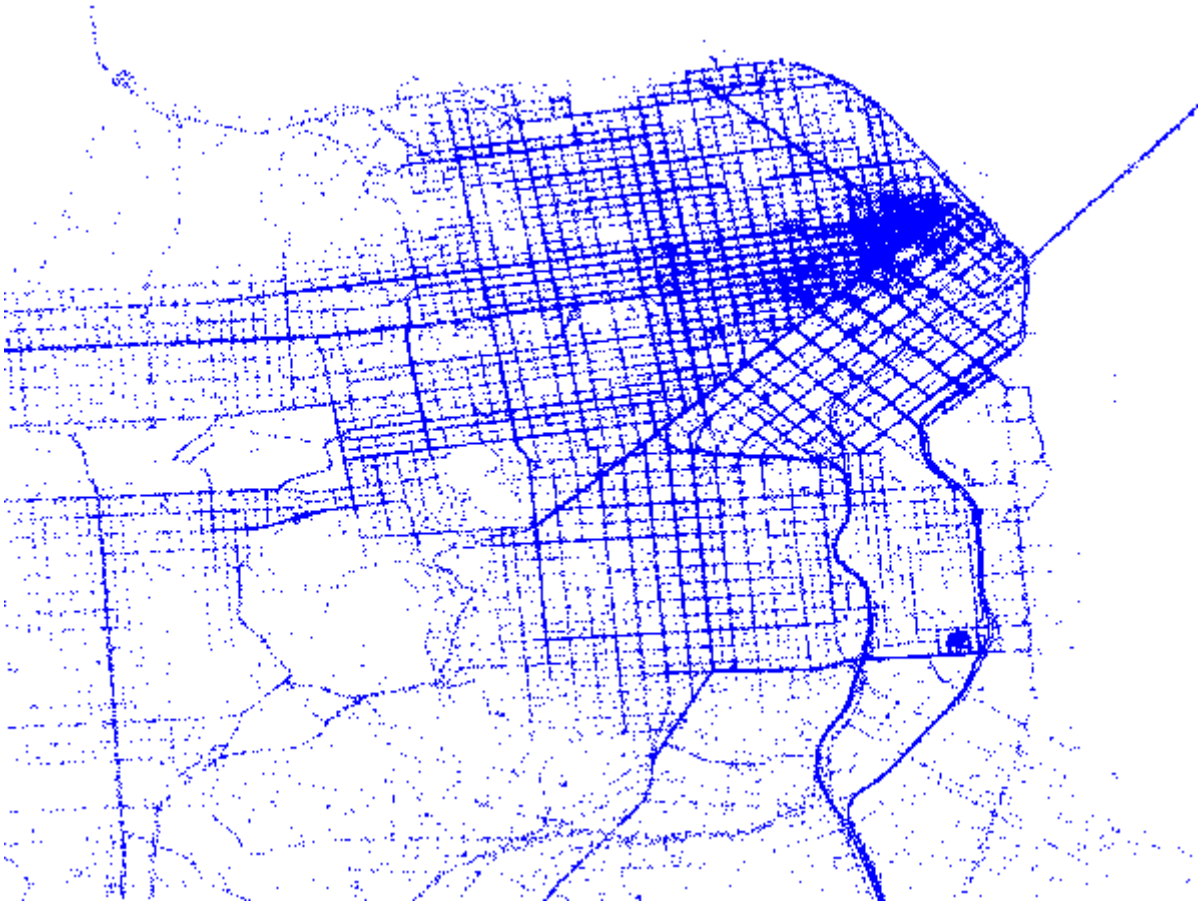


Figure 1.2. One day of sparsely-sampled GPS data from San Francisco taxi drivers as provided by the Cabspotting project.

Sparsely-sampled probe GPS data is currently the most ubiquitous data source on the arterial network. An example of this type of data comes from the Cabspotting project [4], which provides the positions of 500 taxis in the Bay Area approximately once per minute. Figure 1.2 shows one full day of raw data, which demonstrates that even just a single data source such as taxis can provide broad coverage of a city. This data clearly has some privacy issues as it is possible to track the general path of the vehicle. However, the majority of this data today comes from fleets of various sorts (such as UPS, FedEx, taxis, etc.). Most of this data is privately held among several companies, but between all sources there are millions of records per day in many major urban markets. One publicly available source of this kind of data is the Cabspotting project [4]. This project provides one-minute samples of the positions of over 500 taxis in San Francisco, CA. This results in upwards of 500,000 measurements per day. Due to the ubiquity of this data source, it is paramount that it be used in an arterial traffic information system. Indeed, it is the only source that is likely to be available across the arterial network in the next decade.

3.1.10. High-frequency GPS

High-frequency probe GPS data refers to the case where probe vehicles send their current GPS location every few seconds (no more than about every 10 seconds). This kind of data is

generally the most accurate kind of vehicle trajectory data possible, especially when sampling every second with a high-quality GPS chip. From this data, one can directly infer velocities and short distance travel times. The issue of map matching is still present as there can be ambiguity around intersections, but the path is usually easy to determine when examining the entire trace. Figure 1.3 depicts a sample of high-frequency data collected as part of the *Mobile Millennium* project. This figure illustrates the level of detail that can be extracted from high-frequency data, but also shows the relatively low percentage of vehicles that were being tracked as there are occasional gaps of five minutes or more between trajectories.

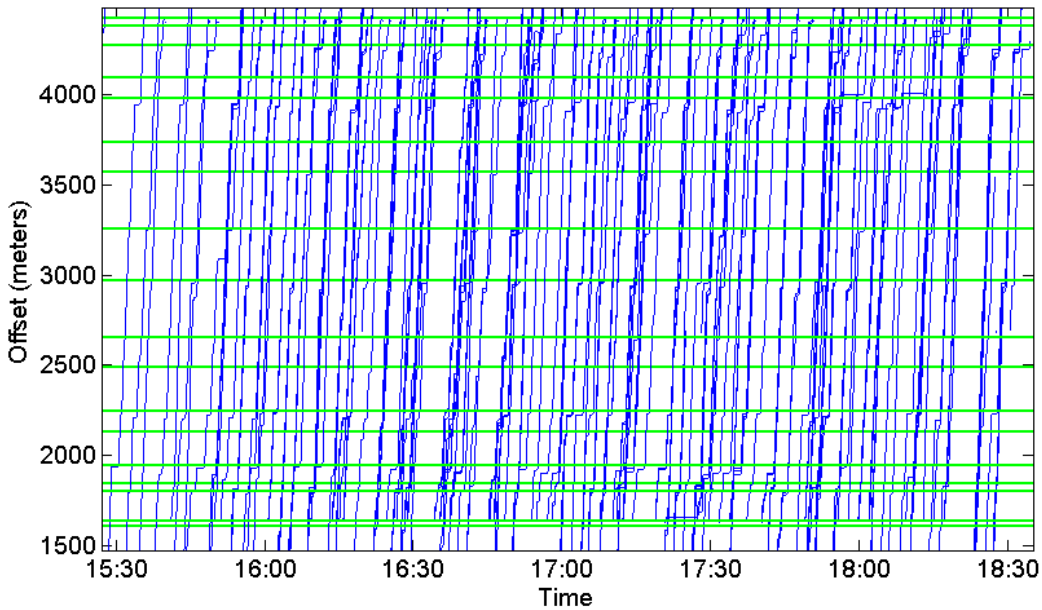


Figure 1.3. Vehicle trajectories from the *Mobile Millennium* evaluation experiment on San Pablo Avenue in Berkeley, Albany and El Cerrito, California. The high-frequency GPS data in this figure is represented as distance (meters) from an arbitrary start point upstream of the experiment location. The horizontal lines represent the locations of the traffic signals along the route.

Sampling a vehicle's position every few seconds is clearly very privacy invasive and it also comes with large communication costs to send the high volume of data. For these reasons, it is not common to receive this data with any kind of regularity. This data is often collected for specific experimental studies, but is not generally available for real-time traffic information systems.

3.2. Practical Considerations for Designing a Traffic Estimation System

In this section, the core issues of practical importance to users of traffic information systems are discussed in detail. This work contributed to the *Mobile Millennium* system by designing and writing software that addressed all of these issues (often with other members of the team). In particular, the network abstraction and map matching functionalities of the system were primarily designed and implemented using this work, and they have been relied upon by more than 20 members of the *Mobile Millennium* team.

3.2.1. Driver Privacy

The expectations of drivers with respect to the privacy of their location measurements varies greatly and is also a generational problem. Some drivers will not be comfortable sharing any data at all, some will be willing to share some data in exchange for value of some kind (real-time routing around traffic, for example), and some would be willing to share any and all data as long as it does not interfere with their ability to use their phone or other GPS device. The *Mobile Millennium* system was designed to accommodate all of these privacy preferences. The system is "opt-in", meaning that drivers who do not wish to provide any data can simply choose to not install the application running on the phone that collects traffic data. For those who wish to participate and receive traffic information on their phone, a spatially-aware sampling system (based on VTLs) was designed to extract information without compromising user anonymity [190]. A technical description of how VTLs work can be found in the sensor taxonomy portion of this chapter (section 8).

There are two primary reasons why VTLs respect driver privacy more than fixed-interval location reporting. First, driver data is collected only at pre-defined locations which only include highways and major arterials, but not residential roads. Second, driver data collected at one location is not re-associated with that same driver's data at another location, except for short distance travel times. These two features combined mean that origin and destination information is unavailable for anyone who has access to the data collected inside the system.

GPS tracking data (both sparse, section 9, and high-frequency, section 10) is not intended to preserve the privacy of the driver. The *Mobile Millennium* system collects this data specifically from sources who have agreed to provide it in that form and are not concerned with privacy of the drivers (the primary function of the data is generally to track service vehicles). This is generally restricted to fleet delivery vehicles or taxis, but if an individual driver wanted to participate in this manner, the data can be collected in that form.

3.2.2. Raw Data Accuracy and Filtering

No data source is perfect and every piece of data received by the *Mobile Millennium* system goes through a specific filtering process. Data from fixed-location sensors generally requires a much different filtering process than GPS data. In the *Mobile Millennium* system today, fixed-location

sensor data is only available on the highways and particular filtering algorithms have been developed specifically for highway traffic estimation algorithms. The basic idea behind fixed-location sensor filtering is to correct the values being reported by the sensors to account for the noise in the measurements. GPS data is the only available data on arterial roads in the *Mobile Millennium* system and this data requires both map matching and path inference (see section 5). These processes represent a different notion of filtering, where instead of correcting values, the data is actually being translated from one spatial reference system (GPS position on the Earth) to another (link identifiers and position along the link using a network representation of the road).

The purpose of mentioning issues of data quality and filtering here is to highlight the importance of the data cleaning process to the overall goal of building a high-quality traffic information system. Furthermore, it is important to have analytical measures for how accurate a particular data source is. The *Mobile Millennium* system provides a valuable framework for comparing data from multiple sources and validating the accuracy of those sources. For example, one way to validate the filtering of sparse GPS data is through the use of high-precision, high-frequency GPS devices, which have been used by drivers hired by the project and for which the correct path can be determined with certainty. By down-sampling this high accuracy data to the level typically received from sparse GPS sources, the reconstruction of the path from the sparse data can be compared with the true known path.

3.2.3. Scalability

Traffic systems are required to produce estimates across the entire network, continually updating themselves by processing thousands of new data records every few seconds. If scalability is not taken into account in the system design, it is quite likely that the system will not be able to keep up with all of the streaming data in real-time. The *Mobile Millennium* system was designed to be modular, so that each component of the system can run independently, potentially on its own server. This means that the various processes (from collecting raw data, filtering, running the estimation algorithms, disseminating estimations to third parties, visualization, validation, monitoring, etc.) can be divided up among all of the server resources available.

Beyond the modular design, it is also important to design estimation algorithms that scale in a reasonable way with the size of the raw data and the size of the network. In fact, a natural way of modeling the arterial network from processing sparse probedata requires solving a non-linear network optimization problem that grows quadratically in the number of links in the network. Given that a typical large city (such as San Francisco) may have on the order of 2,500 links, solving this optimization problem in real-time quickly becomes impractical. For this reason, the algorithms presented later (in chapters 18 and 19) are designed so that the optimization problem is linear in the number of links and therefore computationally tractable.

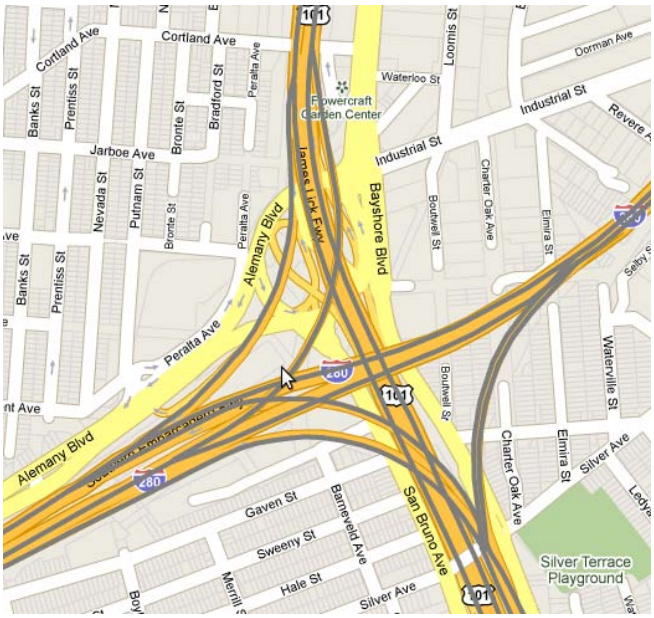
3.2.4. Network Abstraction

The most fundamental, core piece of any traffic information system is the digital representation of the road network. All data is associated with some location and needs to be mapped precisely. Estimation models need precise information about the geometry and physical characteristics of the road network. Separate components of the system need to be able to communicate

information about location information in a universally consistent manner. For these reasons, it is critical to build the traffic system starting with a good digital map of the roadway. A digital map is constructed using the common graph theory notions of *nodes* and *links*. A link is the stretch of road between two nodes and a node represents the intersection of multiple links. The digital map must include the geometry (i.e. the latitude/longitude coordinates) of each link and will generally contain a number of road attributes, such as the number of lanes, the speed limit, the road type (such as highway, arterial, or ramp), the name of the road, etc.

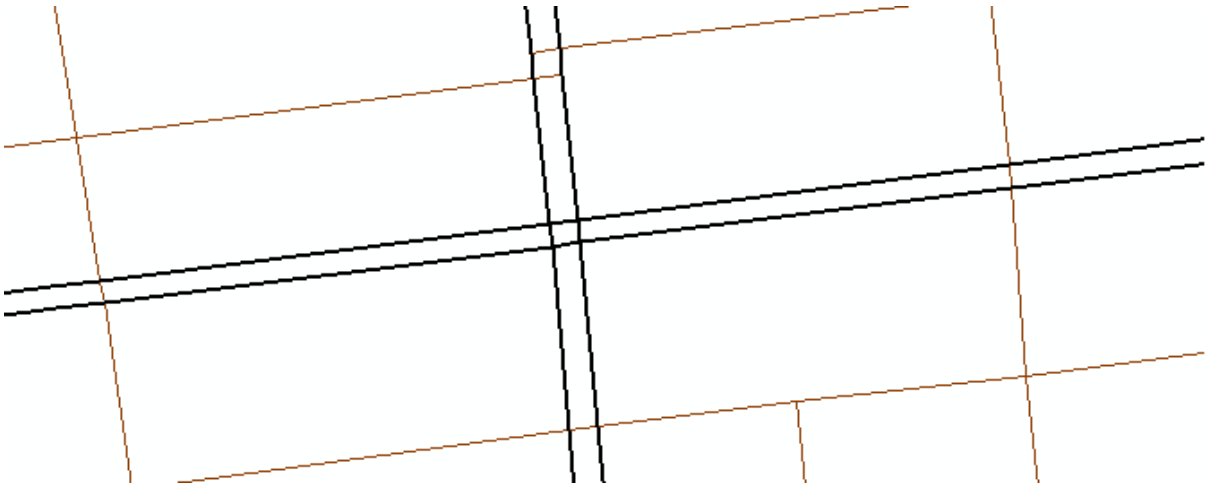
The *Mobile Millennium* system was built using Navteq maps [16] as the underlying representation of the road network. Navteq maps (see figures 2.1 and 2.2) were made available to the *Mobile Millennium* team as part of the Safe Trip21 project, which was funded by the United States Department of Transportation [29] and the California Department of Transportation [5]. Navteq maps provide detailed geometry and numerous road attributes per link (over 100). Traffic models generally assume a directed graph representation of the network and the Navteq map goes beyond this in terms of the complexity of its digital representation. Figure 4 illustrates the level of complexity around an interchange in San Francisco, CA. For highway traffic estimation algorithms, the key pieces of information are simply the points where roads merge or diverge (the nodes of the directed graph) as well as where the on-ramps and off-ramps allow for entering and exiting the highway. The added detail in geometry for the ramps and approaches to the highway add unnecessary complexity to the model and it is therefore better to remove those. The result of the network abstraction algorithm for this part of the highway is shown in figure 4.

(a) Navteq representation of the interchange. The black indicates roads that Navteq designates as part of the highway. The red indicates residential streets.

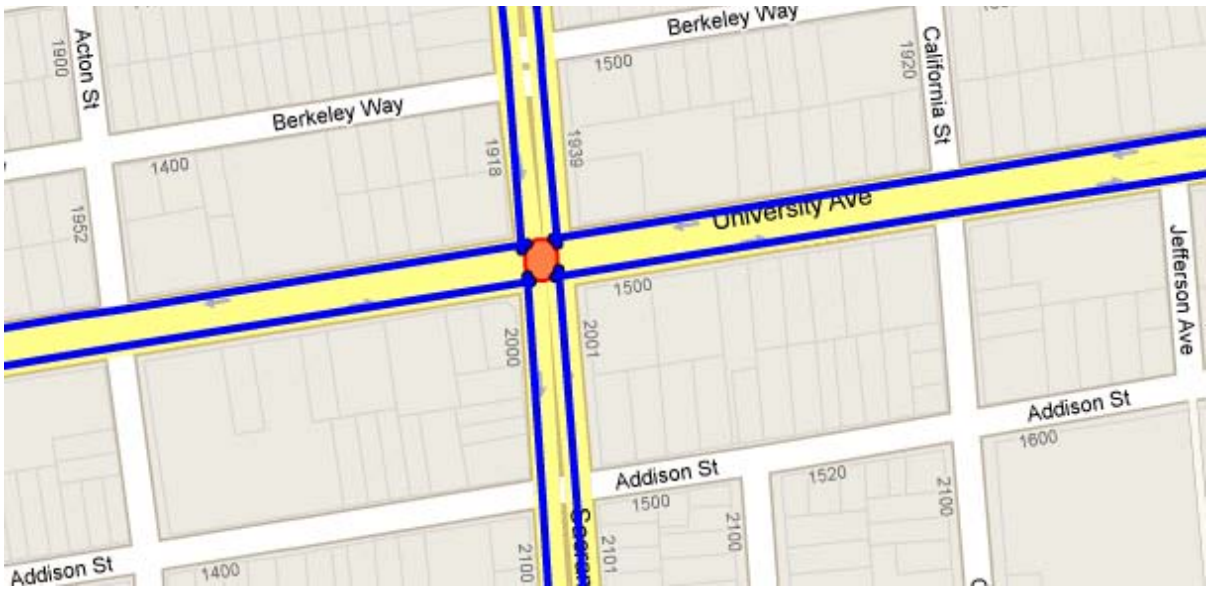


(b) Simplified representation (in gray) of the interchange for traffic estimation algorithms overlayed on a Google map of the area. Although not pictured, the estimation algorithms are aware of the incoming and outgoing ramps, allowing them to account for incoming and exiting traffic while not worrying about modeling the ramp traffic conditions explicitly.

Figure 2.1: Navteq (a) and simplified (b) representations of the I-280, highway 101 inter-change in San Francisco, CA



(a) Highway representations of the I280, highway 101 interchange in San Francisco, CA Navteq representation of the intersection. Note that the four small links forming a square in the middle are all approximately 10 meters long each and just represent the distance from one side of the intersection to the other



(b) Simplified representation of the intersection. The small links have been replaced by an intersection object that has a positive area. All of the connecting links connect only to this one intersection object instead of to each of the "short" links.

Figure 2.2: Navteq (a) and simplified (b) representations of an arterial intersection in Berkeley, CA. The intersection is represented by 4 "short" links in the Navteq database, but for traffic estimation it is more appropriate to have a single intersection object.

Arterial networks experience a different type of problem with the road network representation,

which is illustrated in figure 2.1. The key issue here is that Navteq represents nodes of the graph by a single GPS point with no area and they also often represent each direction of traffic by a separate link. When two roads intersect as in the figure, four “short” links are created a result of the two links for each direction intersecting. These four links are not “real” links, but rather just artifacts of their choice of how to draw the road network. Traffic estimation models assume that this is just one intersection and should be represented as a single node in the graph with all links connecting to it. The result of the network abstraction algorithm is presented in figure 2.1. This simplified representation helps for both the traffic estimation algorithms as well as the map matching and path inference filters.

The final component of the network abstraction algorithm for both highways and arterials is that of link selection and merging. Link selection simply refers to the fact that not all roads should be considered for traffic estimation, particularly residential roads. For highway networks, all nodes that have exactly one incoming and one outgoing link are removed since these are unnecessary for traffic estimation (these nodes often exist in the original map to denote the presence of a physical sign along the side of the road). For arterial networks, traffic estimation algorithms operate on links defined by the stretch of road between signalized intersections (or intersections with stop signs). When a node has only one incoming and one outgoing link and there is no traffic signal or stop sign at this node, then the links are merged together.

The result of this network abstraction procedure is a unified representation of the roadway that all components of the system can use to communicate location information. It also allows for intuitive visualization of the output of the traffic estimation models.

3.2.5. Map Matching and Path Inference

Probe data from GPS devices is often very accurate and easy to place on the digital map of the road network. However, there is enough noise in the data that there are several situations that frequently arise that make directly inferring the correct mapping difficult. One situation that is difficult to deal with on highways is when a frontage road is very close to the highway. In this situation, it can be difficult to distinguish which of the two roads the driver was on. Another difficult situation to deal with on arterials is when an observation occurs directly in the middle of an intersection (like that in figure 2.2). Furthermore, when a vehicle is transmitting its GPS position infrequently, the number of turns made between measurement locations could make it difficult to determine the correct path taken between successive measurements.

To address these difficulties, the *Mobile Millennium* system developed an algorithm that simultaneously performs map matching and path inference for both sparsely-sampled and high-frequency GPS probe data. The map matching component is performed using a spatial database capable of performing *spatially-indexed* queries (which are performed by a PostgreSQL database inside the *Mobile Millennium* system [24]). This speeds up the map matching process by several orders of magnitude by localizing the search for possible mappings to the set of nearby links. Several possible mappings are returned by the first stage of the map matching procedure. The second stage of the algorithm looks at all of the realistic paths between pairs of GPS measurements and determines the most probable path (which includes the most probable mappings on each end) [199]. Figure 2.3 shows a small sample of GPS points (hollow circles)

along with the inferred mapping and traveled path.

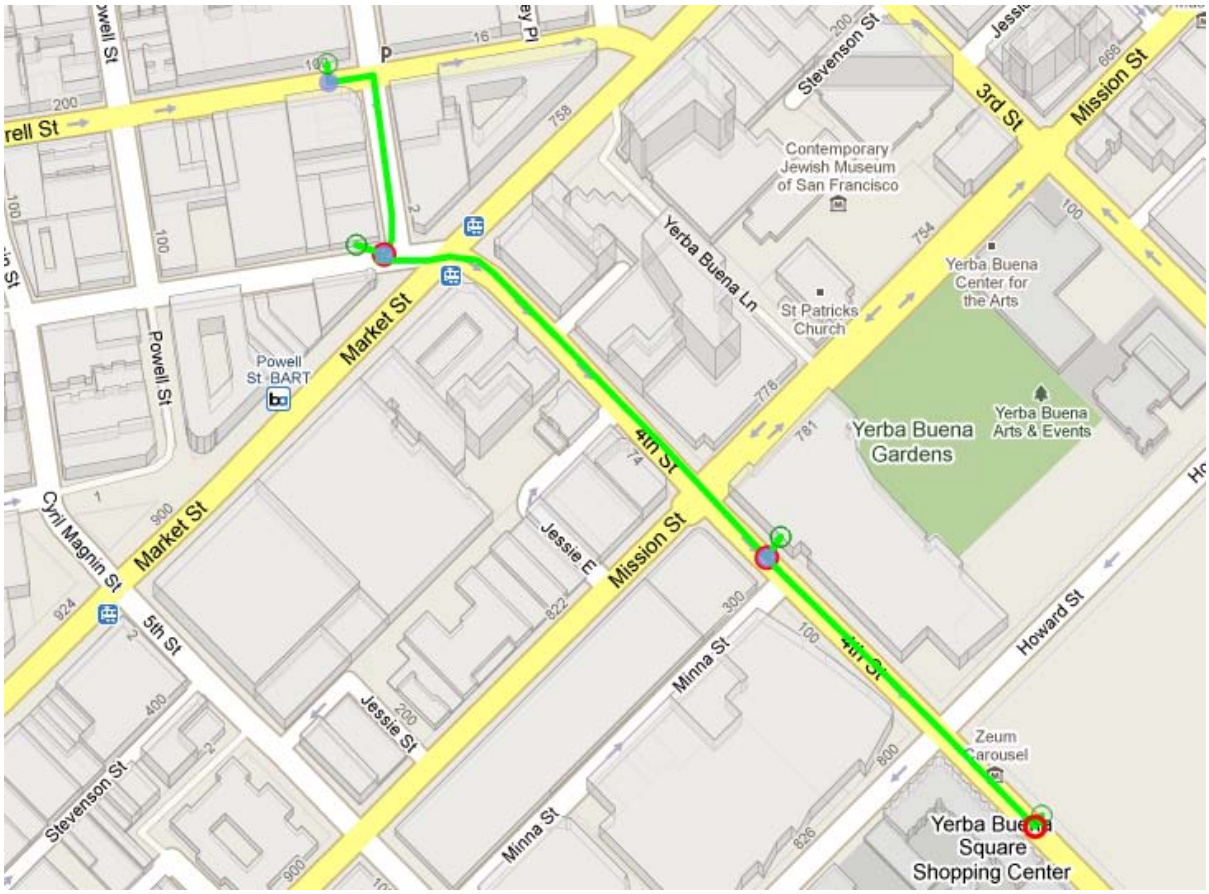


Figure 2.3. An illustration of the *Mobile Millennium* map matching and path inference algorithm. The hollow circles represent the GPS measurement locations. The blue/red circles represent the start/end of a pair of GPS points as mapped to the road. The green lines indicate the inferred path traveled by the probe vehicle.

Fixed-location sensors also require map matching, although the task is generally much easier than for GPS data. For these types of sensors, a spatial database is again required to identify the closest links to the GPS location of the sensor (which is generally how fixed-location sensors are identified). The GPS location often comes with a description of the location as text and this text is used in the case where the GPS location is close to several possible links. In that situation, the text acts as a discriminator for choosing the correct mapping.

3.2.6. Visualization

At the tail end of any traffic information system, there must be some way to visualize and interpret the results of the traffic estimation algorithms and routing services. The “color-coded” map has become the standard way to quickly disseminate real-time traffic estimates to a wide audience. A color map allows anyone to quickly spot the high congestion areas on the route of interest. In addition to the color map, it is also important to visualize other key pieces of information. Displaying travel times along different route choices between the same origin and

destination pair allows drivers to quickly choose the right one. An example of the *Mobile Millennium* system visualizer, which is the public output of the system, is presented in figure 2.4.

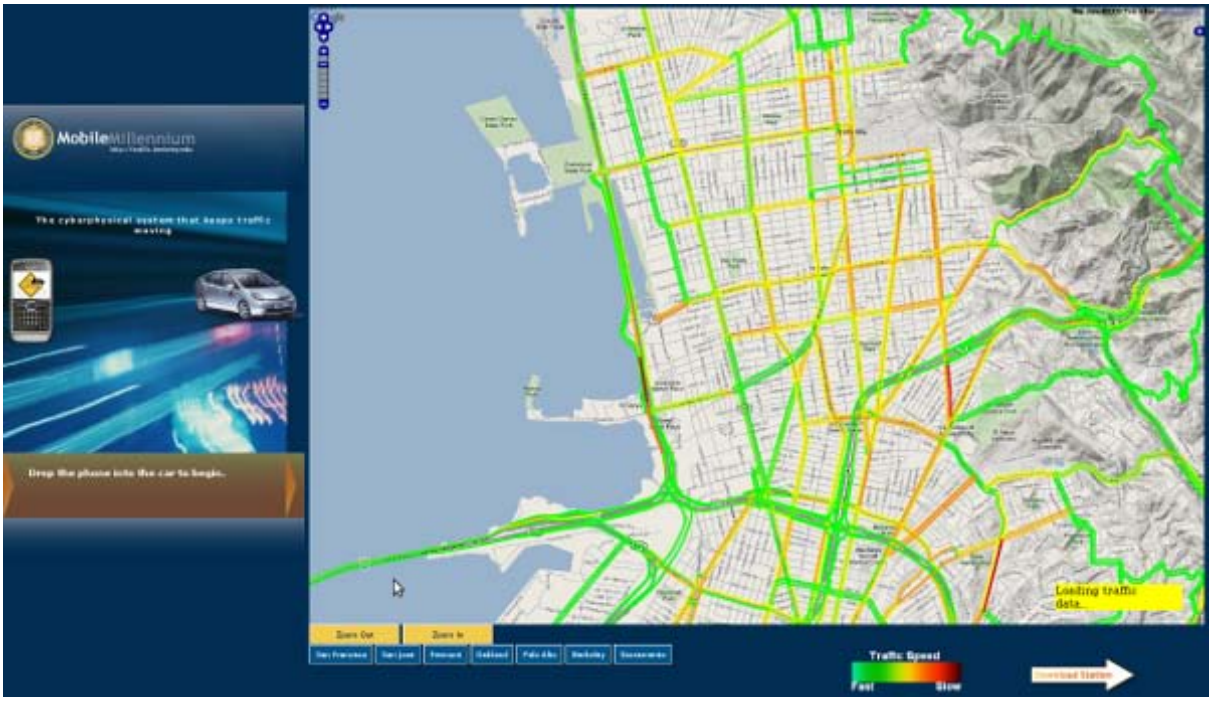


Figure 2.4. *Mobile Millennium* public visualizer showing real-time traffic conditions in Berkeley and Oakland, CA.

In addition to the final output, it is also critical to visualize intermediate components of the traffic estimation process. The *Mobile Millennium* system developed both an internal and external visualizer to allow researchers and the public to view traffic information on a map easily. The internal version of the visualizer allows for detailed insight into how models are producing the estimates. It also allows the researcher to overlay multiple sets of information at once such as fixed sensor locations, portions of GPS traces, or accident information. These visualization tools have become a staple of the *Mobile Millennium* research team and have vastly improved the rate of progress of algorithm development. Figure 2.5 shows an example of some of the layers that are available inside the visualizer (current highway traffic estimates and PeMS loop detector locations).

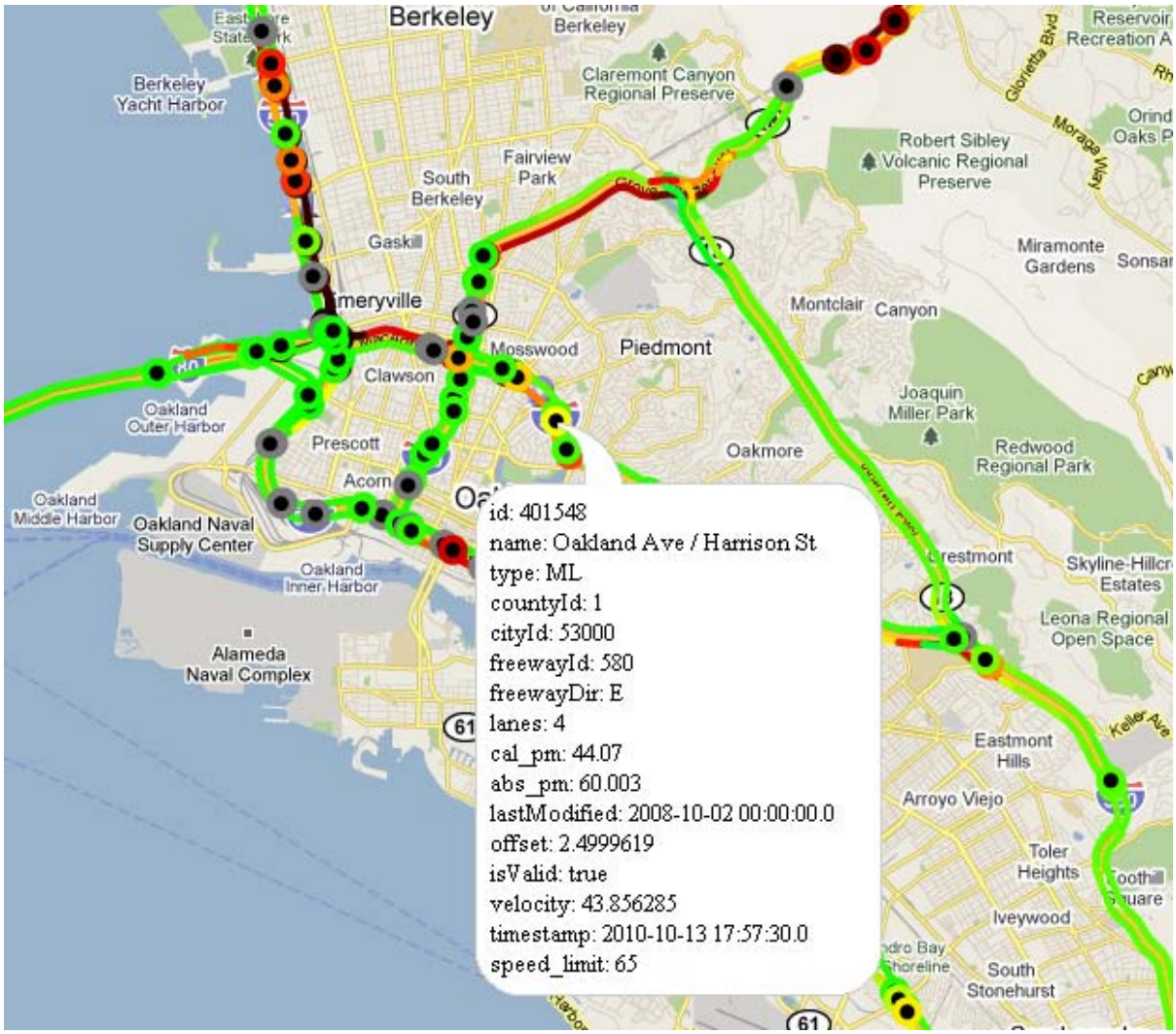


Figure 2.5. *Mobile Millennium* internal visualizer showing model outputs and locations of PeMS loop detectors (hollow circles).

3.2.7. Mobile Client

An increasingly important part of traffic information dissemination is through cell phones. The use of cell phones as part of traffic information systems was the primary inspiration for the *Mobile Millennium* system, which is described in more detail in section 3. Given that the amount of computing power, communication and sensing capabilities in phones is constantly growing, smartphones will continue to be of great value for both providing raw data and for drivers to see real-time conditions while driving.

As part of the *Mobile Millennium* project, Nokia built the first traffic monitoring mobile client that ran on their N95 and E71 series phones (shown in figure 2.6). The requirements of the client were to use the VTL system infrastructure that they had built and also to be able to display live traffic conditions via a color map directly on the phone. This allowed drivers to see current

traffic information while providing data to the system, through the use of VTLs.

Figure 2.6. The *Mobile Millennium* phone client running on a Nokia E71. The phone client displays a color map of current traffic conditions around the driver's location while simultaneously providing VTL data to a central server.

3.2.8. Sensor Deployment

Traffic information systems that use fixed-location sensors as the primary data source inevitably have the problem of *where* and *how many* sensors to deploy. Developing optimal deployment strategies is crucial for public transit agencies building and operating a traffic information system at minimal cost. Historically, sensors have been placed using "rules of thumb" such as every half mile or every third of a mile as is done in different parts of the California highway network [65].

3.3. A Berkeley Prototype: intro to the Mobile Millennium System

This section presents an overview of the *Mobile Millennium* system and highlights the important design decisions that led to the successful execution of the project. These design decisions include using a database-centric approach as opposed to a data queue approach as well as using flexible modules rather than dependently linked processes. The design of the system has allowed for easy scalability as well as the introduction of entirely new research areas to be built into the system seamlessly.

3.3.1. History of the Project

Mobile Millennium began immediately following the successful *Mobile Century* experiment on February 8, 2008 [181]. The initial stated goals of the project were to build a fully operational traffic information system using VTL-based sensing from individual cell phones. The initial partnership was primarily between the Nokia Research Center in Palo Alto, CA and UC Berkeley. Nokia was responsible for developing the software application to go on individual cell phones as well as providing the VTL infrastructure for processing the raw data coming from the phones. UC Berkeley was responsible for building a system capable of using the raw VTL data as input into traffic models that would then output estimates and forecasts of traffic conditions along the major roads in northern California (including both highways and arterials). The traffic estimates were sent back to Nokia so they could be displayed on the cell phones that were running the same software application that was also providing VTL data. This was the first

"participatory sensing" project ever for traffic estimation.

On November 10, 2008, the official phone application was released to the public (shown in figure 2.6). There were more than 5,000 downloads in the first few months, which provided a small amount of data on a daily basis in parts of the Bay Area and Sacramento. VTL data from the phones was supplemented with data from PeMS as well as Navteq radar data to feed the live *Mobile Millennium* system. The first demonstration of the systems capabilities occurred on November 18, 2008 when 20 drivers equipped with cellphones running the official phone application drove for 3 hours in Manhattan, New York. The VTL data was the only source of data available for that experiment and was relied upon entirely for estimating traffic conditions in real-time. The model estimates were displayed live for the attendees of the ITS World Congress that was taking place next to the experiment site. The location of the experiment is shown in figure 3.1.

Figure 3.1. Site of the first *Mobile Millennium* system demonstration in Manhattan, New York on November 18, 2008.

Following the launch of the official *Mobile Millennium* phone application, the project continued to expand the volume of data sources and the sophistication of the traffic estimation models. The goals of the project expanded to include real-time routing algorithms and also for the system to become a central data collection point for many different traffic related data sources. Today, the *Mobile Millennium* system continues to expand its reach to new applications centered around data from mobile devices, including air quality estimation, river flow estimation, and earthquake detection.

3.3.2. System Architecture

There are multiple ways to look at the architecture of the *Mobile Millennium* system. Figure 3.2 illustrates the flow of information through the system from raw data to useful information. In the *Mobile Millennium* system, raw data always goes through at least one filter before being delivered to the models and estimation algorithms. The output of these models is used in a number of applications before being sent to third parties for consumption or analysis. Underneath the flow of data through the system are several components needed for quality analysis and visualization of each step of the process. With this in mind, the *Mobile Millennium* team built an evaluation framework and internal visualizer for comparing and analyzing data from multiple sources using several quality metrics. These allow for quick checking of the data through all steps of the process, from raw data to filtered data to model outputs.

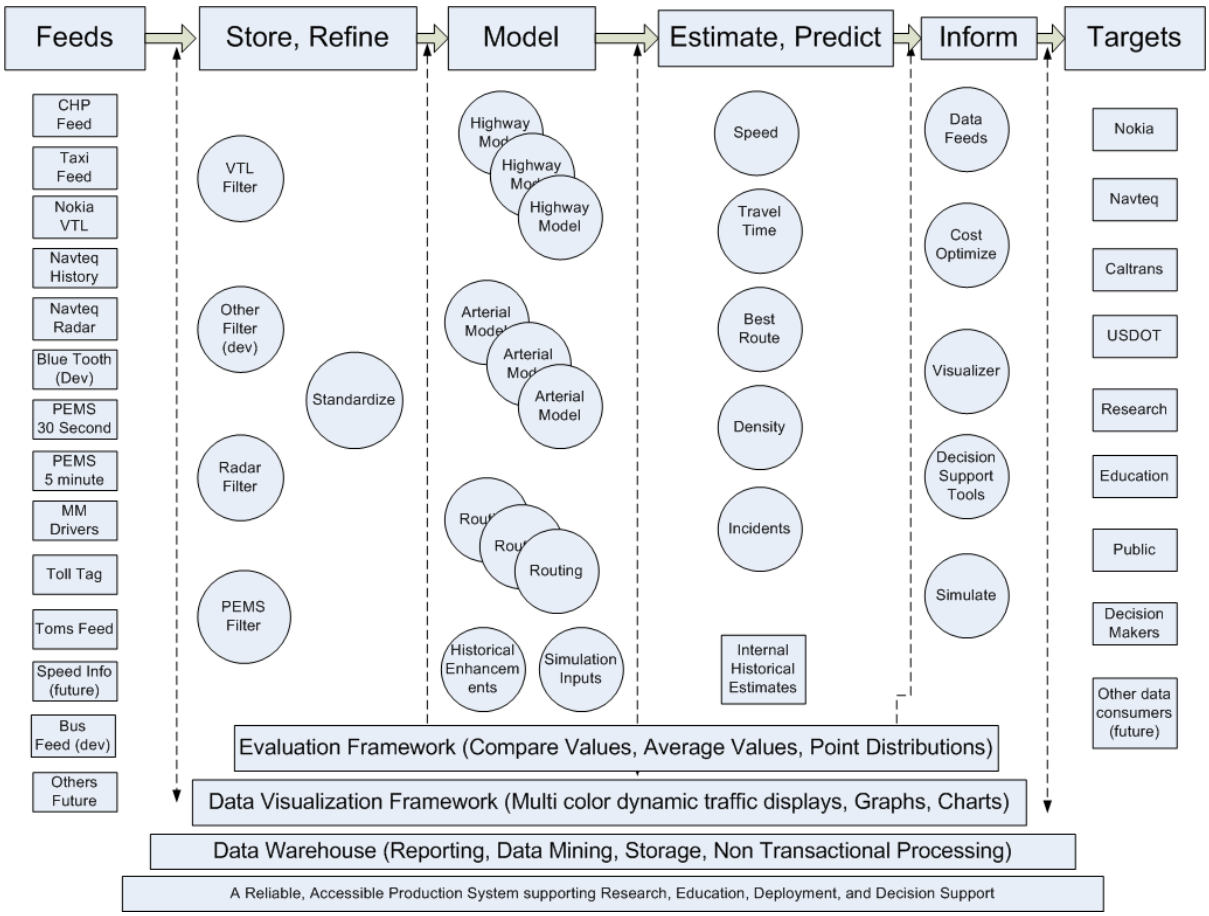


Figure 3.2. An overview of the *Mobile Millennium* system.

Another way of looking at the *Mobile Millennium* system is depicted in figure 3.3. This figure illustrates the way the components of the system interact. In general, the database is the central point for communication between processes. This allows for a modular system where components can function independently without worrying about if another component fails. The system software was designed so that one core module directly interacts with the database and

requires that all requests to receive or send data are passed through that module. There were two members of the *Mobile Millennium* team responsible[2] for maintaining that core module and adding new functionality with regards to reading or writing data. While this places a burden on the people responsible for the core module, it ensures that access to the database is done in a consistent way, which prevents instabilities from occurring. This decision also enabled the project to bring inexperienced student programmers in and have them contribute quickly without having to learn the details of the database design. The core module includes the following basic functions:

- Accessing a simplified representation of the road network for any geographic area of interest. (See section 4 for more details on how this network representation is created.)
- Accessing all raw and filtered data.
- Writing all data or model estimates.

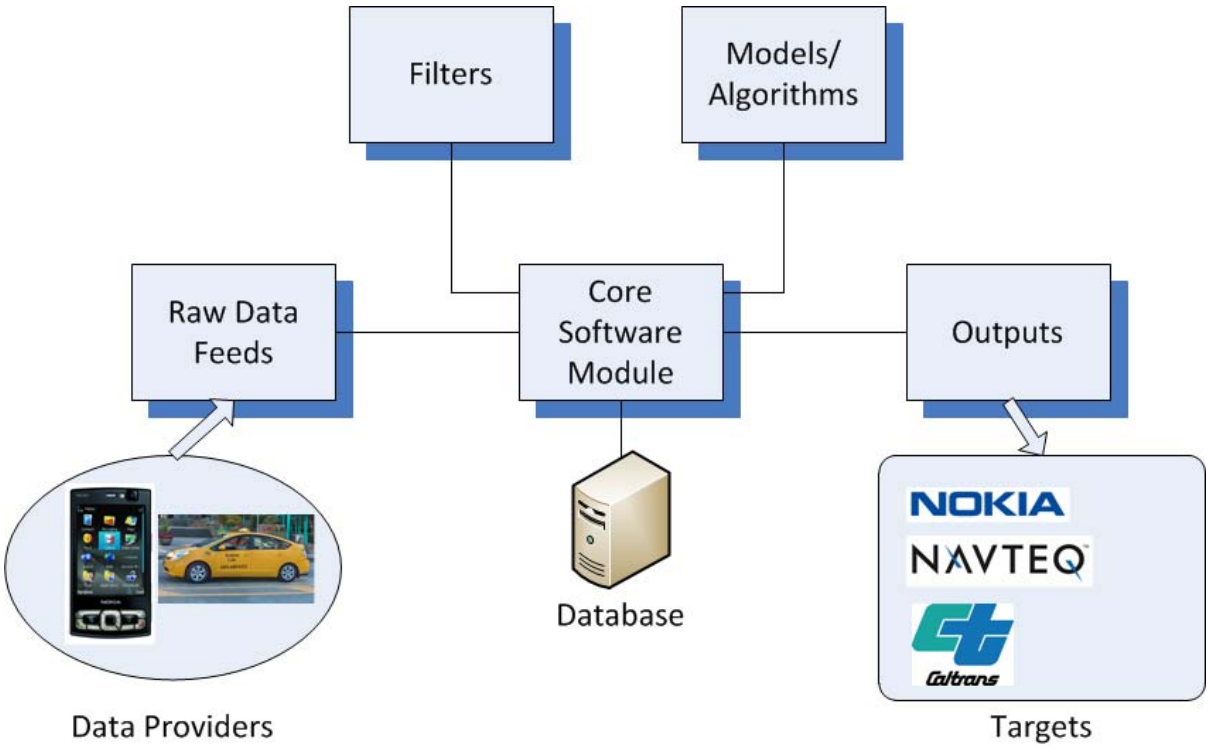


Figure 3.3. *Mobile Millennium* system database-centric architecture.

The disadvantages of this database-centric design are that the input/output bandwidth needs to be high enough to handle all of these requests with minimal delay and if there is any problem with the database, it affects all other processes. The input/output bandwidth constraint is handled by having high-powered machines hooked up together using direct gigabit ethernet links. In general, the importance of the database has led to the team focusing on ways of optimizing database performance and keeping a close eye on database maintenance.

[2]The two team members responsible for the core module were Ryan Herring and Saneesh Apte.

3.3.3. Database Design

The requirements for the database software for the *Mobile Millennium* system were that it be open-source (due to budget constraints) and that it have spatial features (because of the need to do spatial queries on GPS data). These two requirements led to choosing PostgreSQL [24] as the database software with the PostGIS extensions [23] for spatial queries.

The *Mobile Millennium* database's design is centered around the map data provided to the project by Navteq. This data provides the underlying structure of the road network along with a number of key attributes (e.g. lanes, speed limit, etc.). The simplified network construction described in section 4 (called the "Model Graph" in the *Mobile Millennium* system) sits on top of the Navteq map data, meaning that there is a direct relationship from the Model Graph links back to the original Navteq links (known as a foreign key relationship in database terminology). This structure allows any data that is mapped to a Navteq link to be easily converted into a mapping on the associated Model Graph link or vice versa. This is important as the system follows the general rule that raw data sources are mapped to the Navteq map, whereas the traffic estimation models use the Model Graph for inputs and outputs. Translating back and forth is a process that needs to be able to occur very quickly with no delay and the design of the database tables relating the Navteq map and the Model Graph make this possible.

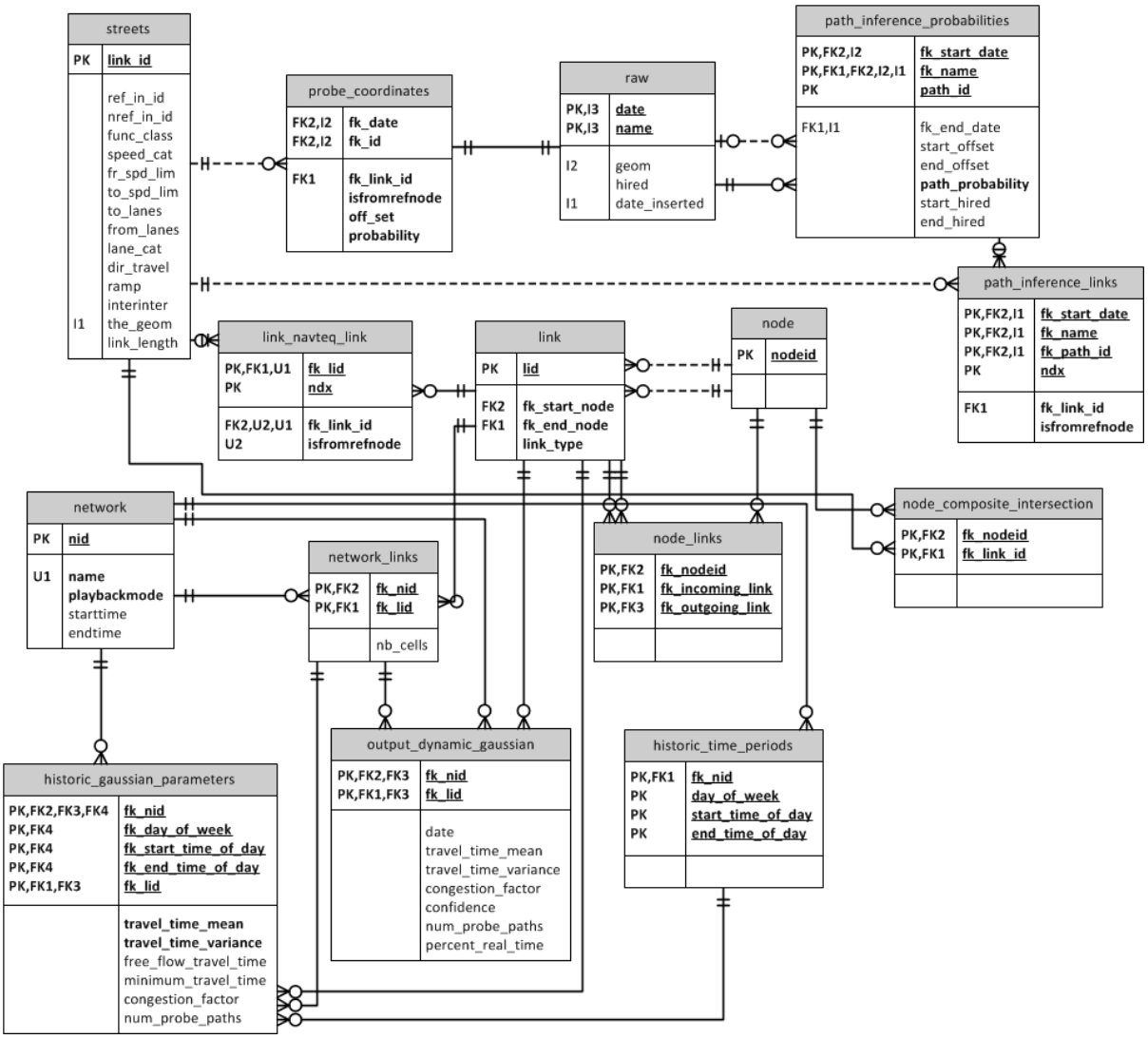


Figure 3.4. An illustration of the relationship between the core tables and a few auxiliary modules of the *Mobile Millennium* database. The symbols between tables represent the relationship between data structures (such as one-to-one, many-to-one, etc.).

Figure 3.4 displays the important tables of the database for the core infrastructure as well as a few example modules that use it (the full database structure is far too big to put in a single graphic). This figure emphasizes the design decision to have raw data feeds dependent upon the Navteq map and the models dependent upon the Model Graph. As the system evolves, new data feeds and traffic estimation models will follow that same pattern.

3.3.4. System Modules

The *Mobile Millennium* system modules can be divided into 4 broad categories: raw data feeds, raw data filters, estimation models/algorithms, and output handlers. Each type of module interacts with the core software module responsible for interacting with the database. In this way,

each of these modules in similar, but each comes with a few specific considerations, which are detailed here. All of these modules are monitored in real-time with alerts sent to key members of the team if any problem occurs.

Raw Data Feeds

All raw data feeds in the *Mobile Millennium* system are constructed using the same principles. First, care is taken to ensure that no raw data is missed in transmission from the source. Most of the feeds into the system are of the *pull* variety, meaning that a mm server queries the data provider every fixed number of seconds for the latest data. This requires checking for duplicates as well as using a rolling window to make sure any late data is still picked up. Some data feeds are *pushed* to the system, meaning that the system must have a process in place to handle any data that is sent by the data provider. This requires enough bandwidth to process any data that is sent without adding delay to the system.

Filters

There are two general classes of filters in the *Mobile Millennium* system and both must satisfy the requirements of locating the data on the map and running with minimal delay so as to provide the models with the data as soon as possible. One filter class encompasses data coming from fixed-location sensors such as loop detectors or radar, the other class is for GPS probe data. The first type of filter requires a static mapping component which places each fixed sensor on the Navteq map and a dynamic component that must process real-time data as quickly after it arrives as possible. These class of filters rely on a fixed map matching procedure developed for fixed sensor data and then a custom filter is built to handle the specifics of the raw data arriving. In terms of the database design, there is a cost savings by having the map matching done once and then only needing to reference a sensor identification number when processing real-time data.

The second class of filters for GPS data requires both filtering in the more traditional sense of the word (removing outliers, smoothing, etc.) as well as map matching for every data point that arrives. This type of filter is much more computationally intensive than the first class as performing map matching is a time consuming path because it relies on a spatial query to the database for each GPS point. This type of filter has been implemented in such a way as to leverage parallel computing technology when it becomes available to the team in the coming months. Choosing this design strategy means that this computationally intensive filter can scale well when the volume of data substantially increases as is expected in the coming years.

Models/Algorithms

The models and algorithms modules in the *Mobile Millennium* system are the most important components of the system from a scientific point of view. They represent the implementation of research ideas that often take years to put into production. Since they are generally significantly bigger modules than the other types, much of the focus of the team is on how to make them run as smoothly as possible. There are two requirements that models must adhere to in the *Mobile Millennium* system. First, it is necessary that any estimation algorithm run fast enough to be used in real-time. This means that the algorithm itself needs to be computationally efficient, but also

means that the systems team needs to ensure that the algorithm has the server resources necessary to run at fast speeds. The second requirement is that all inputs and outputs of the algorithm go through the core software module, which ensures a smooth interaction with the database.

Outputs

The outputs of the system are occasionally directed outward, as is the case when data is sent to Nokia for real-time visualization on cell phones, but these outputs can also be directed toward further research and analysis as well. The team designed an evaluation framework and visualizer to take advantage of the outputs of the models and both of these again rely on the database as the central point of communication. This places a few requirements on the output processes, whose job it is to take the raw model outputs and convert them into whatever form is expected for analysis, visualization or end-user information. Some transformation is always necessary for presenting the model outputs in the correct format, whether it be encoding velocity values in a color scale on a map or computing estimated travel times from a dynamic velocity field. It is precisely the job of the output modules to adapt to whatever the natural outputs of the models are and transform them into the format expected by the final targets.

3.3.5. Field Experiments

Numerous field experiments have been conducted to test the *Mobile Millennium* system. Two that were already mentioned were the original *Mobile Century* experiment in the East Bay, CA followed by the first true *Mobile Millennium* experiment in Manhattan. The following additional experiments were also run:

- **Three experiments in June/July 2008 in Berkeley, CA.** These experiments were run on a small set of arterial roads in Berkeley as preparation for the Manhattan experiment that was run later in 2008. The primary question of interest to be answered by this set of experiments was if the GPS in the phones could provide good enough data on arterials. Additionally, the experiment was designed to specifically study the travel patterns through one of Berkeley's busiest intersections at San Pablo Avenue and University Avenue. Each experiment involved 20 vehicles driving several loops through Berkeley, with all 20 vehicles traversing the segment of University Avenue going west through the San Pablo Avenue intersection. Each test lasted approximately two hours.
- **Three experiments April 27-29, 2010 in the East Bay, CA.** Along with the following set of experiments, this set was designed to be the official test of the production version of the *Mobile Millennium* arterial model. These experiments each had 20 drivers (although data from some of the vehicles was never captured). The goal of this set of experiments was to estimate travel times along a 2.3 mile stretch of San Pablo Avenue that went through Berkeley, Albany and El Cerrito, CA during a three-hour time period. Bluetooth readers collected data intended to be used as ground truth, although the data was not accurate enough to be considered ground truth.
- **Three experiments June 29-July 1, 2010 in San Francisco, CA.** The other half of the official *Mobile Millennium* evaluation. The goal of this set of experiments was to estimate travel times on a 1.1 mile stretch of 3 parallel roads (Van Ness Avenue,

Franklin Street, and Gough Street) over a three-hour time period. 20 vehicles were used in each experiment, split into two loops. Bluetooth readers were used again to collect ground truth, although the same issues prevent it from being considered true ground truth data.

The set of data collected as a result of all of these experiments is large and it has proven to be valuable for research conducted by the *Mobile Millennium* team. It can be argued that the most valuable assets of the entire *Mobile Millennium* project have been the construction of the system described in section 2 and the set of data collected described above. The combination of data and robust system tools make it easy for new researchers that join the project to get started quickly analyzing data and trying out new algorithms, which was one of the primary initial goals of the project.

4. Outreach

4.1. Introduction

"Engage the public and popularize ITS concepts" was a stated goal of the SafeTrip-21 project. The DOT was interested in projects that would quickly show benefit and were likely to survive as commercial offerings going forward. Specifically the ST21 goals were to accelerate ITS research into real-world applications. This would be accomplished by "Piloting a model for ITS field tests that:"

- Leverage existing infrastructure and technologies
- Invite partnerships with shared funding
- Focus on sustainable, market-ready applications
- Deploy and show benefits quickly
- Help focus future Federal research investments

In all of these areas Mobile Millennium was a success. The use of cell phones by commuters to provide location information is now so wide spread that it is hard to remember the world when traffic information could only be provided by fixed sensors placed along a road. Numerous commercial ventures (Google, Navteq, Inrix, Telenav, etc.) are operating in this space. Many of them have hired individuals who worked on the Mobile Millennium project and more still are using the methods, algorithms and protocols that were developed and published as part of the Mobile Millennium project.

This section will discuss the efforts of Mobile Millennium in engaging the public, partnering with private industry and popularizing ITS concepts. We will discuss websites, emails, support forums, news articles, user surveys, academic papers, etc. We will start with a quick review of our partnerships, our messages and our results. The following sections will provide implementation details and analysis of results.

From the public perspective, Mobile Millennium was launched in the Bay Area on November 10, 2008, and demonstrated live in New York City on November 18, 2008, at the 15th World

Congress on Intelligent Transportation Systems. Consumer interaction was managed by NRC, although the free download software was managed by UC Berkeley. The software continued to be available until the public participation phase ended on November 10, 2009.

4.1.1. Partnerships and Funding

The Mobile Millennium project was made possible through partnership with many sponsors, including:

	UC Berkeley College of Engineering is the home of several departments, including the Civil and Environmental Engineering Department, which led the Mobile Century and Mobile Millennium projects.
	California Center for Innovative Transportation, the host organization for Mobile Millennium, works to ensure a harmonious collaboration between government, academia, and industry. The CCIT staff brings key expertise in the areas of traffic engineering, Advanced Traveler Information Systems (ATIS), and hands-on implementation and deployment of operational applications. As of the writing of this report CCIT has merged with PATH (Partners for Transportation Technology)
	The California Department of Transportation (Caltrans), and its Division of Research and Innovation, in cooperation with its partners, has developed a comprehensive program to research, develop, test, and evaluate transportation innovations. This program has partially funded the Mobile Millennium project, and will enable Caltrans to enhance and expand mobility options.
	The US Department of Transportation, supported the Mobile Millennium project through its Safe Trip 21 (ST21) initiative.
	Nokia identified the Mobile Millennium experiment as the next step toward its mobile traffic probe program. In collaboration with the UC Berkeley team, their staff designed, developed, and operated client-side and server-side software used as the engine for traffic data collection and dissemination during the experiment.
	NAVTEQ brought technical know-how and operational expertise to the program as the industry leader in production and distribution of digital maps and real-time traffic information. Navteq's traffic division worked with the Mobile Millennium team to fuse probe data and current traffic information and leverage existing distribution channels.

In addition to project funding from Mobile Millennium's core partners, seed research funds have been provided by various sponsors. Sponsors are listed below in order of their respective contributions (in some cases support includes in-kind contributions).

	The **National Science Foundation** has provided seed money under award 0615299 for the development of viability algorithms, and under contract 0845076 for the development of Lagrangian sensing based inverse modeling algorithms.
	The **Volvo Research and Educational Foundations** has sponsored the development of algorithms for inverse modeling using flow based models for highway systems, through the Volvo Center of Excellence at the University of California, Berkeley (Volvo Center for Future Urban Transport).
	The University of California Research Center has provided seed funding for the development of traffic flow models currently used by the Mobile Millennium system.
	The Finish agency **Tekes** was the first agency to formally fund the Nokia-Berkeley collaboration for the development of algorithms for highway traffic reconstruction.
	CITRIS enabled Mobile Millennium to seal a partnership with Tekes and has been supporting the Nokia Distinguished Lectures on Cyber Physical Systems. Its new building Sutardja Dai Hall is now hosting some key members of the Mobile Millennium team. The arterial traffic display was first unveiled in the CITRIS technology museum at the dedication ceremony of Sutardja Dai Hall, where it can currently be viewed.
	The **VTT Technical Research Center of Finland** has been supporting the project by sending researchers to participate in the creation of the system. VTT has an ongoing research collaboration with Mobile Millennium.
	The **Ministere de l'Ecologie, de l'Energie, du Development Durable et de l'Amenagement du Territoire** funded collaborations between the Universite Paris Dauphine and UC Berkeley, focused on the development of algorithms for traffic flow modeling using viability theory.
	The **VIMADES** is a French company developing software using concepts from viability theory. Technology developed by VIMADES has been used in the development of the numerical software of Mobile Millennium.

As an historical note the partnership between CCIT and NRC can be traced back to 2006, when the National Science Foundation co-funded a joint US/European Union workshop, in Helsinki, the capital of Finland. UC was represented by the Center for Information Technology Research in the Interest of Society (CITRIS), a multi-campus entity with a mission to create information technology solutions for social, environmental, and health care problems. In effect, CITRIS is a broker between academic partners and business.

Contacts made at the Helsinki workshop led to discussions regarding the potential use of mobile phone and navigation technologies to monitor real-time traffic flow, which in turn led to the launch of the Mobile Century project in February 2008. Mobile Century was intended as a proof of concept to test traffic data collection from GPS-equipped cell phones in one hundred vehicles driven on a 10-mile stretch of a highway located in the San Francisco Bay Area.

4.1.2. Awards and Special Ceremonies

California Transportation Foundation's 2009 Tranny Award

The Mobile Century/Mobile Millennium Project won the California Transportation Foundation's 2009 Tranny Award for Traffic Operations/ITS Project of the Year. The awards celebrate transportation achievements in 2008. Project partners Caltrans District 4, UC Berkeley College of Engineering, and Nokia were announced at a June 3 luncheon in Sacramento.

ITS Best Innovative Practices Award for 2008

The Mobile Century project was honored with the "Best Innovative Practices Award" by the Intelligent Transportation Society of America at the annual meeting of the national group in New York City in November, 2008.

Federal awards

With the work performed for Mobile Millennium, the PI of the project, Professor Alexandre Bayen received two prestigious awards from the Federal Government, the CAREER Award from the National Science Foundation, and the PECASE Award from The White House. The Faculty Early Career Development (CAREER) Program is a Foundation-wide activity that offers the National Science Foundation's most prestigious awards in support of junior faculty who exemplify the role of teacher-scholars through outstanding research, excellent education and the integration of education and research within the context of the mission of their organizations. Such activities should build a firm foundation for a lifetime of leadership in integrating education and research. NSF encourages submission of CAREER proposals from junior faculty members at all CAREER-eligible organizations and especially encourages women, members of underrepresented minority groups, and persons with disabilities to apply. The Presidential Early Career Award for Scientists and Engineers (PECASE) is the highest honor bestowed by the United States government on outstanding scientists and engineers in the early stages of their independent research careers. The White House, following recommendations from participating agencies, confers the awards annually. To be eligible for a Presidential Award, an individual must be a U.S. citizen, national or permanent resident. These awards acknowledge the foundational work performed by the Mobile Millennium team.

Student awards

The students gained enormous visibility too through this research program, leading to several awards which they won, for their work within Mobile Millennium. A few of them are listed below: •

- Sebastien Blandin, UCTC Dissertation Grant, UC Transportation Center, 2011
- Aude Hofleitner, Eisenhower Fellow, US Department of Transportation, 2011
- Sebastien Blandin, Finalist Best Student Paper Award , IEEE Conference on Decision and Control, Atlanta, 2010

- Timothee Chamoin, Prix de Stage d'Option, Mathematiques Appliquees, Ecole Polytechnique, France, 2010
- Sebastien Blandin, Eisenhower Fellow, US Department of Transportation, 2010
- Christian Claudel, Leon O Chua Award, UC Berkeley, 2010
- Dan Work, Rodney E. Slater Award, ENO Transportation Foundation, 2010
- Dan Work, ENO Fellow, ENO Transportation Foundation
- Dan Work, Student of the Year Award, UC Transportation Center, 2009
- Dan Work, Eisenhower Fellow, US Department of Transportation, 2009

CITRIS opening

The Mobile Millennium arterial traffic visualizer was an important component of the grand opening of the CITRIS building (Sutardja Dai Hall). As part of the opening ceremony and in the presence of distinguished guests including California Governor Gray Davis, Mobile Millennium's live interactive visualizer was presented to guests of the ceremony. The visualizer is now operational and can be viewed by any visitor of the museum. More about the CITRIS opening can be found at:

`http://www.berkeley.edu/news/media/releases/2009/03/02.citris.shtml`

Figure 1.3. CITRIS opening

AASHTO 2009 Conference

A special client and special messaging from Nokia was developed for participants at the American Association of State Highway and Transportation Officials (AASHTO) conference. A

booth was setup, a large display screen with traffic visualization was provided and many of the participants either tried or watched a demonstration of the phone client.

4.2. Marketing – Message and Medium

One of the important goals for Mobile Millennium was to determine if normal commuters would be willing to provide information on their locations and speed of travel in order to improve the availability of traffic information. As such it was imperative that normal commuters know about the Mobile Millennium effort and understand how to participate in the project. Communicating with a group of anonymous public users and motivating them to work together for a common goal requires a strong marketing effort. Marketing is composed of a message and the medium for delivering that message. Nokia, UC Berkeley and Caltrans public relations departments worked closely together for months to lay out a communication plan.

Figure 2.1. In the front: Ph.D. students Aude Hofleitner and Ryan Herring, helping early users to download the app on their phone on the launch of *Mobile Millennium*, November 10, 2005 in CITRIS headquarters at UC Berkeley. In the back, Greg Merritt and Ph.D. student Christian Claudel.

The Mobile Millennium message was simple. "Download a client to your phone that will provide us with anonymous information on your vehicles location and speed". In return you will:

- Receive a map of the traffic conditions (color coded speeds) on your phone permitting you to make better judgments about your commute.
- Help out society and the environment by improving traffic
- Participate in a cool experiment using new technology

- Be part of a new internet connected community
- Work with UC Berkeley, a great educational institution

The mediums chosen to communicate the message were:

- Press releases
- Press conferences
- Public events
- Company sponsored events
- A new web site
- Speaking engagements
- Academic Review Papers

Figure 2.2. Mobile Millennium Principal Investigator Alexandre Bayen, interviewed by the media.

Mobile Millennium's public launch was in mid-November 2008. The project ended on June 1st, 2010. The launch was a public relations extravaganza. Media (TV, Radio, Print, Blogs, etc) descended on the University to hear the announcement. Listeners and readers were asked to go to a website, enter a zip code and download a phone client. The downloading of clients started immediately. Over the course of Mobile Millennium project there were more than 5000 users.

However the depth to which Mobile Millennium helped modify the public's view of traffic and cell phones is best communicated by noting the over 200 news articles, seminars and other public announcements related to Mobile Millennium. The contribution to research and commercialization is shown by the over 30 research and journal publications and the list of companies (Navteq, Nokia, IBM, Google, Telenev, etc) and Universities (Berkeley, University of Illinois, Klaus in Saudi Arabia) that Mobile Millennium personnel now work or teach at. There were also over 50 students and interns who are currently still at university who worked on the Mobile Millennium project, who are taking their experience in using cell phones into areas such as earthquake research, water tracking, integrated corridor management, etc.

Below is a listing with web links to much of the information mentioned above. This information may also be found on the Mobile Millennium website `www.traffic.berkeley.edu`.

PRESS RELEASES, INTERVIEWS, MEDIA ARTICLES

WIRED	2011-Mar-02	Calling All Cars: Cell Phone Networks and the Future of Traffic
ars technica the art of technology	2011-Mar-02	Calling All Cars: Cell Phone Networks and the Future of Traffic
Technology Review PUBLISHED BY MIT	2011-Feb-02	Social Surveillance Yields Smarter Directions
THE WALL STREET JOURNAL. WSJ.com	2010-Aug-28	The Inconvenient Truth About Traffic Math: Progress Is Slow
Miller-McCune	2010-Feb-10	Tell Me Where It Hurts, Mr. Highway
AT Magazine	2009-Nov-01	La Recherche Sur Les Transports - Made in California
	2009-Nov-01	Transports et environnement : Un partenariat de recherche renforcé avec la Californie
rediff.com rediff BUSINESS	2009-Jul-24	Traffic snarls may soon be history
NewsOn6	2009-Jul-01	GPS systems that give traffic reports
VentureBeat	2009-May-06	UC Berkeley's CITRIS lab: a haven for startups tries to solve big problems
REUTERS	2009-May-06	CITRIS: An Incubator of Green Tech Innovation
The New York Times	2009-May-03	Smarter GPS to Let Cellphones Point the Way

Cal Alumni Association UC Berkeley	2009-May-01	The Connected Commute
NATURE	2009-Apr-23	Phoning In Data
NOKIA Connecting People	2009-Apr-01	Nokia - Open Threads
WELT ONLINE	2009-Mar-18	Das Handy wird zum Navi der Zukunft
IEEE Advancing Technology for Humanity	2009-Feb-23	Intelligent Transportation Systems, Cell Phone Enhancements Improve Mass Transit
IEEE Advancing Technology for Humanity	2009-Feb-18	Cellphones for Science
npr	2009-Jan-27	Who's Calling? It's Your Traffic Report
THE INDUSTRY STANDARD	2009-Jan-27	Researchers use your cellphone to provide real-time traffic information
NEBusiness	2008-Dec-18	Comment: Information must be protected
EXCELSIOR www.excelsior.com.mx	2008-Dec-10	Ring, tráfico al habla
CITRIS	2008-Dec-08	December 2008 CITRIS Newsletter
ITSBerkeley Institute of Transportation Studies	2008-Dec-01	Berkeley Researchers' High Profile at the 15th annual ITS World Congress in NYC
THE SACRAMENTO BEE	2008-Nov-25	Test program guides travelers by cell phone
ars technica the art of technology	2008-Nov-23	Nokia collaboration may keep you out of traffic jams

NewScientist	2008-Nov-22	Cellphone clusters give traffic jams away
VentureBeat	2008-Nov-20	Nokia researchers show off the mobile experiences of the future
U.S.News & WORLD REPORT	2008-Nov-19	Volunteer to Have Your Driving Habits Tracked, Help Reduce Traffic
Wi-Fi Cell Phones	2008-Nov-18	Mobile Millennium - GPS Traffic Mapping
BBC	2008-Nov-18	Tech that trumps traffic tangles
c|net	2008-Nov-18	Nokia shows off real-time traffic application
Dailynews	2008-Nov-18	NAVTEQ dials into new traffic monitoring data
The New York Times	2008-Nov-18	Volunteers Sought for Real-Time Traffic Project
EarthTimes	2008-Nov-17	Groundbreaking Debut of Traffic Probe Data at ITS World Congress
TWICE	2008-Nov-17	Navteq/Nokia Offer Advanced Traffic Updates
GNT generation nouvelles technologies	2008-Nov-14	Mobile Millenium : Nokia teste l'info trafic par mobile GPS
SDA ASIA Information Power for IT Experts	2008-Nov-13	Nokia Research Center puts Mobile Millennium to Curb Traffic Congestion
MercuryNews.com	2008-Nov-13	UC Berkeley software turns cell phones into traffic trackers
CONTRA COSTA TIMES ContraCostaTimes.com	2008-Nov-13	Traffic Study Goes High Tech
NEWSFACTOR.com	2008-Nov-12	New Software Turns Cell Phones into Traffic Trackers

Mobinaute	2008-Nov-12	Mobile Millennium : Nokia utilisera ses mobiles GPS pour éviter les embouteillages
ZDNet	2008-Nov-12	Nokia working to reduce traffic congestion
THE INDEPENDENT	2008-Nov-12	Cyberclinic: Flu outbreaks and traffic jams
Electricpig.co.uk The only tech you need	2008-Nov-12	Nokia launches next-gen traffic study
MSNBC	2008-Nov-12	GPS'li telefonlar trafik bilgisi olu!turacak
The Mainichi Daily News	2008-Nov-11	Nokia、GPSを利用した交通情報収集プロジェクト「Mobile Millenium」を開始
GPS Business News	2008-Nov-11	Nokia, NAVTEQ in large scale trial for traffic information generated by GPS-phones
c|net	2008-Nov-11	Using your cell phone's GPS to map traffic
Future Tense	2008-Nov-11	Software empowers cell phones to fight traffic congestion
CZ	2008-Nov-11	Pilotprojekt erprobt GPS-basierte Verkehrsnachrichten in Echtzeit
ITPRO	2008-Nov-11	Nokia studies traffic with GPS-enabled mobiles
Technology Review PUBLISHED BY MIT	2008-Nov-11	Tracking Traffic with Cell Phones

TMCnet.com	2008-Nov-11	Nokia Intros Mobile Millennium to Help Reduce Recovery Traffic Congestion
snapvoid	2008-Nov-10	Mobile Millenium Project, GPS, And Mobile Phones To Ease Your Commute.
SYMBIAN FREAK	2008-Nov-10	Large scale public pilot to gather and analyze traffic information using GPS-enabled phones
Symbian-Guru.com Notes, Reviews, and Walkthroughs For Symbian	2008-Nov-10	Nokia Launches Public Trial Of Mobile Millenium
techradar.com Deep into technology	2008-Nov-10	Nokia's Mobile Millennium gives free traffic info
cellular-news	2008-Nov-10	Using Mobile Phones to Monitor Road Traffic Congestion
Atelier	2008-Nov-10	Les mobiles GPS recréent le trafic routier de San Francisco
TELECOMPAPER	2008-Nov-10	Open-source-concurrentie in VS voor TomTom HD Traffic
WIRED	2008-Nov-10	Nokia Experiment Turns Phones Into Traffic Sensors
YAHOO!	2008-Nov-10	Nokia Research Center Puts Mobile Millennium in Gear to Help Reduce Traffic Congestion
YAHOO!	2008-Nov-10	Cell Phones Linked to Track Real-Time Traffic (PC Magazine)
San Francisco Chronicle	2008-Nov-10	Cell phones part of traffic monitoring network

sfist	2008-Nov-10	UC/Berkeley Releases Cellphone Program to Help Ease Traffic
NOKIA Connecting People	2008-Nov-10	Nokia Research Center puts Mobile Millennium in gear to help reduce traffic congestion
examiner.com get inside San Francisco	2008-Nov-10	Nokia Research Center Puts Mobile Millennium in Gear to Help Reduce Traffic Congestion
Forbes	2008-Nov-10	Nokia Research Center Puts Mobile Millennium in Gear to Help Reduce Traffic Congestion
COMPUTERWORLD	2008-Nov-10	Project turns GPS phones into traffic reporters
GPS Business News	2008-Nov-10	Nokia, NAVTEQ in large scale trial for traffic information generated by GPS-phones
inside-handy.de ...einfach wissen	2008-Nov-10	Neue Software nutzt Handys mit GPS als Stau-Sensoren
BLOGO WOGO	2008-Nov-10	Free Traffic Info with Nokia's Mobile Millenium
MediaPost	2008-Nov-10	Mobile Program Sends Real-Time Traffic Info
CLOUD COMPUTING JOURNAL WWW.CLOUDCOMPUTING.SYS-CON.COM	2008-Nov-10	Nokia Research Center Puts Mobile Millennium in Gear to Help Reduce Traffic Congestion
MercuryNews.com	2008-Nov-10	UC Berkeley software turns cell phones into traffic trackers
New Mobile Tech Mobile Technology And Wireless Industry News	2008-Nov-10	Mobile Millenium Gives You Traffic Reports, For Free
THE DAILY CALIFORNIAN DAILYCAL.ORG	2008-Nov-10	Researchers' New GPS Software Could Get Drivers Out of a Jam
NBC NEWS	2008-Nov-10	Gridlock Gadget: New Cell Phone Software to Help Drivers Avoid Traffic

mocoNews.net	2008-Nov-10	Mobile Content Bits: Nokia's Big Traffic Plans; Zed Ringtones On Alltel; Sega Mobile; Mobile Billing
PC MAGAZINE	2008-Nov-10	Cell Phones Linked to Track Real-Time Traffic
Pocketlint	2008-Nov-10	Traffic info from GPS devices to be analysed
abc	2008-Nov-10	UC Berkeley teams up with Nokia for traffic
SLASHGEAR	2008-Nov-10	Nokia Mobile Millennium GPS traffic monitoring project seeking volunteers
	2008-Nov-10	Cal Program Uses Cell Phones To Unjam Traffic
AP	2008-Nov-10	Cell Phones Can Help Traffic Flow
impre.com	2008-Nov-10	A olvidarse del caos vial
KTVU.COM	2008-Nov-10	UC Berkeley To Offer Free Cell Phone GPS Download
Mobile Messaging 2.0	2008-Nov-10	Nokia Launches Mobile Millenium for Traffic Updates
San Francisco Chronicle	2008-Nov-09	GPS, Cell Phones Hooked Up to Monitor Traffic
AOL news	2008-Nov-09	Calif. program uses cell phones to unjam traffic
MercuryNews.com	2008-Nov-09	Calif. program uses cell phones to unjam traffic
GIZMODO	2008-Nov-08	Mobile Millennium Project is a Poor Man's Traffic- Relaying GPS

Dr.Dobb's	2008-Nov-07	Turning Mobile Phones into Traffic Cops
engadget	2008-Nov-07	Mobile Millennium project promises to track traffic with cellphones
PC WORLD	2008-Nov-07	Camera Phones and GPS Are for SMBs Too, Says Startup
GPS World UTILITIES & COMMUNICATIONS	2008-Nov-07	NorCal GPS Cell Phone Traffic Probe Project Gets Underway
zimbio	2008-Nov-07	Mobile Millenium project promises to track traffic with cellphones
roadflares.org	2008-Nov-07	Mobile Millenium
Slashdot News for Nerds. Stuff that matters.	2008-Nov-07	Project Turns GPS Phones Into Traffic Reporters
TransID	2008-Nov-07	The mobile millenium project : GPS et informations trafic !
PHYSORG.COM SCIENCE : PHYSICS : TECH : NANO : NEWS	2008-Nov-06	UC Berkeley, Nokia turn mobile phones into traffic probes
COMPUTERWORLD	2008-Nov-06	Project turns GPS phones into traffic reporters
UNIVERSITY of CALIFORNIA	2008-Nov-06	UC Berkeley and Nokia turn mobile phones into traffic probes with launch of pilot traffic-monitoring software
Cal Neighbors	2008-Nov-01	Bay Area Drivers Can Use Cell Phones to Avoid Traffic Snarls
Los Angeles Times	2008-Oct-24	Should Caltrans be Twittering?
VIRTUALIZATION JOURNAL	2008-Oct-23	NAVTEQ to Showcase Leading Map-Enhanced Traffic Solutions at ITS World Congress
les echos.fr le web de l'économie	2008-Oct-02	Pour Nokia, l'avenir passe par les services

San Francisco Chronicle	2008-Jun-26	Plan to avoid traffic jams using cell phones
Oakland Tribune www.oaklandtribune.com	2008-Jun-25	Feds to help world's largest traffic tech test
UCBerkeleyNews	2008-Jun-22	Berkeley In the News
c|net	2008-Jun-20	Mini-subs Exploring Sacramento River
iTRIS	2008-Jun-01	Intelligent Infrastructure: Public Service, Safety, and Security
Technology Review PUBLISHED BY MIT	2008-May-16	A Faster, More Energy-Efficient GPS
PRWeb® PRESS RELEASE NEWSWIRE	2008-Apr-22	Low Cost Fuel Cell Project Wins Clean Tech Innovation Prize Awarded by UC Berkeley's Center for Entrepreneurship & Technology
GPS DAILY connecting anyone to anywhere	2008-Feb-12	Nokia And UC Berkeley Capture Real-Time Traffic Information
c|net	2008-Feb-12	Uni students get stuck in traffic for Nokia
TRADING MARKETS .COM Making Great Traders	2008-Feb-11	Researchers test real-time traffic
npr	2008-Feb-11	Cell Phones: A New Commuter Tool?
Largest Companies	2008-Feb-11	Nokia and UC Berkeley capture real-time traffic information using GPS enabled mobile devices
Dr.Dobb's	2008-Feb-11	Cars and Cell Phones: Maybe They're Not So Bad After All
DVICE imagined by Syfy	2008-Feb-11	Nokia phone will steer you around traffic better than your fancy GPS system
digitoday	2008-Feb-11	Nokia tutkailee tieliikennettä ihmisillä

c|net	2008-Feb-11	Nokia trials N95 as traffic monitor
Machines Like Us	2008-Feb-10	Capturing traffic data using GPS-enabled cell phones
PHYSORG.COM SCIENCE : PHYSICS : TECH : NANO : NEWS	2008-Feb-10	New research project captures traffic data using GPS-enabled cell phones
SLASH PHONE	2008-Feb-10	Nokia and UC Berkeley capture real-time traffic information using GPS enabled mobile devices
GIZMODO	2008-Feb-09	Nokia GPS Phones to Fight the Traffic Plague
techshout.com Beta Shouts out the latest in Technology!	2008-Feb-09	Nokia and UC Berkeley Experts build Technology to offer Real Time Traffic Information
eCanadaNOW	2008-Feb-09	GPS Phones Used To Monitor Traffic
eFluxMedia	2008-Feb-09	Nokia And UC Berkeley Tests GPS Phones As Traffic Sensors
eNews	2008-Feb-09	Nokia Tests A Traffic-Tracking Service
engadget	2008-Feb-09	Nokia trial turns N95s into traffic sensing tools
Los Angeles Times	2008-Feb-09	Using cellphones to beat traffic?
Oakland Tribune www.oaklandtribune.com	2008-Feb-09	GPS-based system would track traffic with phones
Oakland Tribune www.oaklandtribune.com	2008-Feb-09	GPS phone system tested by students

nokiaphoneblog	2008-Feb-09	Nokia Conducts Real Time Traffic Test
Oakland Tribune www.oaklandtribune.com	2008-Feb-09	New GPS phone system tested on Interstate 880
Mobile Whack	2008-Feb-09	Nokia and UC Berkeley capture real-time traffic information using GPS-enabled mobile devices
Oakland Tribune www.oaklandtribune.com	2008-Feb-09	Area study tracks cell phones on highways to monitor traffic
Oakland Tribune www.oaklandtribune.com	2008-Feb-09	Experiment uses phones to track I-880 traffic
SWB Symbian WEB BLOG	2008-Feb-09	Interesting GPS Experiment by Nokia and UC Berkeley
TECHGADGETS.IN	2008-Feb-09	N95 Phone gives Real-time Traffic Info, thanks to Nokia and UC Researchers
Mobiletor.com™ Enabling the Connection	2008-Feb-09	Nokia, UC Researchers capture Real-time Traffic Info using N95 Handset
Mobile Digital	2008-Feb-09	Nokia (Palo Alto), UC Berkeley Capture Real-Time Traffic Information Using GPS Enabled Mobile Devices
MercuryNews.com	2008-Feb-09	Researchers try tracking traffic using cell phone GPS
c|net	2008-Feb-08	Nokia turns people into traffic sensors
	2008-Feb-08	Test GPS Technology For Traffic Info

TG DAILY	2008-Feb-08	Nokia tracks traffic info with gang of GPS feeds
Wireless and Mobile News	2008-Feb-08	UCB & Nokia Test GPS for Traffic Flow and Monitoring
Oakland Tribune www.oaklandtribune.com	2008-Feb-08	Profs test tracking GPS phone to gauge traffic
REUTERS	2008-Feb-08	Nokia and UC Berkeley Capture Real-Time Traffic Information Using GPS Enabled Mobile Devices
MOBILEDIA	2008-Feb-08	Nokia and UC Berkeley Monitors Highway Traffic
iTRIS	2008-Feb-08	Using GPS Mobile Phones as Traffic Sensors
My Digital Life	2008-Feb-08	Trend of Having GPS Enabled Cell Phones for Traffic Monitoring
San Francisco Chronicle	2008-Feb-05	Cell phone test to monitor I-880 traffic flow
MobiFrance	2008-Feb-04	Interview avec Alexandre BAYEN, chercheur et Professeur Français à l'université de Berkeley en Californie
Slashdot News for Nerds. Stuff that matters.	2008-Feb-03	Cellphones to Monitor Highway Traffic
ZDNet	2008-Feb-01	Cellphones to monitor highway traffic
intomobile	2008-Feb-01	UC Berkeley Students to Trial Mobile Phone Based California Traffic-Monitoring - Mobile Century Uses Nokia N95 as Mobile GPS Generator

Press conferences

- *Mobile Millennium*, UC Berkeley, CA November 10, 2008

Held jointly with Nokia, Navteq and the California DOT

- *Safe Trip 21*, Bay Bridge, Oakland, CA June 26, 2008

Held jointly with the US DOT and the California DOT

- *Mobile Century*, Union Landing, CA February 8, 2008

Held jointly with Nokia and the California DOT

Press releases

- *University of California at Berkeley*, "Campus dedicates new state-of-the-arts CITRIS research headquarters," Mar. 2, 2009
- *University of California at Berkeley*, "Dedication of new CITRIS headquarters marks new stage of innovation to help fuel economic growth," Feb. 27, 2009
- *Nokia*, "Nokia Research Center puts Mobile Millennium in gear to help reduce traffic congestion," Nov. 10, 2008
- *University of California at Berkeley*, "UC Berkeley and Nokia turn mobile phones into traffic probes with launch of pilot traffic-monitoring software," Nov. 6, 2008
- *ITS America*, "ITS America announces finalists for the 2008 Best of ITS Awards," Oct. 30, 2008
- *U.S. Department of Transportation*, "U.S. DOT partners with Caltrans to move California drivers one step closer to instant travel information and safety technologies," Jun. 25, 2008
- *Nokia*, "Nokia and UC Berkeley capture real-time traffic information using GPS enabled mobile devices," Feb. 8, 2008
- *University of California at Berkeley*, "Joint Nokia research project captures traffic data using GPS-enabled cell phones," Feb. 8, 2008

TV interviews (selected)

- *OETA News*, “New Levee Bewaching Prevention Techniques,” Nov. 11, 2009
- *CBS’s Smart Planet*, “Alex Bayen, Professor, Systems Engineerng, UC Berkeley,” May 29, 2009, by Jason Pepper
- *NBC News*, “Tech Future in Good Hands at Cal,” May 6, 2009
- *Cnet*, “Nokia shows off real-time traffic application,” Nov. 18, 2008
- *BBC News*,“ Tech that trumps traffic tangles,” Nov. 18, 2008, by Jason Palmer
- *Cnet*, “Using your cell phone’s GPS to map traffic,” Nov. 11, 2008, by Kara Tsuboi
- *KTVU*,“UC Berkeley To Offer Free Cell Phone GPS Download,” Nov. 10, 2008.
- *CBS News*, “Cal Program Uses Cell Phones to Unjam Traffic,” Nov. 10, 2008
- *ABC News* ,“UC Berkeley teams up with Nokia for traffic,” Nov. 10, 2008
- *NBC News*, “Gridlock Gadget: New Cell Phone Software to Help Drivers Avoid Traffic,” Nov. 10, 2008
- *ABC News*, “Real-time traffic information to your cell phone,” Jun. 25, 2008
- *NBC News*, “Bay Area traffic study kicks off,” Jun. 25, 2008
- *Cnet*, “Mobile Sensing-mini subs explore Sacramento,” Jun. 20, 2008
- *Cnet*, “Students get stuck in traffic for Nokia,” Feb. 12, 2008
- *Cnet*, “Nokia trials N95 as traffic monitor,” Feb. 11, 2008
- *CBS News*, “Cal, Nokia test GPS technology for traffic info,” Feb. 8, 2008
- *ABC News*, “Cell phones used to test Easy Bay traffic,” Feb. 8, 2008
- *Cnet*, “Nokia turns people into traffic sensors,” Feb. 8, 2008
- *NBC News*, “Researchers test GPS-cell phone navigation in South Bay,” Feb. 8, 2008
- *Fox News*, “Nokia and UC Berkeley capture real-time traffic information using GPS enabled mobile devices,” Feb. 8, 2008

<u>Radio interviews</u> (selected)

- *Nature-podcast*, “Phoning in Data,” April 23, 2009, by Roberta Kwok
- *KQED*, “Dialing in on Traffic,” Dec. 15, 2008, by David Gorn
- *National Public Radio*, “Who’s Calling? It’s Your Traffic Report,” Jan. 26, 2009, by David Gorn
- *KQED*, “Reporter’s Notes: Dialing in on Traffic,” Dec. 12, 2008, by David Gorn
- *National Public Radio*, “Cell Phones: a new commuter tool?,” Feb. 11, 2008
- *KCBS*, “Researchers road test GPS technology for traffic info,” Feb. 8, 2008

<u>Newspaper interviews or articles</u> (selected)

- *The Wall Street Journal*, “The Inconvenient Truth About Traffic Math: Progress

Is Slow," Aug. 28, 2010, by Carl Bialik

- *California Alumni Magazine*, "The Connected Commute," May/June 2009, by David Downs
- *The New York Times*, "Smarter GPS to Let Cellphones Point the Way," May 3, 2009, by Roy Furchgott
- *Nature*, "Phoning in Data," April 23, 2009, by Roberta Kwok
- *Die Welt*, "Das Handy wird zum Navi der Zukunft," March 18, 2009
- *The Sacramento Bee*, "Test program guides travelers by cell phone," Nov. 25, 2008, by Tony Bizjak
- *New Scientist*, "Cellphone clusters give traffic jams away," Nov. 22, 2008
- *The New York Times*, "Volunteers Sought for Real-Time Traffic Project," Nov. 18, 2008 by Roy Furchgott
- *The Earth Times*, "Groundbreaking Debut of Traffic Probe Data at ITS World Congress," Nov. 17, 2008
- *Contra Costa Times*, "Traffic study goes high tech," Nov. 13, 2008 by Erik Nelson
- *San Jose Mercury News*, "UC Berkeley software turns cell phones into traffic trackers," Nov. 13, 2008, by Dennis Cuff
- *The Daily Californian*, "Researchers' New GPS Software Could Get Drivers Out of a Jam," Nov. 10, 2008
- *San Francisco Chronicle*, "Cell phones part of traffic monitoring network," Nov. 10, 2008
- *Forbes*, "Nokia Research Center Pults Mobile Millennium in Gear to Help Reduce Traffic Congestion," Nov. 10, 2008
- *San Francisco Chronicle*, "GPS Cell phones hooked up to monitor traffic," Nov. 9, 2008
- *San Francisco Chronicle*, "Plan to avoid traffic jams using cell phones," Jun. 26, 2008
- *The Oakland Tribune*, "Feds to help world's largest traffic tech test," Jun. 25, 2008
- *The Oakland Tribune*, "Experiment uses phones to track I-880 traffic," Feb. 9, 2008
- *San Jose Mercury News*, "Researchers try tracking traffic using cell phone GPS," Feb. 9, 2008
- *Los Angeles Times*, "Using cell phones to beat traffic?," Feb. 9, 2008
- *Tri Valley Herald*, "New GPS phone system tested on Interstate 880," Feb. 9, 2008
- *San Mateo County Times*, "GPS-based system would track traffic with phones," Feb. 9, 2008
- *San Francisco Chronicle*, "Cell phone test to monitor I-880 traffic flow," Feb. 5, 2008

<u>Other media outlets</u> (selected)

- *Reuters*,"CITRIS: An Incubator of Green Tech Innovation," May 6, 2009
- *Venture Beat*,"UC Berkeley's CITRIS lab: a haven for startups trying to solve big problems," May 6, 2009, by Dean Takahashi
- *Nokia Open Innovation Newsletter*,"Open Threads," April 2009
- *IEEE*,"Intelligent Transportation Systems, Cell Phone Enhancements Improve Mass Transit," Feb. 23, 2009
- *IEEE Spectrum*,"Cell Phones for Science," February 2009, by Prachi Patel-Predd
- *The Industry Standard*,"Researchers use your cellphone to provide real-time traffic information," Jan. 27, 2009, by Sindya Bhanoo
- *The Journal*,"Comment: Information must be protected," Dec. 18, 2008, by Maitland Hyslop
- *Excelsior*,"Ring, trafico al habla," Dec. 10, 2008, by Carlos Fernandez de Lara
- *CITRIS Report*,"Taming Traffic with Your Phone: The Mobile Millennium Project", Dec. 08, 2008, by Gordy Slack
- *NewsBITS*,"Berkeley Researchers' High Profile at the 15th annual ITS World Congress in NYC," Winter 2008, by Ann Brody-Guy
- *Cal Neighbors*,"Bay Area drivers can use cell phones to avoid traffic snarls," Fall 2008
- *CEE @ Berkeley Connections*,"Mobile Millennium Poised to Expand Bay Area's Reputation as High-tech leader," Fall 2008,
- *ARS Technica*,"Nokia collaboration may keep you out of traffic jams," Nov. 23, 2008, by David Chartier
- *Venture Beat*,"Nokia researchers show off the mobile experiences of the future," Nov. 20, 2008, by Dean Takahashi
- *US News & World Report*,"Volunteer to Have Your Driving Habits Tracked, Help Reduce Traffic," Nov. 19, 2008
- *Daily News Online*,"NAVTEQ dials into new traffic monitoring data," Nov. 18, 2008
- *Wi-Fi Cell Phones*,"Mobile Millennium- GPS Traffic Mapping," Nov. 18, 2008
- *This Week in Consumer Electronics -TWICE*,"Navteq/Nokia Offer Advanced Traffic Updates," Nov. 17, 2008, by Amy Gilroy
- *GNT*,"Mobile Millennium: Nokia teste l'info trafic par mobile GPS," Nov. 14, 2008, by C. Bruno
- *SDA Asia*,"Nokia Research Center puts Mobile Millennium to Curb Traffic Congestion," Nov. 13, 2008
- *ZD Net*,"Nokia working to reduce traffic congestion," Nov. 12, 2008, by Matthew Miller
- *Newsfactor.com*,"New Software Turns Cell Phones into Traffic Trackers," Nov. 12, 2008
- *MSNBC*, "GPS'li telefonar trafik bilgisi olusturacak," Nov. 12, 2008
- *Mobinaute*,"Nokia utilisera ses mobiles GPS pour eviter les embouteillages," Nov. 12, 2008
- *The Independent*,"Flu outbreaks and traffic jams," Nov. 12, 2008
- *Electricpig*,"Nokia launches next-gen traffic studies," Nov. 12, 2008
- *American Public Media*,"Software empowers cell phones to fight traffic

congestion," Nov. 11, 2008

- *TMCnet*,"Nokia Intro Mobile Millennium to Help Reduce Roadway Traffic Congestion," Nov. 11, 2008
- *MIT Technology Review*,"Tracking Traffic with Cell Phones: A new project collects traffic data from GPS-enabled cell phones," Nov. 11, 2008
- *IT Pro*,"Nokia studies traffic with GPS-enabled mobiles," Nov. 11, 2008
- *Computer Zeitung*,"Pilotprojekt erprobt GPS-basierte Verkehrshcarichten in Echtzeit," Nov. 11, 2008
- *GPS Business News*,"Nokia, NAVTEQ in large scale trial for traffic information generated by GPS-phones," Nov. 11, 2008
- *PressDemocrat.com*,"Cell phones can help traffic flow," Nov. 10, 2008
- *impre.com*,"A olvidarse del caos vial," Nov. 10, 2008
- *VOIP IP Technology*,"Mobile Millennium Project, GPS and Mobile Phones To Ease Your Commute," Nov. 10, 2008
- *BlogoWogo*,"Free Traffic Info with Nokia's Mobile Millennium," Nov. 10, 2008
- *L'Atelier*,"Les mobiles GPS recreent le trafic routier de San Francisco," Nov. 10, 2008
- *Cellular-News*,"Using Mobile Phones to Monitor Road Traffic Congestion," Nov. 10, 2008
- *Computer World*,"Project turns GPS phones into traffic reporters," Nov. 10, 2008
- *Inside-handy.de*,"Neue Software nutzt Handys mit GPS als Stau-Sensoren," Nov. 10, 2008
- *Media Post*,"Mobile Program Sends Real-Time Traffic Info," Nov. 10, 2008
- *New Mobile Tech*,"Mobile Millennium Gives You Traffic Reports, For Free," Nov. 10, 2008
- *Pocket-lint*,"Nokia Announces Mobile Millennium Project," Nov. 10, 2008
- *SFist*,"UC Berkeley releases cell phone program to help ease traffic," Nov. 10, 2008
- *Slash Gear*,"Nokia Mobile Millennium GPS traffic monitoring project seeks volunteers," Nov. 10, 2008
- *Symbian Freak*,"Large scale public pilot to gather and analyse traffic information using GPS-enabled mobile devices," Nov. 10, 2008
- *Symbian-guru.com*,"Nokia launches public trial of Mobile Millennium," Nov. 10, 2008
- *TechRadar.com*," Nokia's Mobile Millennium gives free traffic info," Nov. 10, 2008
- *Telecom Paper*,"Open-source-concurrentie in VS voor TomTom HD Traffic," Nov. 10, 2008
- *PC Magazine*,"Cell Phones Linked to Track Real-Time Traffic," Nov. 10, 2008
- *Mobile Messaging 2.0*,"Nokia Launches Mobile Millennium for Traffic Updates," Nov. 10, 2008
- *mocoNews.net*,"Nokia's Big Traffic Plans," Nov. 10, 2008
- *Gizmodo*,"Mobile Millennium Project is a Poor Man's Traffic Relaying GPS," Nov. 8, 2008
- *Dr. Dobb's Portal*,"Turning Mobile Phones into Traffic Cops," Nov. 7, 2008

- *Engadget*,"Mobile Millennium Project promises to track traffic with cellphones," Nov. 7, 2008
- *PC World*,"Camera Phones and GPS Are for SMBs Too, Says Startup," Nov. 7, 2008
- *RoadFlares.org*,"Mobile Millennium," Nov. 7, 2008
- *Slashdot*,"Project Turns GPS Phones into Traffic Reporters," Nov. 7, 2008
- *TransID*,"The Mobile Millennium project: GPS et informations trafic!," Nov. 7, 2008
- *Zimbio*,"Mobile Millennium project promises to track traffic with cellphones," Nov. 7, 2008
- *GPS World*,"NorCal GPS Cell Phone Traffic Probe Project Gets Underway," Nov. 7, 2008
- *CITRIS News*,"UC Berkeley and Nokia turn mobile phones into traffic probes with launch of pilot traffic-monitoring software," Nov. 6, 2008
- *PhysOrg.com*,"UC Berkeley, Nokia turn mobile phones into traffic probes," Nov. 6, 2008
- *UC Berkeley Newsroom*,"UC Berkeley and Nokia turn mobile phones into traffic probes with launch of pilot traffic-monitoring software," Nov. 6, 2008
- *CITRIS News*,"Intelligent Infrastructure: Public Service, Safety, and Security: Floating and Cellular Sensors," Jun. 2008
- *MIT Technology Review*,"New GPS Software," May 16, 2008
- *PR Web*,"Mobile Sensing- Lagrangian Sensor project receives Clean Tech Innovation prize," Apr. 22, 2008
- *GPS Daily*,"Nokia and UC Berkeley capture real-time traffic information," Feb. 12, 2008
- *TechGadgets.in*,"N95 phone gives real-time traffic info, thanks to Nokia and UC researchers," Feb. 12, 2008
- *Dr. Dobb's Portal*,"Cars and cell phones: maybe they're not so bad after all," Feb. 11, 2008
- *Largest Companies*,"Nokia & UC Berkeley capture real-time traffic info using GPS-enabled cell phones," Feb. 11, 2008
- *Dvice*,"Nokia phone will steer you around traffic better than your fancy GPS system," Feb. 11, 2008
- *CITRIS*,"Joint research project to capture traffic data," Feb. 11, 2008
- *PhysOrg*,"New research project captures traffic data using GPS-enable cell phones," Feb. 10, 2008
- *Machines Like Us*,"Capturing traffic data using GPS-enabled cell phones," Feb. 10, 2008
- *TechShout.com*,"Nokia and UC Berkeley experts build technology to offer real time traffic information," Feb. 9, 2008
- *eFluxMedia*,"Nokia and UC Berkeley tests GPS phones as traffic sensors," Feb. 9, 2008
- *Engadget*,"Nokia trial turns N95s into traffic sensing tools," Feb. 9, 2008
- *Gizmodo*,"Nokia GPS phones to fight the traffic plague," Feb. 9, 2008
- *Symbian Web Blog*,"Interesting GPS experiment by Nokia and UC Berkeley,"

Feb. 9, 2008

- *eNews 2.0*,"Nokia tests a traffic-tracking service," Feb. 9, 2008
- *Inside Bay Area*,"Area study tracks cell phones on highways to monitor traffic," Feb. 9, 2008
- *Inside Bay Area*,"GPS phone system tested by students," Feb. 9, 2008
- *TradingMarkets.com*,"Researchers test real-time traffic," Feb. 9, 2008
- *Nokia Phone Blog*,"Nokia conducts real time traffic test," Feb. 9, 2008
- *MobileTor.com*,"Nokia, UC researchers capture real-time traffic info using N95 handset," Feb. 9, 2008
- *HardOCP*,"The Next Traffic Sensor is Your Phone," Feb. 9, 2008
- *CanadaNOW*,"GPS Phones Used to Monitor Traffic," Feb. 9, 2008
- *Reuters*,"Nokia and UC Berkeley capture real-time traffic information using GPS enabled mobile devices," Feb. 8, 2008
- *The Tech Generation Daily*,"Nokia tracks traffic info with gang of GPS feeds," Feb. 8, 2008
- *My Digital Life*,"Trend of having GPS enabled cell phones for traffic monitoring," Feb. 8, 2008
- *IntoMobile*,"Mobile Century uses Nokia N95 as mobile GPS sensor," Feb. 8, 2008
- *Inside Bay Area*,"Profs test tracking GPS phone to gauge traffic," Feb. 8, 2008
- *Wireless and Mobile News*,"UCB & Nokia Test GPS for Traffic Flow and Monitoring," Feb. 8, 2008
- *MobilEdia*,"Nokia and UC Berkeley Monitors Highway Traffic," Feb. 8, 2008
- *MobiFrance*,"Interview avec Alexandre Bayen, chercheur et Professeur Francais at l'universite de Berkeley en Californie," Feb. 4, 2008
- *Slashdot*,"Cellphones to Monitor Highway Traffic," Feb. 3, 2008
- *ZD Net*,"Cell phones to monitor highway traffic," Feb. 1, 2008

In addition, *Mobile Millennium* was presented at various venues, thus increasing its visibility.

TALKS

Plenary / keynote speaker

- *13th Annual Inventor Recognition Banquet*, NAVTEQ, The Rookery, Chicago, June 3, 2010, "Technology innovations at the age of web 2.0 and participatory sensing".
- *ARM* $TechCon^3$, Santa Clara Convention Center, October 21, 2009, "Mobile Millennium: using GPS to reconstruct traffic"
- *NAVTEQ Traffic Symposium*, Jacob K Javits Convention Center, New York, NY. November 17th, 2008, "Mobile Millennium: using GPS to reconstruct traffic"

The NAVTEQ Traffic Symposium coincides with the ITS World Congress and gathers about 200 academics and practitioners in the field of traffic monitoring and modeling.

Invited seminars

- *University of California Office of the President*, Board Meeting, Berkeley, CA, October 27, 2010, Host: Professor Mark Yudof, “Mobile sensing in large scale infrastructure systems”.
- *AMP Lab retreat*, Asilomar, CA, December 8, 2010, Host: Professor Michael Franklin, “Cloud based implementations of machine learning algorithms applied to traffic monitoring”.
- *EECS Colloquium*, UC Berkeley, CA, September 29, 2010, Host: Professor Costas Spanos, “Real-time estimation of distributed parameters systems: application to large scale infrastructure systems”.
- *Ecole Nationale des Ponts et Chaussees (ENPC), INRETS*, Marne la Vallee, France, July 1, 2010, Host: Professor Jean-Patrick Lebacque, “Mobile millennium: using smartphones to monitor traffic in privacy aware environments”.
- *Royal Institute of Technology (KTH), Transportation and Logistics Division*, Stockholm, Sweden, April 16, 2010, Host: Professor Haris Koutsopoulos, “Mobile millennium: using smartphones to monitor traffic in privacy aware environments”.
- *Berkeley Wireless Research Center (BWRC)*, UC Berkeley, Berkeley, CA, March 12, 2010, Host: Dr. Gary Kelson, “Mobile Millennium: using cell phones to monitor traffic".
- *Los Alamos National Laboratories (LANL)*, Los Alamos, NM, December 8th, 2009, Host: Dr. Scott Backhaus, “Mobile Millennium: using cell phones to monitor traffic".
- *UC Berkeley, EECS Department, TRUST Center*, September 10th, 2009, Host: Professor Shankar Sastry, “Mobile Millennium: using cell phones to monitor traffic".
- *Massachusetts Institute of Technology (MIT), Department of Civil and Environmental Engineering*, Cambridge, MA, July 31st, 2009, Host: Professor Moshe Ben-Akiva, “Data assimilation for real time traffic flow reconstruction".
- *Palo Alto Research Center (PARC)*, Palo Alto, CA, July 9th, 2009. Host: Dr. Craig Eldershaw. “Mobile millennium: using smartphones to monitor traffic in privacy aware environments”.
- *University of California Los Angeles (UCLA), Department of Electrical Engineering, Center for Embedded Networked Sensing Seminar*, UCLA, CA, June 19th, 2009. Host: Professor Per Deborah Estrin. “Mobile Millennium" using cell phones to monitor traffic”.
- *California Institute of Technology, Control and Dynamical Systems seminar*, Pasadena, CA, June 18th, 2009. Host: Professor Jerry Marsden. “Mobile Millennium using cell phones to monitor traffic”.
- *Princeton University, Department of Mechanical and Aerospace Engineering, Controls Seminar*, Princeton, NJ, June 16th, 2009. Host: Professor Naomi Leonard. “Mobile Millennium using cell phones to monitor traffic”.
- *Stanford University, Department of Aeronautics and Astronautics, Controls Seminar*, Stanford, CA, May 20th, 2009. Host: Professor Per Enge. “Mobile Millennium using cell phones to monitor traffic”.
- *Microsoft Research Symposium*, Seattle, WA, May 14th, 2009. Host: Dr. Eric

Horvitz. “Mobile Millennium: using cell phones to monitor traffic”.

- *University of California, Davis, Civil and Environmental Engineering Department, Transportation Seminar*, Davis, CA, April 10th, 2009. Host: Professor Michael Zhang. “Mobile Millennium: using cell phones to monitor traffic”.
- *Eidgenossische Technische Hochschule Zurich (ETHZ), Electrical Engineering Department*, Zurich, Switzerland, March 24th, 2009. Host: Professor Manfred Morari. “Mobile Millennium: using cell phones to monitor traffic”.
- *University of Illinois at Urbana Champaign, Electrical Engineering Department, Coordinated Science Laboratory*, Urbana-Champaign, IL, March 18th, 2009. Host: Professor Daniel Liberzon. “Mobile Millennium: using cell phones to monitor traffic”.
- *Georgia Institute of Technology, Decision and Control Laboratory*, Atlanta, GA, March 13th, 2009. Host: Professor Eric Feron. “Mobile Millennium: using cell phones to monitor traffic”.
- *University of Pennsylvania, Electrical Engineering Department, Robotics Seminar*, Philadelphia, PA, March 5th, 2009. Host: Professor George Pappas. “Mobile Millennium: using cell phones to monitor traffic”.
- *UC Berkeley, Mathematics Department, Applied Mathematics Seminar*, Berkeley, CA, February 20th, 2009. Host: Professor Jon Wilkening. “Construction of lower semi continuous solutions to the Hamilton-Jacobi equation with internal boundary conditions: application to highway traffic monitoring”.
- *UCSD, Mechanical and Aerospace Engineering Department, Control Seminar*, La Jolla, CA, February 13th, 2009. Host: Professor Miroslav Krstic. “Mobile Millennium: using cell phones to monitor traffic”.
- *UC Berkeley, EECS-CEE-ME, Control Seminar* Berkeley, CA, February 27th, 2009. Host: Professor Ruzena Bajcsy. “Mobile Millennium: using cell phones to monitor traffic”.
- *Northwestern University, Civil Engineering, Transportation Seminar*, Evanston, IL, December 4th, 2008. Host: Professor Marco Nie. “Mobile Millennium: using cell phones to monitor traffic”.
- *UC Berkeley, CITRIS Research Exchange*, UC Berkeley, CA, April 16th, 2008. Host: Professor Paul Wright. “Integrating Motion into Infrastructure using Cell Phones”.
- *UC Berkeley, CITRIS - ITS seminar*, UC Berkeley, CA, February 8th, 2008. Host: Professor Paul Wright. “Mobile century: using GPS mobile phones as traffic sensors”.
- *UC Berkeley, CEE Department, ITS seminar*, UC Berkeley, CA, December 14th, 2007. Host: Professor Mark Hansen. “Travel time estimation using probe vehicle data: the Nokia N95 experience”.

Industry and government talks

- *Banatao Board Meeting*, Palo Alto, CA, November 23, 2010, “Mobile sensing in large scale infrastructure systems”.
- *California Department of Transportation*, Sacramento, CA, November 17, 2010,

"Future of data procurement policies".

- *Orange Institute*, San Francisco, CA, November 15, 2010, "Mobile sensing in large scale infrastructure systems".
- *INRIA-Berkeley meeting*. UC Berkeley, CA, November 12, 2010, "Mobile sensing in large scale infrastructure systems".
- *Ericsson-CITRIS meeting*, Berkeley, CA, October 15, 2010, "Mobile Millennium, using phones as traffic sensors".
- *The Bohemian Club*, San Francisco, CA, October 13, 2010, "Mobile Millennium, using phones as traffic sensors".
- *IBM-Caltrans Meeting*, UC Berkeley, Setpember 16, 2010, "Data fusion for traffic monitoring".
- *Telenav*, Santa Clara, CA, September 15, 2010, "Data fusion for traffic monitoring".
- *HP-CITRIS Meeting*, UC Berkeley, CA, September 10, 2010, "Mobile Millennium, using phones as traffic sensors".
- *IBM-CITRIS Meeting*, UC Berkeley, CA, August 27, 2010, "Mobile Millennium, using phones as traffic sensors".
- *Swedish DOT Meeting*, UC Berkeley, CA, August 5, 2010, "Mobile Millennium Stockholm".
- *Department of Homeland Security*, Moffett Field, CA, February 11, 2010, "Mobile Sensing for traffic, environmental monitoring and emergency response".
- *Agilent Technologies*, Santa Clara, CA, December 11th, 2009, "Mobile Millennium, using phones as traffic sensors".
- *Polaris*, Santa Clara, CA, November 19th, 2009, "Mobile Millennium, using phones as traffic sensors".
- *VOLPE Center (US DOT)*, Boston, MA, November 13th, 2009, "Mobile Millennium".
- *NAVTEQ*, Chicago, IL, September 28th, 2009, "Mobile Millennium".
- *T-Mobile labs*, Mountain View, CA, September 18th, 2009. "Mobile Millennium, using phones as traffic sensors".
- *NAVTEQ*, Chicago, IL, July 23rd, 2009. "Mobile Millennium".
- *BMW*, Palo Alto, CA, July 21th, 2009, "Mobile millennium: using smartphones to monitor traffic in privacy aware environments".
- *NAVTEQ - UC Berkeley traffic workshop*, UC Berkeley, CA, July 10th, 2009, "Mobile millennium: using smartphones to monitor traffic in privacy aware environments".
- *Energy Efficiency; Cyber-Physical Systems; Medical Devices & Systems*, Siemens Corporate Headquarters, Munich, Germany, May 28th, 2009. "Mobile Phones as Sensors for Improved Energy Efficiency".
- *Siemens – Berkeley Day*, Siemens Corporate Headquarters, Munich, Germany, May 27th, 2009. "Mobile Phones as Sensors for Improved Energy Efficiency".
- *VOLVO Centers of Excellence Symposium*, Gothenborg, Sweden, April 19th, 2009. "Mobile Millennium: using cell phones to monitor traffic".
- *South Bay Traffic Officials Association (SBTOA)*, San Jose, CA, March 10th, 2009. "Mobile Millennium: using cell phones to monitor traffic".

- *NAVTEQ*, Chicago, IL, December 4th, 2008. "Mobile Millennium: using cell phones to monitor traffic".
- *NAVTEQ Traffic Symposium*, "Mobile Millennium: using GPS to reconstruct traffic", New York, NY. November 17th, 2008.
- *California DOT meeting*, "Mobile Millennium: using cell phones to monitor traffic", Richmond Field Station, CA, August 12, 2008.
- *ITS Board of Directors meeting*, "Mobile Millennium: using cell phones to monitor traffic", Richmond Field Station, CA, August 6, 2008.

Talks at workshops, conferences, or meetings

- *Hyperbolic systems and control in networks*, Institut Henri Poincaré, Paris, France, October 20, 2010, "Optimization formulations for inverse modeling problems, with applications to Mobile Sensing"
- *CITRIS review*, Berkeley, CA, October 25, 2010, "Mobile sensing in large scale infrastructure systems".
- *ICRA10 workshop on Robotics and Intelligent Transportation Systems*, Anchorage, AK, May 7th, 2010, "Mobile Sensing for traffic and environmental monitoring" [Talk delivered by Andrew Tinka].
- *PATH-Tsinghua workshop*, PATH, Richmond Field Station, Richmond, CA, April 7th, 2010, "Mobile Millennium: using cell phones to monitor traffic".
- *AAAI Spring Symposia Series, Embedded Reasoning: Intelligence in Embedded Systems*, Stanford, CA, March 24th, 2010, "Mobile Millennium: using cell phones to monitor traffic".
- *2009 IEEE Conference on Decision and Control, Special SIAM session*, Shanghai, China, October 21rd, 2009, "Dirichlet Problems for Some Hamilton-Jacobi Equations with Inequality Constraints".
- *Position and Time, 3rd Annual Symposium*, Stanford University, Stanford Center for Navigation, Stanford, CA, October 21rd, 2009, "Mobile Millennium: using cell phones to monitor traffic".
- *TTI - Vanguard "More from Less"*, Jersey City, NJ, October 1st, 2009, "Mobile Millennium, using smartphones to monitor traffic".
- *2009 National Highway Data Workshop and Conference*, California Department of Transportation, Oakland, CA, September 23rd, 2009. "Mobile Millennium, using phones as traffic sensors
- *CalDay College of Engineering Speaker*, UC Berkeley, Berkeley, CA, April 18th, 2009. "Mobile Millennium: using GPS to reconstruct traffic".
- *15th World Congress on ITS*, Safe trip 21 session, New York, NY, November 18th, 2008, "Mobile Millennium: using GPS to reconstruct traffic".
- *SUPERB seminar*, UC Berkeley, Berkeley, CA, July 3rd, 2008. "Mobile Millennium: using GPS to reconstruct traffic".
- *CalDay*, UC Berkeley, Berkeley, CA, April 11th, 2008. "Mobile Century Traffic Project: GPS in your cell phone".
- *Vincent Lo & Shanghai – CITRIS*, UC Berkeley, Berkeley, CA, November 14th, 2007. "Large scale infrastructure systems monitoring using cellular phones".

- *Nokia delegation meeting – CITRIS*, UC Berkeley, Berkeley, CA, November 7th, 2007. "Large scale infrastructure systems monitoring using cellular phones".
- *CITRIS - Tekes meeting*, UC Berkeley, Berkeley, CA, October 1, 2007 . "Large scale infrastructure monitoring using mobile sensor networks".

4.3. Users, Phones and Functions

As a preamble we wish to thank Keith Jasper and his team for their contributions. Certain information from the National Evaluation of the Safe Trip-21 Initiative: *Draft* Evaluation Report: Mobile Millennium (as provided by Keith) has been used in this section.

4.3.1. User Downloads

The name Mobile Millennium was chosen because the project's goal was to enlist thousands of users (hence the use of the word Millennium). The project was also the next logical step from the Mobile Century project where 100 paid drivers participated in a test where their location and speed was tracked in order to estimate traffic conditions. The overall goal was to get as many people to voluntarily participate as possible. Thousands of users would provide statistical significance to results and would also explore voluntary usage of the traffic pilot software. The ultimate goal was 10,000 users permitting studies to test the system at scale and permitting informed decisions about technology and business.

At the end of the experiment it is estimated that around 5000 users used the pilot. These included users from the Berkeley website, Caltrans employees and users at Navteq and Nokia. About 2/3 of this number are registered users for whom we captured certain demographic details. This was only done for users within the Bay Area. All partners and sponsors were pleased with the number of participants.

For any particular time period, Nokia was able to generate usage reports out of system. Some examples include: how many times the application is started per week, what phones it comes from, how the application is actually opened, how much time the application is open for and did the phone get a GPS lock (phones were either turned off to quickly or were in an area where GPS coverage was unavailable). This is internal data to Nokia and is not available in this report.

Users registered for and received emails with download instruction from the UC Berkeley website. The following is a screenshot of the user registration page. In the side-bar on the left hand side of the registration page under the heading "Tested Devices" is a list of the mobile phones supported by Mobile Millennium.

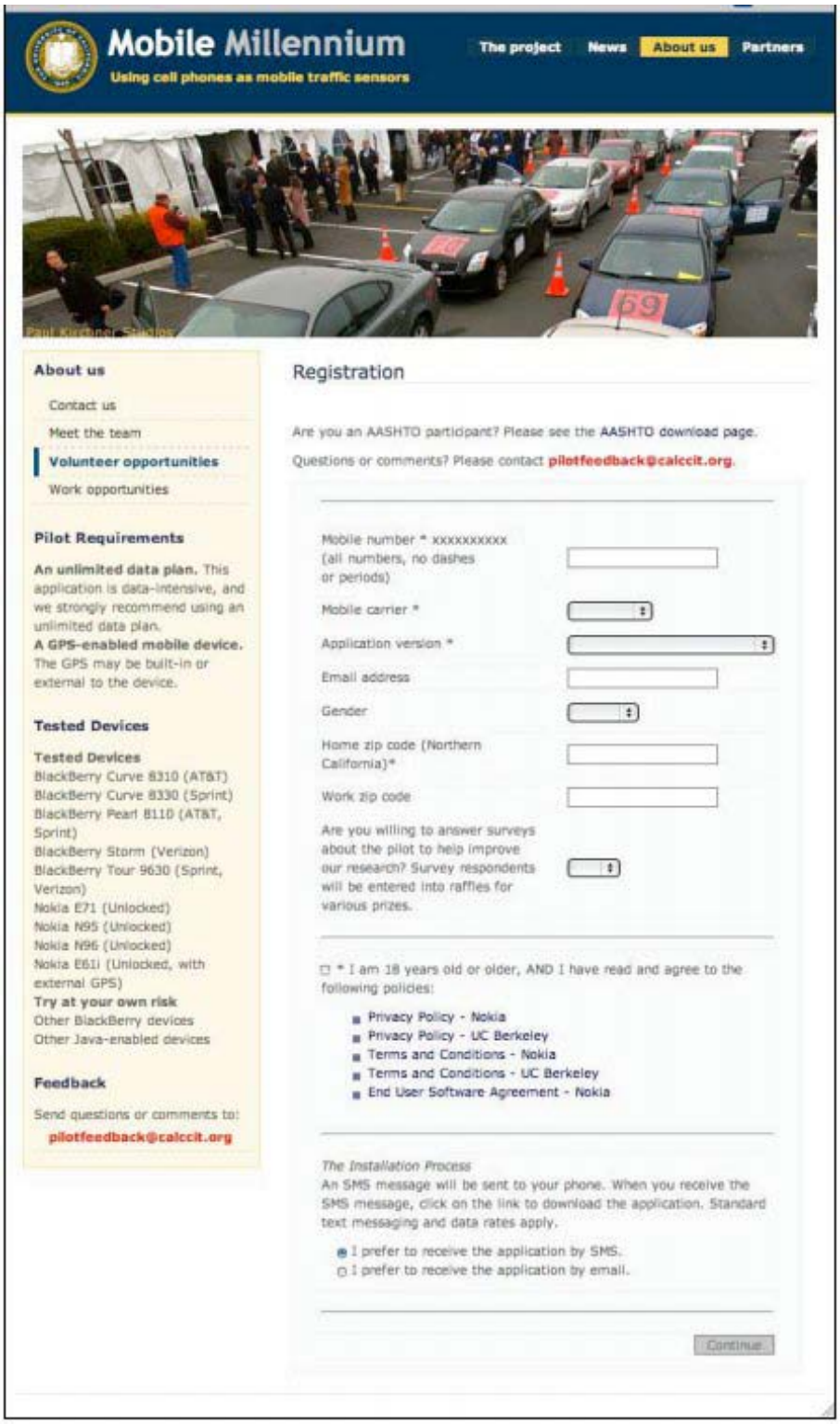


Figure 3.1. Mobile Millennium registration form.

When users registered for Mobile Millennium and downloaded the traffic application software, in addition to confirming they were 18 years old or older, and agreeing to various policy

statements, they were required to provide the telephone number for their mobile device, their mobile phone service carrier, the make/model of their mobile phone, and their home zip code. They were also invited to provide their email address, gender, and work zip code, although this was voluntary.

Registration information was collected and stored by CCIT, and forms the basis for the analysis of usage statistics. Prior to providing usage statistics, CCIT ensured that the database did not contain personally identifiable information, and also eliminated selected software download records that were known to relate to development team members' testing activities.

Usage statistics indicate that the software application was downloaded 2,241 times to unique mobile phone numbers between November 10, 2008 and November 10, 2009. The monthly distribution of downloads is provided in Figure 3.2. This analysis ignores additional downloads in early November 2008, prior to the official launch. It is understood that these downloads were by members of the development team, or by others affiliated with the partners.

The daily rate of downloads peaked at more than 70 per day in November 2008 (post-launch), tailing off to less than 4 downloads per day in each of the final eleven months. Given that the primary outreach effort was concentrated on the initial launch period, this is not unexpected. Indeed apart from some minor outreach by NRC and CCIT at the 2009 AASHTO Annual Meeting during late October 2009 in Palm Desert, California, there were no specific outreach campaigns after the launch. Despite the absence of any additional outreach efforts, users continued to download the application software throughout the operational period.

Month	Days downloads were available	Downloads per month	Cumulative Downloads	Avg. Downloads per day
November 2008	21	1498	66.8	71.3
December 2008	31	276	79.2	8.9
January 2009	31	114	84.2	3.7
Febuary 2009	28	58	86.8	2.1
March 2009	31	40	88.6	1.3
April 2009	30	58	91.2	1.9
May 2009	31	46	93.3	1.5
June 2009	30	43	95.2	1.4
July 2009	31	38	96.9	1.2
August 2009	31	20	97.8	0.6
September 2009	30	20	98.7	0.7
October 2009	31	25	99.8	0.8
November 2009	10	5	100.0	0.5

Figure 3.2. Download statistics.

Home zip code was a required field when users registered for the application software. The vast majority of downloads were by users who declared their home zip code to be in a range from

936xx to 959xx (see Figure 3.3). This corresponds to the Bay Area/Sacramento region. In addition, there were a small number of downloads from further afield, mostly in California, but with a few out of state. Downloads are spread across more than 400 zip codes, with the greatest concentration around the immediate Bay Area.

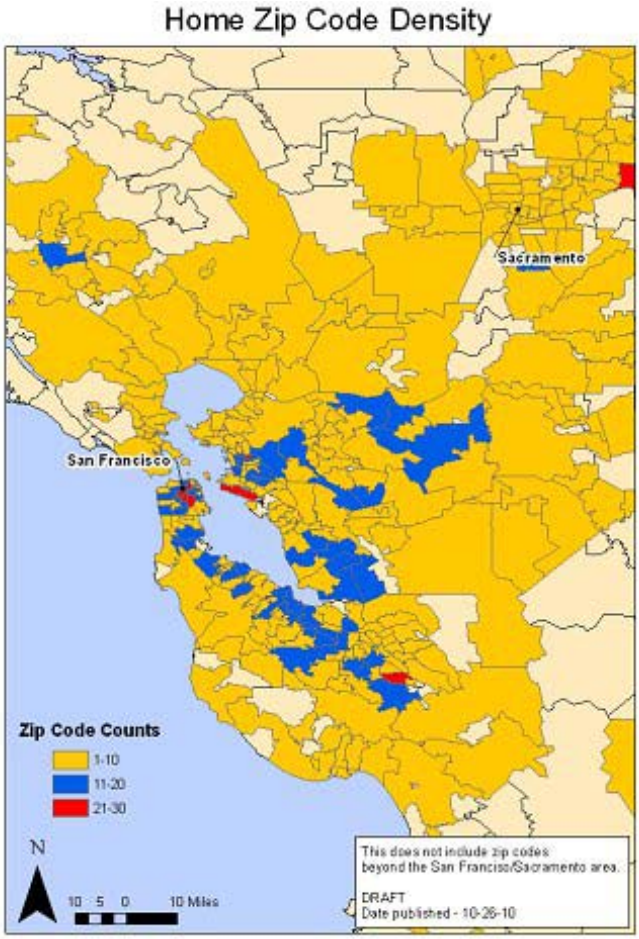


Figure 3.3. Number of registered users for each home zip code.

Work zip code was an optional field when users registered for the application software, and were not provided for 400 of the application software downloads. Work zip codes were more concentrated than home zip codes, with the greatest concentration around the immediate Bay Area, notably in the San Jose area (see Figure 3.4). Interestingly, downloads are spread across more than 485 zip codes, which is more than for home zip codes. Work zip codes associated with more than 160 downloads were for locations outside of the Bay Area/Sacramento region, i.e. not in the zip code range from 936xx to 959xx. Many of these were for out of state locations.

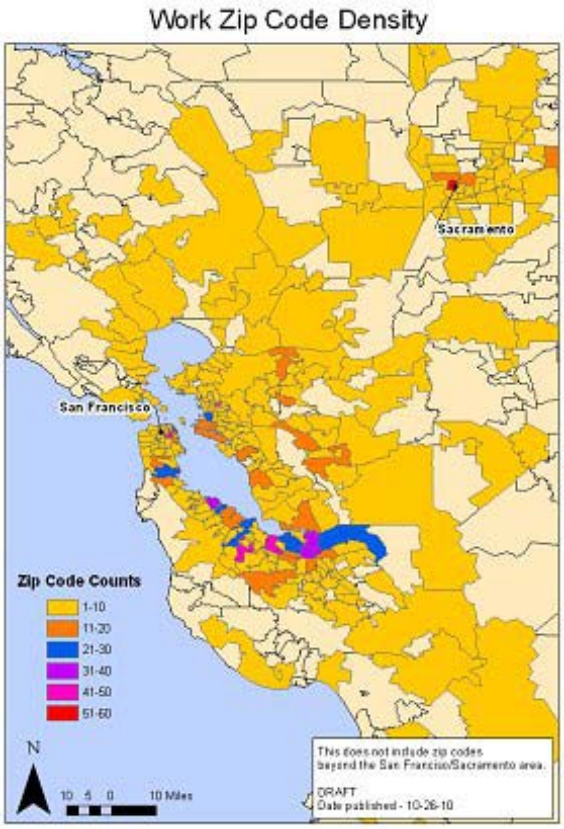


Figure 3.4. Number of registered users for each work zip code.

High numbers of downloads are associated with Palo Alto, Berkeley, and Sacramento, where NRC, CCIT and Caltrans are located. Whether this highlights increased awareness of Mobile Millennium or possible distortion in the usage statistics related to development team activity cannot be determined by the Evaluation Team without gaining access to personally identifiable information.

4.3.2. Phones and Carriers supported

At the time Mobile Millennium started cell phone technology was just approaching the "smart phone" era. There were four requirements that needed to be met in order for a phone to be included in the pilot.

- The ability to run a Java application.
- The ability to display a traffic map.
- The ability for the Java application to access GPS information. This is not just a technical issue. Some carriers control what hardware an application can use.
- Approval from the phone companies and carriers as appropriate.

This last proved to be difficult to meet with regard to the Apple iPhone. The Mobile Millennium project had a strong desire to provide a client application for the Apple iPhone. However for indeterminate reasons (possible due to legal wrangling between Nokia and Apple) and despite continuing efforts we were not able to get the phone client application approved for the iPhone. This was unfortunate as we would likely have reached larger download numbers if the iPhone was supported. Throughout the project a list registrants with iPhones was maintained in the hope that they could be contacted when the iPhone application became available.

The following phones were supported:

- BlackBerry
    - Curve 8310 - AT&T
    - Curve 8330 - Sprint
    - Pearl 8110 - AT&T & Sprint
    - Storm - Verizon
    - Tour 9630 - Sprint and Verizon
- Nokia
    - E71
    - N95
    - N96
    - E61
- Possible
    - Other Blackberry devices
    - Other Java-enabled devices

4.3.3. System Functionality

The basic functions of the system were that the client would read GPS locations, save the time and position data and then send information when privacy preserving virtual trip lines (VTLs) were crossed. VTLs are discussed in more detail in other parts of this report. The data would be fed to UC Berkeley software that would estimate traffic speed. These estimates would be mapped to colors (green for fast, red for slow), overlaid on a map of roads of the bay area and then the map would be presented in real time to the commuter on the phone.

There were a number of user interface challenges with this approach and not all of them were resolved (as the surveys discussed later show).

4.4. Legal Items

There were a number of legal items to be worked out as part of the Mobile Millennium project's outreach effort with users. These include:

- Human Subjects Review: As required by federal law and university policy, the University of California, Berkeley has a Committee for the Protection of Human Subjects (CPHS) that serves as an Institutional Review Board (IRB) to review all research projects involving human subjects. The test protocols for Mobile Millennium and the Networked Traveler projects were reviewed and approved by CPHS.
- UC Berkeley Privacy Policy: This document publicly stated how any information collected on the web site would be used.
- Nokia's Privacy Policy: This document explained what types of information Nokia collected in connection with its products and services and how it processed such information.
- UC Berkeley Web Site Terms of Service: This document basically states that the UC is not responsible for any negative side effects (viruses for example) that result from using the site.
- Nokia's Terms of Service: Basic information on usage of the client. For example you must be over 13 years of age.
- Nokia's End User Software Agreement: This is a standard software usage document.

The full text of each of these documents follows.

4.4.1. Human Subjects Review (UC)

UNIVERSITY OF CALIFORNIA AT BERKELEY

BERKELEY • DAVIS • IRVINE • LOS ANGELES • MERCED • RIVERSIDE • SAN DIEGO

SAN FRANCISCO • SANTA BARBARA • SANTA CRUZ

OFFICE FOR THE PROTECTION OF HUMAN SUBJECTS
University of California, Berkeley
2150 Shattuck Avenue, Suite 313
Berkeley, CA 94704 -5940

(510) 642-7461
Fax: (510) 643-6272
Website: http://cphs.berkeley.edu
FWA#00006252

December 17, 2008

ALEXANDRE BAYEN (bayen@berkeley.edu)
Civil and Env. Eng.
711 Davis Hall, CEE MC# 1710
Berkeley, CA 94720

RE: **CPHS Protocol #2008-11-7**
"Mobile Millennium" - Faculty Research - Caltrans RTA 65A0301 - Civil and Env. Eng.

Dear Professor BAYEN:

Thank you for the statement and request for exemption that you submitted to the Committee for the above-referenced project. Your submission has been reviewed and granted exemption, as it satisfies the Committee's requirements under category 2b of the federal regulations. Accordingly, the project is exempt from full Committee review provided that there are no changes in the use of human subjects.

Please note that although your research has been deemed exempt from full committee and subcommittee review, you still have a responsibility to protect your subjects, and the research should be conducted in accordance with the principles of the Belmont Report. Download the Belmont Report at this link:
http://www.hhs.gov/ohrp/humansubjects/guidance/belmont.htm.

If you have any questions about this matter, please contact the OPHS staff at 642-7461; FAX 643-6272; E-Mail ophs@berkeley.edu.

Sincerely,

Rebecca Armstrong, D.V.M, Ph.D
Director, Office for the Protection of Human Subjects

RA: AT

Cc: Graduate Division (degrees@berkeley.edu)
Tom West (tomwest@calccit.org)
Dan Work (dbwork@berkeley.edu)

Figure 4.1. Protocol submission (response from the CPHS).

Privacy Policy (UC)

Privacy Statement for University of California, Berkeley Websites

Policy

The University of California, Berkeley is committed to protecting the privacy and accuracy of your confidential information to the extent possible, subject to provisions of state and federal law. Other than as required by laws that guarantee public access to certain types of information, or in response to subpoenas or other legal instruments that authorize access, personally-identifiable information is not actively shared. In particular, we do not re-distribute or sell personal information collected on our web servers.

Information collected

UC Berkeley websites may collect personal information such as name, address, e-mail address, telephone number(s), and/or educational interests. Such personal information may be requested by us for research, public service or teaching programs, or for administrative purposes. Additional personal information, such as credit card account information, may be requested for purchases or enrollment purposes.

Web servers typically collect, at least temporarily, the following information: Internet Protocol (IP) address of computer being used; web pages requested; referring web page; browser used; date and time. UC Berkeley may collect statistics identifying particular IP addresses from which our websites are accessed.

Use of collected information

UC Berkeley may use personal information collected from websites for the purpose of future communication back to online enrollees, in order to keep you informed of such activities as campus programs, symposia and/or special events, but only if you are provided the opportunity to opt out of that type of use.

UC Berkeley may use browser-IP-address information and anonymous-browser history to report information about site accesses and for profiling purposes. This information is generally used to improve Web presentation and utilization. The campus also may use IP address information for troubleshooting purposes.

Some UC Berkeley online activity sites may use “cookies” in order to deliver web content specific to individual users’ interests or to keep track of online purchasing transactions. Sensitive personal information is not stored within cookies.

Distribution of collected information:

By using this site, you are consenting to the release of your personal information to Nokia for

purposes of this project. Nokia's Privacy Policy is at

`http://traffic.berkeley.edu/pilot/nokia_privacy.html`. Please do not use this site if you do not wish your information to be released by UC Berkeley to Nokia for these purposes. Other than to Nokia, UC Berkeley will not distribute or sell personal information to third-party organizations.

Other than to Nokia, UC Berkeley will not disclose, without your consent, personal information collected about you, except for certain explicit circumstances in which disclosure is required by law.

Individual choice

Individuals who wish to use methods other than online enrollment may submit requests by email or U.S. mail addressed to the UC Berkeley organization responsible for the website.

Access to your own information

Questions regarding users' rights to review, modify or delete their previously provided personal information should be directed to the campus organization to which they provided the information. Any disputes will be resolved under existing records regulations applicable to UC Berkeley.

Additional Information

For more detailed information about requirements for campus online activities see the *e-Berkeley Policy* section on *Privacy and Confidentiality of Information*.

Responsibility for sites linked to

While using a UC Berkeley website, you may encounter hypertext links to the Web pages of organizations not directly affiliated with UC Berkeley. UC Berkeley does not control the content or information practices of external organizations. We recommend you review the privacy statements of these organizations.

4.4.2. Privacy Policy - Nokia

WE CARE ABOUT YOUR PRIVACY

Nokia is committed to protecting your privacy and to complying with applicable data protection and privacy laws. We hope that this Privacy Policy ("Policy") helps you understand what kind of information we collect in connection with our products and services and how we process such information. "Nokia" refers to Nokia Corporation, including its affiliates (also referred to as "we", "us", or "our"). This Policy applies to the Mobile Millennium project. It is not applicable to other Nokia services, products or projects, unless incorporated by specific reference to this Policy.

Privacy by Design™

The Mobile Millennium project employs a holistic approach to the collection, use and disclosure of your personal information. Privacy by Design follows three major principles:

- Build privacy into the system.
- Minimize the amount of sensitive information that is collected, transferred, and stored.
- Discard unnecessary or revealing information at every step.

Why are we collecting GPS information?

The purpose of this mobile map client application is to develop accurate, fine-grained traffic models. Currently, traffic information is collected by sensors on, around, and underneath major roads. These sensors are expensive to install and maintain and, in many cases, are not present on many roads.

The purpose of the Mobile Millennium project is to explore the possibility of monitoring traffic conditions through GPS–enabled phones. When combined with our mobile map client application, your GPS-enabled phone can act as traffic sensor. If there are many phones collecting traffic data, we can build better traffic models. As a result, you (and other drivers) can receive more comprehensive, up-to-date traffic information.

What information do we collect?

When you register on Berkeley's Mobile Millennium website (`http://www.traffic.berkeley.edu`), Berkeley collects some personal information, such as your phone number and email address ("Registration Information"). Berkeley's treatment of the Registration Information is governed by Berkeley's privacy policy, located at

`http://traffic.berkeley.edu/pilot/UCB_privacy.html`. Berkeley shares Registration Information with Nokia. Nokia's treatment of the Registration Information is governed by this privacy policy. Nokia uses the Registration Information to send you, via SMS

or email, a link to a web page from which you can download Nokia's mobile traffic client. If you opted-in to receiving surveys about the Mobile Millennium Traffic Pilot during Registration, we may also contact you when we run these surveys.

When you use the Mobile Millennium Traffic Pilot, your mobile device will be sending anonymous GPS data to us. GPS data includes your GPS coordinates, velocity, and a time stamp ("GPS Data"). Nokia does not link your GPS Data to your Registration Information.

GPS Data is only gathered when you cross a Trip Line™. A Trip Line is a virtual location that is important for traffic, such as an intersection or a freeway exit. Trip Lines are placed on major roads only; there are no Trip Lines on small residential roads.

From Nokia's point of view, we only know that someone crossed a Trip Line at a certain time and speed. For example, we may know that a user crossed the intersection of University and Shattuck at 2 pm at 25 mph. If there is only one user in Berkeley at that time, one could speculate that the same person crossed the intersection of University and Sacramento a few minutes later. However, there is no way to associate the GPS Data with the Registration Information in order to identify the person as Joe the Plumber. However, if there is more than one user in the same area, we cannot distinguish between who continued to University and Sacramento and who turned off University onto MLK and who turned off MLK onto University and who drove on MLK past University.

After collected GPS Data is sent to our servers, it is periodically and automatically deleted from your mobile device. Your mobile device does not store GPS Data on a persistent basis. When you exit the application, the location shown on your screen will be stored. This location may or may not be related to your actual physical location.

In the future, we may offer personalized traffic reports. Personalization (which is optional) requires sharing personally identifying data to fulfill your requests.

Finally, we may collect anonymous usage statistics solely in order to help us improve the application and our services, such as the installation date of the application, the number of times that the application has been opened since installation, and the total amount of time that the application has been opened, that GPS is on, that GPS is on and location data is not available, the number of network errors the application encounters, and other usage information. Usage statistics are very general. Nokia does not link usage data to your Registration Information or GPS Data.

Can anyone link my GPS Data back to me?

- We encrypt your GPS Data before it is sent to us from your mobile device in order to prevent access to your GPS Data in transit, even if someone is eavesdropping on the transmission.
- The GPS Data that is gathered is first anonymized and then ultimately aggregated There is no way to recreate the GPS Data from any given individual once it has been aggregated. We never link your Registration Information to your GPS Data.

- It is not possible to infer where you live or the origin or destination of your trips. The way the system is designed (e.g. anonymized GPS Data, limited number of TripLines), we never see the actual origin or destination point of your journeys. Also, with a reasonable number of participants in a geographical area, it is not possible to associate any single individual with a trip from point A to point B.

We are working with a major university on the Mobile Millennium pilot project that incorporates the privacy features described in this policy. Other experts who feel they could contribute to the project are welcome to contact us to participate in this analysis.

With whom will the GPS Data be shared?

The GPS Data that is gathered from you and other users is anonymized and then aggregated and cannot be linked back to any individual. Aggregated data may be shared with the user community, e.g., as the red, yellow, and green traffic lines that you see in your application. Aggregated data may also be shared with or sold to other entities, such as the Department of Transportation ("DOT") or the California Center for Innovative Transportation ("CCIT") for research and analytical purposes.

With whom will your Registration Information be shared?

We do not sell, lease, rent or otherwise disclose your Registration Information to third parties unless otherwise stated below.

- *Consent* We may share your Registration Information if we have your consent to do so.
- *Nokia companies and authorized third parties* We may share your Registration Information with other Nokia companies or authorized third parties who process Registration Information for Nokia for the purposes described in this Policy. Such parties are not permitted to use your Registration Information for other purposes, and we require them to act consistently with this Policy and to use appropriate security measures to protect your Registration Information.
- *Mandatory disclosures* We may be obligated by mandatory law to disclose your Registration Information to certain authorities or other third parties, for example, to law enforcement agencies in the countries where we or third parties acting on our behalf operate. We may also disclose and otherwise process your Registration Information in accordance with applicable law to defend Nokia's legitimate interests, for example, in civil or criminal legal proceedings.

The Purposes for which We Process Your Registration Information

Nokia processes end user Registration Information for the purposes described in this Policy and/or the Privacy by Design description. Please note that one or more purposes may apply simultaneously and in the event of a conflict between the Privacy by Design description and this Policy, the Policy shall control.

- *Provision of products and services* We may use your Registration Information to fulfill your requests, process your order or as otherwise may be necessary to perform or enforce the contract between you and Nokia, to ensure the functionality and security of our products and services, to identify you and to prevent and investigate fraud and other misuses.
- *Development of products and services* We may use your Registration Information to develop our products and/or services. However, for the most part we only use aggregate and statistical information in the development of our products and services. We may create aggregate and statistical information based on your Registration Information.
- *Communicating with you* We may use your Registration Information to communicate with you, for example, to provide information relating to our products and/or services you are using or to contact you for customer satisfaction queries. We may use your Registration Information for research purposes; however, will not disclose your Registration Information to third parties for non-research purposes without your prior consent.

Data Quality

We take reasonable steps to keep the Registration Information we possess accurate and up-to-date and to delete out of date or otherwise incorrect or unnecessary Registration Information.

Security

While there are always risks associated with providing Registration Information, whether in person, by phone, via the Internet or otherwise, and no technology is completely safe or "tamper" or "hacker" proof, Nokia takes appropriate technical and organizational information security measures to prevent and minimize such risks.

Such measures include, where appropriate, the use of firewalls, secure server facilities, encryption, implementing proper access rights management systems and processes, careful selection of processors and other technically and commercially reasonable measures to provide appropriate protection for your Registration Information against unauthorized use or disclosure. Where appropriate, we may also take back-up copies and use other such means to prevent accidental damage or destruction to your Registration Information. If a particular part of a Nokia website supports on-line transactions, we will use an industry standard security measure, such as the one available through "Secure Sockets Layer" ("SSL"), to protect the confidentiality and security of online transactions.

Your Rights

In case you wish to know what Registration Information we hold about you or you wish to replenish, rectify, anonymized or delete any incomplete, incorrect or outdated Registration Information, or you wish us to cease processing your Registration Information for the purpose of sending promotional materials or direct marketing or for the performance of market research or on other compelling legal grounds, you may, as appropriate and in accordance with applicable

law, exercise such rights by contacting us through the contact points referred to below. In some cases, especially if you wish us to delete or cease the processing of your Registration Information, this may also mean that we may not be able to continue to provide the services to you. We encourage you to use available profile management tools for the above purposes as such tools often provide you with direct access to your Registration Information and allow you to effectively manage it.

Please note that Nokia may need to identify you and to ask for additional information in order to be able to fulfill your above request. Please also note that applicable law may contain restrictions and other provisions that relate to your above rights.

The Controller of Your Registration Information and Contact Details

Nokia Corporation of Keilalahdentie 4, 02150 Espoo, Finland shall be the controller of your Registration Information.

In matters pertaining to Nokia's privacy practices you may also contact us at:

Nokia Corporation
c/o Privacy
Keilalahdentie 4
02150 Espoo
Finland

4.4.3. Terms of Service - UC

Terms and Conditions of Use

University of California, Berkeley

1. Terms and Conditions of Use:

By using this site and all other Regents of the University of California sites, referred to as “these sites,” you agree to these terms of use. If you do not agree to these terms of use, please do not use these sites. These sites are owned and operated by Regents of the University of California (referred to as “University,” “we,” “us,” or “our” herein). We reserve the right, at our discretion, to change, modify, add, or remove portions of these terms at any time. Please check these terms periodically for changes. Your continued use of these sites following the posting of changes to these terms will mean you accept those changes.

2. Disclaimer:

THE REGENTS OF THE UNIVERSITY OF CALIFORNIA MAKE NO REPRESENTATION OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND SPECIFICALLY DISCLAIM ANY IMPLIED WARRANTIES OR

MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE. Further, the Regents of the University of California reserve the right to provide revised software and/or documentation and to make changes from time to time in the content hereof without obligation of the Regents of the University of California to notify any person of such revision or change.

3. University Internet Sites:

The majority of University web sites do not represent the University itself in any way. Any content contained or accessible from these sites, regardless of access method (http, ftp, etc.), does not reflect the views of the Regents of the University of California. The Regents does not endorse, warrant, or otherwise take responsibility for the contents of any material accessible from these sites.

4. Endorsements:

Links from any web site located on a University server to any non-University site do not imply University endorsement of the site’s products or services. References to non-University products, services, or organizations do not imply University endorsement of the products, services, or organizations.

5. General Liability:

The materials on University’s Internet site are provided “as is” and without warranties of any

kind either express or implied. To the fullest extent permissible pursuant to applicable law, University disclaims all warranties, express or implied, including, but not limited to, implied warranties of merchantability and fitness for a particular purpose. University does not warrant that the services will meet your requirements, the functions contained in the materials will be uninterrupted or error-free, that defects will be corrected, or that this site or the server that makes it available are free of viruses or other harmful components. University does not warrant or make any representations regarding the use or the results of the use of the materials in this site in terms of their correctness, accuracy, quality, reliability, appropriateness for a particular task or application, or otherwise. No oral or written information or advice given by service provider or its authorized representatives shall create a warranty. You are entirely responsible for and assume all risk for use of the service and the materials on this site. In no event will University be liable for any special, indirect, incidental, or consequential damages even if University has been advised of the possibility of such damages. You should not use the service, web site or the material contained therein where damage could result if an error occurred. University does not warrant or represent that its security procedures will prevent the loss of or improper access to your data. University is not responsible for transmission errors or corruption or security of information carried over telecommunication/data lines.

6. Limitation of Liability:

This disclaimer of liability applies to any general, special or consequential damages or injury caused by any negligence, failure of performance, error, omission, interruption, deletion, defect, delay in operation or transmission, computer virus, communication line failure, theft or destruction or unauthorized access to, alteration, or use, whether for breach of contract, tortuous behavior, negligence, or under any other cause of action. You specifically acknowledge that University is not liable for any defamatory, offensive, infringing or illegal materials or conduct on the part of, or attributable to, any third party. University reserves the right to remove such materials from its web site without liability.

Under no circumstances, including, but not limited to, negligence, shall University be liable for any special or consequential damages that result from the use of, or the inability to use, the materials in this site, even if University or a University authorized representative has been advised of the possibility of such damages. Applicable law may not allow the limitation or exclusion of liability or incidental or consequential damages, so the above limitation or exclusion may not apply to you. In no event shall University's total liability to you for all damages, losses, and causes of action (whether in contract, tort including, but not limited to, negligence, or otherwise) exceed the amount paid by you, if any, for accessing this site.

7. Disclaimer of Responsibility for Link:

University's web site contains links to sites that are not maintained by University. University is not responsible for the content of those sites. Other linked sites may change without our knowledge. The inclusion of such links and frames in the University web site does not imply University's endorsement of the linked or framed sites or their content.

University makes no representations whatsoever about any other web site that you may access

through this one. When you access a non- University site, that site is independent from University. University has no control over the content on that web site. In addition, a link to a non-University site does not mean that University endorses or accepts any responsibility for the content, or the use, of such site. It is your responsibility to take precautions to ensure that whatever you select for your use is free of such items as viruses, worms, Trojan horses and other items of a destructive nature. In no event will University be liable to any party for any direct, indirect, special or other consequential damages for any use of this website, or on any other linked website, including, without limitation, any lost profits, business interruption, loss of programs or other data on your information handling system or otherwise, even if we are expressly advised of the possibility of such damages.

In these terms of service, “University”, “we”, “us”, and “our” includes Nokia Corporation and its affiliates (“Nokia”), to the extent the terms are applied to the Mobile Millenium Traffic Pilot and the website related to the pilot, provided that Nokia does not take on any obligations with respect to such website. Nokia is cooperating with the University in running the pilot.

4.4.4. Terms of Service - Nokia

1. These terms and conditions (the "Terms") and the Nokia privacy policy available at http://traffic.berkeley.edu/pilot/nokia_privacy.html("Privacy Policy") shall govern the use by you of the Nokia mobile traffic navigation services (the "Service"), operated and provided by Nokia Corporation, its subsidiaries, or affiliates ("Nokia") and together shall constitute an agreement between you and Nokia of Espoo, Finland. These Terms are only applicable to the Service and do not apply to any other Nokia services, websites, business channels or other business operations, unless explicitly so stated. By accessing and using the Service, you expressly agree to the following Terms. If you do not agree to the following Terms, please note that you are not allowed to use the Service.

2. The Service utilizes mobile map client software ("Software") which operates on GPS-equipped mobile phones in order to collect data for purposes of processing and rendering relevant, live traffic information. The Software and its servers and other supporting infrastructure will collect, process, collate, and render for end users a large volume of data concerning such end users' traffic activities and their use of traffic and other information. In order to use the Services, you need to make sure that the Software is installed and operating on your GPS-equipped mobile phone and that your GPS is turned on. Please note that GPS-related services may be very data intensive and that you are responsible for all costs incurred in connection with your use of the Software and Service, including but not limited to any phone, data plan or other usage charges. Nokia strongly recommends that you subscribe to an unlimited data plan. The Software which you use may from time to time automatically download and install updates from Nokia. These updates are designed to improve the Services and may take the form of bug fixes, enhanced functions, and/or new software versions. You agree to receive such updates as part of your use of the Service.

3. The content of the Service is ©2008 Nokia Corporation (or the respective Nokia suppliers or other third parties). Any rights not expressly granted herein are reserved. Your use of the Service and the content therein, is restricted to private, non-commercial use. Reproduction, transfer, distribution or storage of part or all of the Service or its contents in any form without the prior written permission of Nokia is prohibited.

4. For your easy accessibility, Nokia may include links to sites on the Internet that are owned or operated by third parties. Upon following a link to such third-party site, you will need to review and comply with that site's rules of use before using such site. You should be aware that third-party content presented to you as part of the Service may be protected by intellectual property rights which are owned by the third parties who provide that content to Nokia. You may not modify, sell, distribute or create derivative works based on this content unless you have been specifically permitted to do so in a separate agreement by Nokia or by the owners of that content, as the case may be. You agree that Nokia has no control over the third party content of third-party sites and does not assume any responsibility for services provided or material created or published by such sites. You acknowledge and agree that Nokia is not liable for any loss or damage which may be incurred by you as a result of the availability of those external sites or resources, or as a result of any reliance placed by you on the completeness, accuracy or existence of

any advertising, products or other materials on, or available from, such sites or resources. A link to a third-party site does not imply that Nokia endorses the site or the products or services referenced in the site.

5. To use the Service, you must be at least thirteen years of age. If you are at least thirteen years of age but a minor where you live, you must have permission from your parent or legal guardian to use the Service.

6. You agree to provide and maintain accurate, current and complete information when registering for the Service.

7. Nokia is a registered trademark of Nokia Corporation. Nokia's product names are either trademarks or registered trademarks of Nokia. Other product and company names mentioned herein may be trademarks or trade names of their respective owners. The Service and related software are protected under U.S. and international copyright laws and you are hereby notified that copyrights are claimed by Nokia Corporation and/or its affiliates. Your access to the Service should not be construed as granting, by implication, estoppel or otherwise, any license or right to use any marks appearing on the Service without the prior written consent of Nokia or the third party owner thereof. You agree to abide by any and all copyright notices displayed on the Service. You also agree not to attempt to decompile or disassemble, reverse engineer or otherwise attempt to discover any source code contained in the Service.

8. In connection with the Service and Software, you: (a) hereby consent to the data being used and processed by Nokia in order to improve the Software and Service and related mobile traffic software; (b) you will not use the Software or Service to perform any unauthorized data collection, extraction or mining or gain or attempt to gain unauthorized access to Nokia's or any Service visitor's computer system; (c) will not submit a virus, Trojan horse, "sniffer" routine, backdoors, "robots", "spiders", worms, time bombs, bots, or other harmful software code, file, program or programming routine, or other contaminating or destructive features; (d) will not impose an unreasonably large load on the Service or Nokia's computer systems, or use a computer programming routine, file or device to damage or interfere with the operation of the Service; (e) will not spam Nokia; and (f) will comply with all applicable laws, including all applicable export control laws and not to transfer or make available to a destination prohibited by such laws any content, software, encryption or materials subject to restrictions under such laws.

9. THE SERVICE IS PROVIDED ON AN "AS IS" AND "AS AVAILABLE" BASIS. NOKIA DOES NOT WARRANT THAT THE SERVICE WILL BE UNINTERRUPTED OR ERROR OR VIRUS-FREE. NO WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO WARRANTIES OF TITLE OR NON-INFRINGEMENT OR IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, IS MADE IN RELATION TO THE AVAILABILITY, ACCURACY, RELIABILITY OR CONTENT OF THE SERVICE. YOU EXPRESSLY AGREE AND ACKNOWLEDGE THAT THE USE OF THE SERVICE IS AT YOUR SOLE RISK AND THAT YOU MAY BE EXPOSED TO CONTENT FROM VARIOUS SOURCES. NOKIA ASSUMES NO LIABILITY OR RESPONSIBILITY FOR ANY CONTENT OR INFORMATION PROVIDED BY OTHER USERS OF THE SERVICES.

10. NOKIA SHALL NOT BE LIABLE FOR ANY DIRECT, INDIRECT,

INCIDENTAL, SPECIAL, PUNITIVE OR CONSEQUENTIAL OR OTHER DAMAGES, LOST PROFITS, BUSINESS REVENUE, GOODWILL, ANTICIPATED SAVINGS, OR DATA, CAUSED BY THE USE OF OR INABILITY TO USE THE SERVICE EVEN IF THE POSSIBILITY OF SUCH DAMAGES HAS BEEN ADVISED. SOME JURISDICTIONS DO NOT ALLOW EXCLUSION OF CERTAIN WARRANTIES OR LIMITATIONS OF LIABILITY, SO THE EXCLUSIONS OR LIMITATIONS IN THE TERMS MAY NOT APPLY TO YOU. THE LIABILITY IS IN SUCH CASE LIMITED TO THE GREATEST EXTENT PERMITTED BY LAW.

11. The Privacy Policy and additional provisions in the Terms govern use of your personal data. As explained in more detail in the Privacy Policy, your personal data will be processed for the following purposes: to improve Nokia products and services; to communicate with you (for example, to provide surveys for research purposes); to personalize Nokia's offering; to comply with mandatory legal requirements and in connection with law enforcement, civil or criminal legal proceedings. If Nokia decides to sell, buy, merge or otherwise reorganize its businesses in some countries, this may involve Nokia disclosing personal data to prospective or actual purchasers, or receiving it from sellers, or their advisers. Transfers of your personal data as described above may include transfers or disclosures outside the European Economic Area and in the United States of America or other regions where adequate protection for your personal data may not be guaranteed. In such a case Nokia takes appropriate measures to provide adequate protection for your personal data.

If you want to access, update or delete any personal data, or you want Nokia to stop processing your personal data for marketing purposes, contact us at:

Nokia Corporation
c/o Privacy
Keilalahdentie 4
02150 Espoo
Finland

12. The Service may not be available in some countries and may be provided only in selected languages. The Service may be network dependent, contact your network service provider for more information. Nokia reserves the right, in its sole discretion, to change, improve and correct the Service. The Service may not be available during maintenance breaks and other times. Nokia may prevent your access to the Service or any part thereof if it believes you have breached the Terms. Nokia may also decide to discontinue the Service or any part thereof in its sole discretion. In such case you will be provided a prior notification.

13. YOU AGREE TO DEFEND, INDEMNIFY AND HOLD HARMLESS NOKIA, OFFICERS, DIRECTORS, EMPLOYEES AND AGENTS FROM AND AGAINST ANY AND ALL THIRD PARTY CLAIMS AND ALL LIABILITIES, ASSESSMENTS, LOSSES, COSTS OR DAMAGES RESULTING FROM OR ARISING OUT OF: I) YOUR BREACH OF THE TERMS; OR II) YOUR INFRINGEMENT OR VIOLATION OF ANY INTELLECTUAL PROPERTY, OTHER RIGHTS OR PRIVACY OF A THIRD PARTY.

14. Nokia reserves the right to modify the Terms at any time without prior notice. Such changed Terms will be effective once posted to the "About" or equivalent area of the Service, or as otherwise noticed to you. You are responsible for regularly reviewing the Terms.

You understand and agree that if you use the Service after the date on which the Terms have changed, Nokia will treat your use as acceptance of the updated Terms.

15. You may stop using the Service at any time. You do not need to specifically inform Nokia when you stop using the Service. The Terms will continue to apply until terminated by Nokia as set out below. Nokia may at any time suspend Service and/or terminate its legal agreement with you if: (a) you have breached any provision of the Terms; (b) Nokia is required to do so by law; (c) Nokia is planning to no longer provide the Service to users in the country in which you are located or use the service; or (d) the provision of the Service is, in Nokia's opinion, no longer commercially viable.

16. You may not assign any rights or obligations under these Terms without Nokia's prior written consent. Nokia may assign these Terms or its rights and obligations hereunder to a third party without your prior notice or approval.

17. The Terms shall be governed by the laws of New York without regard to its conflicts of law provisions. Any and all claims, except claims for monies due to Nokia, arising out of or relating to the purchase of products, content or services by you shall be barred unless an action or legal proceeding is commenced within eighteen (18) months after the date you or Nokia knew or should have known of the facts giving rise to such claim. Any dispute relating in any way to your use of the Service, or your materials, product or service order from the Service, shall be submitted (together with any counterclaims and disputes under or in connection with other transactions or agreements between you and Nokia) to final and binding, confidential arbitration in Westchester County, New York, except that Nokia may seek injunctive or other appropriate relief if you have violated or threatened to violate any intellectual property rights. All matters relating to arbitration shall be governed by the Federal Arbitration Act (9 U.S.C. §1 et. seq.). Arbitration shall be conducted by a single arbitrator under the then prevailing Wireless Arbitration Rules of the American Arbitration Association ("AAA") (except as such rules may be modified by the provisions of the Terms), Each Party shall submit or file any claim which would constitute a compulsory counterclaim (as defined by Rule 13 of the Federal Rules of Civil Procedure) within the same proceeding as the claim to which it relates. Any such claim which is not submitted or filed in such proceeding shall be barred. Subject to any terms contained in the Terms limiting or excluding damages, the arbitrator may award any relief that the arbitrator deems proper, including without limitation equitable relief, provided that no award of exemplary, special, consequential or punitive damages shall be permitted. The prevailing party, as determined by the arbitrator, shall pay the AAA arbitration fees and the arbitrator's fees and expenses, as applicable. The arbitrator's award shall be binding and may be entered as a judgment and enforceable in any court of competent jurisdiction. To the fullest extent permitted by applicable law, the arbitration shall be conducted on an individual, not a class-wide basis, and no arbitration under the Terms shall be consolidated with or joined to an arbitration involving any other person or entity, whether through class arbitration proceedings or

otherwise, without the prior written consent of you and Nokia.

18. The Terms shall neither exclude nor limit any of your mandatory rights in your country of residence. If a provision of the Terms is found to be invalid, the validity of the remaining provisions shall not be affected and the invalid provision shall be replaced with a valid provision that comes closest to the result and purpose of the Terms. If there is any conflict between these Service Terms and Privacy Policy, the provisions of these Service Terms shall prevail. The provisions of the Terms that are intended to survive termination shall remain valid after any termination.

4.4.5. End User Software Agreement (Nokia)

IMPORTANT: READ CAREFULLY BEFORE INSTALLING, DOWNLOADING, OR USING THE SOFTWARE

NOKIA CORPORATION END-USER SOFTWARE AGREEMENT

This Software Agreement ("Agreement") is between You (either an individual or an entity), the End User, and Nokia Corporation ("Nokia"). The Agreement authorizes You to use the Software specified in Clause 1 below, which may be stored on a CD-ROM, sent to You by electronic mail, or downloaded from Nokia's Web pages or Servers or from other sources under the terms and conditions set forth below. This is an agreement on end-user rights and not an agreement for sale. Nokia continues to own the copy of the Software and the physical media contained in the sales package and any other copy that You are authorized to make pursuant to this Agreement.

Read this Agreement carefully before installing, downloading, or using the Software. By clicking on the "I Accept" button while installing, downloading, and/or using the Software, You agree to the terms and conditions of this Agreement. If you do not agree to all of the terms and conditions of this Agreement, promptly click the "Decline" or "I Do Not Accept" button, cancel the installation or downloading, or destroy or return the Software and accompanying documentation to Nokia. YOU AGREE THAT YOUR USE OF THE SOFTWARE ACKNOWLEDGES THAT YOU HAVE READ THIS AGREEMENT, UNDERSTAND IT, AND AGREE TO BE BOUND BY ITS TERMS AND CONDITIONS.

1. SOFTWARE. As used in this Agreement, the term "Software" means, collectively: (i) the software product identified above (ii) all the contents of the disk(s), CD-ROM(s), electronic mail and its file attachments, or other media with which this Agreement is provided, including the object code form of the software delivered via a CD-ROM, electronic mail, or Web page (iii) digital images, stock photographs, clip art, or other artistic works ("Stock Files") (iv) related explanatory written materials and any other possible documentation related thereto ("Documentation"); (v) fonts, and (vi) upgrades, modified versions, updates, additions, and copies of the Software (collectively "Updates"), if any, licensed to You by Nokia under this Agreement.
2. END-USER RIGHTS AND USE. Nokia grants to You non-exclusive, non-transferable end-user rights to install the Software on the local hard disk(s) or other permanent storage media of one computer and use the Software on a single computer or terminal at a time.
3. LIMITATIONS ON END-USER RIGHTS. You may not copy, distribute, or make derivative works of the Software except as follows:
   (a) You may make one copy of the Software on magnetic media as an archival backup copy, provided Your archival backup copy is not installed or used on any computer. Any other copies You make of the Software are in violation of this Agreement.
   (b) You may not use, modify, translate, reproduce, or transfer the right to use the Software or copy the Software except as expressly provided in this Agreement.

(c) You may not resell, sublicense, rent, lease, or lend the Software.

(d) You may not reverse engineer, reverse compile, disassemble, or otherwise attempt to discover the source code of the Software (except to the extent that this restriction is expressly prohibited by law) or create derivative works based on the Software.

(e) Unless stated otherwise in the Documentation, You shall not display, modify, reproduce, or distribute any of the Stock Files included with the Software. In the event that the Documentation allows You to display the Stock Files, You shall not distribute the Stock Files on a stand-alone basis, i.e., in circumstances in which the Stock Files constitute the primary value of the product being distributed. You should review the "Readme" files associated with the Stock Files that You use to ascertain what rights You have with respect to such materials. Stock Files may not be used in the production of libelous, defamatory, fraudulent, infringing, lewd, obscene, or pornographic material or in any otherwise illegal manner. You may not register or claim any rights in the Stock Files or derivative works thereof.

(f) You agree that You shall only use the Software in a manner that complies with all applicable laws in the jurisdiction in which You use the Software, including, but not limited to, applicable restrictions concerning copyright and other intellectual property rights.

4. COPYRIGHT. The Software and all rights, without limitation including proprietary rights therein, are owned by Nokia and/or its licensors and affiliates and are protected by international treaty provisions and all other applicable national laws of the country in which it is being used. The structure, organization, and code of the Software are the valuable trade secrets and confidential information of Nokia and/or its licensors and affiliates. You must not copy the Software, except as set forth in clause 3 (Limitations On End-User Rights). Any copies which You are permitted to make pursuant to this Agreement must contain the same copyright and other proprietary notices that appear on the Software.

5. MULTIPLE ENVIRONMENT SOFTWARE /MULTIPLE LANGUAGE SOFTWARE /DUAL MEDIA SOFTWARE /MULTIPLE COPIES /UPDATES. If the Software supports multiple platforms or languages, if You receive the Software on multiple media, or if You otherwise receive multiple copies of the Software, the number of computers on which all versions of the Software are installed shall be one computer. You may not rent, lease, sublicense, lend, or transfer versions or copies of the Software You do not use. If the Software is an Update to a previous version of the Software, You must possess valid end-user rights to such a previous version in order to use the Update, and You may use the previous version for ninety (90) days after You receive the Update in order to assist You in the transition to the Update. After such time You no longer have a right to use the previous version, except for the sole purpose of enabling You to install the Update.

6. COMMENCEMENT & TERMINATION. This Agreement is effective from the first date You install the Software. You may terminate this Agreement at any time by permanently deleting, destroying, and returning, at Your own costs, the Software, all backup copies, and all related materials provided by Nokia. Your end-user rights automatically and immediately terminate without notice from Nokia if You fail to comply

with any provision of this Agreement. In such an event, You must immediately delete, destroy, or return at Your own cost, the Software, all backup copies, and all related material to Nokia.

7. YOU ACKNOWLEDGE THAT THE SOFTWARE IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW NEITHER NOKIA, ITS LICENSORS OR AFFILIATES, NOR THE COPYRIGHT HOLDERS MAKE ANY REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE OR THAT THE SOFTWARE WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS, OR OTHER RIGHTS. THERE IS NO WARRANTY BY NOKIA OR BY ANY OTHER PARTY THAT THE FUNCTIONS CONTAINED IN THE SOFTWARE WILL MEET YOUR REQUIREMENTS OR THAT THE OPERATION OF THE SOFTWARE WILL BE UNINTERRUPTED OR ERROR-FREE. YOU ASSUME ALL RESPONSIBILITY AND RISK FOR THE SELECTION OF THE SOFTWARE TO ACHIEVE YOUR INTENDED RESULTS AND FOR THE INSTALLATION, USE, AND RESULTS OBTAINED FROM IT.

8. NO OTHER OBLIGATIONS. This Agreement creates no obligations on the part of Nokia other than as specifically set forth herein.

9. LIMITATION OF LIABILITY. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL NOKIA, ITS EMPLOYEES OR LICENSORS OR AFFILIATES BE LIABLE FOR ANY LOST PROFITS, REVENUE, SALES, DATA, OR COSTS OF PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES, PROPERTY DAMAGE, PERSONAL INJURY, INTERRUPTION OF BUSINESS, LOSS OF BUSINESS INFORMATION, OR FOR ANY SPECIAL, DIRECT, INDIRECT, INCIDENTAL, ECONOMIC, COVER, PUNITIVE, SPECIAL, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND WHETHER ARISING UNDER CONTRACT, TORT, NEGLIGENCE, OR OTHER THEORY OF LIABILITY ARISING OUT OF THE USE OF OR INABILITY TO USE THE SOFTWARE, EVEN IF NOKIA OR ITS LICENSORS OR AFFILIATES ARE ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. BECAUSE SOME COUNTRIES/STATES/JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF LIABILITY, BUT MAY ALLOW LIABILITY TO BE LIMITED, IN SUCH CASES, NOKIA, ITS EMPLOYEES OR LICENSORS OR AFFILIATES' LIABILITY SHALL BE LIMITED TO U.S. $50.

Nothing contained in this Agreement shall prejudice the statutory rights of any party dealing as a consumer. Nothing contained in this Agreement limits Nokia's liability to You in the event of death or personal injury resulting from Nokia's negligence. Nokia is acting on behalf of its employees and licensors or affiliates for the purpose of disclaiming, excluding, and/or restricting obligations, warranties, and liability as provided in this clause 9, but in no other respects and for no other purpose.

10. TECHNICAL SUPPORT. Nokia has no obligation to furnish You with technical support unless separately agreed in writing between You and Nokia.

11. EXPORT CONTROL. The Software, including technical data, includes cryptographic software subject to export controls under the U.S. Export Administration Regulations ("EAR") and may be subject to import or export controls in other countries. The EAR prohibits the use of the Software and technical data by a Government End User, as defined hereafter, without a license from the U.S. government. A Government End User is defined in Part 772 of the EAR as "any foreign central, regional, or local government department, agency, or other entity performing governmental functions; including governmental research institutions, governmental corporations, or their separate business units (as defined in part 772 of the EAR) which are engaged in the manufacture or distribution of items or services controlled on the Wassenaar Munitions List, and international governmental organizations. This term does not include: utilities (telecommunications companies and Internet service providers; banks and financial institutions; transportation; broadcast or entertainment; educational organizations; civil health and medical organizations; retail or wholesale firms; and manufacturing or industrial entities not engaged in the manufacture or distribution of items or services controlled on the Wassenaar Munitions List.)" You agree to strictly comply with all applicable import and export regulations and acknowledge that You have the responsibility to obtain licenses to export, re-export, transfer, or import the Software. You further represent that You are not a Government End User as defined above, and You will not transfer the Software to any Government End User without a license.

12. NOTICES. All notices and return of the Software and Documentation should be delivered to:

NOKIA CORPORATION
P.O. Box 100
FIN-00045 NOKIA GROUP
FINLAND

13. APPLICABLE LAW & GENERAL PROVISIONS.

This Agreement is governed by the laws of Finland. All disputes arising from or relating to this Agreement shall be settled by a single arbitrator appointed by the Central Chamber of Commerce of Finland. The arbitration procedure shall take place in Helsinki, Finland in the English language. If any part of this Agreement is found void and unenforceable, it will not affect the validity of the balance of the Agreement, which shall remain valid and enforceable according to its terms. This Agreement may only be modified in writing by an authorized officer of Nokia.

This is the entire agreement between Nokia and You relating to the Software, and it supersedes any prior representations, discussions, undertakings, end-user agreements, communications, or advertising relating to the Software.

PLEASE SUBMIT ANY ACCOMPANYING REGISTRATION FORMS TO RECEIVE REGISTRATION BENEFITS.

4.5. Web Interfaces

Web access was important to the Mobile Millennium project. Our participants were expected to be web savvy technology adopters. Participants registered for pilot software from the website, reviewed information and requested support. There was also web application that visualized traffic data for use at special events.

4.5.1. The Web Site

The main website is located at `www.traffic.berkeley.edu` . The website went through frequent maintenance and two major revisions. The website is a basic HTML website with some PHP. There have been 20-30 pages on the website varying with project activities. Below are several versions of the front web page. The first is before the launch, the second after the launch, and the third is the final version of the website as of June 2nd 2011.

Figure 5.1. Front page of the Mobile Millennium website.

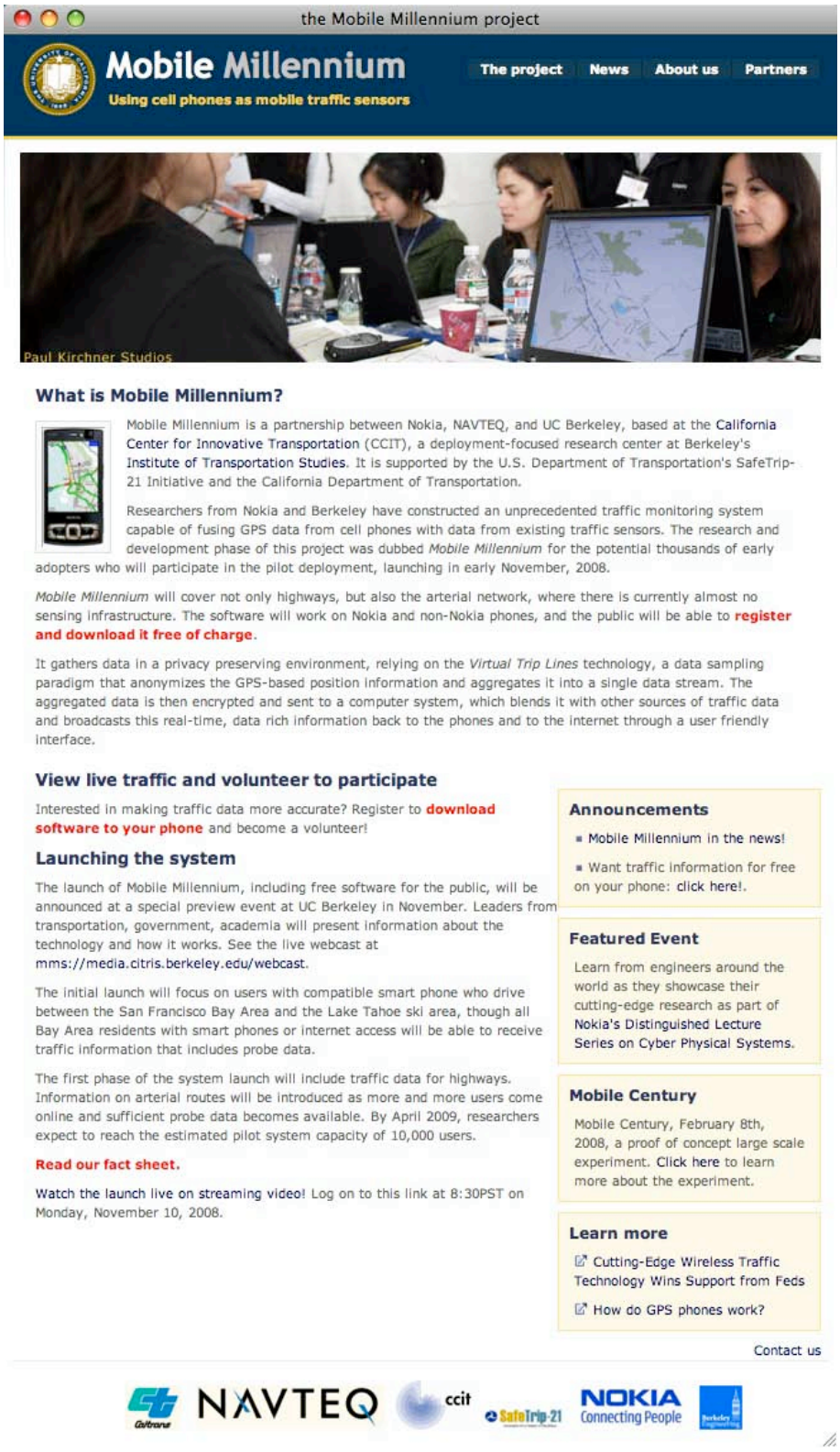


Figure 5.2. Mobile Millennium website.

Figure 5.3. Download page of the Mobile Millennium website.

4.5.2. Support Forum

UC Berkeley setup a support forum where users would post questions and answers, where known, would be posted. The following are some examples of the entries in the support forum.

ERROR MESSAGES (11)

Error 907 (2)

Error message 907 invalid COD (2)

Error 910 (7)

Installation failed, errno=910 Application authorization failure (7)

I am very interested in the project and I think that you guys have a great idea.

I tried to install it today on my phone Samsung Blackjack II AT&T phone.

I selected the "Other Java enabled devices" option for the software.

The "TrafficPilot.jar" downloads, I get a prompt that "This MIDlet does not come from a trusted source.", I approve the install, the installation runs through its steps then ends with the failure message "**Installation failed, errno=910 Application authorization failure"**

i downloaded the software for öther blackberry devices" and i have a blackberry 8820 with t mobile...does not appear to work....gets**error message indicating connection failure**...i am able to use the gps on my phone without errors...any ideas/suggestions?

I had been using this on T-Mobile Blackberry 8120 for months, and currently am getting **networkconnection error message**.

When installing Traffic Pilot I get the following **error message:**

"The suite Traffic Pilot Mobile Client cannot be installed because the permission javax.microedition.location. Location is not available on this device. -58 Please contact your operator for more information" This problem occurs both when downloading you app directly to the ATT Tilt (through Opera 9.5) and when transferring downloaded file from PC to phone.

I finally tried to down load the application again, and it worked.

The application comes up with a map centered around Berkeley. Once I try moving the map toward the peninsula, where I do my commuting, I get an **error:Uncaught exception:**

Application trafficpilot_v1_0_3_bb8310b(213) is not responding process terminated. It happened twice in the first couple of minutes. I will feedback as I get further data.

DEVICE ISSUES (3)

I did a Google search after my note and it appears the vendor (ATT) installed Java VM is either locked or somehow limited. Others have worked around this problem by installing an IBM version. I could not easily find the IBM version so I gave up.

OPERATING ISSUES (5)

Installed on my Sprint 8130 World Blackberry... works perfectly so you may want to add that as a drop-down for SMS notifications but **do I need to keep the application active in order for it to report back to you?** For example **if I start the application, switch to doing e-mails and then make a phone call, will the application continue reporting my location and velocity?**

I have dowloaded the software twice, and I am currently using this for travel. The traffic data seems to very accurate. The problem I am having, is **the system is not being able to locate me. I noticed when it does locate me, it show a blue dot. That dot does not appear consistently after I begin my trip. Actually it only shows up after I have driven for approximately 20 minutes. I am testing this out on the NOKIA N95 phone.**

I have been following your project with great interest.**On your website I found and downloaded something that resolved out to launch.jnlp version 0.5.5. Running this software on my desktop or laptop has been really great until the past week or two when it stopped working. Should I be downloading another version or has this feature been disabled?**

NEW PLATFORM / VERSION ISSUES (14)

you guys planning on **releasing a Windows Mobile version? (10)**

WEBSITE REGISTRATION ISSUES (1)

I think you guys overlooked something on your web site. When I tried to sign up for the software, **it would not let me get past that part**

about the Northern CA zip code. After trying several times to enter my zip code (45106) in Ohio, I kept getting an error referring me back to that one section. I am assuming, then, that the only area involved in this project is Northern CA. If that is true, you should probably state that on the first page, especially if you get national attention for your project. I would dearly love to have such a project here in the Cincy area. I have Telenav, and one of its features is to warn of traffic problems along my route. The main problem I have found is that, other than construction delays, it warns me of delays when I am pretty much already aware that I am in the midst of a delay (and cursing Telenav). I am ususally informed too late to take alternate routes.

4.5.3. Newsletter

A newsletter was sent out to all registered users who had indicated that it was acceptable to contact them. The newsletter follows:

Read All About It

Sure, you're living it, but have you read about what others are saying about our landmark field test? Follow the links below for some high-profile press coverage. You'll find a feature on Mobile Millennium's principal investigator Alexandre Bayen on the new CBS website Smart Planet (`http://www.smartplanet.com/people/video/alex-bayen-professor-systems-engineering-uc-berkeley/306654/`).
And go to `http://traffic.berkeley.edu/media` page for more links to print and multimedia coverage from around the world.

Survey Note

Thanks to all of you who replied to our recent survey. You're comments will help us understand the best way to adjust the Traffic Pilot software in this critical development stage. We'll announce a Nokia phone winner via email by July 1.

New Release

A new version of Traffic Pilot will be out in a few weeks—this version will include improvement based on the the last round of analysis and user feedback. Look for an SMS message with an update to the current software.

Blackberry Storm: New Options is all this correct?

We've heard that Blackberry Storm users have had problems receiving the SMS message containing the link to the application. We have corrected this problem, so update your software—just return to `http://traffic.berkeley.edu` and re-register.

Here's another option, to make sure you receive the software:

1. Re-register, but this time select "email" as the way you'd like to receive your application.
2. Accessing your email via your smart phone, double-click on the applications you receive via email attachment.

One of these two methods should install the application and have on your way to benefiting from the most up-to-date traffic information available.

Special AT&T Fix

AT&T service plan users:, you have a new benefit: You know all of those extra permissions screens you have to say "OK" to before the Traffic Pilot map comes to life? Gone. Now when you launch the application, you'll go straight to traffic information. To get the update, just go to the http://traffic.berkeley.edu and sign up for the software again. The application you receive will be the updated version.

We HEART Commuters

Think your company would be interested in participating in Mobile Millennium *en masse*? We're looking to work with a community of commuters assist in the validation of the system.

If you have a group of 25 to 100 co-workers who are interested and have qualifying smart phones and data plans, we'd love to hear from you. Contact Steve Andrews at `sansdrews@calccit.org`. If your group qualifies, a CCIT team will come to your location to explain the technology, answer questions, and help get you signed up.

We welcome comments about this newsletter, the Traffic Pilot software, or the Mobile Millennium project. Write us at `trafficpilot@calccit.org`.

4.5.4. Visualization

Visualization of traffic information was vital to the project. There is a separate chapter in this document that discusses the main tool we used to visualize traffic data. This tool is referred to as the "visualizer" In this section we will only focus on the outreach effects of the visualization and not on the implementation or usage details.

The visualization tool was used in several venues:

1. At conferences and seminars where it was used to demonstrate real time traffic for highways and arterials
2. In the DRI office where it was used to demonstrate the Mobile Millennium project, often described as one of the most successful projects DRI has funded.
3. In the CITRIS display area. CITRIS is a new venue on campus where governments (both US and international), industry leaders and academic luminaries frequently gather to discuss new joint ventures with the University of California. There were also a number of press interviews performed at CITRIS in front of the visualizer.

The image below is of the visualizer. The main image is of traffic in Berkeley California. Traffic travelling at the speed limit is shown as green, slower traffic as yellow and very slow or stopped traffic as red. One of the exciting items is that traffic is shown both for highways and arterials. The map maybe zoomed in or out and scrolled to show traffic throughout the bay area.

The panel on the left is an interactive video with sound. A user drags a cell phone into a car and starts a presentation on Mobile Millennium.

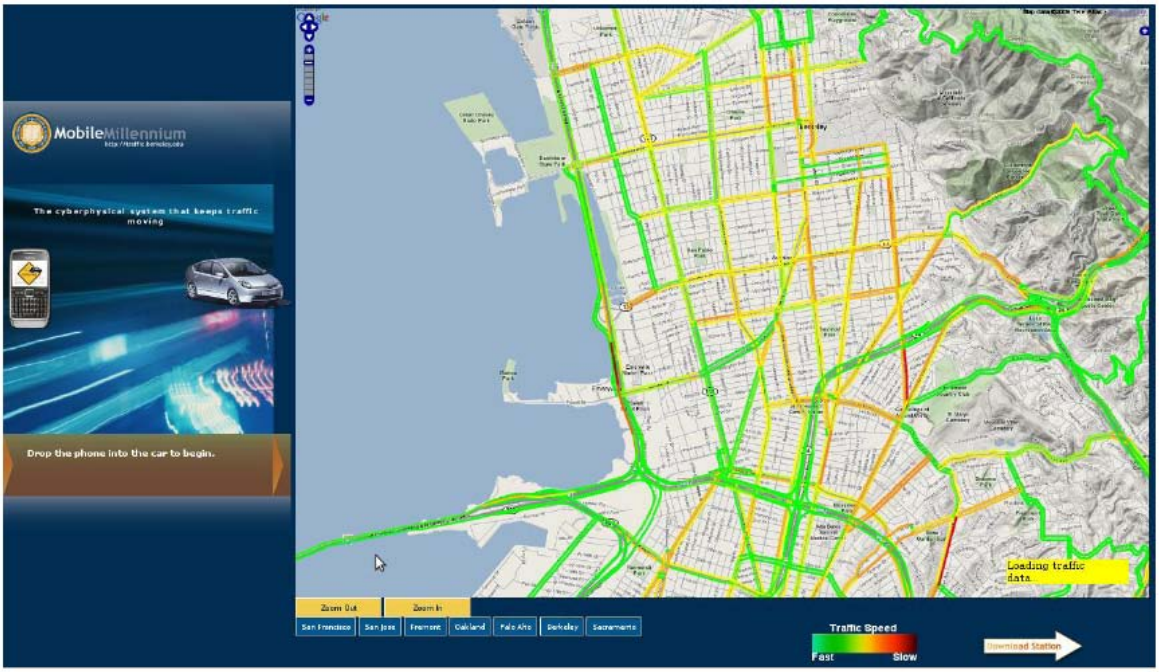


Figure 5.4. Mobile Millennium interface, available on the web and with touchscreen in CITRIS.

The visualizer is also capable of showing more distinct information on traffic as shown below. The first picture displays the individual information available for individual sections of road. The second is an example of the routing capability of the Mobile Millennium software. Given two locations the software would provide you the best route between the destinations. More discussion on this will be found in the section on routing.

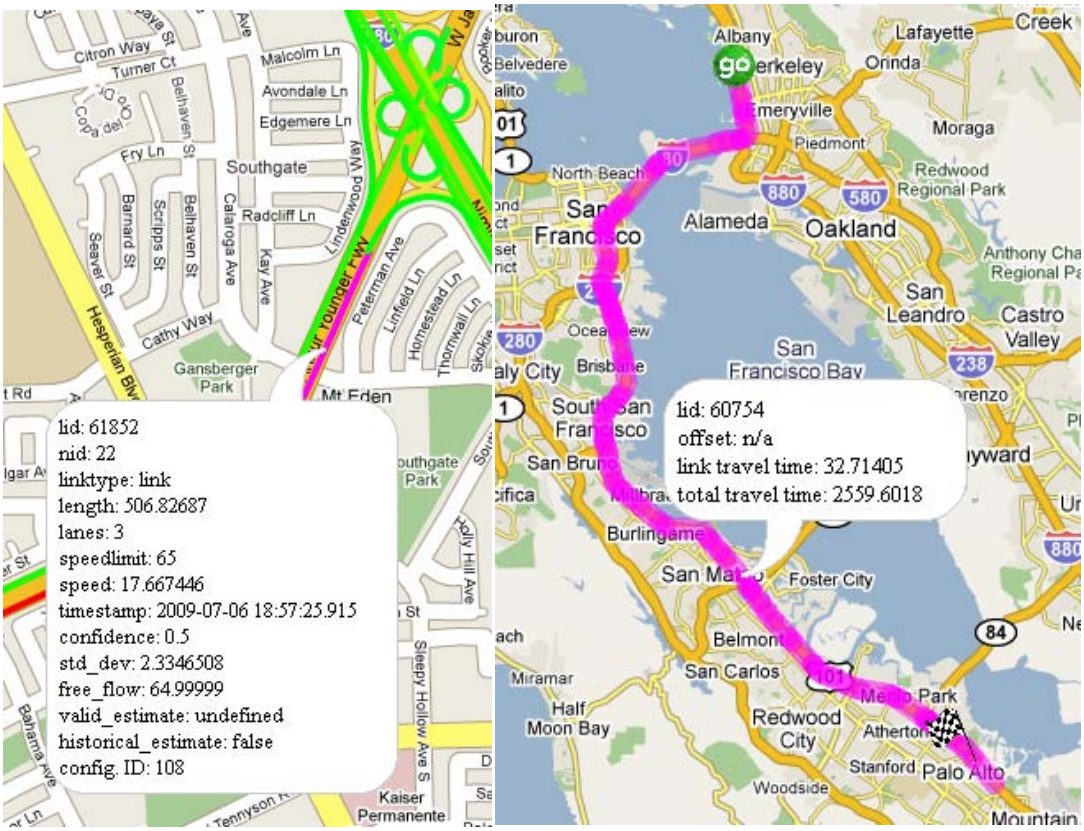


Figure 5.5. Example of map attributes and live trafic data accessible through the Mobile Millennium visualization tool.

Overall the visualizer has likely been viewed by millions of people through the TV, Press, Internet and in-person presentations on Mobile Millennium. The Mobile Millennium live traffic map was one of the first traffic maps to include both highway and arterial traffic estimations. Today traffic maps are all the norm but this was not the case when the vizualizer was launched.

4.6. Survey Process

4.6.1. Summary

The usage statistics provide a quantitative insight to the adoption of Mobile Millennium. However this is an incomplete picture because it reveals nothing about the characteristics of users or the actual pattern of use of the system. The user surveys conducted by NRC go some way towards providing insights as to who the users are, how they used Mobile Millennium and their perceptions about using the system and perhaps how it can be improved. The normal registration process for Traffic Pilot results in anonymous users, just requiring a zip code. Some users gave authorization to contact them (opted in for surveys and gave email address). These participants were sent surveys.

The user surveys were intended to support NRC's product development process related to traffic information services. NRC received no federal or state funding related to its participation in Mobile Millennium. Indeed NRC invested its own resources and expertise to the test, and these surveys would normally be considered proprietary. However NRC has agreed to release these results as part of the final report. The results should be viewed with a clear understanding that the results are from questions asked regarding a 1st generation phone client and not a commercial offering. Phone applications and associated operating systems and hardware have been significantly updated since the Mobile Millennium pilot. Today the survey results are valuable both from an historical perspective as well as from a detailing of issues that are still very likely of interest to commuters and industry application vendors.

Understanding how the phone client and the provided traffic data were being used by travelers was an important part of the Mobile Millennium project. Would users be willing to provide location data, would they feel their privacy was being respected, did they find the traffic estimations useful? These and other questions needed clarification in order to understand the ability to grow the Mobile Millennium technology into a full featured product for use by the general public. Nokia designed the surveys and UC Berkeley distributed them. From Nokia's perspective the objects of the survey were Test of market viability. Can Nokia make a commercial product out of this technology?

The surveys were conducted in the first half of January 2009 and through most of June 2009, two and seven months respectively after the launch of Mobile Millennium. As mentioned previously, participation was optional, and approximately three quarters of users agreed to participate in future surveys at the time of registration. An email was sent to the email address of users who 'opted-in' to participate in surveys prior to each survey. The email invited users to participate in one or both user surveys (depending on when they registered) and contained a link to the website hosting each survey, which was then taken online. The user surveys were developed and analyzed by NRC.

After a review and summary of the results the actual survey questions and answers will be provided as charts and graphs.

1. First Survey Summary

- Ran from 01/06/09 – 01/16/09
- Survey request sent to ~1300 registered users
  - Email
  - Users opted in to be contacted for surveys during registration
- Received ~370 responses
- Incentive: raffle for a N95

January 2009 survey questions were used for basic information gathering:

- Who is provider?
- What type of mobile device are you using?
- How do you prefer to download? Was website a good way? (Alternatives – sms, etc.). If not, what problems did you have? Delivery mechanism. Website download was a popular way.
- Frequency of use, how often, would use again in the future (if not, why not?)
- What other features?

2. Second Survey Summary

- Survey open from June 2 – June 26
- 1550+ emails sent out
- 286 unique respondents

The June survey was a follow-up to the first survey asking questions about the market, how to approach the user, ultimately asking questions on how to make a business out of providing traffic information:

- Intended to be a follow up to January survey.
- Fill in some of the gaps that we've seen
- Test some hypotheses regarding of Traffic Pilot. For example, we have seen in usage stats that some users are opening and not getting GPS locks – are users just turning it on to look at current conditions and then just turning it off? If so, this is not good – not contributing data.
- When and why are you using it?

3. Summary Analysis

Almost 80% of all Mobile Millennium users had registered by the time of the January 2009 user survey. More than 90 percent of all Mobile Millennium users had registered by the time of the June 2009 user survey

The surveys mostly asked different questions, so there is limited scope for any trend analysis.

	January 2009	June 2009
Survey Date	January 6 to January 16	June 2 to June 26
Registered Users	1774 (by End December 2008)	2090 (by end May 2009)
Responses Invitation	370 (1300)	286 (1550)
Responses Rates	28.5%	15.8%
Male/Female	N/A	86% / 14%
Age	N/A	18-25 (5%) 26-35 (24%) 36-45 (34%) 46-55 (22%) 56+ (15%)

One similar question area in both surveys related to frequency of use. In both surveys it was apparent that the majority of users were infrequent users. The minority of users that used Mobile Millennium daily declined from 33% in January 2009 to 11% (6 or more uses in the past 7 days) in June 2009.

\

	January 2009	June 2009
Use on a Daily Basis (Yes or No)	33% / 67%	
How many uses in the past 7 days?		0 (40%) 1-2 (30%) 3-5 (19%) 5-7 (7%) 11+ (15%)

In the January 2009 survey, 9% of respondents indicated that they had not actually used Mobile Millennium, mostly because "my phone is not supported." In the June 2009 survey, "It does not work on my phone" remained the top reason for never using the system.

For the January 2009 survey, respondents who had decided not to use the system in the future suggested that the following improvements would convince them to try the system again:

- Better user experience (16 comments)
- More accurate travel data (10 comments)
- Higher quality traffic incident data (10 comments)
- Traffic data on more roads (8 comments)
- Personalized traffic information for the routes I drive(7 comments)
- Driving directions (7 comments)

For the June 2009 survey, the focus was more on how respondents use the system. For example, nearly half of users opened the application to view traffic information prior to getting in their car (24% in a building, 23% on their way to the car), one third (34%) of users opened the application when they were "in my car as I started driving," and 19% opened the application "while I was driving, when I encountered (or became concerned about) traffic." Frequent users of Mobile Millennium were least likely to open the application while driving.

One behavior that emerged from the June 2009 survey was that while almost two thirds of users kept the application open for at least the duration of their trip, the remaining third closed the application after viewing traffic conditions. This is important because by closing the application,

the mobile phone no longer acts as a traffic probe, significantly reducing probe penetration and most likely reducing the accuracy of the traffic information provided by Mobile Millennium. This behavior occurred more often among less frequent users, so it is likely that less than one third of trips are affected. The reasons for users closing the application are unknown, but it may suggest that they had no intention of viewing (or being distracted by) the application while driving, or perhaps it indicates some concern about battery life, or being tracked while driving.

On the subject of privacy, 68 percent of respondents trusted the system to protect the privacy of their travel data, 7 percent did not trust the system, and 25 percent either did not know or did not care.

4.6.2. Survey Related Emails

1. First Survey – Sent 1/07/09

Emails were sent out with the survey. These emails follow.

2. First Survey – Sent 1/07/09

Subject:
Berkeley/Nokia Research Update: Win a Nokia N95!

Body:
Dear Traffic Pilot User:

Thank you for participating in our traffic research!

We would like to learn about your experiences with the Mobile Millennium Traffic Pilot – both good and bad. Your feedback will help us improve this developing technology. We have designed the survey to take less than five minutes of your time.

In appreciation for your participation, responses received before midnight on Tuesday, January 13, 2009, may be eligible for a raffle of a new Nokia N95 phone, packed with smart-phone features and accessible through AT&T or T-Mobile.

Thank you for helping to advance our research on the Mobile Millennium traffic information project.

Start the survey now by following this link:
http://survey.nokiapaloalto.com/TrafficPilotSurvey/.

- The Mobile Millennium Team

P.S. You must be 18 or over to participate in the survey.

——
You are receiving this email because you agreed to be contacted about our surveys when you registered on the Traffic Pilot website, http://traffic.berkeley.edu/pilot. If you would like to stop receiving survey requests, go to

http://traffic.berkeley.edu/remove.html. Please allow one week to

fully remove your address from the system.

3. Second survey – Sent 6/2/09

Dear Traffic Pilot User:

Thank you for participating in our traffic research!

We are conducting a follow-up survey to learn more about how you use the Mobile Millennium Traffic Pilot. Your feedback during our January survey was very valuable, and we hope that you will continue to help us improve this developing technology. We have designed the survey to take about five minutes of your time.

In appreciation for your participation, responses received before midnight on Tuesday, June 16, 2009, may be eligible for a raffle for two new (unlocked) Nokia E71 phones, packed with smart-phone features and accessible through AT&T or T-Mobile. ne
Thank you for helping to advance our research on the Mobile Millennium traffic information project.
Start the survey now by following this link:

http://survey.nokiapaloalto.com/TrafficPilotSurvey2/

- The Mobile Millennium Team

P.S. You must be 18 or over to participate in the survey.

——
You are receiving this email because you agreed to be contacted about our surveys when you registered on the Traffic Pilot website, http://traffic.berkeley.edu/pilot. If you would like to stop receiving survey requests, go to

http://traffic.berkeley.edu/remove.html. Please allow one week to fully remove your address from the system.

4. Second Survey Reminder – Sent 6/16/09

Subject:
Berkeley/Nokia Research Deadline: Respond by Midnight for a Chance to Win a Nokia E71

Body:
Dear Traffic Pilot User:

If you have already responded to our survey, thank you for your

constructive feedback!

If you have not yet responded, today is your last chance. All responses received by 11:59 p.m. tonight, Tuesday, June 16, 2009, may be eligible for a raffle for two new Nokia E71 phones, packed with smart-phone features and accessible through AT&T or T-Mobile.

The survey should take about five minutes of your time. Start it now by following this link:

http://survey.nokiapaloalto.com/TrafficPilotSurvey2/

This is the last reminder that we will send out. Thank you for helping to advance our traffic research!

- The Mobile Millennium Team

P.S. You must be 18 or over to participate in the survey.

——
You are receiving this email because you agreed to be contacted about our surveys when you registered on the Traffic Pilot website, http://traffic.berkeley.edu/pilot. If you would like to stop receiving survey requests, go to

http://traffic.berkeley.edu/remove.html. Please allow one week to fully remove your address from the system.

4.7. Survey Results

Each survey had a number of questions. We will present, generally in a graphical format, the results of each survey question. We will then provide a short discussion on the results. This analysis is purely one interpretation of the data. Without speaking with the participants it is not possible to validate our opinions. Thus our analysis should be considered food for thought and not a definitive statement.

4.7.1. First Survey

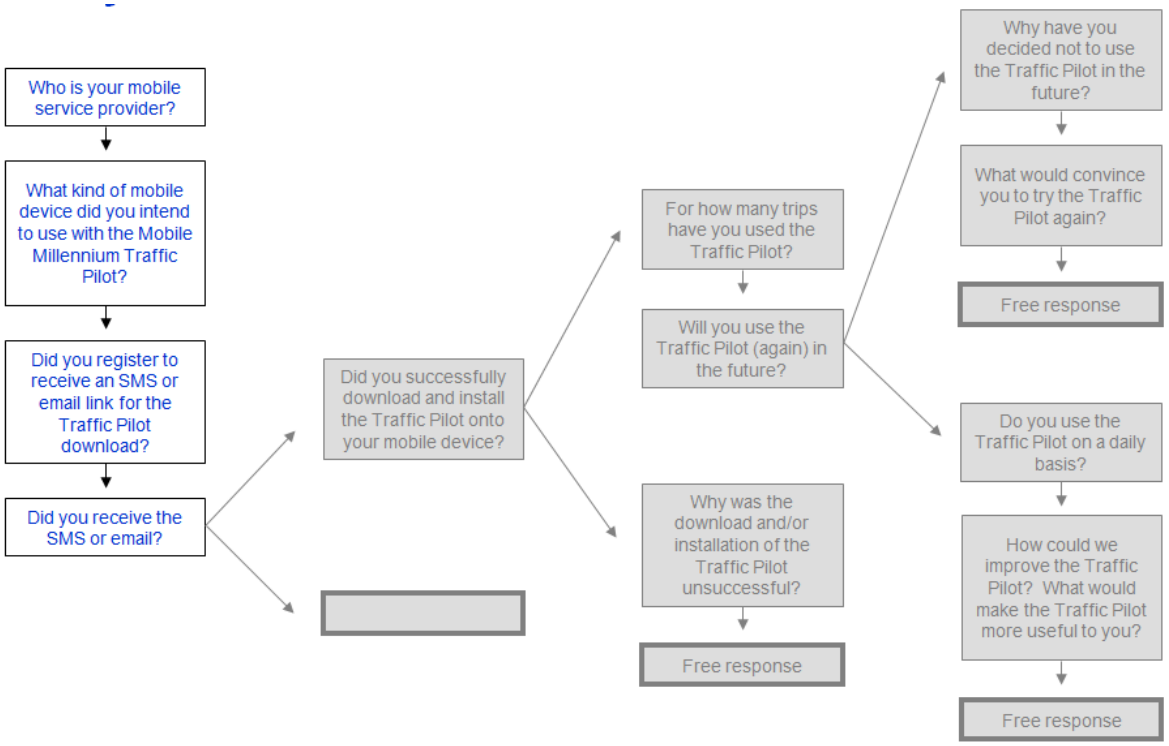


Figure 7.1. Survey Flow

This survey was a directed graph with a given response determining what the next survey questions would be.

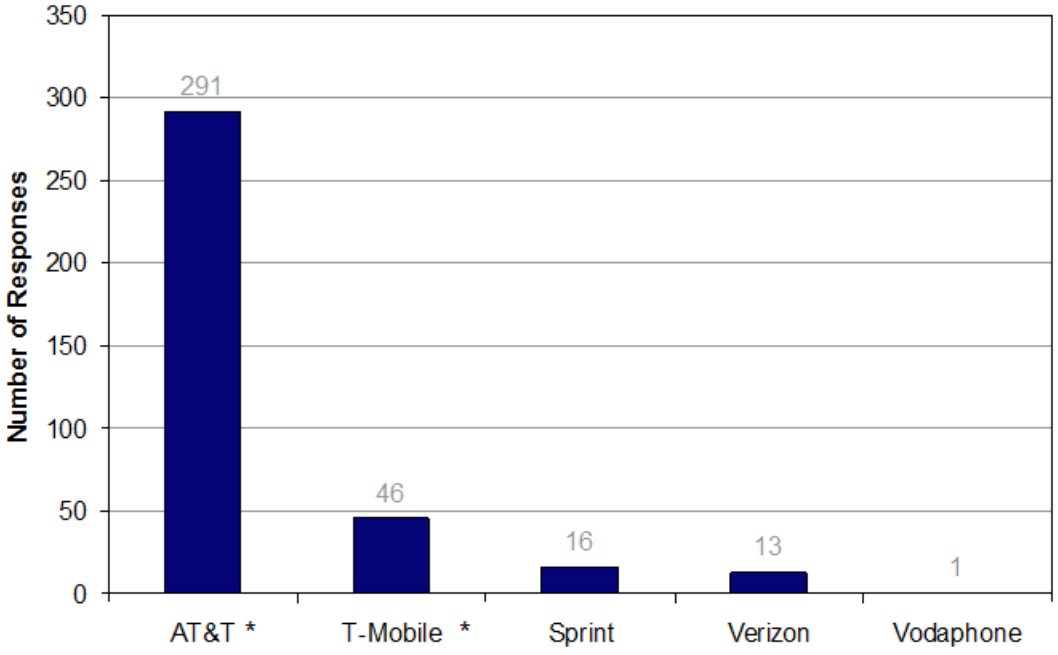


Figure 7.2. Mobile Service Provider

Most respondents used AT&T. We believe that AT&T was the majority provider in the Bay Area at this time for smart phones.

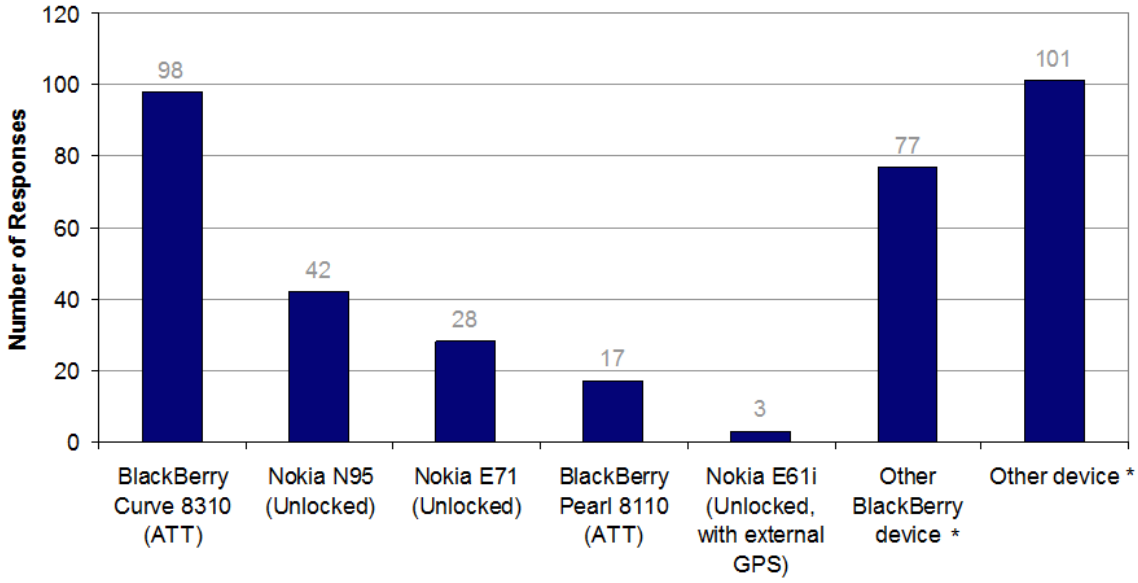


Figure 7.3. Mobile Device

At the time of this survey BlackBerrys were very popular smart phones and perhaps more importantly iPhones were not supported. iPhones probably comprised >20% share of the Bay Area smart phone market at this time.

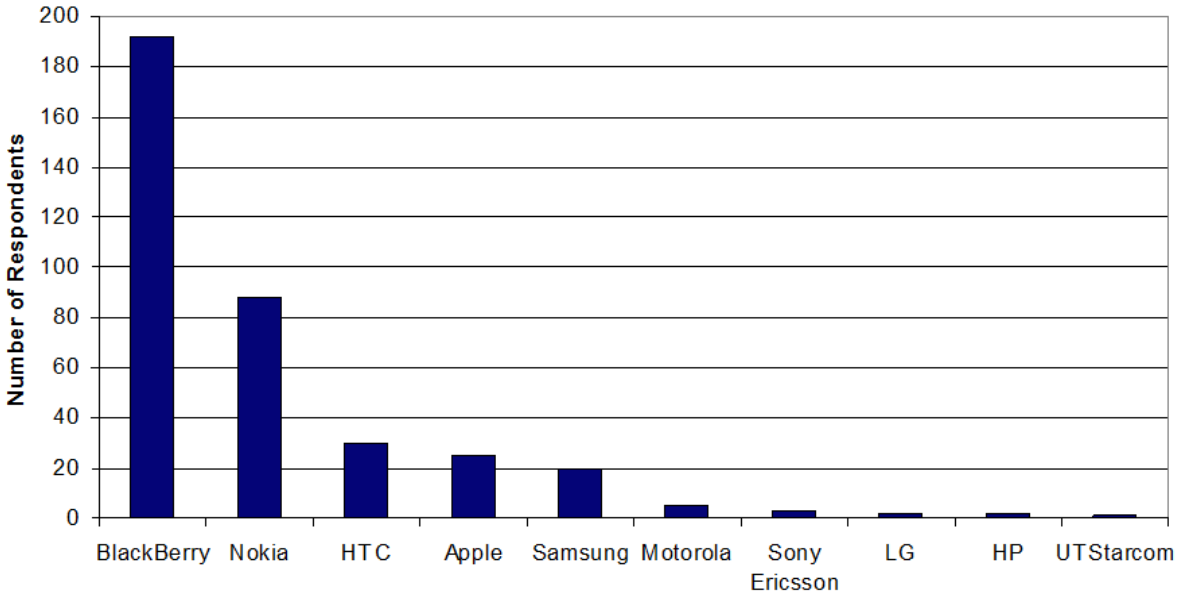


Figure 7.4. Devices By Manufacturer

Once again, Blackberry was a popular smartphone.

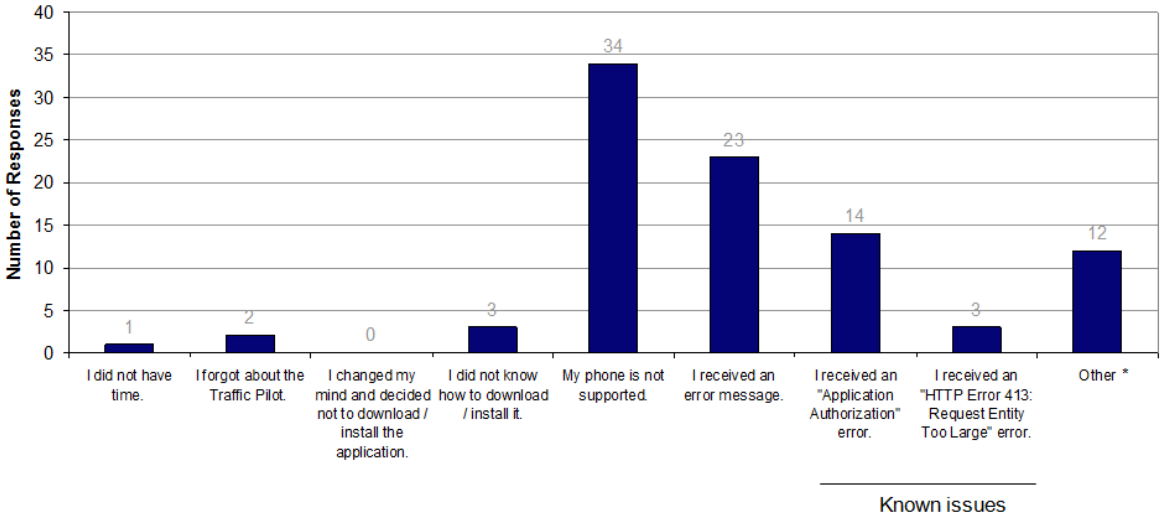


Figure 7.5. Why was Pilot unsuccessful

For folks who thought the pilot was unsuccessful (only a quarter of the responses) the issues were mostly related to software and hardware. This is not surprising given the level of smart phone capabilities at the time of the survey. Of note very few changed their minds regarding the idea of the pilot.

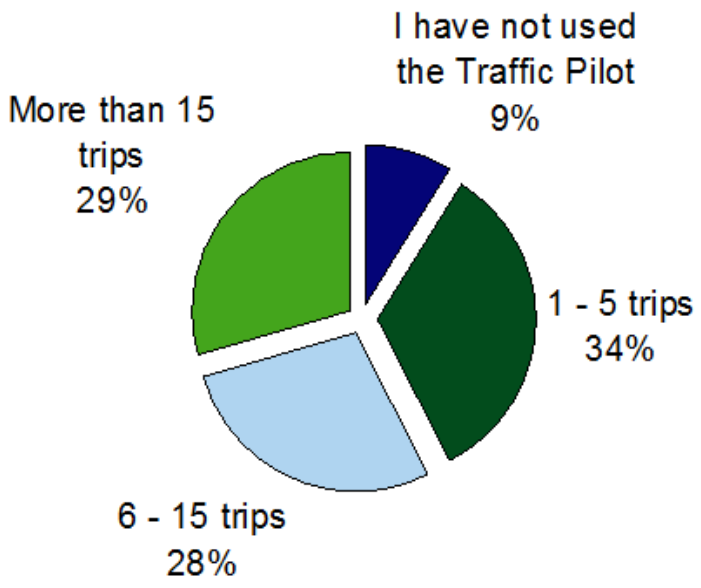


Figure 7.6. How many trips have you used the Traffic Pilot?

This survey was sent out at the beginning of January. The pilot was started in November. Considering all the holidays the result that close to a third of responders used it more than 15 times seems to indicate interest and support for the project.

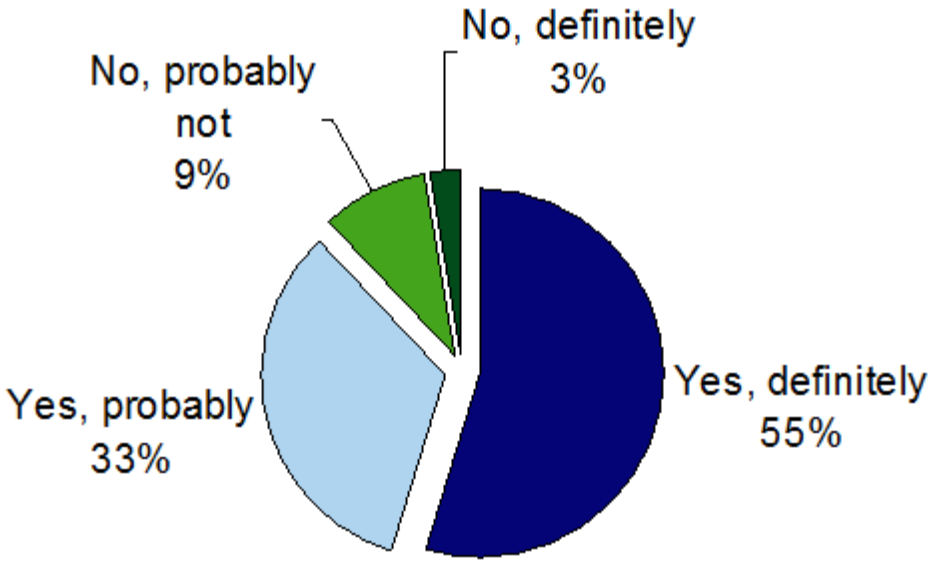


Figure 7.7. Will you use the traffic pilot again?

A high majority of users were planning on using the pilot again. Considering the early stages of the product development and the challenges inherent in early smart phones this is a great response.

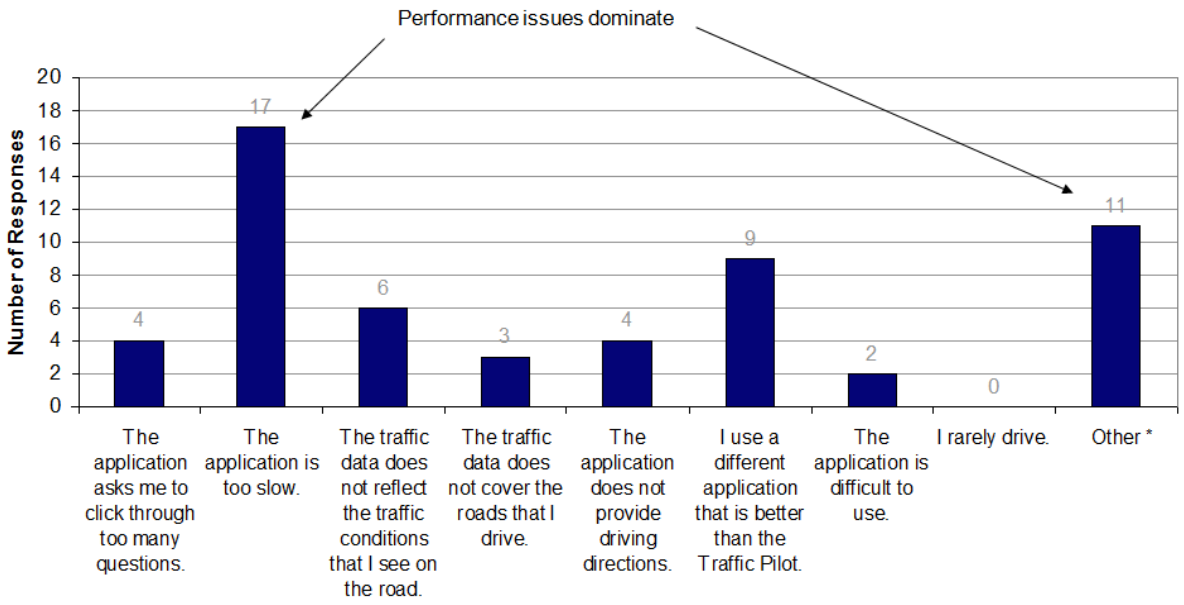


Figure 7.8. Why will you not use it?

Again in this case most folks are using it. But even the criticisms provide implicit support of the concept.

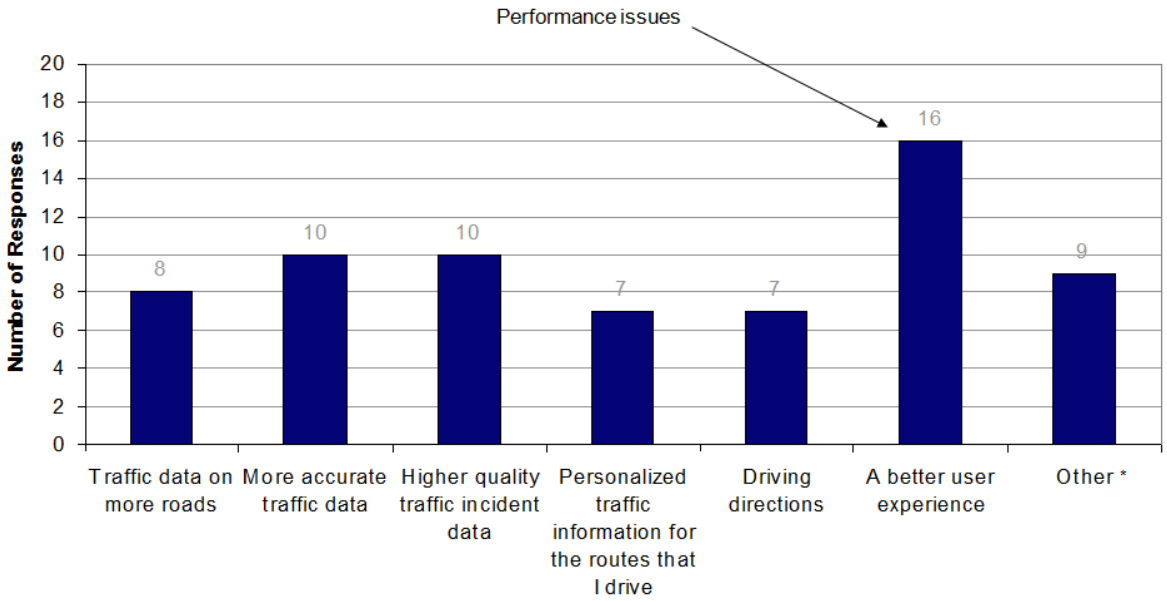


Figure 7.9. What would convince you to try the pilot again?

In this case the performance and UI issues were paramount.

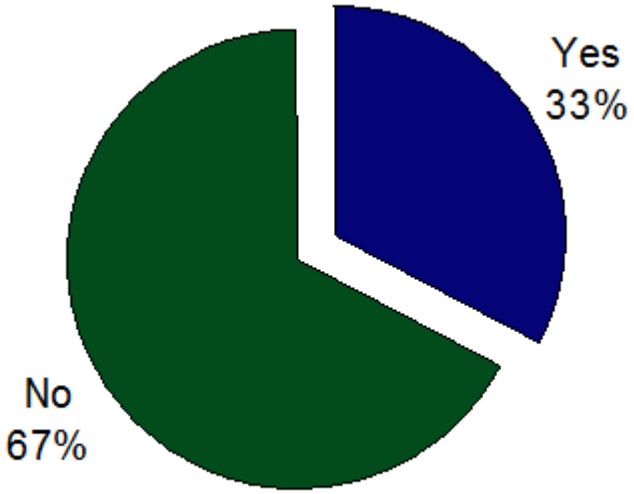


Figure 7.10. Do you use the traffic pilot on a daily basis?

This is quite a good number especially considering the UI difficulties.

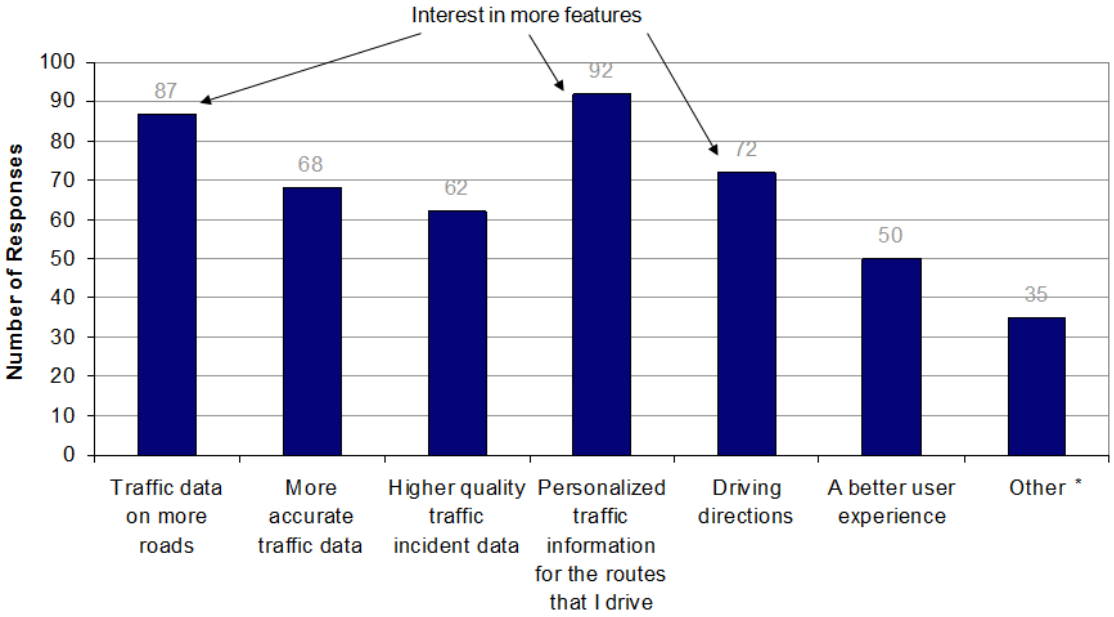


Figure 7.11. How could we improve the traffic pilot?

These are all positive items indicating an expansion of scope. No indication the pilot is a bad idea.

4.7.2. Second Survey

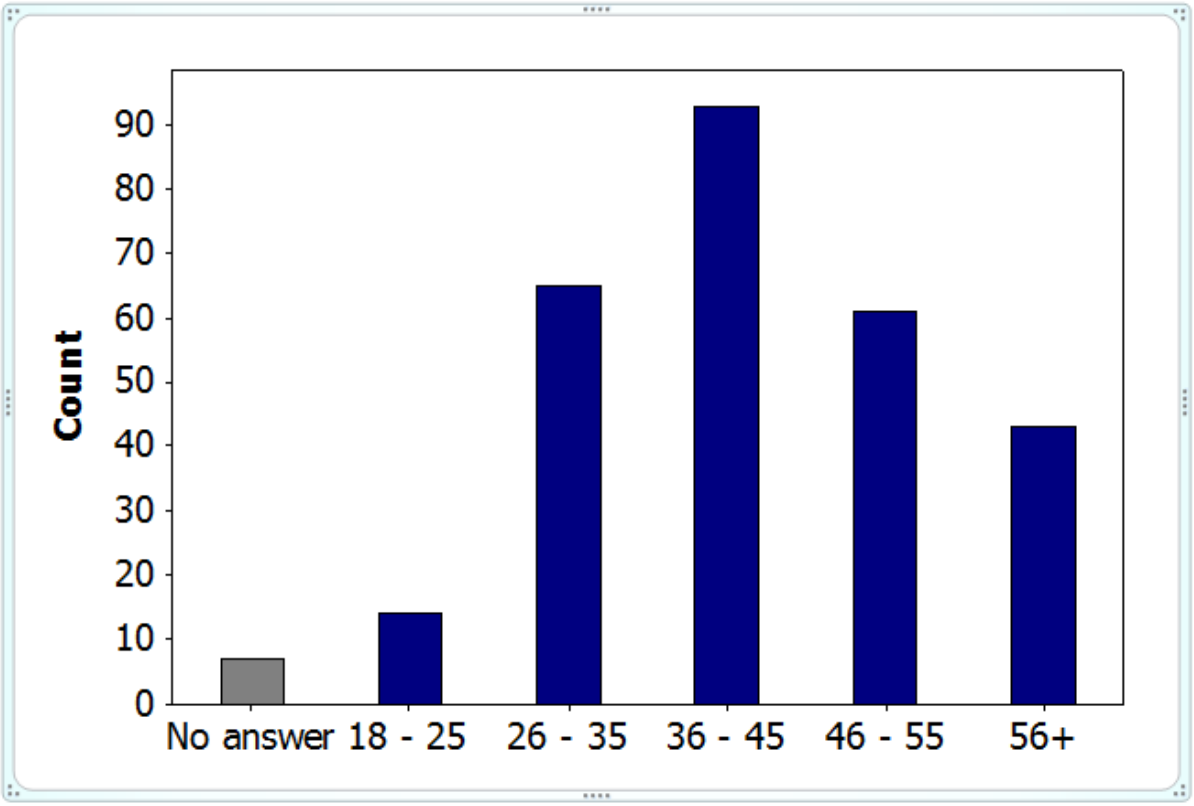


Figure 7.12. Respondents Age Range

It seems that we have picked up commuters here who could afford smart phones. Also iPhone smart users may have been younger. This may explain why there are more middle age users than younger users.

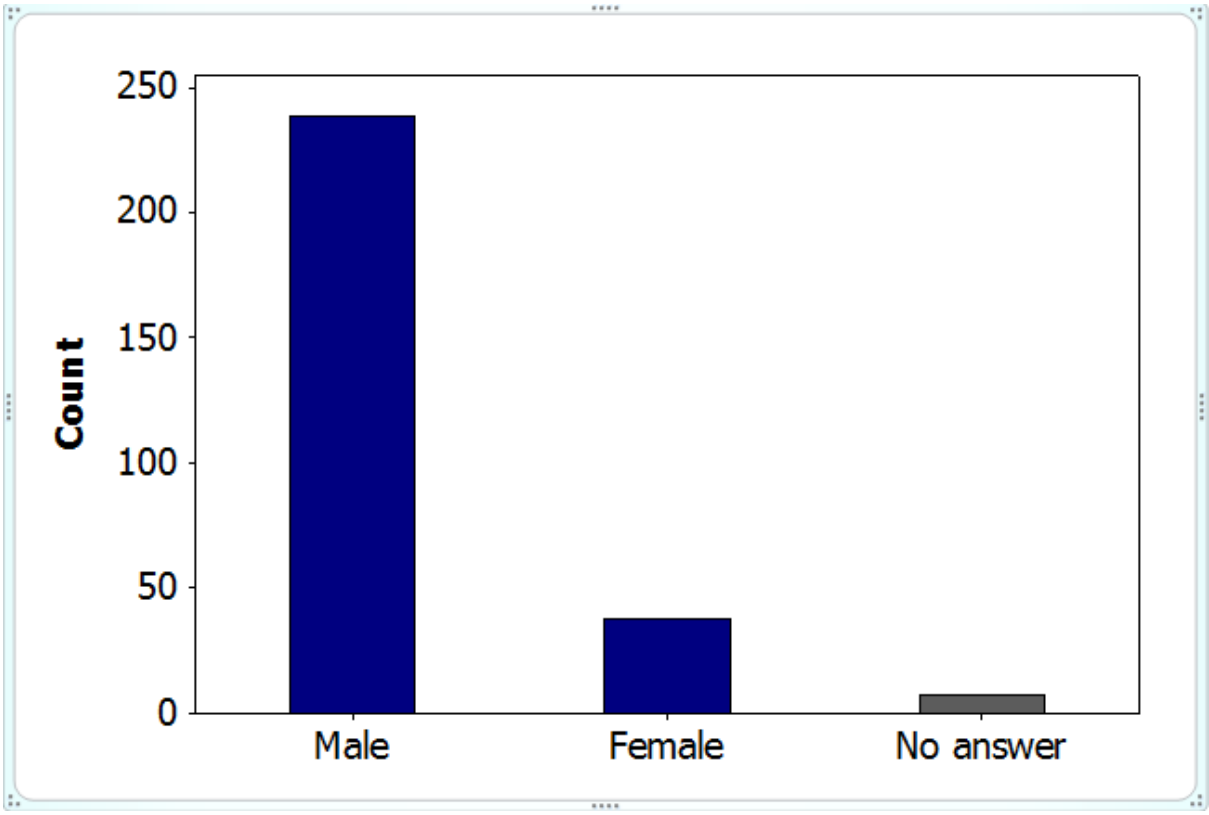


Figure 7.13. Respondents Gender

Men generally are more interested in new techie items.

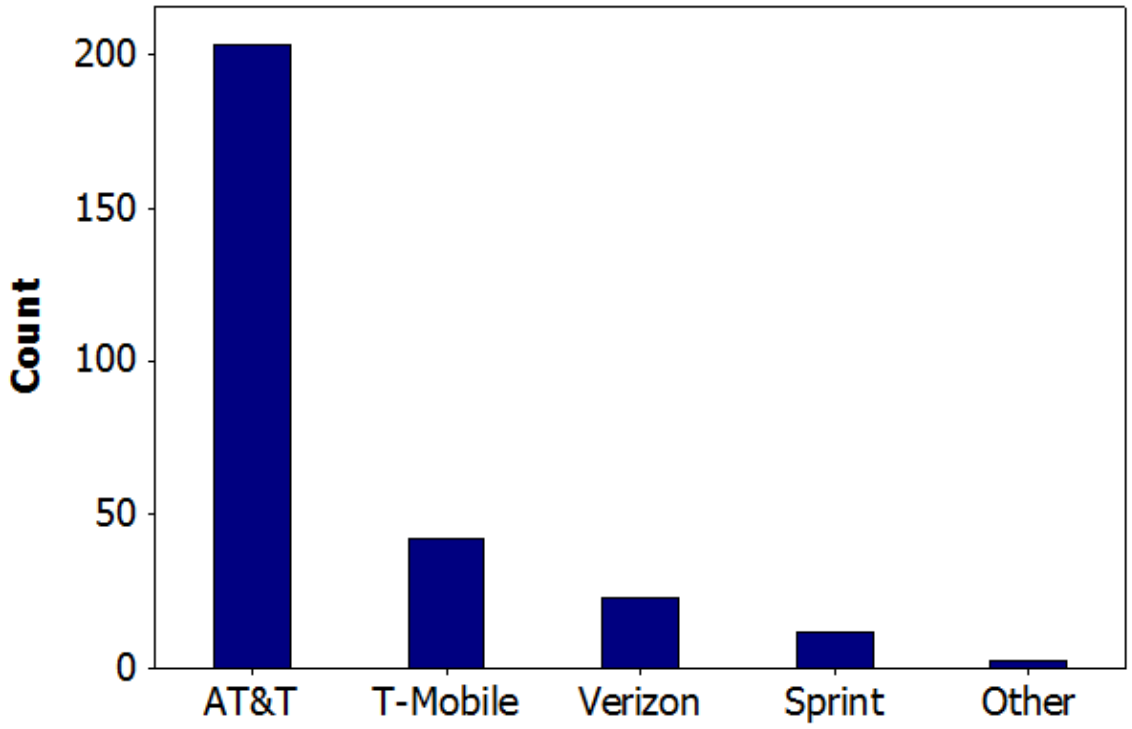


Figure 7.14. Respondents Mobile Service Provider

This has not significantly changed from the first survey

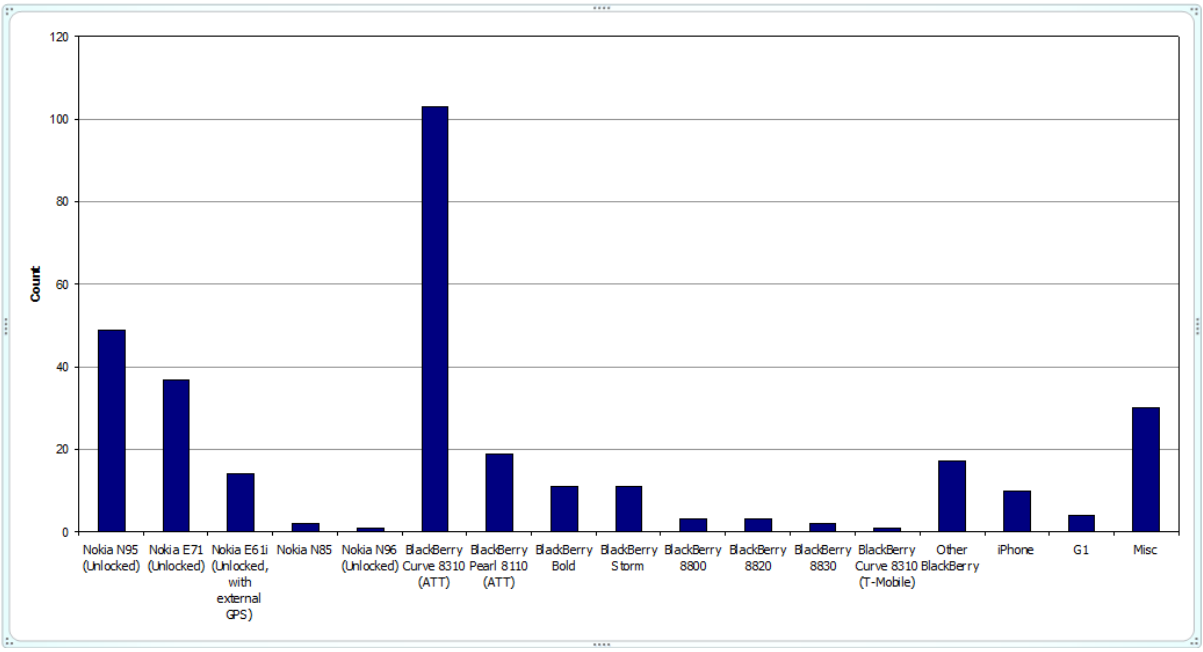


Figure 7.15. Respondents Device

Once again this has not changed significantly from the first survey

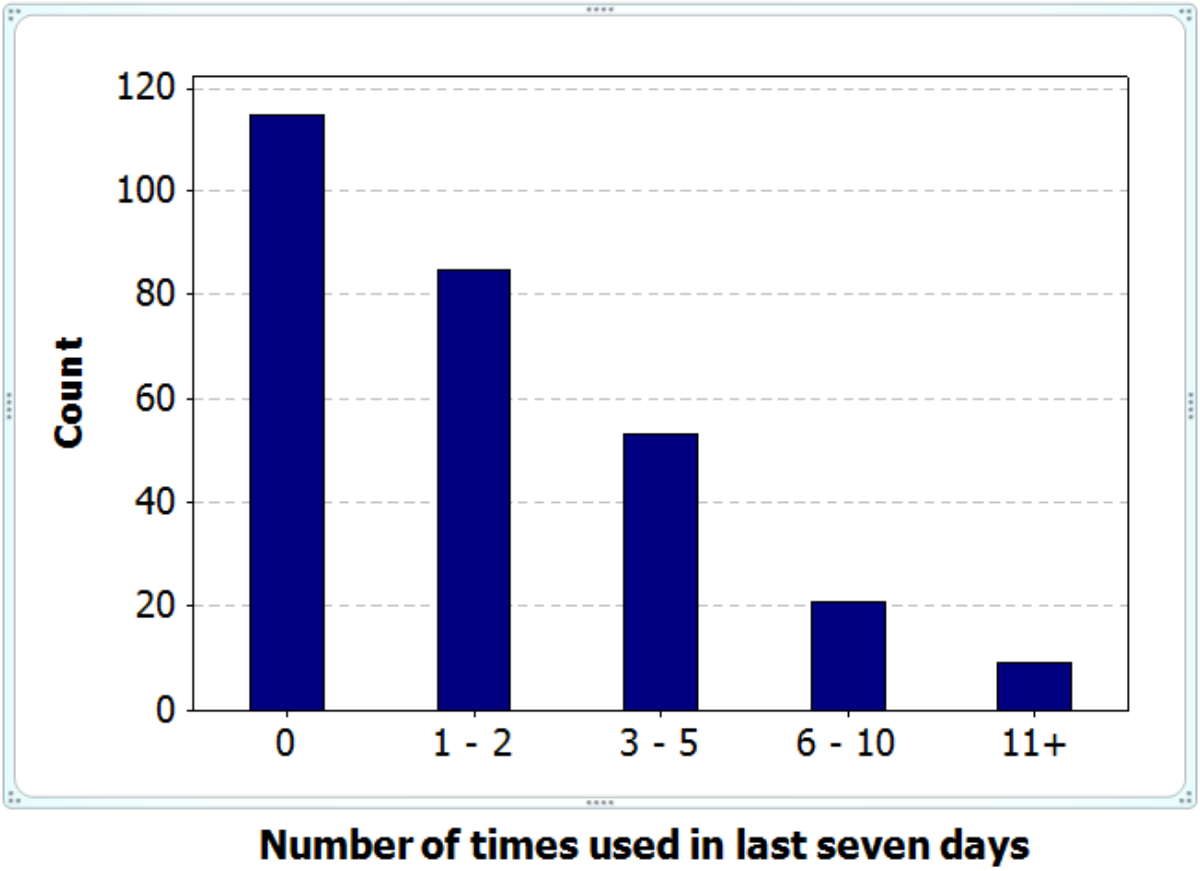


Figure 7.16. How many times have respondents used the client in the last seven days?

This shows the drop off in the interest in the pilot and also we believe in the challenges (battery drain, etc) in using the pilot frequently.

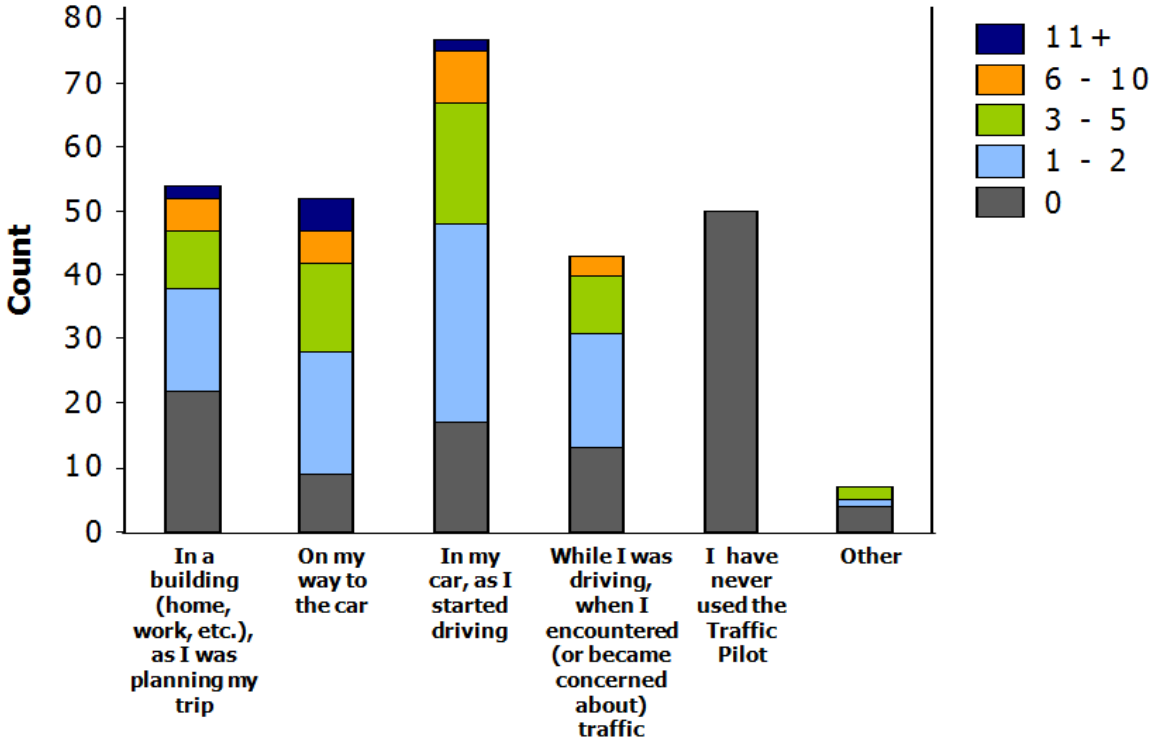


Figure 7.17. When did the respondent last use the traffic pilot?

Answers are pretty evenly distributed among the choices with use in the car the highest as would be expected.

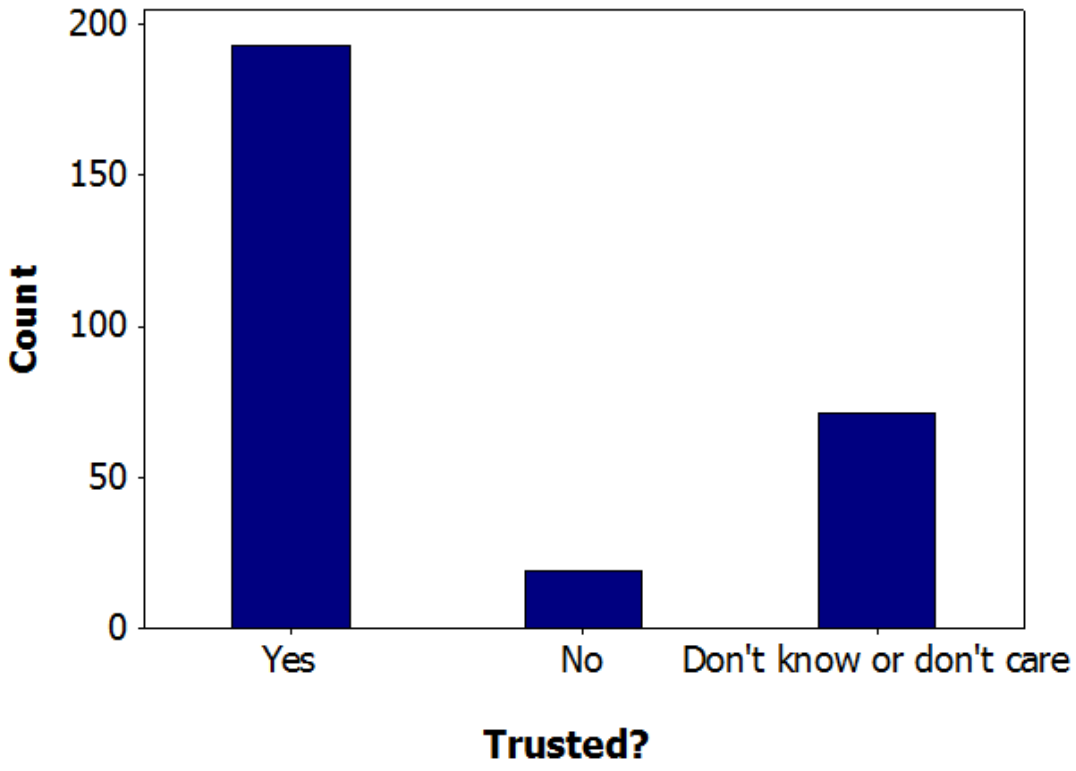


Figure 7.18. Privacy

This is important for two reasons. One, most folks trusted the application a primary goal of the pilot. However, the number who don't care is interesting. This seems to reflect current location aware applications where folks are willing to share their location without worry.

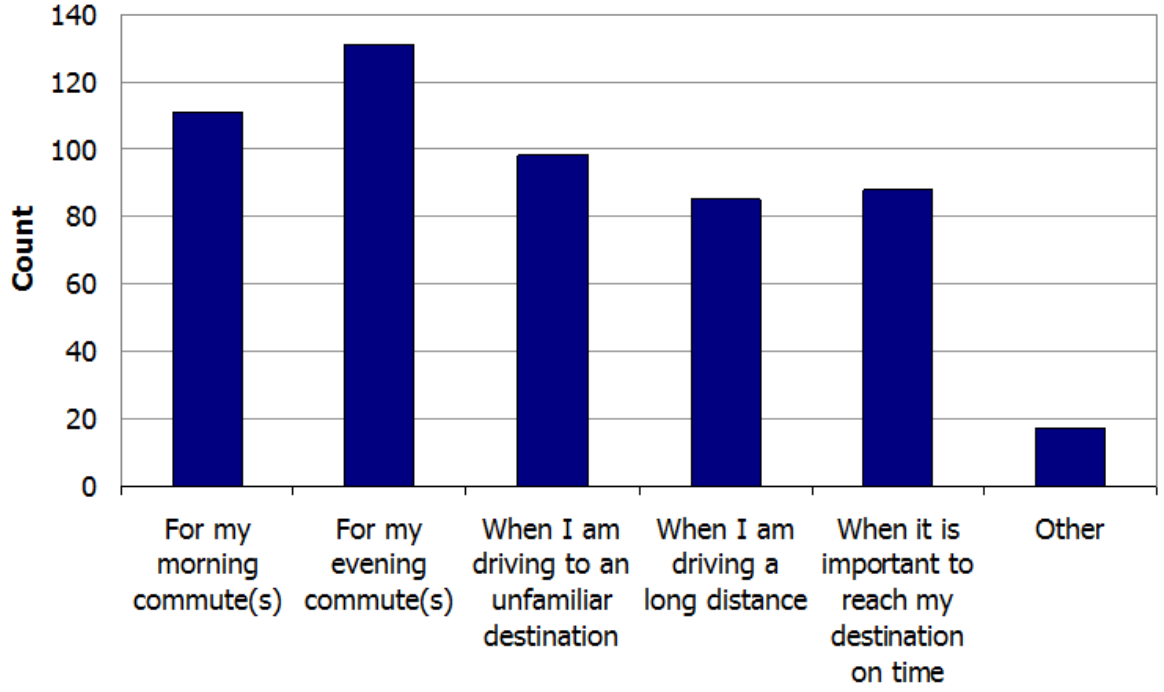


Figure 7.19. When do you use the traffic pilot?

Respondents use the application for a number of reasons. It appears to be a useful application.

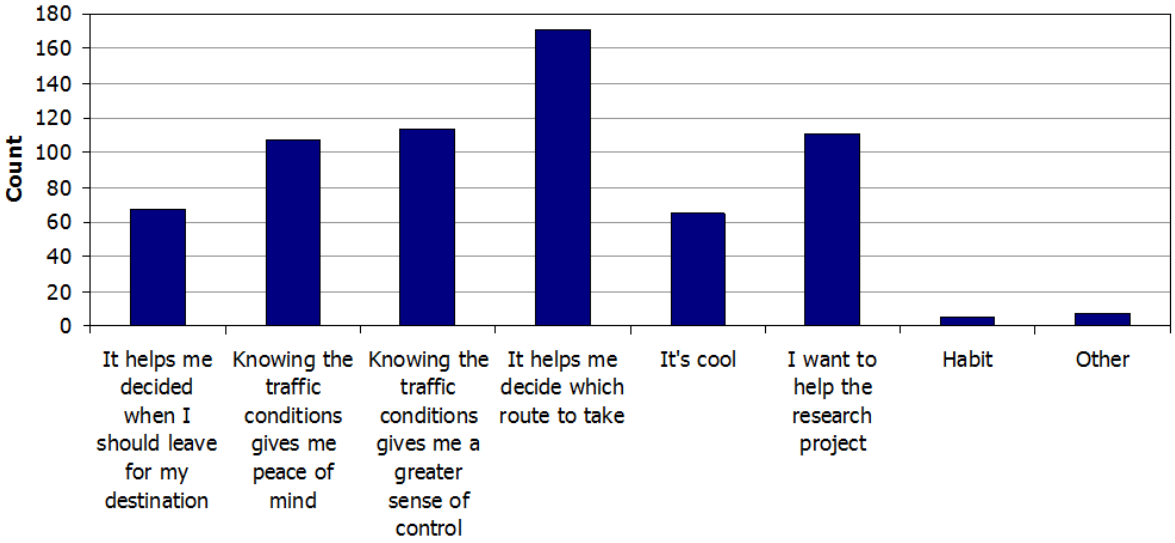


Figure 7.20. Why do you use the traffic pilot?

Respondents expressed that [the application] provides useful information and it is cool.

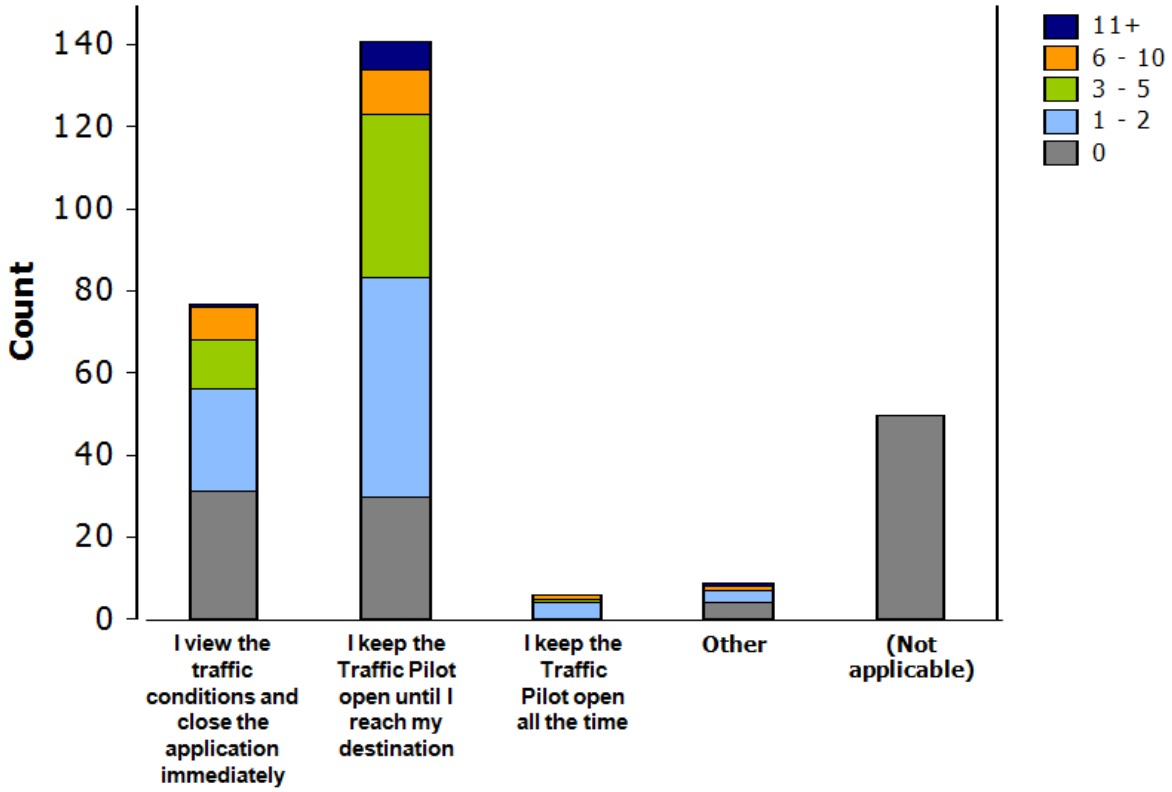


Figure 7.21. How long do you keep the application open?

Most respondents kept looking at the map in the car. This turned out to be a distracting driving issue.

- It does not work on my phone (37)
  - iPhone (9)
  - Unknown (8)
  - BlackBerry (4)
  - Samsung Blackjack II (3)
  - Windows Mobile device (2)
  - AT&T Tilt (2)
  - LG Vu (1)
  - Cingular 2125 (1)
  - Sprint Mogul (1)
  - Samsung Epix AT&T (1)
  - Palm Treo 650 (1)
  - HTC G1 (1)
  - Nokia N82 (1)
  - T-Mobile Dash (1)
  - Nokia AT&T Simbios60 (1)
- I was not able to download it (4)
- I do not know how (2)
- Other (7)
  - it was too slow and sucked the battery out of my phone.
  - tried 2 use but incountered problems, no return calls when i wanted info
  - Verizon
  - My smart phone died
  - It crashed when loading, tried twice, crashed twice, gave up.
  - not available in my area
  - i don't have GPS on my Blackberry

Figure 7.22. I have never used the traffic pilot because...

People would have used it but it did not work on their device.

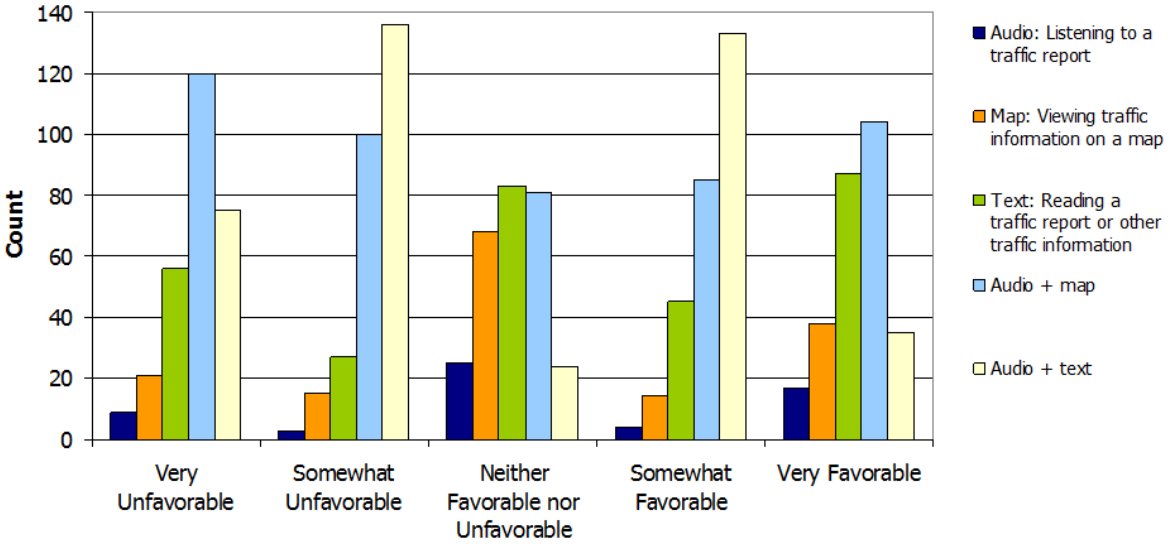


Figure 7.23. Opinion on form factors for receiving traffic information

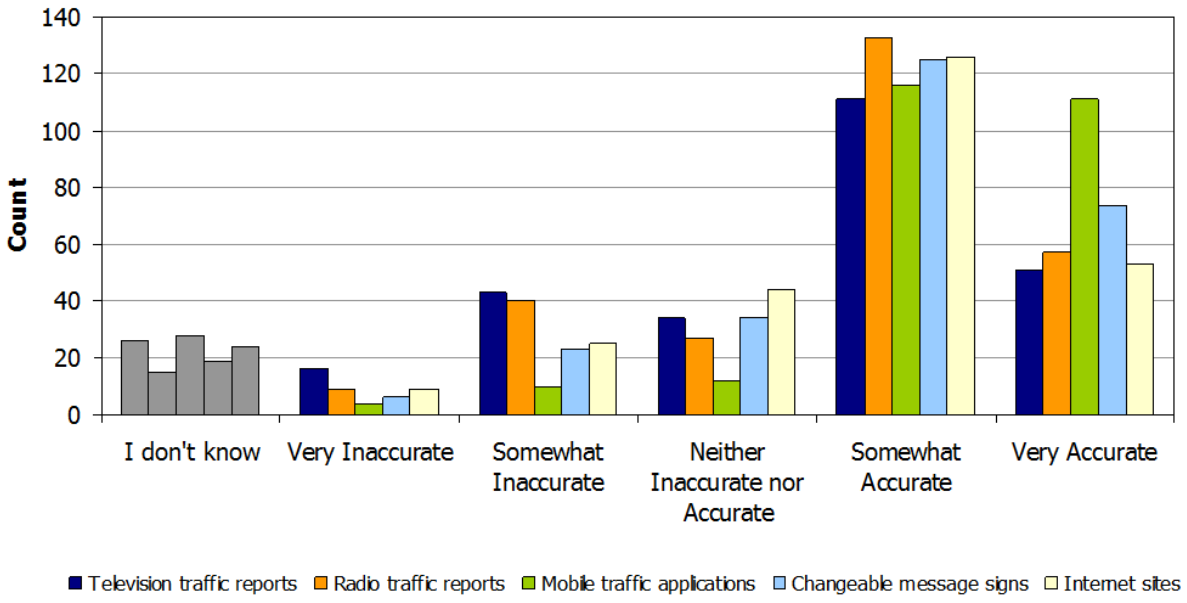


Figure 7.24. Opinion on accuracy of traffic information types

So most people think that all sources are somewhat accurate and the pilot users feel that they are using the more accurate method. One would hope this would be the opinion of the users.

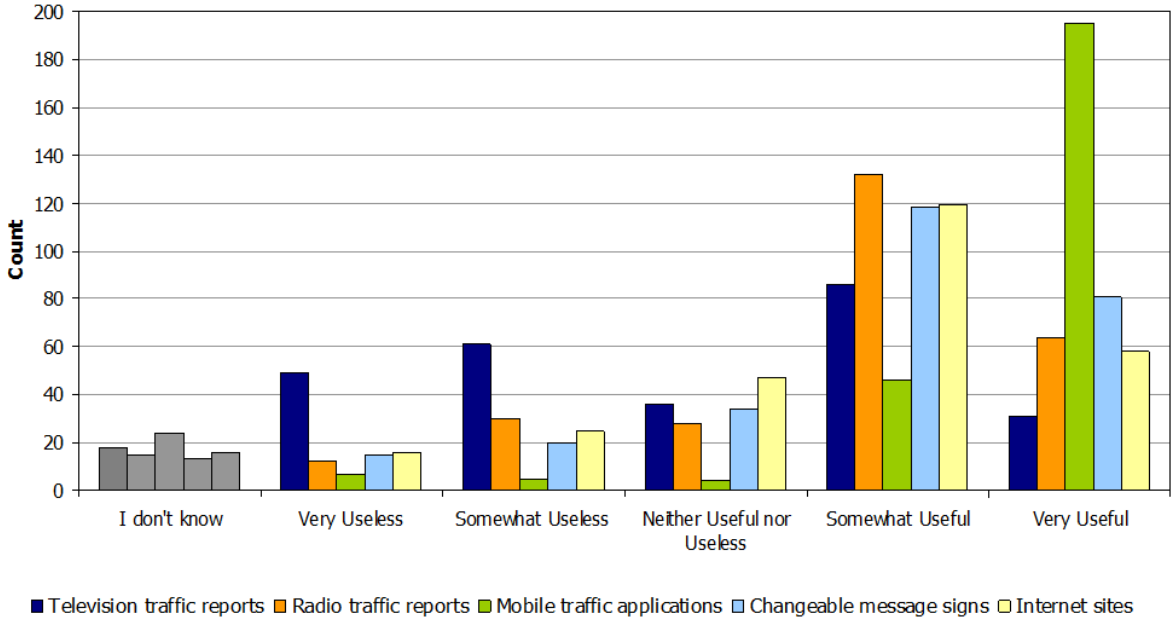


Figure 7.25. Opinion on usefulness of traffic types

This seems self selected for mobile pilot users.

4.8. Conclusion

Overall the Mobile Millennium project met its goals to “Engage the public and popularize ITS concepts”. It laid a solid scientific foundation for estimation of traffic conditions using speed data collected by cell phones in commuter cars. Traffic maps from user provided location data may seem mundane today but Mobile Millennium helped get it started and ensured that the results were accurate, privacy safe and attractive to a large number of public users. In terms of pure outreach it is hard to find a research project that received more worldwide attention by more governments, industries and consumers.

5. Demos and Field Experiments

5.1. Introduction

This chapter addresses the logistics associated with moving from Mobile Century, the 100 vehicle proof-of-concept experiment using GPS smartphones as traffic probes, to Mobile Millennium, the 5,000+ vehicle pilot deployment of a fully operational traffic information system utilizing data collected from individual smartphones. Summarized here is field operational test information–the nitty-gritty that connects algorithms and software to cell phones, drivers, and vehicles.

5.1.1. Moving from concept to field operational test

The tremendous success of the 1-day, 100-vehicle Mobile Century experiment on February 8, 2008, was quickly followed by a unanimous call to accelerate scale-up to a pilot deployment aimed at demonstrating the viability of utilizing VTL-based sensing data from individual smart phones to be the core of a real-time mobile traffic information system. The pilot deployment, aptly named Mobile Millennium, was envisioned to include a minimum of 1,000 vehicles operated by "Early Adopters" who used smart phones downloaded with the traffic system software (the client).

Driving acceleration of the scale-up effort was the desire by the research partners (Caltrans, U.S. DOT, RITA-SafeTrip 21, Nokia, NAVTEQ, and UC Berkeley) to showcase a fully operational GPS smart phone traffic information system at the 15th World Congress on Intelligent Transportation Systems held in New York City in November 2008.

Accomplishing this objective would require the development and execution of a carefully orchestrated series of real-time road tests that included both highway and arterial routes. Adding to the logistical complexity was the need to perform road tests in both the San Francisco Bay Area and in New York City concurrently with development and debugging of the system. Section 3 provides a chronology of the operational tests that led to the public launch of Mobile Millennium on November 10, 2008 in Berkeley, CA and the successful real-time demonstration of the system on November 19, 2008 in New York City, NY.

The launch was followed by more than six months of real-time system operation in the San Francisco-Sacramento, CA regional area that included the voluntary participation of more than 5,000 smart phone users. To say that the success of Mobile Millennium rested on the satisfactory execution of thousands of details, often performed on the fly in spite of thoughtful and exhaustive planning, is a considerable understatement.

Having little more than eight months before the scheduled Millennium demonstration at the ITS-World Congress required a quick evaluation of the field procedures used for the Century experiment and an assessment of their suitability for scale-up.

Key among these was the recruitment and training of test vehicle drivers, the availability of vehicles and areas to stage them, the selection of routes both in the San Francisco Bay Area and New York City, and, coming to grips with the costs associated with the rubber meeting the road. Perhaps most important of all, however, was development of the plan to identify, recruit, and retain thousands of "Early Adopters" who would agree to utilize the system during the six-month pilot period. Each of these will be given attention in Section 2 on field test components.

Before moving to a discussion of the field components however, it is important to understand the purpose of the pilot deployment, discussed in Section 2, and to have an appreciation for the incredibly compressed timeframe in which it was accomplished, summarized in the chronology provided in Section 3.

5.1.2. Purpose of the field operational field (FOT)

The typical ITS innovation follows a linear progression that can take years, even decades, to move from research and development to proof-of-concept to pilot deployment and ultimately to a product.

In the case of Mobile Millennium, however, the entire process was fast-tracked to operational status in just eight months to meet the requirements of the U.S. DOT-RITA SafeTrip-21 program – namely that a robust, market-ready, real-time traffic application that was accessible and easy to use by the traveling public be showcased at the ITS-World Congress in November 2008.

With early November 2008 established as the "go live" target, chief among the objectives of the partnership was to quickly leverage the "tech buzz" produced by Mobile Century and to translate that interest into recruitment of so-called Early Adopters (volunteer participants) who would become regular users of the Mobile Millennium traffic software.

Mobile Millennium accelerated the process from research and development to a fully operational system scalable for widespread geographic implementation in less than one year. Central to the partnership achieving accelerated deployment of Millennium was differentiation of the FOT along two tracks – one focusing on the operation of the system, and the other addressing system adoption by consumers (Early Adopters). Employing two distinct tracks for the FOT enabled the partners to concurrently address system development, scale-up modifications, refinement of the models, promotion to consumers, and outreach.

In effect, the FOT served to improve the rigor of the Mobile Millennium system for daily, twenty-four hour operational use by thousands of participants as well as to build interest in, and enthusiasm for, participatory sensing.

5.1.3. Chronology

This section provides a chronology of the field tests and demonstrations undertaken to advance from Mobile Century to Mobile Millennium.

- **February 8, 2008** Mobile Century proof-of-concept experiment completed.
- **July, 2008** Review of Mobile Century experiment logistics undertaken to begin preparation of the Mobile Millennium field test plan.
- **August, 2008** A series of 3-hour, 20 vehicle road tests using the traffic client are planned for Berkeley-San Francisco, CA and New York City, NY.
- **August 2-20, 2008** Drivers are recruited and trained for the Berkeley-San Francisco road tests.
- **August 21, 2008** Pre-test of the Berkeley-San Francisco highway and arterial routes.
- **September 2008** Recruitment of volunteer participants "Early Adopters" begins with outreach to Caltrans, University of California, Nokia, and NAVTEQ personnel.

The Mobile Millennium website (`fhttp://traffic.berkeley.edu`) goes live.

Enterprise Rent-A-Car, Manhattan is contracted to supply vehicles and drivers for the New York City road tests and ITS World Congress demonstration.

- **September 3, 2008** The first 3-hour, 20 vehicle Berkeley to San Francisco highway and arterial model test is performed.
- **September 17, 2008** The second 3-hour, 20 vehicle Berkeley to San Francisco highway and arterial model test is performed.
- **September 24, 2008** The first 3-hour, 20 vehicle New York City arterial test is performed.

Figure 1.1. A Mobile Millennium test vehicle ready for field test deployment in Berkeley, CA. Note that the vehicle includes a driver and an observer (passenger) responsible for recording traffic information along the route.

- **October 15, 2008** The second 3-hour, 20 vehicle New York City arterial test is performed.
- **October 29, 2008** The third and final 3-hour, 20 vehicle New York City arterial test is performed.
- **November 10, 2008** Mobile Millennium Pilot Launch in Berkeley, CA. Early adopters are given access to download the traffic client software to their smartphone.
- **November 19, 2008** A live, real-time, 3 hour, 20 vehicle demonstration of Mobile Millennium is presented at the 15th ITS World Congress in New York City, NY.

Figure 1.2. Early adopters at the November 10, 2008 Berkeley, CA launch register to receive the Mobile Millennium software on their smartphones.

- **November 19, 2008** The second wave of Early Adopters are given access to download the traffic client software at the ITS World Congress in New York City, NY
- **March-April, 2009** Weekly, weekday, 3 hour, 20 vehicle road tests using 5 different overlapping routes begin in Berkeley, CA to validate the arterial model
- **April 13, 2009** Mobile Millennium is demonstrated at the IEEE HSCC/RTAS Conference in San Francisco, CA
- **October 16-26, 2009** Mobile Millennium is demonstrated in real-time at the annual AASHTO conference in Palm Desert, CA
- **November, 2009** Downloading of the Mobile Millennium traffic client concludes
- **February, 2010** U.S. DOT Evaluation of Mobile Millennium

5.2. Field test components

The Mobile Millennium FOT was implemented using two tracks, (i) a system operations track, focused on road testing in the San Francisco Bay Area and New York City, and (ii) a consumer

adoption track focused on the recruitment and retention of Early Adopters (consumers). Both tracks are highlighted in this section.

5.2.1 System operation

The focus of the system operation track was to validate the accuracy of the highway and arterial model algorithms developed by the UC Berkeley research team. This included road tests in both the San Francisco Bay Area and New York City. What follows is a summary of the system operation components divided by geographic location.

5.2.2 System operation testing: Berkeley-San Francisco, CA

- Test route identification - Berkeley-San Francisco

  - Routes proximate to the UC Berkeley campus research team
  - Routes with an estimated free-flow travel time of 15-20 minutes per cycle (loop)
  - Routes (loops) unidirectional with signal-controlled intersections and left and right turn combinations
  - Routes where a penetration of 2-5% could be maintained throughout the test

- Recruitment and training of vehicle drivers - Berkeley-San Francisco

  - Drivers are graduate and undergraduate UC Berkeley students at least 18 years of age and in possession of a valid U.S. driver's license
  - Drivers must be English speaking
  - Drivers must have a clean driving record
  - Drivers must be familiar with California Department of Motor Vehicle regulations
  - Drivers must attend an orientation and safety briefing
  - Drivers must be able to drive continuously for 3 hours following a prescribed route
  - Drivers must be available to drive weekdays during peak morning (6-10 AM) and afternoon (3-7 PM) commute hours
  - Drivers must be familiar with Bay Area roadways
  - Driver's briefed on route directions, safety, and accident reporting procedures prior to each deployment
  - Driver's acknowledge that they have not consumed alcohol or drugs for at least 24 hours prior to commencement of each field test
  - Driver's complete pre- and post-field test checklists

Figure 2.1. An example of a flyer used to recruit graduate student drivers for Mobile Millennium field tests.

- Pre- and post-field test checklists - Berkeley-San Francisco
  - Pre-deployment checklist

    1. Driver acknowledges that s/he has not consumed alcohol or taken drugs within the last 24 hours.
    2. Driver receives key, fuel card, and bridge toll for assigned vehicle
    3. Driver inspects interior and exterior of vehicle and notes any damage or deficiency
    4. Driver confirms that magnetic decals (hood number and side door identification) are safely affixed to vehicle
    5. Driver adjusts seat and mirrors
    6. Driver checks that phone cradle, charger, and phone are installed and secure

  - Post-deployment checklist

    1. Driver returns vehicle refueled to pre-deployment level
    2. Driver removes phone, phone cradle, and charger from vehicle
    3. Driver removes all personal possessions from vehicle
    4. Driver removes magnetic decals from vehicle
    5. Driver inspects interior and exterior of vehicle and notes any damage or deficiency
    6. Driver parks and secures vehicle for release to Enterprise RAC
    7. Driver surrenders vehicle key, fuel card, fuel and bridge toll receipts

- Rental and staging of test vehicles - Berkeley-San Francisco
  - Vehicle rental via Enterprise-Rent-A-Car per University of California vendor agreement
  - Vehicles assembled by Enterprise at designated staging area (Shattuck Ave @ Derby, Berkeley, CA) at least one hour prior to scheduled deployment
  - Each vehicle given a pre- and post-deployment inspection checked by driver
  - Each vehicle equipped with a magnetic number decal for placement on the vehicle's hood, and two side door identification decals
  - Each test vehicle equipped with phone cradle secured to lower passenger side of windshield
  - Each vehicle equipped with a Nokia N95 phone (installed with the latest version of the traffic client) and auxiliary power phone charger
  - Each vehicle time-released from staging area at 1-minute intervals
  - Vehicles refueled by driver's prior to re-assembling at staging area

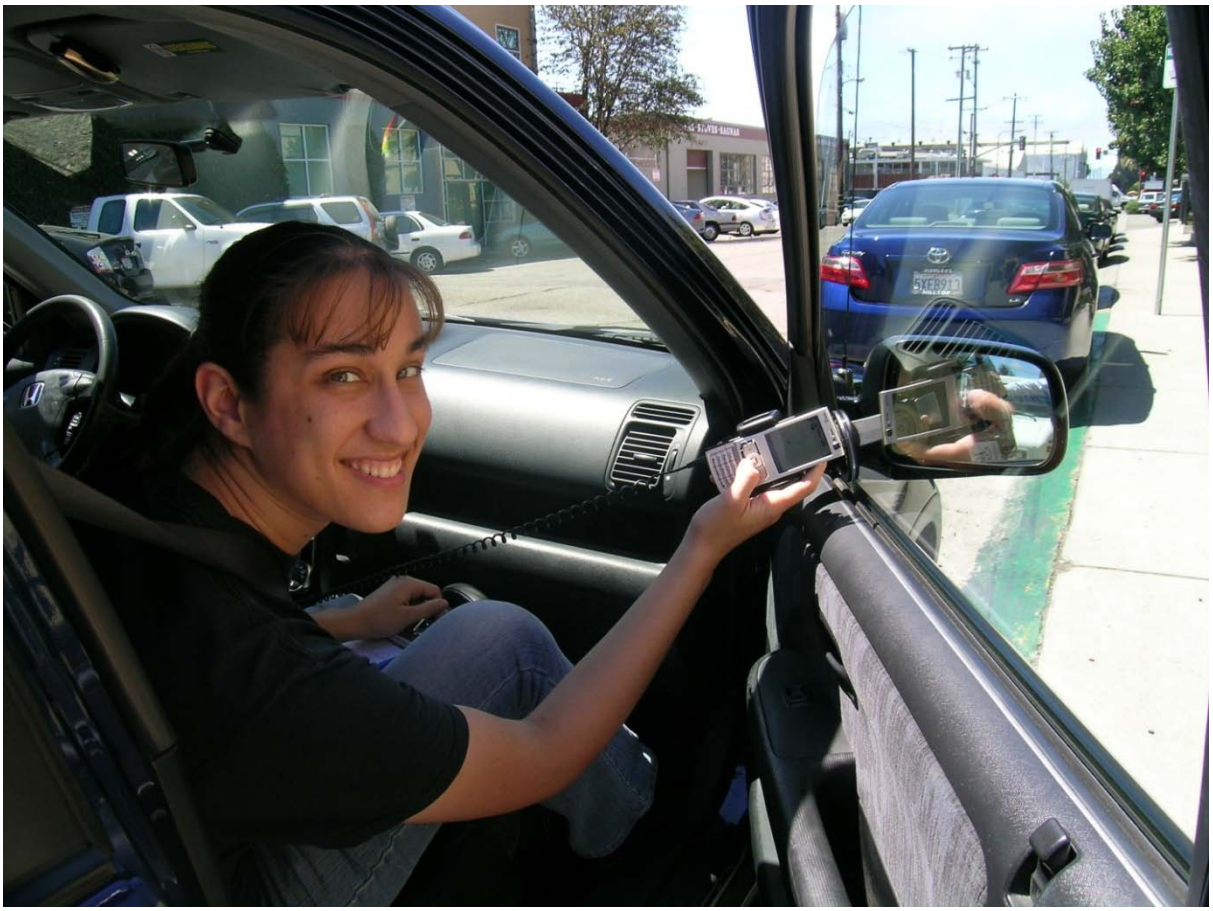


Figure 2.2. Student researcher Sarah Stern updates a Nokia N95 phone with the latest version of the Mobile Millennium traffic client software prior to a Berkeley-San Francisco field test.

- Download of the latest version of the Millennium client software to test phones
  - Nokia N95 phones downloaded with the latest version of the traffic client software one hour prior to each deployment (this procedure typically required 1-2 minutes per phone)
  - Each Nokia N95 phone checked for GPS acquisition prior and operating traffic client immediately prior to release from field test staging area (this procedure typically took 20-60 seconds per phone)

5.2.3 Berkeley-San Francisco, CA road tests

(a) Berkeley-San Francisco Pre-test, August 21, 2008

Type: Highway & arterial pre-test, Loops 1-3

Scope: 21 vehicles, (20 rental cars + 1 SUV service vehicle); Berkeley to San Francisco and return - via I 80 Bay Bridge; vehicles staged at Enterprise Rent-A-Car overflow lot, Shattuck and Derby.

Drivers/monitors: 21 pre-screened student drivers + monitoring station - 4 students (laptops, counters, power supply, table/chairs, safety gear); drivers and monitors shuttled from Hearst Mining Circle, UC Berkeley to Shattuck/Derby staging area and returned to campus at conclusion of pre-test.

Route: Leave Berkeley across Oakland/Bay Bridge to San Francisco via Fremont exit to Embarcadero-North Point-Columbus-Broadway left turn loop.

Vehicle prep: In accordance with pre-deployment checklists.

Schedule: Depart Berkeley staging area at 12:00 PM PST. Arterial loop drive time 1 to 1.5 hours. Highway drive time 1 to 1.5 hours.

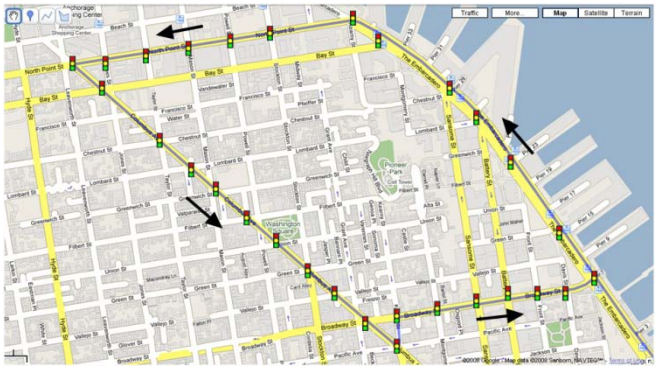


Figure 2.3. Map of Loop 1, the downtown San Francisco arterial portion (Embarcadero-North Point-Columbus-Broadway) of the Berkeley-San Francisco route pre-test August 21, 2008.

Monitor notes and observations (unedited):

1. Proceeded with travel along the designated route from Berkeley staging area across SF-Bay Bridge to Embarcadero.
2. Along Embarcadero there is a parking lane to the far right and a bike lane in between the through lanes and parking lane. Embarcadero switches between three and two lanes. There are also two lane left turn pockets that appear along Embarcadero.
3. Left turn onto North Point is long. There is not a large noticeable elevation change on North Point. North point is two lanes on one side and one lane on the other. Parking lanes are on the far side of both sides.
4. Congestion always experienced on Columbus around Stockton.
5. Left turn onto Kearney is not a good turn. Recommend alternate route using Bay.
6. Major traffic on Broadway caused by construction. We missed three light cycles at Broadway and Sansome. Along Broadway, signals at Battery, Front, and Davis appear to be synchronized. Start time: 12:59p. End time: 1:18p. Total time: ~19 min.

Figure 2.4. An example of downtown San Francisco traffic flow along Columbus during the route selection pre-test.

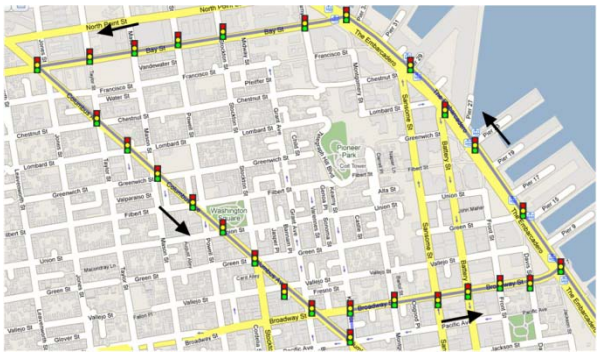


Figure 2.5. Map of Loop 2, showing the downtown San Francisco portion of the Berkeley-San Francisco route pre-test August 21, 2008.

Monitor notes and observations (unedited):

1. Repeat of Embarcadero loop.
2. No change along Embarcadero.
3. The left turn onto North Point was green this time.
4. Electric buses along Columbus cause congestion because they block a lane of traffic since they cannot pull over all the way. We did not make it through the light cycle at Union and Columbus because of taxi congestion and a guy trying to make a left turn, which held up traffic.
5. Congestion continued at Stockton.
6. Left turn onto Kearney is not recommended.
7. The construction on Broadway did not cause congestion this time. We breezed through Broadway.
8. Start time: 1:18p. End time: 1:35p. Total time: ~17 min.

Figure 2.6. Map of the downtown San Francisco portion (Embarcadero-Bay-Columbus-Broadway) of Loop 3 used for the Berkeley-San Francisco route pre-test August 21, 2008.

Monitor notes and observations (unedited):

1. Modified Embarcadero loop to shorten route and use Bay instead of North Point because the left turn is more natural.
2. Everything along Embarcadero is consistent.

3. Left turn onto Bay was smooth; it was green. There was minor construction on Bay.

4. Congestion always experienced on Columbus around Stockton. A delivery truck caused slight congestion on Columbus.
5. Recommendation for changing left turn on Kearney is the same.
6. There were no problems along Broadway.
7. Start time: 1:35p. End time: 2:15p. Detour: 1:41-2:04p. Total time: ~17 min (excluding detour).

(b) Berkeley-San Francisco highway/arterial test 1, September 3, 2008

Type: Highway & arterial (penetration rate 2-5%)

Scope: 21 vehicles, (20 rental cars + 1 SUV service vehicle); Berkeley to San Francisco and return - via I 80 Bay Bridge; vehicles staged at Enterprise Rent-A-Car overflow lot, Shattuck and Derby.

Drivers/monitors: 21 pre-screened student drivers + monitoring station - 4 students (laptops, counters, power supply, table/chairs, safety gear); drivers and monitors shuttled from Hearst Mining Circle, UC Berkeley to Shattuck/Derby staging area and returned to campus at conclusion of test.

Route: Leave Berkeley across Oakland/Bay Bridge to San Francisco via Fremont exit to Embarcadero-North Point-Columbus-Broadway left turn loop.

Vehicle prep: In accordance with pre-deployment checklists.

Schedule: Depart Berkeley staging area 11:00 a.m. Complete final loop cycle by 3:00 p.m. and return to staging area via Bay Bridge.

Figure 2.7. An example of traffic congestion experienced during a field test.

(c) Berkeley-San Francisco highway/arterial test 2, September 17, 2008

Type: Highway & arterial (penetration rate 2-5%)

Scope: 21 vehicles, (20 rental cars + 1 SUV service vehicle); Berkeley to San Francisco and return - via I 80 Bay Bridge; vehicles staged at Enterprise Rent-A-Car overflow lot, Shattuck and Derby.

Drivers/monitors: 21 pre-screened student drivers + monitoring station - 4 students (laptops, counters, power supply, table/chairs, safety gear); drivers and monitors shuttled from Hearst Mining Circle, UC Berkeley to Shattuck/Derby staging area and returned to campus at conclusion of test.

Route: Leave Berkeley across Oakland/Bay Bridge to San Francisco via Fremont exit to Market-Mission left turn loop.

Vehicle prep: In accordance with pre-deployment checklists.

Schedule: Depart Berkeley staging area 11:00 a.m. Complete final loop cycle by 2:00 p.m. and

return to staging area via Bay Bridge.

Stoplights Direction of traffic

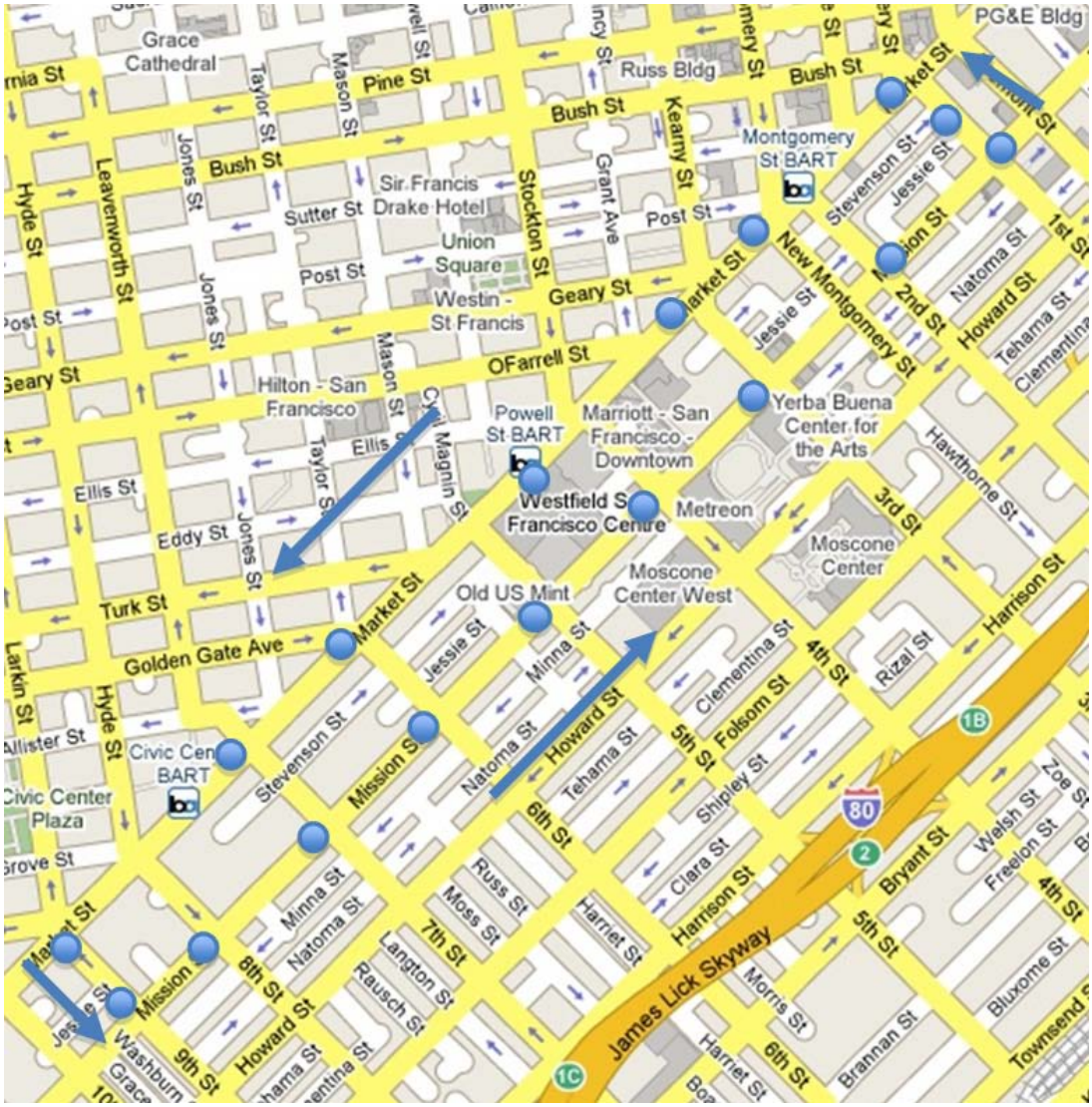


Figure 2.8. Map of the Mission-Market left turn loop utilized for highway/arterial model test 2. Dots indicate traffic signals; arrows indicate the direction of traffic.

(d) Berkeley-San Francisco arterial model validation tests, March-April, 2009

Type: Arterial validation (penetration rate 2-5%)

Scope: 21 vehicles, (20 rental cars + 1 SUV service vehicle); vehicles staged at Enterprise Rent-A-Car overflow lot, Shattuck and Derby. Weekly tests of 3 hours each performed alternately (MWF and TuTh).

Drivers/monitors: 21 pre-screened student drivers + monitoring station - 4 students (laptops, counters, power supply, table/chairs, safety gear); drivers and monitors shuttled from Hearst Mining Circle, UC Berkeley to Shattuck/Derby staging area and returned to campus at conclusion of test.

Routes: All vehicles depart from staging area for deployment along 5 separate overlapping routes - 4 vehicles assigned to each route (Figures 2.9 through 2.13) - near downtown Berkeley.

Vehicle prep: In accordance with pre-deployment checklists.

Schedule: Depart Berkeley staging area 12:00 PM PST, complete final loop cycle by 3:00 p.m. and return to Shattuck/Derby staging area. Weekly tests performed alternately on MWF and TuTh.

Week 1: WMF, routes 1-5, 4 vehicles each route, depart 12 PM, final loop 3 PM

Week 2: TuTh, routes 1-5, 4 vehicles each route, depart 12 PM, final loop 3 PM

Week 3: MWF, routes 1-5, 4 vehicles each route, depart 12 PM, final loop 3 PM

Week 4: TuTh, routes 1-5, 4 vehicles each route, depart 12 PM, final loop 3 PM

Week 5: MWF, routes 1-5, 4 vehicles each route, depart 12 PM, final loop 3 PM

Week 6: TuTh, routes 1-5, 4 vehicles each route, depart 12 PM, final loop 3 PM

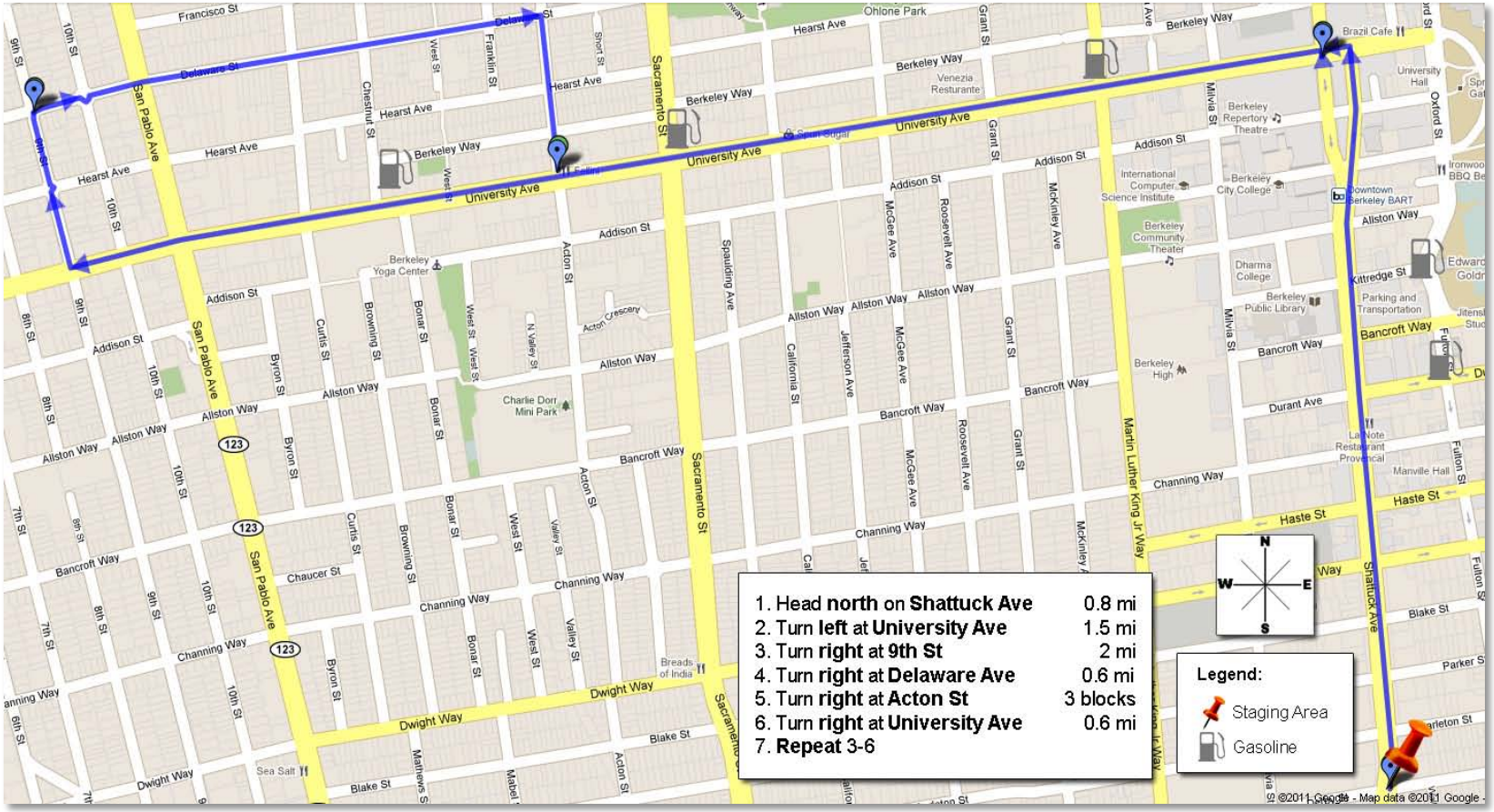


Figure 2.9: Berkeley Arterial Validation Route 1

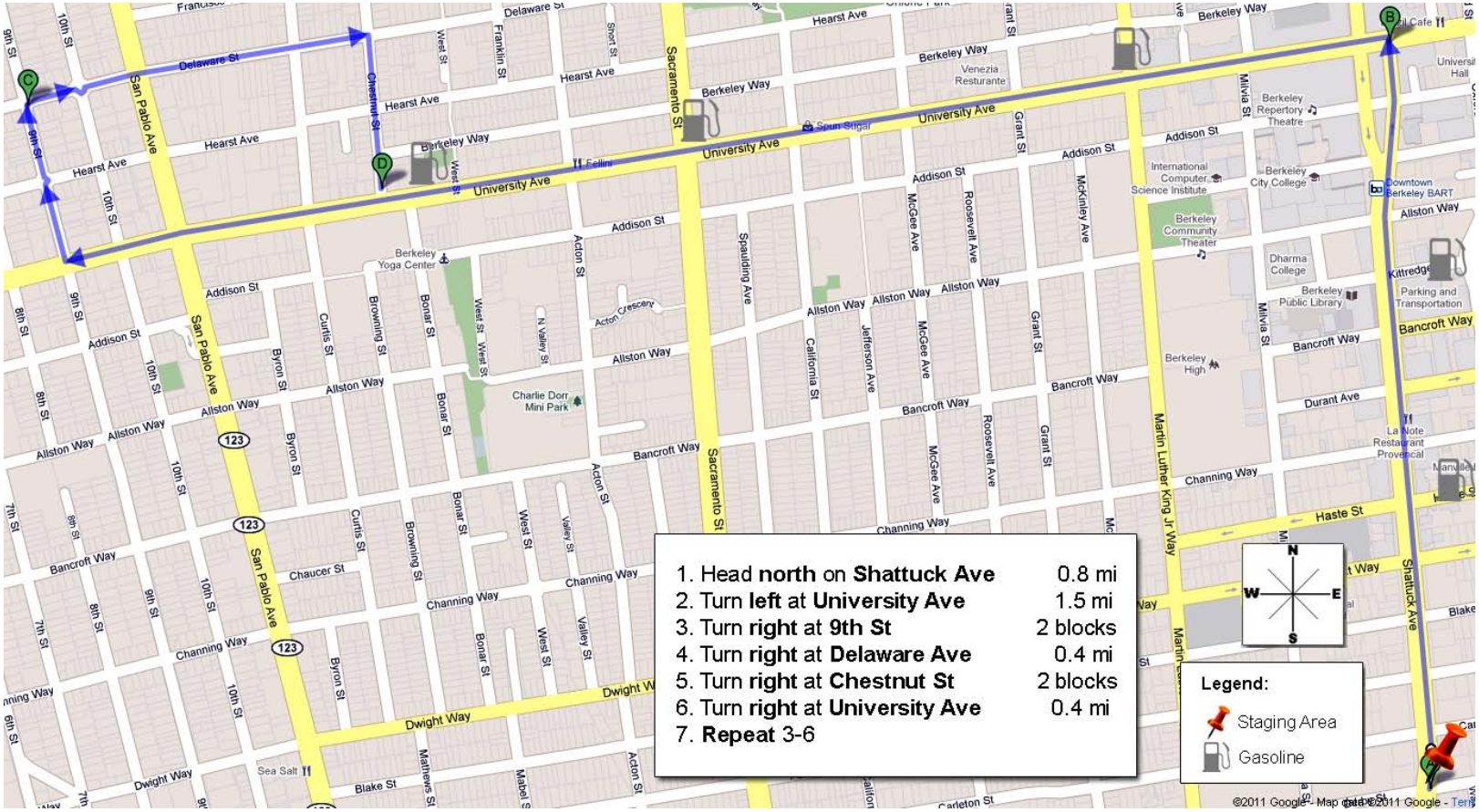


Figure 2.10: Berkeley Arterial Validation Route 2

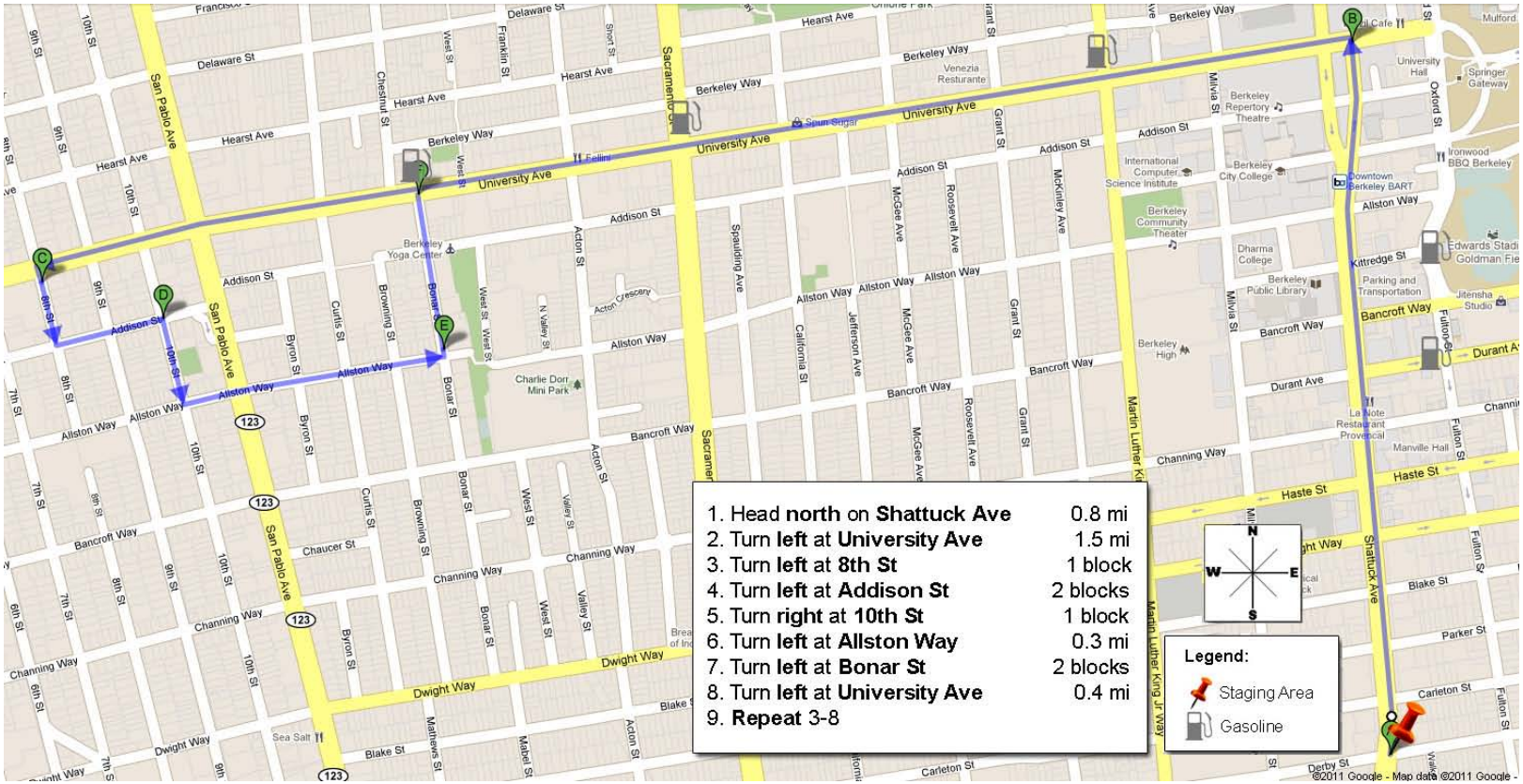


Figure 2.11: Berkeley Arterial Validation Route 3

Figure 2.12: Berkeley Arterial Validation Route 4

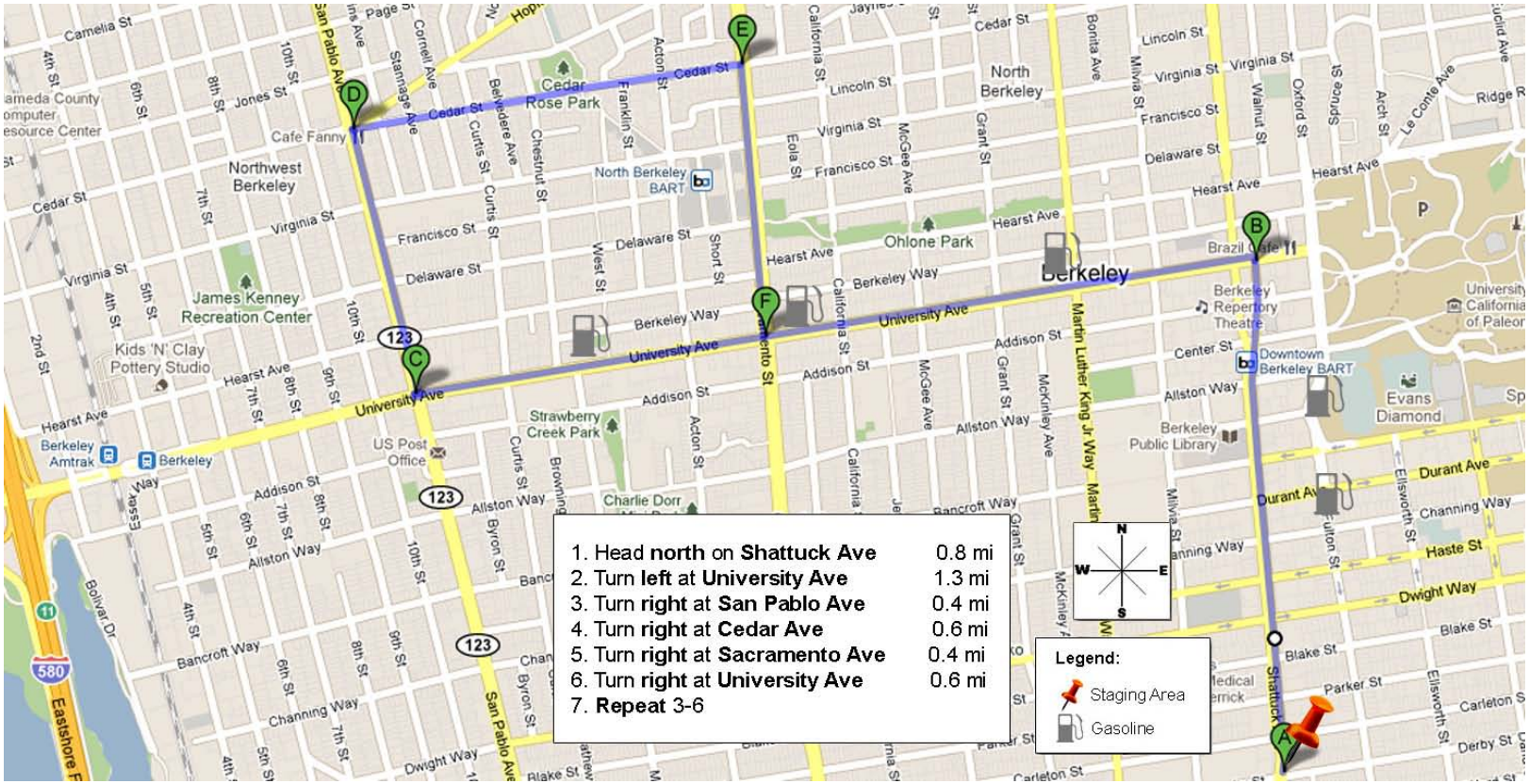


Figure 2.13: Berkeley Arterial Validation Route 5

5.2.4. System operation testing: New York City, NY

- Identification of test route - New York City

  - Route proximate to the Javits Convention Center complex
  - Route with an estimated free-flow travel time of 10-15 minutes per cycle (loop)
  - Route (loop) unidirectional with signal-controlled intersections
  - Route where a penetration of 2-5% could be maintained throughout the test

- Recruitment and training of vehicle drivers - New York City

  - All drivers are employees of Enterprise RAC - 5 drivers from each NYC office designated to participate in road tests. UC negotiates a per vehicle price for each road test inclusive of vehicle rental, vehicle delivery and removal, staging facility, and driver.
  - Drivers must be English speaking
  - Drivers must have a clean driving record
  - Drivers must be familiar with New York Department of Motor Vehicle regulations
  - Drivers must be able to drive continuously for 3 hours following a prescribed route
  - Drivers must be available to drive weekdays during morning commute period (6AM-12PM)
  - Drivers must be familiar with New York City roadways
  - Driver's briefed on route directions, safety, and accident reporting procedures prior to each deployment
  - Driver's acknowledge that they have not consumed alcohol or drugs for at least 24 hours prior to commencement of each field test
  - Driver's complete pre- and post-field test checklists

    - Pre- and post-field test checklists - New York City

      - Pre-deployment checklist

        1. Driver acknowledges that s/he has not consumed alcohol or taken drugs within the last 24 hours.
        2. Driver receives key and refueling card
        3. Driver inspects interior and exterior of vehicle and notes any damage or deficiency
        4. Driver confirms that magnetic decals (hood number and side door identification) are safely affixed to the vehicle
        5. Driver adjusts seat and mirrors
        6. Driver checks that phone cradle, charger, and phone are installed and secure prior to departure for from staging facility

      - Post-deployment checklist

1. Driver returns vehicle refueled to pre-deployment level
2. Driver removes phone, phone cradle, and charger from vehicle
3. Driver removes all personal possessions from vehicle
4. Driver removes magnetic decals from vehicle
5. Driver inspects interior and exterior of vehicle and notes any damage or deficiency
6. Driver parks and secures vehicle in designated area of Enterprise RAC facility
7. Driver surrenders vehicle key, fuel card, and fuel receipts

5.2.5. New York City, NY arterial testing

(a) NYC test 1, September 24, 2008

Type: Arterial (penetration rate 2-5%)

Scope: 20 vehicles staged at Enterprise Rent-A-Car Manhattan indoor facility. Enterprise contacts: Watt, Peterson, Friedman.

Drivers/monitors: 20 Enterprise RAC drivers + 3 members of the research team. Drivers and monitors assemble at Enterprise staging area. Final safety and route briefing given by UC research personnel prior to deployment.

Route: Leave staging area for 10th Ave., 34th St, 12 Ave loop and return. (Figure 2.14) Arterial loop drive time: 3 hours. Distance: 2.3 miles per loop.

Vehicle prep: In accordance with pre-deployment checklists.

Deployment schedule:

7:00 AM	UC Research personnel arrive at Enterprise RAC indoor staging facility
7:15 AM	UC Research personnel download traffic client to all test phones and acquire GPS
7:30 AM	Enterprise RAC - Area 1 drivers and test vehicles arrive; pre-deployment activities begin per checklist
7:45 AM	Enterprise RAC Area 2 drivers and test vehicles arrive; pre-deployment activities begin per checklist
8:00 AM	Enterprise RAC Area 3 drivers and test vehicles arrive; pre-deployment activities begin per checklist
8:15 AM	Enterprise RAC Area 4 drivers and test vehicles arrive; pre-deployment activities begin per checklist
8:30 AM	Enterprise RAC Area 5 drivers and test vehicles arrive; pre-deployment activities begin per checklist
8:45 AM	Enterprise RAC drivers given final route and safety instructions.
9:00 AM	Test vehicles depart Enterprise RAC staging facility at 1-minute interval
12:00 PM	Enterprise RAC drivers and vehicles begin returning to staging facility

1:00 PM All drivers and vehicles complete post-deployment checklist
Field note (unedited):

1. Map: Blue and red dots represent traffic signals; black arrows are for actual movements at each signal. Along 10th Ave (one-way), there are 6 lanes, with the left and right lanes designated for parking.
2. There is a traffic signal on 34th St. in front of the Javits Center (red), which caused spillover and a 2-minute delay. This problem may be more severe during the demonstration given that the conference will be in progress at Javits.
3. There were some queues along 12th Ave., almost no queue on 10th Ave. The queue on 34th St. before turning on 12th Ave. was long. The signal at Javits contributes to this.
4. Actual loop time: 15-17 minutes (approximately 4.5 minutes per loop segment). All segments were stable.

Indicates traffic signals

Indicates number of lanes and direction of traffic

Figure 2.14. Map of NYC arterial route used for each NYC road test and the ITS World Congress demonstration. The red dot indicates the traffic signal nearest to the Javits Convention Center.

(b) NYC test 2, October 15

Type: Arterial (penetration rate 2-5%)

Scope: 20 vehicles staged at Enterprise Rent-A-Car Manhattan indoor facility. Enterprise contacts: Watt, Peterson, Friedman.

Drivers/monitors: 20 Enterprise RAC drivers + 3 members of the research team. Drivers and monitors assemble at Enterprise staging area. Final safety and route briefing given by UC research personnel prior to deployment.

Route: Leave staging area for 10th Ave., 34th St, 12 Ave loop and return. (Figure 2.14) Arterial loop drive time: 3 hours. Distance: 2.3 miles per loop.

Vehicle prep: In accordance with pre-deployment checklists.

Deployment schedule:

7:00 AM	UC Research personnel arrive at Enterprise RAC indoor staging facility
7:15 AM	UC Research personnel download traffic client to all test phones and acquire GPS
7:30 AM	Enterprise RAC - Area 1 drivers and test vehicles arrive; pre-deployment activities begin per checklist
7:45 AM	Enterprise RAC Area 2 drivers and test vehicles arrive; pre-deployment activities begin per checklist
8:00 AM	Enterprise RAC Area 3 drivers and test vehicles arrive; pre-deployment activities begin per checklist
8:15 AM	Enterprise RAC Area 4 drivers and test vehicles arrive; pre-deployment activities begin per checklist
8:30 AM	Enterprise RAC Area 5 drivers and test vehicles arrive; pre-deployment activities begin per checklist
8:45 AM	Enterprise RAC drivers given final route and safety instructions.
9:00 AM	Test vehicles depart Enterprise RAC staging facility at 1-minute interval
12:00 PM	Enterprise RAC drivers and vehicles begin returning to staging facility
1:00 PM	All drivers and vehicles complete post-deployment checklist

(c) NYC test 3, October 29

Type: Arterial (penetration rate 2-5%)

Scope: 20 vehicles staged at Enterprise Rent-A-Car Manhattan indoor facility. Enterprise contacts: Watt, Peterson, Friedman.

Drivers/monitors: 20 Enterprise RAC drivers + 3 members of the research team. Drivers and monitors assemble at Enterprise staging area. Final safety and route briefing given by UC research personnel prior to deployment.

Route: Leave staging area for 10th Ave., 34th St, 12 Ave loop and return. (Figure 2.14) Arterial loop drive time: 3 hours. Distance: 2.3 miles per loop.

Vehicle prep: In accordance with pre-deployment checklists.

Deployment schedule:

7:00 AM	UC Research personnel arrive at Enterprise RAC indoor staging facility
7:15 AM	UC Research personnel download traffic client to all test phones and acquire GPS
7:30 AM	Enterprise RAC - Area 1 drivers and test vehicles arrive; pre-deployment activities begin per checklist
7:45 AM	Enterprise RAC Area 2 drivers and test vehicles arrive; pre-deployment activities begin per checklist
8:00 AM	Enterprise RAC Area 3 drivers and test vehicles arrive; pre-deployment activities begin per checklist
8:15 AM	Enterprise RAC Area 4 drivers and test vehicles arrive; pre-deployment activities begin per checklist
8:30 AM	Enterprise RAC Area 5 drivers and test vehicles arrive; pre-deployment activities begin per checklist
8:45 AM	Enterprise RAC drivers given final route and safety instructions.
9:00 AM	Test vehicles depart Enterprise RAC staging facility at 1-minute interval
12:00 PM	Enterprise RAC drivers and vehicles begin returning to staging facility
1:00 PM	All drivers and vehicles complete post-deployment checklist

(d) NYC test 4, November 19

Type: Arterial (penetration rate 2-5%)

Scope: 20 vehicles staged at Enterprise Rent-A-Car Manhattan indoor facility. Enterprise contacts: Watt, Peterson, Friedman.

Drivers/monitors: 20 Enterprise RAC drivers + 3 members of the research team. Drivers and monitors assemble at Enterprise staging area. Final safety and route briefing given by UC research personnel prior to deployment.

Route: Leave staging area for 10th Ave., 34th St, 12 Ave loop and return. (Figure 2.14) Arterial loop drive time: 3 hours. Distance: 2.3 miles per loop.

Vehicle prep: In accordance with pre-deployment checklists.

Deployment schedule:

7:00 AM	UC Research personnel arrive at Enterprise RAC indoor staging facility
7:15 AM	UC Research personnel download traffic client to all test phones and acquire GPS
7:30 AM	Enterprise RAC - Area 1 drivers and test vehicles arrive; pre-deployment activities begin per checklist

7:45 AM	Enterprise RAC Area 2 drivers and test vehicles arrive; pre-deployment activities begin per checklist
8:00 AM	Enterprise RAC Area 3 drivers and test vehicles arrive; pre-deployment activities begin per checklist
8:15 AM	Enterprise RAC Area 4 drivers and test vehicles arrive; pre-deployment activities begin per checklist
8:30 AM	Enterprise RAC Area 5 drivers and test vehicles arrive; pre-deployment activities begin per checklist
8:45 AM	Enterprise RAC drivers given final route and safety instructions.
9:00 AM	Test vehicles depart Enterprise RAC staging facility at 1-minute interval
12:00 PM	Enterprise RAC drivers and vehicles begin returning to staging facility
1:00 PM	All drivers and vehicles complete post-deployment checklist

5.2.6. Consumer adoption

The focus of the consumer adoption (Early Adopters) track of the FOT was to test interest in and use of a real-time probe-based traffic system by the traveling public as well as to determine the perceived value of traffic probe information.

Much of the consumer adoption effort relied upon outreach and marketing of Mobile Millennium discussed separately in this final report.

From a logistics perspective, however, consumer adoption was dependent upon the Mobile Millennium website[3] being fully functional from the date of the launch on November 10, 2008 until completion of the free download period in November 2009.

Early Adopters needed to quickly navigate the website to register their mobile phone and to download the software. To facilitate a smooth and quick downloading process on the November 10 launch date, a "tech bar" comprised of a crew of ten Mobile Millennium research team members was assembled to provide personal registration and download assistance.

The "tech bar" concept was also employed at the Nokia and NAVTEQ booths during the ITS World Congress as well as IEEE and ASSHTO conferences.

(a) IEEE HSCC/RTAS Conference, San Francisco

Poster Abstract: Mobile Millennium-Traffic Monitoring with GPS Phones
Date: April 13, 2009
Time: 6-8:30 pm
Location: 55 Parc Hotel, Room: Cyril Magnin III

Registration info: Linda Buss 715-235-0487 lindabuss@mac.com
½ day $135 / full day $225 tutorials

[3] http://traffic.berkeley.edu

student member = $400
student non-member = $500
early registration deadline – 3/31/09

Poster: Sarah/AlfredDeliver draft to SA by 4/3

Dan's NSF/Washington Poster template:	Deliver to SA by 3/27
Addition of arterials content Ryan/Aude:	Deliver to SA by 3/31
New poster w/arterial Sarah/Alfred:	Deliver to SA by 4/3
Review/revisions Sarah/Alfred:	Deliver to SA by 4/7
Poster printing/mounting:	Delivery by 4/10
Easel	

Demo: [Repeat of CITRIS]

Touch screen delivery–JED	Delivery to CCIT by 4/6
Set-up – DANIEL/SANEESH	Delivery by 4/10
Touch screen	
Touch screen stand	
Laptop + power cord + cables + mouse	
Extension cords/power strip	
Duct/gaffers tape	
Padding/packaging	

Site logistics – special set arrangements-DAN	Deliver to CPS by 3/27
Table	
Electric power	
Wireless	

Minivan rental

(b) AASHTO

Dates: Thursday 10/22- Monday, 10/26/09
Palm Springs, CA

Hotel:
Desert Springs, A JW Marriott Resort & Spa
74855 Country Club Drive
Palm Desert, CA 92 10260

Offering:

- Mobile Millennium real-time traffic application with new traffic alerts
  - "Simulated" alerts to go off every 30 minutes during show
  - Participants will have the ability to simulate their own alerts acting as a TMC operator from the Mobile Millennium booth
- Participants will also be able to look at real-time traffic in their home locale

Objectives:

- 200+representatives having downloaded the application
- 100 “loaner phones” out for the conference to over 75% of the states
- Special recognition from Randy as to success/influence of MM technology
- Demonstrate CA’s national leadership in ITS

Mobile Millennium Application Download, Training and Alert Creation Opportunities

- Pre-Show
    - On “Mobile Millennium at AASHTO” website as of October 1
- At Show
    - At the Mobile Millennium Hospitality Suite at the Desert Springs Hotel, Thursday, 10/22, 11-5
    - At the Mobile Millennium Trade Fair Booth
        - Friday, 10/23 9:30 to 4:30
        - Saturday, 10/24, 9:30 to 3:30

Communications to registrants in advance of the show and at show

- Pre-Show
    - Invitation letter from Randy explaining program (9/16)
    - Email announcement to all registrants that MM website is open (10/1)
    - Email reminder to all registrants (1 week prior to show)
- At Show
    - Randy will announce from podium in his welcome speech
    - Hotel Lobby will have Poster
    - Hospitality suite
    - Trade show booth

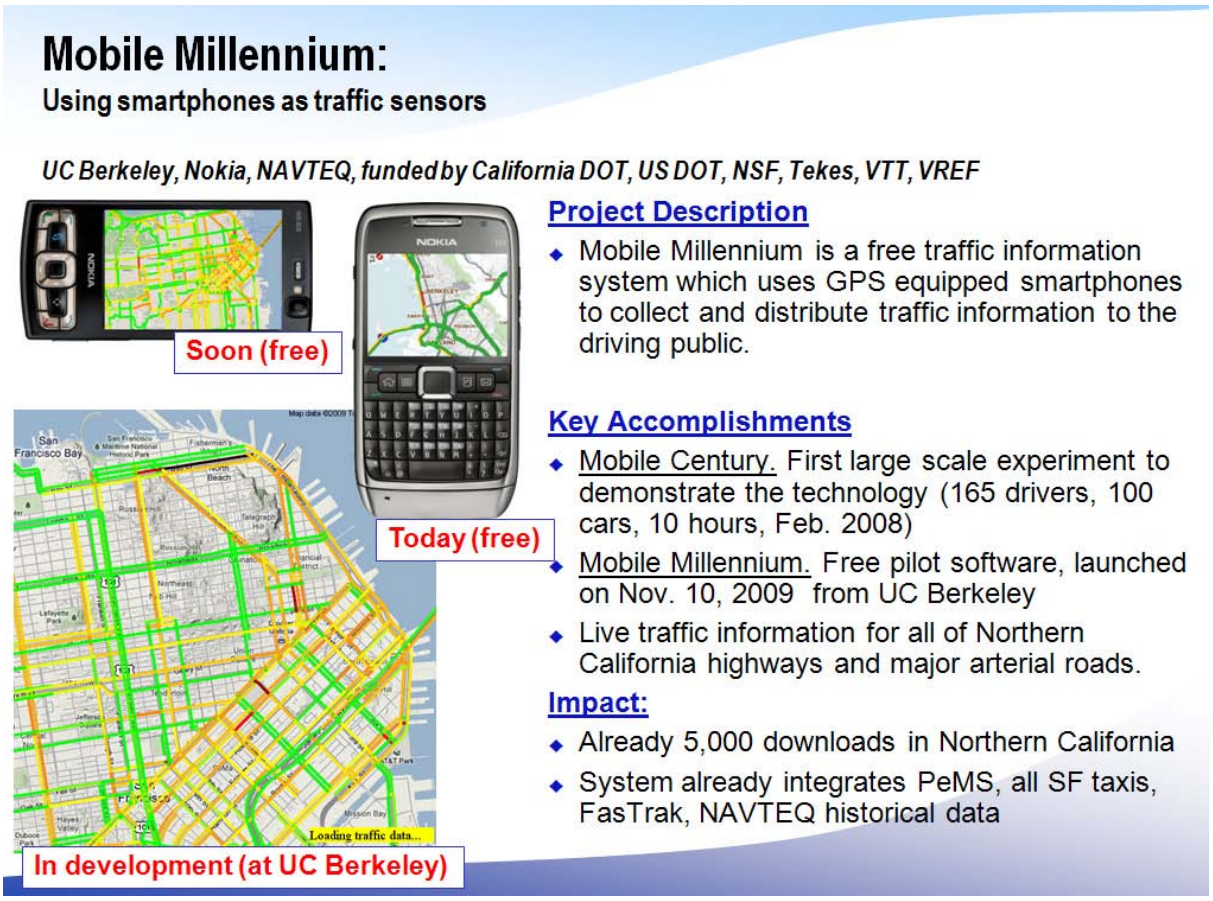


Figure 2.15. The Mobile Millennium poster presented next to the live demonstration in the Mobile Millennium booth.

Deliverables by Partner:

- **NAVTEQ**
  - Overall Project management
  - Secure booth and hospitality suite
  - Oversee Booth and hospitality suite theme (collateral, posters, technology), coordinating closely with NOKIA
  - Coordinate necessary staff for booth and suite
  - Coordinate communication with AASHTO
  - Coordinate communication with Caltrans
  - Coordinate potential sponsorship of cocktail reception (review pricing, availability, options)
  - Manage design and production of Mobile Millennium keepsake (do we want to do this?)
- **NOKIA**
  - Revise application to include alerts
  - Add on-site ability to add alerts

    - Manage technology for booth
    - Work with NAVTEQ on booth theme, wording, signage , technology
    - Contribute NOKIA collateral and posters
    - Co-sponsorship of potential reception, if available (?)
- **CCIT**
    - Create new landing page to link with MM download site
    - Staff email response center
    - Manage follow-up emails to come from ccit address
    - CCIT presence at booth (materials)
    - Coordinate this program with other MM appearances at AASHTO (IMS?)
- **CALTRANS**
    - Approve letter
    - Recognition in Randy's speech
    - Use of phones by all Caltrans registrants
    - Confirm Caltrans interest in a presence at booth (?)
    - Advocacy among other registrants

Schedule:

By 9/4/09

- Confirmation from CCIT of their role and contact emails
- Concept of operations complete
- Booth and suite confirmation from AASHTO

By 9/11/09

- Approval for Randy's invite letter
- Inclusion on AASHTO Agenda
- Determine participation, if any, in IMS (with CCIT)
- Determine if we want to create a "take-away" keepsake
- Reservations made for key staff

By 9/18/09

- Review Design of New "cover" web page describing AASHTO event
- Review New application download page and draft script for downloading application
- Test ccit email address
- Randy's Invite Letter sent out to all registrants

By 9/25/09

- Communication/Outreach/application to initial set of registrants who respond to invitation
- Staffing for booth confirmed
- Process developed to ensure that no one needs to wait at the booth for download
- BOOTH THEME finalized
- Signage/posters for booth confirmed
- Collateral confirmed

- Approval of script for email invitation announcing website open

On October 1

- Updated MM application sent to CCIT
- Mobile Millennium at AASHTO website launch and ready for download
- CCIT staff ready to receive calls/emails
- Email sent to all registrants inviting them to download app

By 10/9/09

- Continued communication/Active outreach to key CEOs who respond to invitation
- Goal to have _% downloaded prior to event start
- Approval of script for reminder email

By 10/16/09

- Reminder email sent to all registrants
- List of Participating CEOs compiled
- Collateral/posters mailed out for set-up
- Positive quote from CEO on his/her use of application
- Inclusion in Randy's speech
- Vehicle rental reservation completed w/Enterprise #67CHY9 [Scott Heath]
- Marriott accommodations confirmed for State rate attendees (Daniel 84060305/Samitha 84060537) [Randy Woolley]
- Credentials confirmed for trade fair only and poster session only attendees [Mandy Chu]
- Electric power order submitted and confirmed for booth #518 [Nate Jordan/Brudvik]
- Internet connection confirmed for booth #518 [Dirk Spaulding]
- Banner p/u Dean's Signs {Zaz / Jed BluCard]
- Final brochure approval / production] [Ann Guy]
- All posters in press [Jay Sullivan]

MON 10/19 -- Prep

- Brief attendees on travel and logistics [-West & Ritter / free agents]
- Poster p/u RFS
- Posters mounted
- Load repeating slideshow on laptop (WoCo slide presentation)
- Assemble equipment and materials for loading

TUE 10/20 -- Prep & Loading

- Vehicle p/u @ 3 pm
- Load equipment & materials [See checklist 10/21]
- Issue Fleet Services card & Risk Mngt accident packet
- Issue C scratch off permit
- Brochure p/u from printer

October 21

- Booth set-up
- Hospitality suite set-up
- Staff arrival
- Dry-runs

WED 10/21 -- Travel Day

- Transport vehicles depart for Palm Desert [Steve + Daniel & Ali]
- Visualizer (plasma screen) + touch screen + cables
- 2 laptops/mice/cables
- Second plasma screen + cables + speakers (for Rose @ CT)
- 4 easels
- 5 mounted posters (HOV/ATIS/CMS/(2)MM
- Extension cords/gaffers tape/table covers/clips/furniture pads/duct tape/tool kit
- Printed materials (brochure/poster handouts/IQ Magazine reprints}
- Banners (CCIT lrg/CCIT sm/CT sm/Explore MM/partners med)
- SWAG [canvas bags / ear buds 150 ea.]

October 22

- Staff hospitality suite

THU 10/22 – Booth Set-up, 8a- 5p only
Booth credentials – Steve, Daniel, Samitha

- CCIT booth (#518) set-up (10' x 10') table + 2 chairs
- Check install of 120V duplex outlet for electric power
- Check install of internet connection
- Install visualizer / touch screen / laptops (test all connections)
- Hang banners
- Place printed materials
- Store poster session materials in booth

FRI 10/23 – Trade Fair 9a-6:30p
Booth credentials – Daniel and Samitha

- MM visualizer demonstration (repeated throughout day)
- Assist with new traffic pilot download as needed
- Meet and greet delegates – provide CCIT brochure
- Staffing: 2 persons / rotate break every 2 hours
- SWAG distributed only to those who provide their business contact information (canvas bags / ear buds)

October 23-24

- Staff booth

SAT 10/24 – Trade Fair & Poster Session

9:00a-12:30p Trade Fair

Booth credentials – Daniel and Samitha

- Same as 10/23 + break down and load vehicle
- Staffing: 2 persons / rotate break every 2 hours

9:00a-4:00p Mobility Showcase [outdoor tent]

- Banner + MM poster
- Staffing: not required

2:00p-4:00p Poster Session [MM/ATIS/CMS/HOV]

- Poster session credentials – Manju, Samitha (Ali 2-day registration)
- Set-up easels/mount posters/place fact sheets
- Staffing: 3 persons / 1 per poster

4:30p Load all equipment and materials for return

October 25

- Tear down and ship home

SUN 10/25

- Depart for Bay Area
- MM taken by Caltrans – Rebecca Boyer for mounting @ CTHQ

MON 10/26

- Unload vehicle (use C scratch-off permit to park)
- Refuel vehicle (fill tank to check-out level only)
- Return vehicle to Enterprise Office @ University/Oxford
- Leave all receipts and paperwork in Steve's in-box

CONTACT INFORMATION

Steve Andrews CCIT-AASHTO logistics	510-501-7919 (cell)
Scott Heath, Enterprise Rent-A-Car #67CHY9	510-705-8989
Nate Jordan, Brudvik Electric – booth power	760-320-4429
Greg Larson, Poster Session Coordinator	916-217-3946 (cell)
Kevin Hanley, Caltrans AASHTO Coordinator	916-716-9087 (cell)
Dirk Spaulding, Caltrans internet connectivity	909-383-7995
Randy Woolley, Caltrans IMS2 Tent	949-756-4930
Rose Melgoza, Caltrans, Booth 520 w/plasma + speakers	909-383-6477
Tom West	510-289-7661

5.2.7. System evaluation

DOT evaluation

In order to evaluate the quality of the Mobile Millennium arterial traffic estimation model, two

field tests were completed in the East Bay and San Francisco April and May of 2010. This section describes the details of the routes driven, the methods for collecting probe vehicle data and validation data, and the logistics protocol for running both field tests.

Two tests were performed. The primary focus of the first was San Pablo Avenue in Berkeley, Albany, and El Cerrito. The primary focus of the second was Van Ness Avenue near downtown San Francisco, with a secondary emphasis on estimating traffic conditions on a small network of streets around Van Ness.

Figure 2.16 shows a map of the San Pablo Avenue route driven. The test involved 20 drivers repeatedly looping north and south along San Pablo. Each direction is approximately 2.3 miles of road. The Mobile Millennium arterial model estimated segment travel times between the bluetooth readers placed at the locations of the pins on the map. To see this route on an interactive Google Map, visit:

`http://maps.google.com/maps/ms?ie = UTF8&hl = en&msa = 0&msid=10140966`

`1672838926341.00047dc718529cdee794e&z=13`

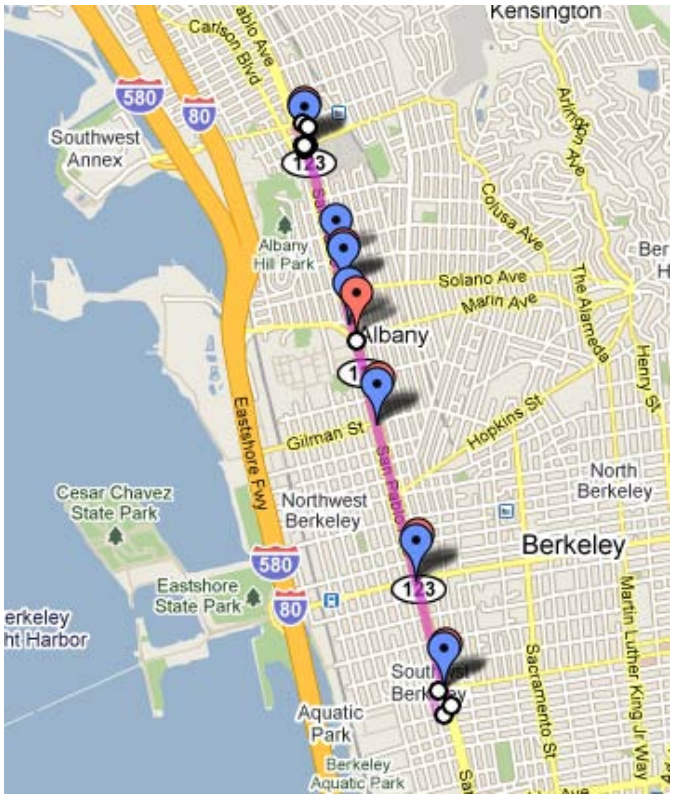


Figure 2.16. Map of the San Pablo Avenue evaluation route.

Figure 2.17 shows a map of the San Francisco route driven. There were 5 different routes, each with a different color. The primary focus was on Van Ness Avenue, in both directions. The

secondary focus was studying the travel patterns in this small network. Franklin and Gough are known to local San Francisco drivers as good alternatives to Van Ness and the research team wanted to study this phenomenon as well as understand the dynamics of traffic over a network (including one-way and two-way streets). The Mobile Millennium arterial model estimated segment travel times between the bluetooth reader locations indicated by pins on the map. Specifically, the streets to be modeled included:

- Van Ness, northbound and southbound
- Franklin, northbound
- Gough, southbound
- Hyde, southbound
- Leavenworth, northbound
- California, eastbound
- Pine, westbound
- Turk, westbound
- Golden Gate, eastbound

To see this route on an interactive Google Map, visit:

`http://maps.google.com/maps/ms?ie=UTF8&hl=en&msa=0&msid=10140966`

`1672838926341.00047dc673c0b1f4c9716&z=15`

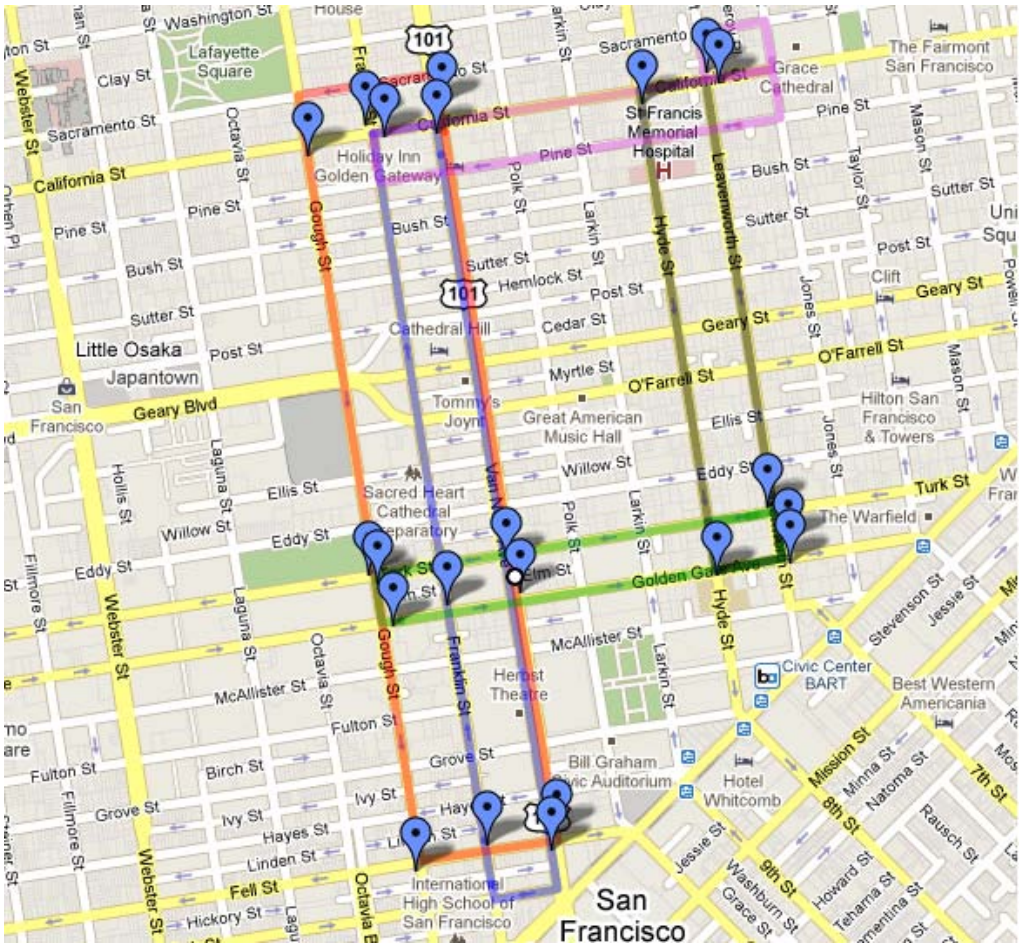


Figure 2.17. Figure 4.19: Map of the five downtown San Francisco evaluation routes.

- **Times/Durations:**

San Pablo Avenue: A weekday test performed during evening commute hours from 3:30pm to 6:30pm.

Van Ness and SF Network: A weekday test performed during evening commute hours from 3:30pm to 6:30pm.

- **Data Collection**

Probe Vehicle Data: Each test drivers was equipped with a GPS device recording the vehicle location every few seconds. This was used to generate sample probe data (either VTL-based or time-sampling based) used by the model. The GPS device was also bluetooth enabled allowing the team to test the bluetooth readers.

Bluetooth Data: Bluetooth readers were placed along the routes. The intent of collecting this data was to ground truth travel time measurements for validating the model results.

Part II

Mobile Millennium System

6. Core Systems

6.1. System Overview

A schematic of the overall Mobile Millennium system is shown below in Figure 1.1.

The overall system is a combination of phones, cellular networks and back end software. As the diagram shows, processing was split between Nokia and UC Berkeley. A cell phone user would download a client from the Berkeley web site. This client would connect over the cellular network to servers at Nokia. Nokia would download VTLs or virtual trip lines to the phone. The virtual trip lines determined at which points the cell phone client would measure velocity and location. As a phone crossed a trip line this information was sent back to the Nokia servers where it was cleaned of all user identifiable information

The anonymous information was then sent to UC Berkeley where it was cleaned, filtered, associated with an actual roadway in the Navstreets (Navteq) database, fused with other data sources and passed to algorithms that used the data to estimate the speed of traffic on Bay Area roads. These speed estimates were sent back to the Nokia servers where they were used to construct traffic maps where the color of a road corresponded to the average speed of vehicles on the road. This process occurred in real time every week day for over a year.

The estimation algorithms and the virtual trip line technology are discussed later in this document. The details of the Navteq processing steps are proprietary and are not further discussed in this document. The remainder of the chapter will discuss the Mobile Millennium system as it was developed and run at Berkeley. The majority of the data processing and complex algorithm execution occurred on the Berkeley servers.

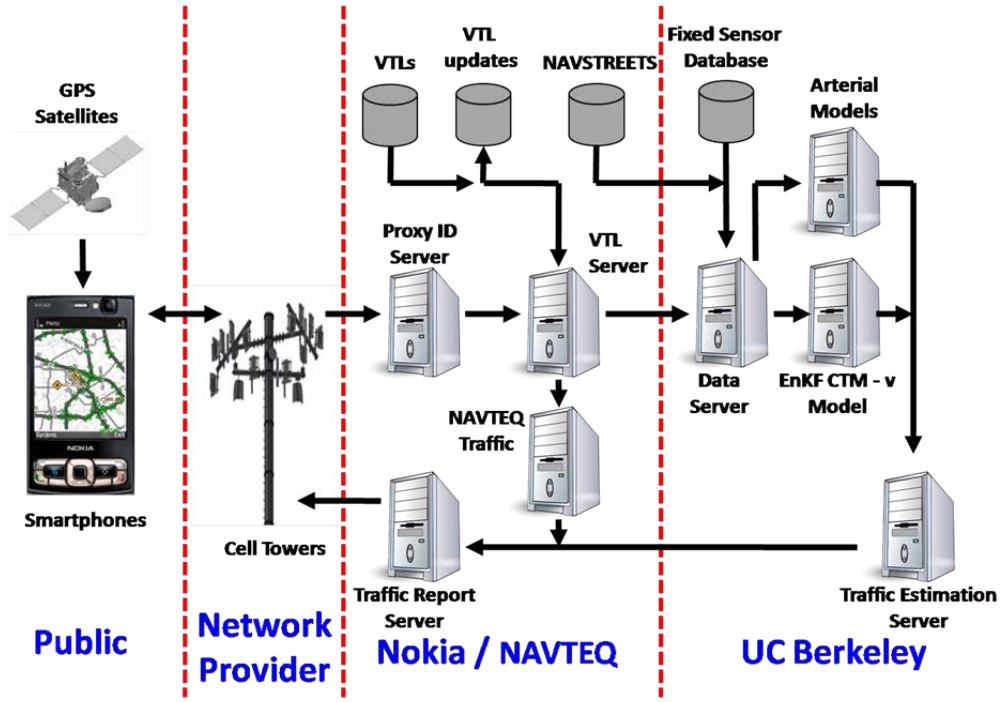


Figure 1.1. Mobile Millennium System Architecture (Nokia and Berkeley)

The Mobile Millennium system at UC Berkeley is a combination of software, hardware, procedures and personnel all focused on providing high volume, high reliability, real-time data processing. It was developed and used by over 50 students, interns, engineers and scientists. It is composed of roughly half a million lines of code, over 100 database tables, multiple hardware servers and several development environments.

As data flows through the system, numerous data feeds are filtered, the refined output fed to state of the art estimation and fusion algorithms and the resulting information visualized and presented for data exploration. Figure 1.2 is an early data flow diagram for the system. Its components will be discussed in more detail later in this section. The headings across the top are major processing steps, the small rectangles on either side are data sources and sinks, the circles within are individual processes and the rectangles at the bottom are cross cutting functionalities.

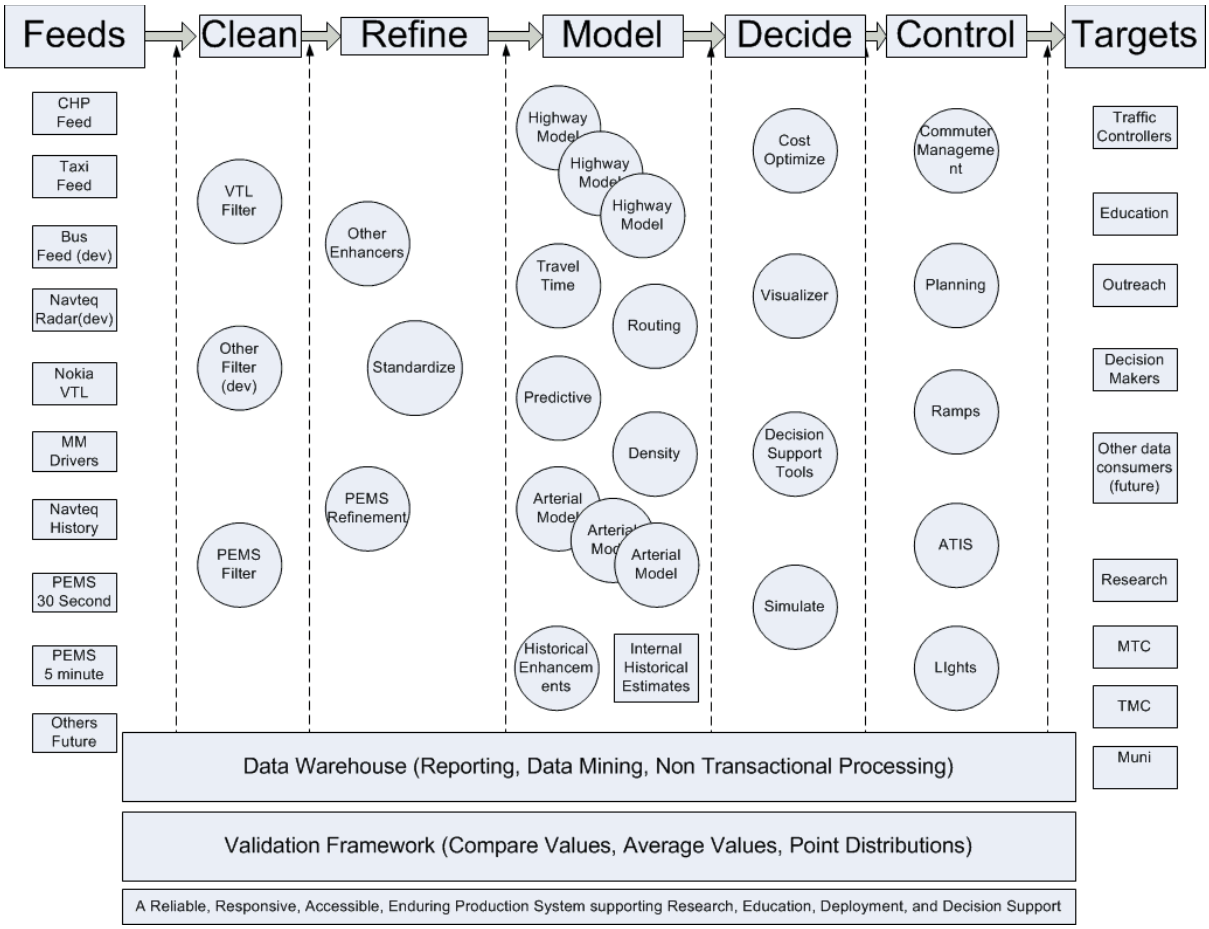


Figure 1.2. Early Mobile Millennium data flow diagram

There are many large commercial companies that provide this type of production data processing. The Mobile Millennium system and support structure differed in several significant ways from commercial environments. These differences include:

- Mission and support structure provided by the parent organization
- Experience and background of the personnel developing the system
- Funding levels and paradigms
- Evolution rate of the system components

These combined to create a challenging, interesting and instructive management problem. As one of the goals of this report is to assist others who wish to carry this research forward we will discuss each of these challenges and how we for the most part successfully responded to them.

Mission and support structure provided by the parent organization

The Mobile Millennium system was developed at the University of California at Berkeley, a

large educational institution not traditionally known for its production data processing capabilities. Within the University the Mobile Millennium system was directly built by students and staff associated with The California Center for Innovative Transportation (CCIT). CCIT has since merged with the Partners for Advanced Transportation Technology (PATH). The faculty and students were also principally associated with the Civil Engineering department at the University. Later personnel from the Computer Science and Electrical Engineering departments joined the project.

CCIT had experience building smaller software systems including infrastructure for managing Changeable Message signs (CMS). However no project with the scope and requirements of Mobile Millennium had been attempted there. Within the University even the scope of the CMS sign effort was considered unusual. As such little resource or experiential support could be provided by the University. This included a lack of organizationally supported computing resources. All hardware had to be purchased for the project and, at least during the initial phases of the project, stored on site.

These challenges were alleviated in several ways. The principle investigator was given leeway in building a new organization and in repurposing existing personnel. Equally important, existing senior management did not attempt to manage the technical aspects of the project. Finally exceptional steps were taken by management to obtain funding from Caltrans for the hiring of external consultants to manage, structure and organize the effort.

Lessons Learned – Flexible Management and aggressive acquisition of proper resources are essential. If we had it to do again we would have provided more and additional funding earlier in the process for professional services.

6.1.1. Experience and background of the personnel developing the system

The Mobile Millennium system required the standard set of expertise used in building and maintaining systems. These areas of expertise included:

- Software Engineering Management
- Software Architectural Design
- Software Engineering
- Database Management
- Quality Control
- Version and Configuration Management
- System Administration Support
- Project Management
- Technical Writing

At the start of the project no professional software management was available and there was one competent software engineer who doubled as the system administrator and architect. There were many civil engineering students available. These included master’s students, PhD students and

interns. These students were from multiple countries and English was often not their first language. As was to be expected of students none had experience with professional software development and were unfamiliar with basic concepts of shared development, source control, security, reliability, testability, and maintainability. Most students had some experience with MATLAB but had not worked with a database management system. They were accustomed to using free tools and thus only had experience with open source products.

The challenges to building a production system with students went beyond simple lack of experience. As students, their principal motivation was the achievement of their degree and professional acknowledgement in their field. Proving that a technology can be commercialized, communicating problems/failures and building a production like system were at odds with their requirement to mathematically prototype many different "crazy ideas" and to publish papers only on the ideas that bore fruit. Learning software engineering techniques, standardizing on software tools/approaches and writing maintainable software was understandably low on their list of priorities.

The early realization that professional assistance was required saved the project. A small group of professional software developers were hired as consultants. A professional developer from one of our partners was also hired and moved to Berkeley from Finland. Equally fortunate and essential, the students at Berkeley are highly intelligent, motivated, emotionally well developed and willing to completely commit, body and soul to the success of a project. This permitted a small group of professional developers to educate and guide them in the building of this system. Several of the students turned out to be excellent software engineers in the making.

The professional staff consisted of

- Software Development Manager with 20+ years of experience. He also functioned as the project manager and shared the role of software architect with one of the engineers
- Three software engineers with beginning to mid level experience
- One developer who had experience in Geographic Information Systems (GIS). This was important as traffic must be mapped to physical locations.
- For a short time an architect/database administrator

The project was continually challenged by the lack of a professional database administrator and a lack of expertise in web development.

Lessons Learned – Bring in professional support as early as possible. We should have provided more focus on initial training and proper motivation of project participants.

6.1.2. Funding levels and paradigms

Both the University and Caltrans Department of Research and Innovation (DRI) were accustomed to funding and supporting research. Mobile Millennium was a combination of research and preliminary implementation. In many ways Mobile Millenniums goal was to prove that implementation of traffic estimation was possible at scale in real-time. This was a new goal for research organizations.

Neither the DRI funding mechanisms nor the University support structures were accustomed to building and maintaining professional software. There was organizational and sponsor pressure to not explicitly fund ongoing core software systems or commercial software as these were not considered important to research. In many ways this is understandable given previous research paradigms. Normally systems support structures were built up and torn down for each project. This was acceptable for analysis of small amounts of data. However it is excessively expensive and logistically prohibitive given the massive amounts of data now being generated by ubiquitous sensors. Data processing and exploration have become computationally complex and are now a base requirement for accelerated research and deployment.

Most companies who perform data analysis have a core group of systems and software personnel that are shared among projects. While it is true that CCIT had a core group of staff engineers these were not trained software engineering and database personnel. Even more importantly the fundamental core hardware systems and core software cross cutting layers (discussed later) were not in place. This was exacerbated by the lack of experience within the research community with the sharing of systems infrastructure.

It is interesting to note that ongoing systems are more readily supported at Caltrans in the Operations group (the PEMS system for example) where the goal of research that leads quickly to implementation is forefront. However the fundamental driver for more computing infrastructure is not the need for rapid implementation, it is the advent of large amounts of data requiring processing in real-time that cannot be managed with traditional tools and processes.

Another area of challenge was the hiring process, including contractors, consultants and employees. In order to hire contractors there is a minimum of a 6 month cycle for contract modifications, bids, purchasing, etc. This process is considered extremely slow in the world of software engineering. The hiring of employees is also problematic from several perspectives. University salaries are not competitive in the Silicon Valley, job security is project based and thus susceptible to yearly layoff notices and there is little funding allocated for professional development. Additionally, the hiring process at the University is based on proving funds will be available for at least a year. Thus it is difficult to build a base development organization.

Lessons Learned – Acknowledge big data has arrived. Build the appropriate system support structures. Understand the research is not production and educate all sponsors about the new requirements for success. Start hiring early.

Evolution Rate of the system components and personnel

Another challenging requirement was the high degree of change of system components and personnel. One of the pleasures of working on Mobile Millennium was the continual goal of improvement and the notion that there were no failures, only opportunities to push the limit of what is possible. Mobile Millennium was a combination of research and production. These two paradigms do not sit easily together. However, in this case, the dynamic tension between the two resulted in both better research and better systems. Research required quick development of new mathematical models, filters, and analysis methods. These mathematical formulae needed to be coded into software, integrated into a production system and quickly tested. Production required

tested software, solid configuration management and predictable update schedules.

Several management decisions were made in order to meld these two opposing requirements:

- The professionals controlled the system. The principle investigator made it clear to his students that they were on a team and that they were to do what the professionals told them to do. The importance of this cannot be overstated. The willingness of the principle investigator to cede authority on these topics to the development professionals was essential.
- The professionals were required to show patience and a relaxing of standards to enable a meeting in the middle on the development process. Working with students who did not understand software development and even at times felt it was beneath them as civil engineers was stressful and trying. However the rewards were clear. Many of the students developed into reasonable developers, the system functioned well most of the time and there was a significant feeling of accomplishment by the professionals and the students on the project
- The professionals would build an architecture including a core set of classes and tools that the students would be required to use. This took time and was a struggle but was worth it.
- Our system would be highly modular and relatively easy to debug. Thus we traded performance for ease of use and understandability. This was a core design of the system that we explicitly discussed and implemented. The basic concept was that there would be no interprocess communication; all information would be transferred through database tables. This resulted in simple modularity but reduced performance. One of our mistakes was not hiring a database administrator.

Lessons Learned – Ensure the existence of strong crosscutting layers, availability of experts in all relevant software fields, leadership by experienced management and an overall desire by all the project members to work in a learning/training environment.

6.1.3. System Description – languages, Infrastructure, Etc

As previously mentioned we had three significant constraints in place when we chose our software and our hardware.

- Fiscal Constraints – We did not have sufficient funds to pay for commercial software or anything but fairly generic hardware.
- The students were determined to use open source software and operating systems
- Our students had no experience with integrated development environments, management tools or databases. This limited us to simpler tools.

Within these constraints we choose the following infrastructure components for our system

- Programming Languages
  (a) Java - Java was an open source, tested language. We felt that there were many open source development tools that worked with Java and that it would be

accepted by the students. It was also object oriented, a requirement for our work.

(b) Scala - Later we also began development in the Scala language. It provided functional characteristics that were becoming more important in our high performance computing environment. We also felt that we should be teaching the most recent programming techniques to our students.

(c) Matlab – No production part of the system was written in Matlab, however many students used it for data analysis. It has superior rapid visualization tools and we effectively had a site license. We built Matlab consumable libraries that permitted Matlab script to access our databases safely.

(d) We also used various scripting languages but only for system management and not for core software

- Database Management Software

  (a) Postgis - We needed an open source database that supported GIS processing. We had significant requirements to map points to locations and to determine points within bounding boxes.

  (b) PGAdmin – A useful GUI that eases Postgis database management

- Integrated Development Environments (IDEs)

  (a) Netbeans – Netbeans is an open source IDE and fairly simple to use. It worked reasonably well.

  (b) IntelliJ – We started out with this IDE (even paid for some licenses). However it was too complex for our students to understand. It was well liked by the professionals.

- Operating Systems

  (a) Linux – For deployment. Works well and is open source.

  (b) Windows - For development and for documentation purposes

- High performance computing

  (a) We used several high performance computing environments. Please refer to the chapter on HPC for more details.

- Data Warehouse software

  (a) Pentaho – Open source. We used various ETL (Extract, Transform and Load) libraries and various visualization tools. Ultimately this proved too complex for our organization to fully implement and support.

- Web Aps

  (a) Tom Cat Server

  (b) Simple Java Script

  (c) Basic HTML

- Mapping Software and Data

  (a) Navteq Maps – Navteq provided us with their street maps. These worked very well. They contained detailed information on both road connectivity and roadway characteristics (free flow speed for example) for all the roads in the bay area. This information was essential to our models and without this the project would likely not have succeeded

  (b) Open Layers on the web client

- Other Support Software

  (a) Configuration Management – SVN

  (b) Unit and System Tests - We used tools to run unit and system tests nightly

(c) Tunneling tools - Various

We would have liked to use more sophisticated code coverage, profiling and testing tools but were unable to bring the organization to a sufficient level of knowledge and maturity to do so.

As previously stated, our architecture was based on all data being passed between processes using database tables. While slow at times it permitted intermediate results to be debugged and was a quickly understandable architecture for our students. We also asked the professional developers to build robust cross cutting layers. This strategy is one of the reasons we were able to meet our deliverables. We abstracted out as many functions as we could so that the student developers were only responsible for algorithms.

6.1.4. System Architecture – Data Flow and Cross Cutting Layers

Figure 1.2, above is from the early stages of Mobile Millennium. All the basic processing steps are in place but the cross cutting layers are in their early stages. Figure 1.3 shows the system during its most active quarters. Figure 1.4 shows how the system blossomed to include cross cutting layers for parallelization (high performance computing), network management (road descriptions) as well as applications that helped to estimate information related to waterways, air quality and earthquakes. By the end of the project the system was being used by multiple groups outside of the core Mobile Millennium project. We viewed this as a mark of success for both the system itself and the organizations associated with the projects.

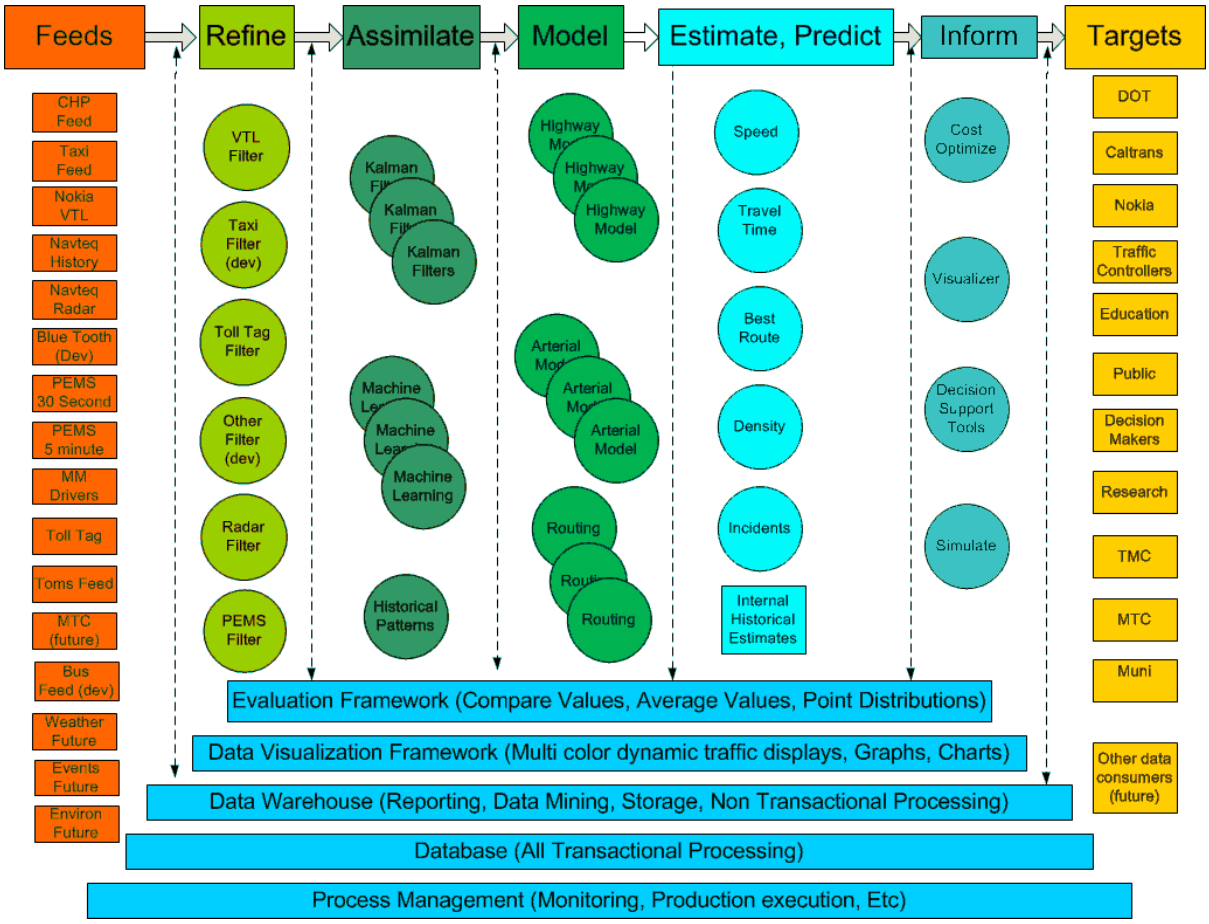


Figure 1.3. Mid-project Mobile Millennium based traffic data collection and processing system

Following is a description of each of the system's components and their contribution to research, analyses and deployment. We first discuss the vertical columns which represent the flow of data from raw input feeds to useful information for traffic data consumers.

Feeds – Traffic data feeds come from many sources. The system receives software from the California Highway Patrol (incidents), Nokia/NAVTEQ/Traffic.com (probe data and radar data), toll tag readers, inductive loops (PEMS), SF Taxi data and other feeds as needed. The architecture for the system is generic and we can integrate new feeds in a relatively quick and easy manner. Raw data from feeds is time stamped, geo coded and saved in raw data tables in our operational database.

Refinement – All raw data feeds require filtering for obvious errors caused by faulty measurement devices and transmission errors. Data also needs refinement for more subtle errors caused by slightly malfunctioning devices or general calibration or device attribute errors. Previous work has shown that this refinement step is absolutely essential to the generation of good traffic information.

Assimilation and Modeling – Raw traffic data represents point in time and point in space measurements from a representative sample of road locations. From this sample the overall conditions of traffic at any point in the road network is desired. Models are used to take measurements of initial roadway conditions and produce estimates and predictions of traffic information. The algorithms for combining actual physical measurements with the estimates from the models are called data assimilation algorithms.

Estimate and Predict – Estimation is the process of determining probable values for traffic metrics as they exist now. Prediction determines probable values for traffic metrics in the future. The system used both flow and machine learning models to perform estimation and prediction of speed, travel time and best route metrics. Multiple versions of each model type were often in use simultaneously.

Inform – Traffic information is not an end in itself, it must be used to help improve traffic conditions through both effective management of current roadways and in the development of new roadway systems. The system currently has an excellent visualization tool used to show traffic data estimates and predictions. Several versions of the visualization tools exist. The version used by researchers permits data drill down into individual measurement points and roadway segments.

We now describe the horizontal cross cutting layers at the bottom of the diagram:

Process Management – All real time systems require monitoring. The system monitors up time, database items, response time, cpu/disk usage etc. If problems are seen notifications are sent and a real time system status management web page is updated. There are scripts for starting and stopping processes on development and production machines.

Database – The system contains operational/development databases and a data warehouse. The operational system is tuned for response time. Each night we would run an ETL (Extract, Transform and Load) to move data from our operational system to our data warehouse. We use PostgreSQL with GIS extensions as our database.

Data Visualization – The system has an entire subsystem and web based user interface dedicated to visualizing traffic data. There is both a user mode and an expert researcher mode. In the expert mode the system permits drill down into sensor and traffic data details. We can also visualize best travel routes and air quality near roadways.

Evaluation – It is crucial to be able to compare outputs from one group of (inputs, filters, models) with a different group. This is especially true if one group represents ground truth. The system has developed methods and interfaces for ensuring that output from models can be compared.

Additional cross cutting layers added after this diagram was completed include:

Network and Geocoding – We need to create many different networks. We refer to these as model graphs and they represent a named subset of the bay area network. These can be used as

part of estimation or evaluation processes. This layer provides tools and interfaces for creating and accessing these networks. It also includes numerous functions for fundamental location aware operations.

Parallelization – As our algorithms became more CPU intensive (principally for machine learning and statistically significant data assimilation) we needed to enable access to high performance computing. This layer defines the interfaces and methods for running jobs in parallel on computing clusters.

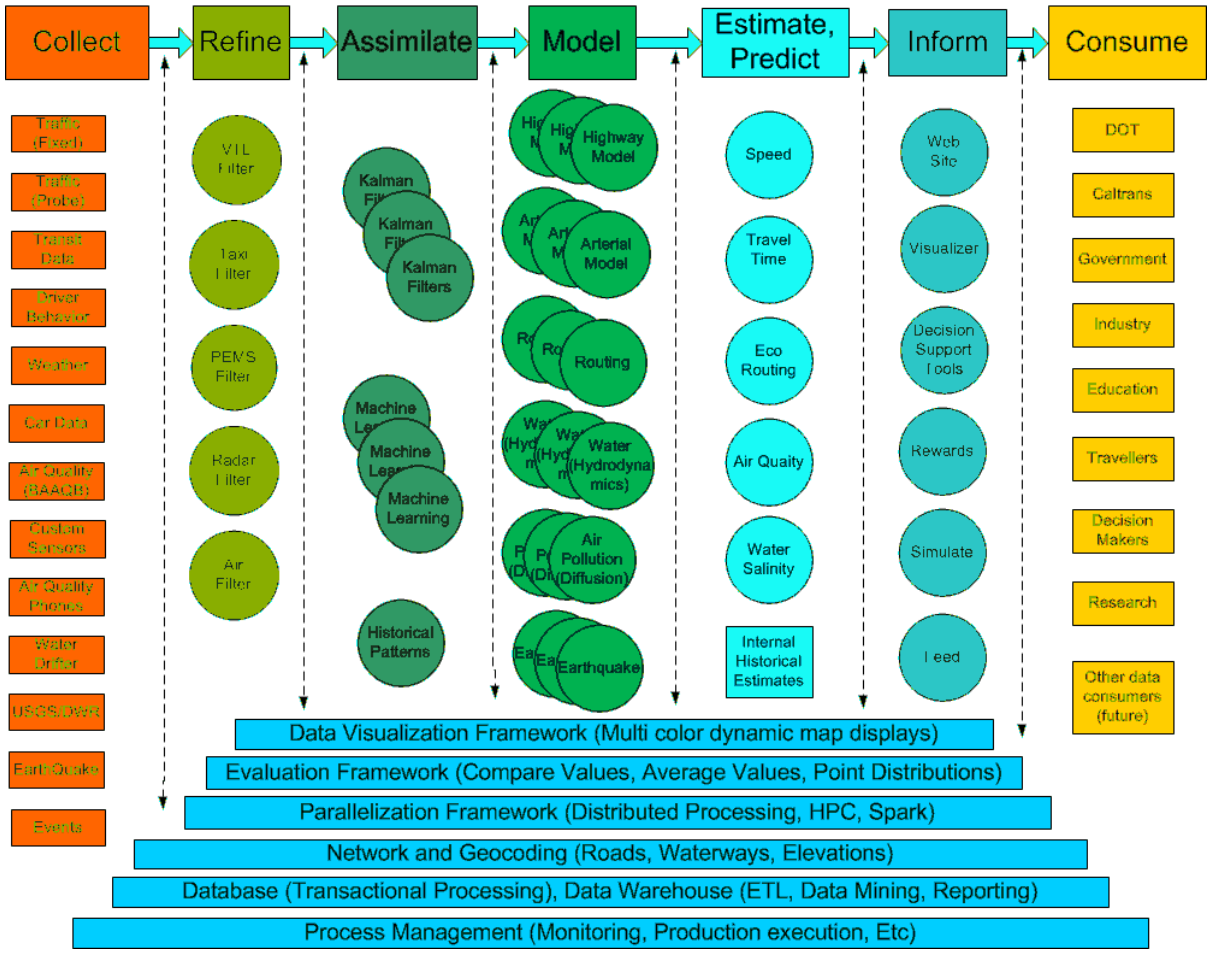


Figure 1.4. Final Mobile Millennium expanded data flow diagram

The next set of sections in this chapter will describe various aspects of the system in more detail. Research chapters will discuss refinement, assimilation, modeling and estimation algorithms.

The *Mobile Millennium* system is a platform for collecting, filtering, processing, analyzing and visualizing traffic data. The primary purpose of the system is to enable researchers to easily

access data and software tools for building new algorithms for estimating traffic conditions.

This chapter describes the basic architecture of the system and gives details about each of the primary components. The intended audience includes readers who want to gain a high-level understanding of the system as well as researchers who want to learn how to use the system to build new applications.

6.2. Important terms and definitions

This section presents a list of terms that are frequently used throughout the document along with a description of what they mean.

- **Network**

A network is a mathematical representation of the road and is defined as a directed graph with a set of nodes and a set of links. The definitions used in the *Mobile Millennium* system are standard in the transportation community.

(a) **Node**

The end point of one or more links, which acts as either the starting or ending point of a link. This corresponds to an intersection in the road network.

(b) **Link**

A one-way stretch of road between two nodes. A two-way street in the road network is represented by two one-way links in the graph.

- **Model**

In the *Mobile Millennium* system, the term “model” refers to a traffic estimation framework and algorithm. All of the algorithms used to produce traffic estimates have some underlying physical model of the dynamic nature of traffic and the term “model” is used to encompass the combination of that physical model with the estimation algorithm.

- **Navteq Map Data**

Navteq is a digital map provider. Their map data consists of many components, the most important of which for the *Mobile Millennium* system are the geometry of the road network (i.e. the latitude/longitude coordinates that describe each link of the road network) as well as various attributes associated with the road network (e.g. number of lanes, speed limit, etc.).

- **Model Graph**

The Model Graph is a construction of the *Mobile Millennium* system. It is an alternate representation of the road network (as opposed to the Navteq map) that is a more appropriate

level of abstraction for running traffic estimation algorithms. It contains only the subset of the road network that is desired for traffic estimation (it excludes residential streets) and it is a representation that makes communicating data across applications easier. The process of simplifying the road network involves combining links together when there is no intermediate intersection as well as better handling of intersections compared to the way they are represented in the Navteq map. Full details of the Model Graph generation are found in section 7.

- **Core**

The term core is used in two different ways in the *Mobile Millennium* system. The first way is to refer to the "core *Mobile Millennium* system", which refers to the parts of the system that are useful for all applications, including the Model Graph and functions to read/write from the database. The second reference to core refers to the **CORE** sub-module, which is generally written in all capital letters to distinguish it from the first reference. The CORE sub-module contains non-traffic-specific data types, helper utilities for reading/writing data to the database and monitoring functions.

- **NETCONFIG**

Generally written in all caps to highlight that it refers to the **NETCONFIG** sub-module. This sub-module contains the traffic-specific data types used in the system and utilities pertaining to manipulating data on the road network. It also contains the interface to the Model Graph.

- **Virtual Trip Line (VTL)**

A VTL is simply a line drawn on the map, meaning that it is completely defined by its start and end GPS coordinates. The purpose of a VTL is to detect when a vehicle crosses a specific point in the road. A mobile phone (or other client examining a GPS trace) detects when a VTL has been crossed and then records (and possibly sends) the crossing information (time, speed of crossing, heading, etc.).

6.3. Conventions used in this document

This section lists the conventions used in both the system and this document.

- A key word in **bold ALL CAPS** generally refers to a NetBeans project by that name. For example, **NETCONFIG** is used to refer to the NetBeans project by that name.
- Class, field and method names are in `typewriter` font.

6.4. System requirements

This section describes the requirements that needed to be met when the *Mobile Millennium* system was being designed.

- Provide a simple, intuitive graph representation of the road network.
- Collect data feeds from many traffic data sources.
  - (a) GPS-enabled cell phones via VTLs
  - (b) PeMS
  - (c) Radar
  - (d) Fleet probe data
- Provide interfaces for accessing filtered data with location information (i.e. what point on the Model Graph does the data belong to?).
- Provide interfaces for storing traffic estimates to a database.
- Provide output feeds of traffic estimates to third parties.
- Provide visualization and data analysis tools for traffic estimates.

6.5. System architecture overview

The *Mobile Millennium* system is composed of various components, or modules. Figure 5.1 depicts the modules of the system graphically, where the lines between modules represent a two way exchange of information.

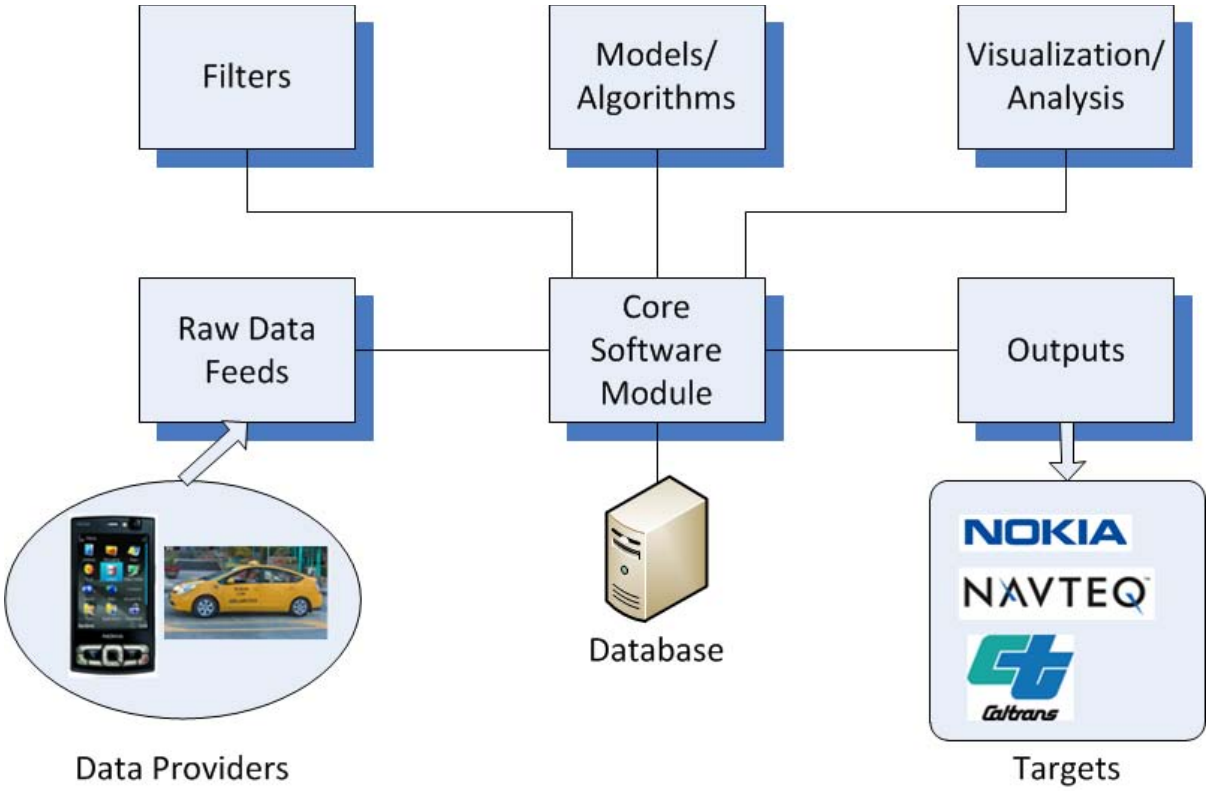


Figure 5.1. Overview of the *Mobile Millennium* system architecture.

Each of the modules is summarized with a brief description of its responsibilities and visibility to the other modules. The responsibilities of a module are the tasks that the module must execute correctly to fulfill its role in the system. The visibility of each module indicates which other modules can access various features of it. Section 6 goes into more detail about the specifics of each module.

6.5.1. Database

- **Responsibilities** Store raw data, filtered data, traffic estimates. Maintain indexes for fast access to data.
- **Visibility** Only the core system module interacts directly with the database. No other modules can see the database.

6.5.2. Core system

This module is broken up into two sub-modules:

CORE

- **Responsibilities** Provide read/write access to the database and provide generic (non-traffic-specific) data types. Also provides a monitoring interface for any module to report various kinds of information about its run-time performance such as the duration of an operation, the number of records processed, or whether an exception has occurred to cause the program to stop running. This information is stored in its own database and the monitoring front-end is responsible for executing alerts basedon certain criteria (e.g. program is taking too long to run, the feed from PeMS has gone down, etc.).
- **Visibility** Read/write access to the database is only available to the **NETCONFIG** sub-module. Generic data types are available to all modules. Monitoring functionality is visible to all other modules.

NETCONFIG

- **Responsibilities** Provide a graph representation of the road network, provide interfaces for accessing data on the graph, provide interfaces for writing traffic estimates to the database and provide traffic-specific data types.
- **Visibility** This sub-module is visible to all other modules.

Input Feeds

- **Responsibilities** Get data from providers and write it to the database (using the interfaces provided by **NETCONFIG**).
- **Visibility** This module is **not** visible to the rest of the system.

Filters

- **Responsibilities** Get raw data, filter it, and store the result in the database (both reading/writing uses the interfaces provided by **NETCONFIG**).
- **Visibility** This module is **not** visible to the rest of the system.

Models

There are several traffic models in the *Mobile Millennium* system (Highway, Arterial, Routing), which are each a sub-module of this module. Each traffic model has the same basic set of responsibilities and visibility.

- **Responsibilities** Get filtered data, produce traffic estimates and write them to the database (using the interfaces provided by **NETCONFIG**).
- **Visibility** This module is **not** visible to the rest of the system.

Output Feeds

- **Responsibilities** Get traffic estimates and send them to a third party.

- **Visibility** This module is **not** visible to the rest of the system.

Analysis

This module consists of two sub-modules:

Visualization

- **Responsibilities** Get raw data, filtered data, and traffic estimates and display them on a map.
- **Visibility** This sub-module is **not** visible to the rest of the system.

Model Evaluation and Raw Data Analysis

- **Responsibilities** First use case is to get data (raw, filtered or traffic estimates) and display informative summary statistics or plots of the data as a time series. Second use case is to get data (raw, filtered or traffic estimates) from two sources and compare them using standard statistical tests as well as display informative summary statistics and plots.
- **Visibility** This sub-module is **not** visible to the rest of the system.

6.6. System architecture detailed module descriptions

Each of the modules mentioned in the previous section is detailed here. The level of detail provided here is intended to give the reader everything they need to get started coding in the *Mobile Millennium* system. This section bridges the gap between the high-level architecture and the detailed code documentation (javadoc).

6.6.1. Database

The *Mobile Millennium* system uses a PostgreSQL [24] database with PostGIS [23] extensions for spatial indexing and queries. Both products are open source with a large group of users and good online documentation.

The database is organized into a number of schemas, essentially one per type of application in the system. The primary schemas of interest to any developer in the system are `dca` and `model_graph`. The `dca` schema contains all ofthe data provided to the system by NAVTEQ and is named due to NAVTEQ's convention of breaking the country into "DCA" regional

blocks. The `model_graph` schema contains all of the data needed to represent the Model Graph (see the Key Terms in section 2). Beyond these two primary schemas, there is generally one schema per application or data source. For instance, the `pems_feed` schema contains the raw data obtained from the PeMS system as well as sensor location information and a filtered version of the PeMS sensor data.

Each schema of the database has its own user profile which must be specified when running any Java program. Only programs run using the user profile of the given schema are allowed to make any changes to the data within that schema. For example, in order to be able to write filtered data to the `pems_feed` schema, the Java program that runs the filter must be run using the `pems_feed` user profile. All programs have read-access to any data from any other schema.

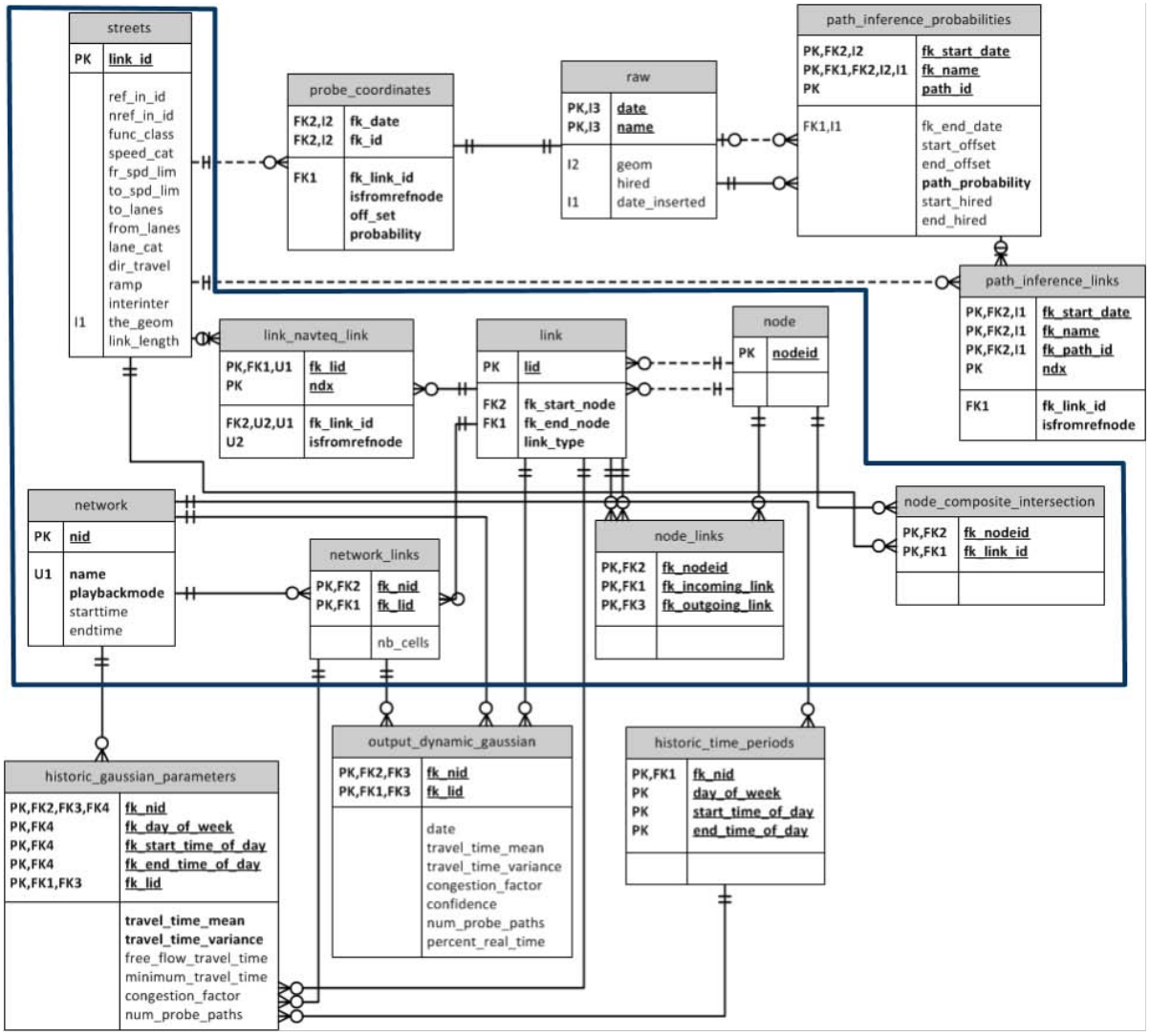


Figure 6.1. An illustration of the relationship between the core tables (those within the thick solid line) and a few auxiliary modules of the *Mobile Millennium* database. The symbols between tables represent the relationship between data structures (such as one-to-one, many-to-one, etc.).

Figure 6.1 displays the important tables of the database for the core infrastructure as well as a few example modules that use it (the full database structure is far too big to put in a single graphic). This figure emphasizes the design decision to have raw data feeds dependent upon the NAVTEQ map and the models dependent upon the Model Graph. As the system evolves, new data feeds and traffic estimation models follow that same pattern.

6.6.2. Core system

The code for the *Mobile Millennium* system is primarily in Java using NetBeans as the Integrated Development Environment (IDE). The system has started to incorporate Scala into the newer system modules. Scala is a relatively new programming language that compiles to Java bytecode, making it easy to incorporate into the existing Java code base. For more information on Scala, refer to [25].

The core system module is comprised of two sub-modules, which are each their own NetBeans projects: **CORE** and **NETCONFIG**. The basic division between the two sub-modules is that any types or functions that are traffic specific are placed in **NETCONFIG**, while the fundamental types and functions that are generic to any project go in CORE. The **CORE** project also contains a number of libraries that are used throughout the code. These include the PostgreSQL and PostGIS java libraries as well as a number of other third-party libraries used in specific contexts (such as math libraries, graph libraries, etc.).

CORE

There is only one package in the **CORE** project, named `core`. The primary classes in this package are `Database` and its sub-classes, `DatabaseReader` and `DatabaseWriter`. `Database` is responsible for all basic interactions with the PostgreSQL database, but it is an abstract class. Users are required to use on of the sub-classes, each of which only allows for reading or writing, respectively. The `DatabaseException` class is used for any error that occurs communicating with the database.

The other most commonly used class in the `core` package is `Monitor`, which provides logging, debugging, error display and monitoring/instrumentation for the entire *Mobile Millennium* code base. This class performs operations on a separate thread in the background so that other programs can take advantage of the functionality without worrying about the monitoring functions slowing down their process. The functions in the `Monitor` class can be used to write information to a dedicated monitoring database (which is separate from the database described in the previous sub-section). This monitoring database provides the back-end to our web-based monitoring system that indicates the current health of all of the programs currently running and some general performance characteristics (how fast, how much data, etc.).

The remainder of the classes in the `core` package are various utility types and functions. The following is a brief description of their organization and content.

- Geographic types
  - `Coordinate` - represents a "GPS point", which means a latitude/longitude coordinate on the Earth for some spatial referencing projection system, such as WGS 84
  - `GeoMultiLine` - represents a "line" on the Earth, which is represented by a sequence of `Coordinate` objects
- Time types
  - `Time` - represents a specific moment in time
  - `TimeInterval` - represents a block of time defined by a start `Time` object and an end `Time` object
  - `TimeOfDay` - represents a generic time of the day, but for no specific day
  - `HistoricTimeInterval` - represents a generic time interval of a generic week, i.e., Mondays from 9am to 10am
- Utilities
  - `Exceptions` - contains static methods for extracting useful information from any exception
  - `StringFormat` - a number of convenience functions for manipulating strings
  - `Tasks` - utilities for creating new threads and other task-related functions
  -

NETCONFIG

There are three packages in this project, named `netconfig`, `model_graph`, and `util`. The `netconfig` package is the primary package used by most application developers in the *Mobile Millennium* system. It provides access to the Model Graph (see the Key Terms in section 2) and traffic data (localized to the Model Graph). The `model_graph` package contains all of the code needed for building the Model Graph as well as the code that is used to create a new network. The `util` package provides miscellaneous utilities that developers have found useful in a number of contexts (such as printing to files in a structured manner, analyzing the topology of a network, etc.).

The primary classes in the `netconfig` package are `Network`, `DataType`, and `Datum`. The `Network` class provides a simple representation of a given area of the road network. To use the `Network` object, one needs to know the `nid` (unique **n**etwork **id**entifier). The `nid` is defined whenever a new network is created (which is described in the next paragraph). A `Network` object can represent either a Model Graph network or a NAVTEQ network. A NAVTEQ network is a direct subset of the NAVTEQ map for a given region. The `Network` class has a number of utility functions for accessing various features of the road network, searching the graph, or determining the closest point on the network to a given GPS coordinate. The `DataType` class is used for reading traffic data from the database or for writing traffic estimates to the database. It is divided into static inner classes for each type of data that is received from or written to the database. For example, if a developer wants to use PeMS data in a highway traffic estimation model, they would use `DataType.PeMS` to retrieve data for a specific time interval

for use in their model. The `Datum` class provides the data structures that are passed between the `DataType` class to the developer's model. In the PeMS example, the return type for a function in `DataType.PeMS` that retrieves data from the database is `Datum.SensorPeMS`. The remainder of the classes in the `netconfig` package represent the building blocks of the `Network` class by providing data structures such as `Link` (a link of the road network), `Node` (an intersection in the road network), `Spot` (a specific location on the road network), and `Route` (a path through the road network), among others.

The primary classes in the `model_graph` package are `BuildModelGraph` and `CreateNetwork`. The `BuildModelGraph` class has a main function in it that loads all of the necessary information from the NAVTEQ database, which it then uses to build the simpler representation of the road network known as the Model Graph. This involves removing all residential streets and removing unnecessary nodes from the graph. The process also constructs one-way links for all parts of the roadnetwork (where NAVTEQ generally uses a single link to represent both directions of traffic) and classifies each link as being a highway, arterial or ramp link. The full details of the algorithm for building the model graph are found in section 7. The `CreateNetwork` class is used to create a new network in the database. A network is defined as a subset of the Model Graph or as a subset of the NAVTEQ map, but the `CreateNetwork` class is only used for creating Model Graph networks (because the NAVTEQ database already has all of the information needed for constructing a NAVTEQ network). The `CreateNetwork` class provides a few different ways of constructing the network. One can specify a bounding polygon, a list of NAVTEQ link IDs, or a list of Model Graph link IDs as well as whether to include highways, arterials or both as a means of selecting the links that are included in the network. Upon successful completion, the process returns a new nid that can be used to instantiate a `Network` object as described in the previous paragraph. The remainder of the classes in the `model_graph` package are used by the `BuildModelGraph` class in the process of constructing the Model Graph.

All of the classes in the `util` package are constructed on an "as needed" basis when several developers all require the same functionality. Examples include `NAVTEQDatabaseConversions` (which provides methods for interpreting the NAVTEQ "speed category" and "lane category" types used in the NAVTEQ database) and `SpatialDatabaseQueries` (which provides access to the spheroid definition of the Earth and some useful values about the dimensions of the Earth).

6.6.3. Input Feeds

Input feeds are individual programs designed to run independently of the rest of the system. The basic requirements of each input feed are to establish a connection with the data provider, collect the raw data and write it to the database. The program also logs (to the monitoring database) the amount of time needed to get each batch of data as well as how many records were obtained. The input feeds are located within the **INPUT_MANAGER** NetBeans project.

6.6.4. Filters

Filters are individual programs designed to run independently of the rest of the system, although they rely on the input feeds correct operation in order to produce valid results. The basic requirements of a filter are to read raw input data from the database, process the data into a prescribed format and output it back to the database. The other requirement is that the filter establish the location of the data on the NAVTEQ map. There are two general types of filters in the *Mobile Millennium* system. The first operates on data coming from fixed-location sensors such as loop detectors or radar and the second operates on GPS probe data. The first type of filter requires a static mapping component which places each fixed sensor on the NAVTEQ map and a dynamic component that processes incoming raw data as quickly as it arrives. These types of filters rely on a fixed map matching procedure developed for fixed sensor data and then a custom filter is built to handle the specifics of the raw data arriving. In terms of the database design, there is a cost savings by having the map matching done once and then only needing to reference a sensor identification number when processing real-time data.

The second type of filter for GPS data requires both filtering in the more traditional sense of the word (removing outliers, smoothing, etc.) as well as map matching for every data point that arrives. This type of filter is much more computationally intensive than the first type as performing map matching is a time consuming process because it relies on a spatial query to the database for each GPS point. This filter has been implemented in Scala and in such a way as to leverage parallel computing technology as more hardware resources become available to the system. Choosing this design strategy means that this computationally intensive filter can scale well when the volume of data substantially increases as is expected in the coming years.

The filters exist in several different NetBeans projects. **PATH_INFERENCE** is the project for processing GPS probe data (written in Scala). Other filters are located in the **INPUT_MANAGER** project.

6.6.5. Models

These modules in the *Mobile Millennium* system are the most important components of the system from a scientific point of view. They represent the implementation of research ideas that often take years to put into production. Since they are generally significantly bigger modules than the other types, much of the focus of the systems team is on how to make them run as smoothly as possible. There are two requirements that models must adhere to in the *Mobile Millennium* system. First, it is necessary that any estimation model run fast enough to be used in real-time. This means that the model itself needs to be computationally efficient, but also means that the systems team needs to ensure that the model has the server resources necessary to run at fast speeds. The second requirement is that all inputs and outputs of the model go through the core software module, which ensures a smooth interaction with the database. These modules take filtered input from the database, run it through a traffic estimation model and write out the resulting traffic estimates to the database.

Examples of these types of modules include the **HIGHWAY** and **ARTERIAL** NetBeans projects, which contain the traffic estimation algorithms for highway and arterial roads,

respectively.

6.6.6. Output Feeds

The output feeds in the *Mobile Millennium* system are stored in the **OUTPUT** NetBeans project. The basic requirement of an output feed is to extract traffic estimates or related traffic data from the database and send it to a third-party system.

6.6.7. Analysis

This module consists of the visualization and evaluation sub-modules.

Visualization

The code for the *Mobile Millennium* visualizer is in the **DIVA** NetBeans project. The visualizer is responsible for querying the database and finding an informative depiction of the data overlayed on a digital map. This includes raw data, filtered data and traffic estimates. Figure 6.2 displays the visualizer used for internal use within the *Mobile Millennium* team, showing highway traffic estimates alongside raw PeMS data.

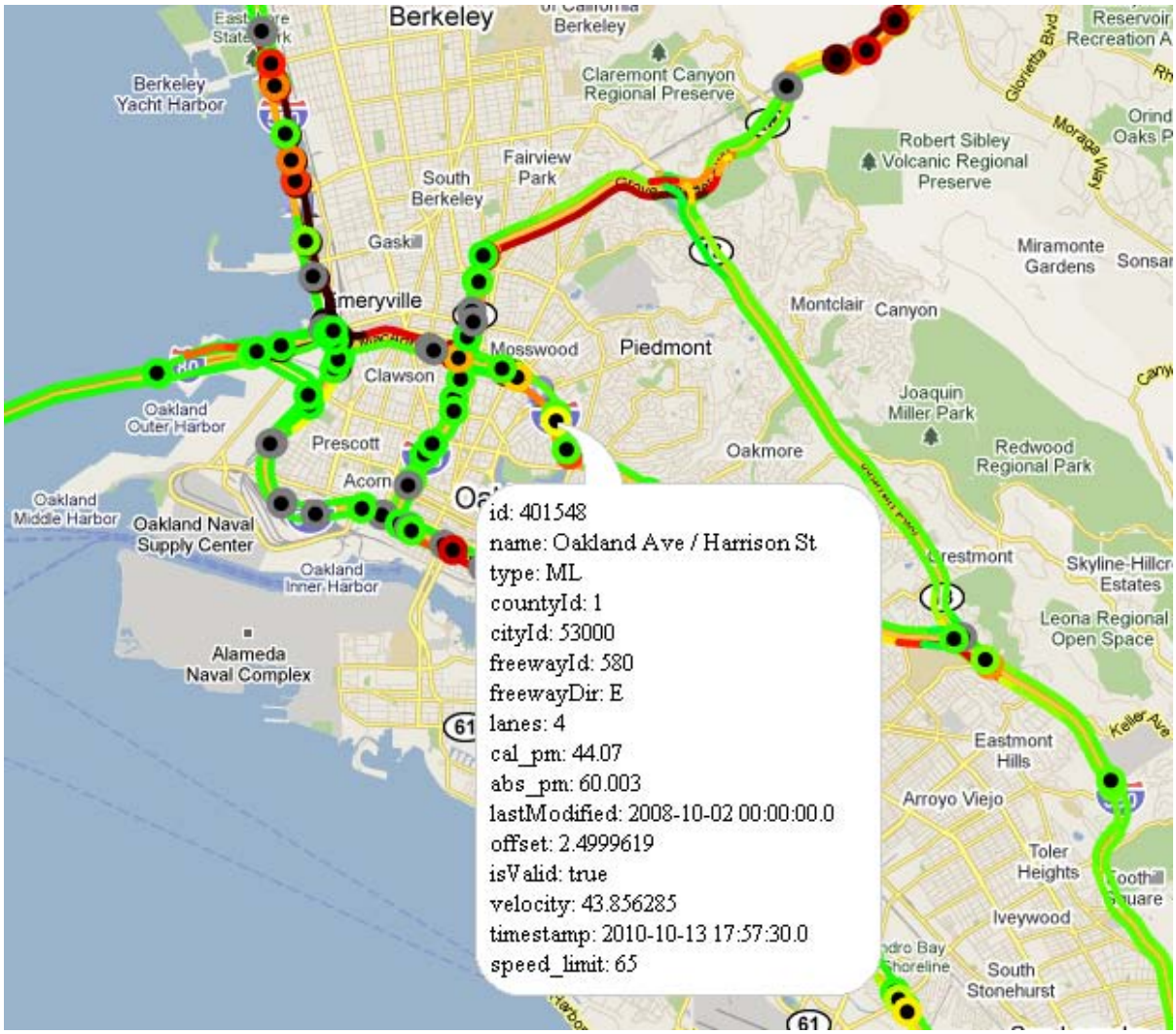


Figure 6.2. ***Mobile Millennium*** **internal visualizer showing model outputs and locations of PeMS loop detectors (hollow circles).**

Model Evaluation and Raw Data Analysis

The code for performing model evaluation and analysis of input data is in the **VALIDATION** NetBeans project. The evaluation module is responsible for two different types of analysis. The first is to provide descriptive statistics and plots about a single data source or traffic estimates. The second is to provide metrics and plots that compare two different data sources or traffic estimates. Examples of the former include showing the average speed of a particular loop detector over time on a plot or to provide the typical mean speed value observed on all Tuesdays at 9am. Examples of the second type include a plot showing travel times collected from video cameras compared with estimated travel times from a model using GPS probe data as input or computing the root mean squared error between the estimates and the observed travel times.

The intention of the evaluation module is to be able to specify any data source or any two data

sources, specify a metric or plot and then generate the desired output (i.e. the value of the metric, a plot showing the comparison, etc.). The system is not currently equipped with all possibilities of data sources, metrics and plots, but it is capable of a subset of these.

6.7. Model Graph Specification

The Navteq map contains a lot of short links useful for navigation, but irrelevant for traffic modeling. The goal of the model graph is to either remove or merge these short links where appropriate. Additionally, the model graph only considers class 1-4roads so that we do not have to worry about the class 5 residential roads, which make up the bulk of the Navteq database. The model graph has the following qualities:

- Each link will be comprised of one or more Navteq links
- Each link will be "unidirectional", meaning that traffic flows in only direction along the link

There are several types of operations in creating the model graph: link removal, node removal/link merging, "unidirectionalizing" links, "composite intersection" consolidation, and link shortening. These operations are performed in the order listed when constructing the model graph.

- **Link removal**

This is the simple operation of not including a link that is in the Navteq database in our model graph. Class 5 roads will not be included.

- **Node removal/link merging**

This operation effectively removes a node that was present in the Navteq database and merges the links that were attached to it. Rules for node removal:

  - The node must have exactly two adjacent links that are both bi-directional (two-way) or both unidirectional (one-way) and in the latter case, there must be one incoming and one outgoing link for this node.
  - Special rules for highways also require that the two adjacent links must have the same number of lanes and same speed limit.

- **Composite intersection consolidation**

This operation takes a "composite intersection" (defined as being an intersection of doubly digitized roads that creates "phantom" links) and reduces it to a single node in the model graph. The "phantom" links will be removed only if they are links from two or more intersecting roads where additional links were created solely due to this intersection. See the Navteq map documentation for full details on why these phantom links are present in the Navteq map.

- **Unidirectionalizing Links**

This operation takes all bidirectional links that are present in the model graph after running all of operations 1-3 above and makes those links unidirectional by creating two links for each bidirectional link. The two unidirectional links will have reverse geometries and will have opposite start and end nodes.

- **Link Shortening (Highways)**

This operation takes highway links that exceed a certain threshold and breaks them into as few number of links as possible while keeping each link under the threshold and maintaining link breaks at the boundary of Navteq links. The value for this threshold is 1500 meters.

Database Tables

The database contains a schema called `model_graph`, so all of the tables below will be `model_graph.table_name`. The schemas for each of the tables below are in the repository under `NETCONFIG/scripts/dev2/model_graph/`.

- **`link`** Each row in this table stores necessary data for a model graph link.
- **`link_navteq_link`** This table is used as an intermediary to encode the one to many relationship that exists between our links and Navteq links.
- **`node`** Each row in this table stores necessary data for a model graph node.
- **`node_in_links`** Each row in this table stores an incoming link for a node.
- **`node_out_links`** Each row in this table stores an outgoing link for a node.
- **`node_links`** Each row is a "node id/model graph incoming link id/model graph outgoing link id" triple where one can get allowable routes (incoming/outgoing link pairs) through a node by selecting all rows with the specified node id.
- **`node_composite_intersection`** Each row is a "node id/Navteq link id" pair where the entry means that the node is a composite node which contains the specified Navteq link.
- **`traffic_flow`** Each row represents a traffic source or sink.
- **`modified_navteq_links`** Each row in this table represents an override of attributes in the Navteq (DCA) streets table .
- **`network`** Each row in this table represents a network, which is a subset of links of the link table.
- **`network_links`** Each row in this table is a "network id/model graph link id" pair. A network is described by collecting all model graph links for a given network id.

6.8. Hardware

6.8.1. Introduction

This section describes the various pieces of computing hardware used for the Mobile Millennium

system. Many of the individual pieces can be grouped to form separate distinct systems. Some of the systems underwent changes in both configuration and usage as the project took shape. First we will mention the original deployment plan and after we will mention how some of the systems ended up changing.

It's worth noting that nearly all of these machines are now housed in the UC Berkeley Colocation (colo) facility. During the first half of the project, a number of the systems were housed on site. The colo provides a number of advantages over housing the servers in our offices. These are:

- Controlled physical access and closed circuit video surveillance.
- Conditioned power with backup UPS powered by diesel generators.
- Fire suppression system.
- HVAC system with filtering to remove dust and contaminants.
- Network connection provided by two 10 Gigabit trunks that connect to the core campus network via separate paths.

6.8.2. Live System

Responsibilities: Runs the current stable version of the mobile millennium software. Stability is the focus for this system. No development occurs on this system and updates are deployed in a very controlled manner after thorough testing has been performed. Only data that is necessary for the functioning of the mobile millennium software is stored here. Any data that we wish to keep is moved nightly to the data warehouse via the ETL process.

Description: Consists of two machines. The first is a compute server that runs the various software components for the Mobile Millennium system. The second is a database server that manages the data generated on the compute server.

Compute server specs:

- Processor: 2 x Intel(R) Xeon(R) CPU X5460 @ 3.16 GHz
- RAM: 8 x 4 GB ECC DDR2 @ 667 MHz
- Disk: 2 x 73.5 GB 15000 RPM, SAS @ 3.0 Gbps
- RAID: Hardware - SAS 6/iR, RAID-1
- OS: Linux

Database server specs:

- Processor: 2 x Intel(R) Xeon(R) CPU E5335 @ 2.00GHz
- RAM: 8 x 4 GB ECC DDR2 @ 667 MHz
- Disk: 4 x 1 TB 7200 RPM, SATA II @ 3.0 Gbps
- RAID: Software - Linux, RAID-5
- OS: Linux

6.8.3. Development System

Responsibilities: Runs the latest unstable version of the mobile millennium software. The focus for this system is to provide an environment that is similar to the live system so that researchers can easily add new functionality to the software and test to make sure new software is running correctly. Updates are always performed on this system first before they are performed on the live system.

Description: Consists of two machines. The first is a compute server that runs the various software components for the mobile millennium system. The second is a database server that manages the data generated on the compute server.

Compute server specs:

- Processor: 4 x Dual-Core AMD Opteron(tm) Processor 8214 @ 2.20 GHz
- RAM: 16 x 4 GB ECC DDR2 @ 667 MHz
- Disk: 3 x 400 GB 10000 RPM, SAS @ 3.0 Gbps
- RAID: Hardware - PERC 5/i, RAID-5
- OS: Linux

Database server specs:

- Processor: 2 x Intel(R) Xeon(R) CPU E5335 @ 2.00GHz
- RAM: 16 x 4 GB ECC DDR2 @ 667 MHz
- Disk: 6 x 1 TB 7200 RPM, SATA II @ 3.0 Gbps
- RAID: Software - Linux, RAID-5
- OS: Linux

6.8.4. Data Warehouse

Responsibilities: Stores all of the data received or generated by the mobile millennium project that researchers wish to keep. This involves raw unprocessed data from our various data feeds that would be difficult to reobtain, and whatever processed data researchers think they might need at a later time. This data is populated by the ETL process from the live and development systems nightly. Once the ETL has completed, the data that was transferred is removed from the live and development systems. This is necessary to keep the databases for the live and development systems small and fast so that those systems can process data in real time.

Description: The data warehouse consists of a server with a direct attached storage device connected via two fiber channel cards. The database files are housed on the direct attached storage device.

Server specs:

- Processor: 2 x Quad-Core AMD Opteron(tm) Processor 2356 @ 2.30 GHz
- RAM: 16 x 4 GB ECC DDR2 @ 667 MHz
- Disk: 2 x 137 GB 10000 RPM, SAS @ 3.0 Gbps
- RAID: Hardware - SAS 6/iR, RAID-1
- OS: Linux

Direct attached storage specs:

- Dell PowerVault MD3000
  - 15 x 750 GB 7200 RPM, SATA @ 3.0 Gbps
  - RAID-6 with 2 Hot Spares

6.8.5. Experiment Machines

Responsibilities: Not strictly part of the mobile millennium system. This represents a collection of machines that are "given" to researchers who want to perform mobile millennium related tasks that would likely disrupt the functioning of the development system. These are tasks such as using a different database table configuration, running a long compute job, or needing data that is not present on the development system.

Description: These machines are whatever we might happen to have on hand. 10 of them are older machines donated to the University by other companies, and then distributed to various departments that requested them for free. They are added and removed as necessary and their operating systems are configured to match whatever the researchers request. If the researchers requested data to be loaded on the machines, it would be copied from the data warehouse.

Specs: These vary, but in general they are all roughly equivalent.

- Processor: 2 x AMD Opteron(tm) Processor 248 @ 2.20 GHz
- RAM: 8 x 2 GB ECC DDR @ 400 MHz
- Disk: 2 x 1 TB 7200 RPM, SATA @ 3.0 Gbps
- RAID: Software - Linux, RAID-1
- OS: Linux

6.8.6. Mobile Millennium Wiki Machine

Responsibilities: Performs three main functions. First, it serves up a wiki that is used as a central point for sharing information about the project. Second it houses the subversion software repository that is used to hold all of the code for the project. Lastly, it holds the arterial binary datastore which represents files used by the arterial team that were not suitable for being stored in the subversion repository.

Description: The wiki machine has very modest requirements only needing to be able to store files and serve up webpages. As such we used an older desktop machine and inserted a SATA card to add modern disks.

Specs:

- Processor: 1 x Intel(R) Pentium(R) 4 @ 1.7 GHz
- RAM: 2 x 256 MB RDRAM @ 400 MHz
- Disk: 1 x 1 TB 7200 RPM, SATA @ 3.0 Gbps
- RAID: none
- OS: Linux

6.8.7. IBM PeMS Machine

Responsibilities: Provides filtered PeMS data to IBM to aid in the development of their Traffic Prediction Tool (TPT). The TPT uses filtered PeMS data as input and generates speed and flow estimates for 5, 10, 15, 30, 45 and 60 minutes into the future. The predictions are provided to PATH for analysis.

Description: This machine has very modest requirements. A 5 year old machine donated by another department was used.

Specs:

- Processor: 2 x Intel(R) Xeon(TM) CPU @ 2.80 GHz
- RAM: 4 x 1 GB ECC DDR @ 200 MHz
- Disk: 5 x 36 GB 10000 RPM, Ultra SCSI 320
- RAID: Software - Linux, RAID-6
- OS: Linux

6.8.8. traffic.calccit.org Webserver

Responsibilities: Serves the public facing website for the Mobile Millennium project. It was used to distribute any information that the researchers wanted the public to know about the project. The public used this site to download the Nokia phone client so they could participate. The site was also used to distribute any data or highway model ouput that was released by the project.

Description: This machine only had to support the website, so it had very modest requirements. A small 1U server was used.

Specs:

- Processor: 1 x Intel(R) Celeron(R) CPU 430 @ 1.80GHz
- RAM: 4 x 1 GB ECC DDR2 @ 800 MHz
- Disk: 2 x 750 GB 7200 RPM, SAS @ 3.0 Gbps
- RAID: Hardware - SAS 6/iR, RAID-1
- OS: Linux

6.8.9. Mobile Millennium Gateway Machine

Responsibilities: Handles services such as e-mail and time synchronization for other machines being used on the project that do not have public IP addresses.

Description: The requirements for the gateway machine are very modest. We used a machine lent to us by Nokia.

Specs:

- Processor: 1 x Intel(R) Xeon(R) CPU E5410 @ 2.33GHz
- RAM: 2 x 2 GB ECC DDR2 @ 667 MHz
- Disk: 1 x 500 GB 7200 RPM, SATA @ 3.0 Gbps
- RAID: none
- OS: Linux

6.8.10. Mobile Millennium Monitor Machine

Responsibilities: Performs three functions. First it serves the monitoring database which holds all the data generated by the various mobile millennium processes via the monitor core class. Second, it runs the actual monitoring system which communicates with the monitoring database to generate alarms if the system is not functioning properly. Lastly, the machine runs Hudson, an automated build tool, which ensures that the mobile millennium software compiles properly.

Description: The requirements for the monitor machine are very modest. We used a machine lent to us by Nokia.

Specs:

- Processor: 1 x Intel(R) Xeon(R) CPU E5410 @ 2.33GHz
- RAM: 2 x 2 GB ECC DDR2 @ 667 MHz
- Disk: 1 x 500 GB 7200 RPM, SATA @ 3.0 Gbps
- RAID: none
- OS: Linux

6.8.11. CITRIS Visualizer Machine

Responsibilities: Used to display the visualizer at the CITRIS (The Center for Information Technology Research in the Interest of Society) tech museum.

Description: This machine had two requirements. It had to be small, and it had to be able to run a web browser. We decided to use an Apple Mac Mini.

Specs:

- Processor: 1 x Intel Core 2 Duo T5600 @ 1.83 GHz
- RAM: 2x512 DDR2 @ 667 MHz
- Disk: 1x80 GB 7200 RPM, SATA @ 3.0 Gbps
- RAID: none
- OS: Windows XP

6.8.12. Conclusion

Throughout the project we stayed pretty close to the hardware planned as laid out above. However, we did not anticipate the load that the databases would place on the servers. If we were to plan this again, we would better optimize the database servers for I/O. That would mean using hardware raid controllers with battery backup units and write back caches, and separate disks to hold different components of the database system. In addition we would use RAID-10 for the database machines instead of RAID-5. In fact, toward the end of the project we converted the RAID on the live database from RAID-5 to RAID-10. In addition, researchers ended up wanting a lot more data on the development server than was originally anticipated. This made the warehouse redundant. We decided to use the data warehouse machine as the development database server because of its capacity and much faster disk array.

6.9. Mobile Millennium Run Scripts

6.9.1. Introduction

The Mobile Millennium Run Scripts (mmscripts) were developed to make it much easier for researchers to work with the Mobile Millennium software. They do this by providing a consistent interface for working with each program while ensuring that there are no system permission problems from having several different people all interacting with the same software at the same time. Researchers use the mmscripts to start, stop, update, and monitor various Mobile Millennium software processes.

The mmscripts consist of three parts. The first part is a sudoers configuration file that allows everything to run as the same user regardless of which researcher is using it. The second part is a bash script that ensures necessary environment variable shave been set and then launches the mm manager. The third is the mm manager which does the actual work.

6.9.2. Use Instructions

A researcher uses the mmscripts by simply typing `mm` plus whatever arguments they need. Typing `mm` by itself prints out help documentation.

This is the syntax of the `mm` command:

```
mm <COMMAND> [PROGRAM] [DB_ENV] [NID] [MAIN_FUNCTION_NAME]
```

The COMMAND argument is used to specify what you want to do, such as start or stop something. The PROGRAM argument is used to indicate which program you are applying the command to. The DB_ENV argument is used to specify which database environment is being used. Generally this indicates which hardware stack you are using. Programs running on the live stack most often use the live database environment, programs running on the development stack most often use the dev database environment. The NID argument is used to specify the network ID. The MAIN_FUNCTION_NAME argument is used for the path inference filter which has several different run configurations based on what it is you want to do.

Which arguments are required depends on the command and program being used. If you fail to give the necessary arguments for a command or program, the mm scripts will inform you of which arguments are required.

This is a list of the available commands and what they do:

- `help`
  - Prints help documentation.
- `log <PROGRAM> [DB_ENV] [NID] [MAIN_FUNCTION_NAME]`
  - Displays the log file for the specified program (runs the linux command cat).
- `logtail <PROGRAM> [DB_ENV] [NID] [MAIN_FUNCTION_NAME]`
  - Continuously displays new lines added to the log file for the specified program (runs the linux command tail -f).
- `restart <PROGRAM> [DB_ENV] [NID] [MAIN_FUNCTION_NAME]`
  - Stops and then starts the specified program.
- `run <PROGRAM> [DB_ENV] [NID] [MAIN_FUNCTION_NAME]`
  - Runs the specified program in the foreground, and all output is displayed on your screen and not written to a log file.
- `start <PROGRAM> [DB_ENV] [NID] [MAIN_FUNCTION_NAME]`
  - Runs the specified program in the background, and all ouput is written to a log file.
- `startall`
  - Starts all programs that were stopped by the last stopall.
- `status`
  - Shows all of the programs currently running.
- `stop <PROGRAM> [DB_ENV] [NID] [MAIN_FUNCTION_NAME]`
  - Stops the specified program.
- `stopall`
  - Stops all running programs. Nothing else can be started or run until a startall command is given.
- `update`
  - Updates code from SVN repository and compiles it.

6.9.3. Configuration

The mmscripts are designed to make it easy for researchers to add additional programs for the scripts to control. To add a new new program to the mmscripts, a researcher must make two modifications. The first modification is to add the name of the program to the programs list in the `printPrograms` function. The second modification is to add the program configuration to the `getProgramConfig` function. Each configuration definition is represented as a case statement where the selection is based on the name of the program. The following is the format for a new program configuration:

```
case "prog_name" =>
   new ProgramConfig (
      db_env = Boolean,              // specifies whether or not the da
tabase environment needs to be specified
      nid = Boolean,                 // specifies whether or not a netw
ork id needs to be specified
      main_function_name = Boolean, // specifies whether or not a main
 function needs to be specified
      classpath = Array[String],    // A string array which contains e
ach item in the java/scala classpath
      java_opts = Array[String],    // A string array which contains a
ny arguments being passed to the jvm
      className = String            // A string that lists the name of
 the class the jvm will be running
   )
```

After the program has been added to the program list and its configuration has been added, the mm manager must be recompiled. This is done by running `make` in the directory where the mm manager is located.

6.10. Mobile Millennium Monitoring System

6.10.1. Introduction

The Mobile Millennium Monitoring System provides real-time automatic monitoring of the various pieces of Mobile Millennium software. If a problem is detected, it sends email alerting researchers of the problem so that they can take corrective measures to fix it. The system consists of three parts. These are the dashboard or front-end, the monitoring database, and the back-end. Careful consideration was used to keep the parts independent. This makes it possible to make changes to each individual piece without having to completely rewrite the whole system.

6.10.2. Dashboard

The dashboard is the web based front-end to the Mobile Millennium Monitoring System, as shown in Figure 10.1.

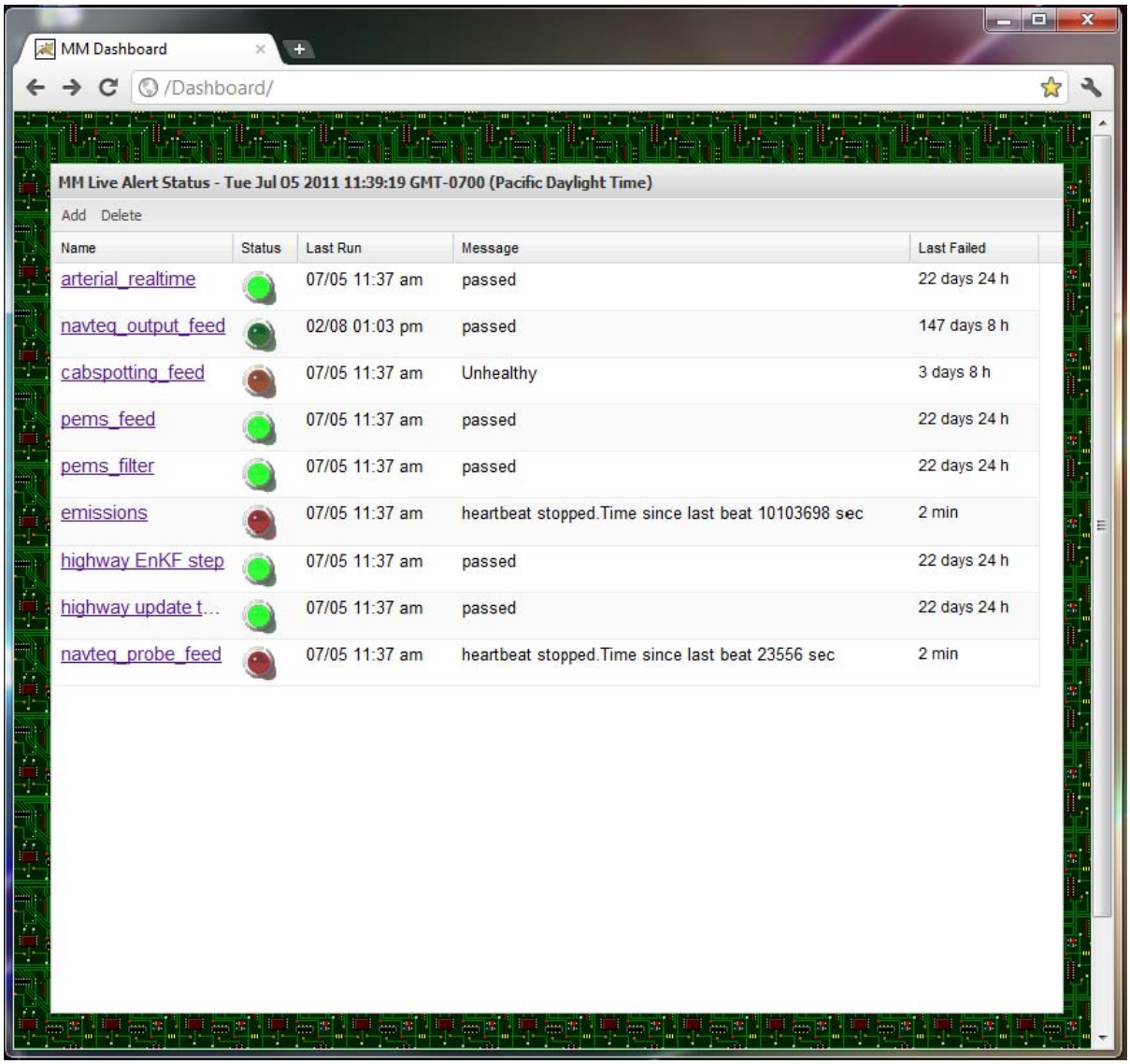


Figure 10.1. Screenshot of the Mobile Millennium dashboard.

It is used to view the current status of the system and to configure alarms for the system.

The dashboard's display consists of 5 columns:

- Name - A descriptive name given when the alarm is created.
- Status - A green, yellow, or red indicator that indicates the health of the service. Green is healthy, yellow is a warning, and red means the service is not currently working.
- Last Run - The last time the alarm was checked.
- Message - Information logged by the service to the monitoring database.
- Last Failed - How long it has been since the last time an alarm was triggered.

From the dashboard it is possible to add or configure alarms. To add a new alarm, a researcher

would click “Add” in the menu bar. This simply adds a new entry to the alarm display which can then be configured. Unfortunately the functionality to delete an alarm was never completed. Therefore even though there is a “Delete” menu item, it is currently not functional. Clicking on the alarm’s name in the name column brings up the configuration dialog for that alarm, as shown in Figure 10.2.

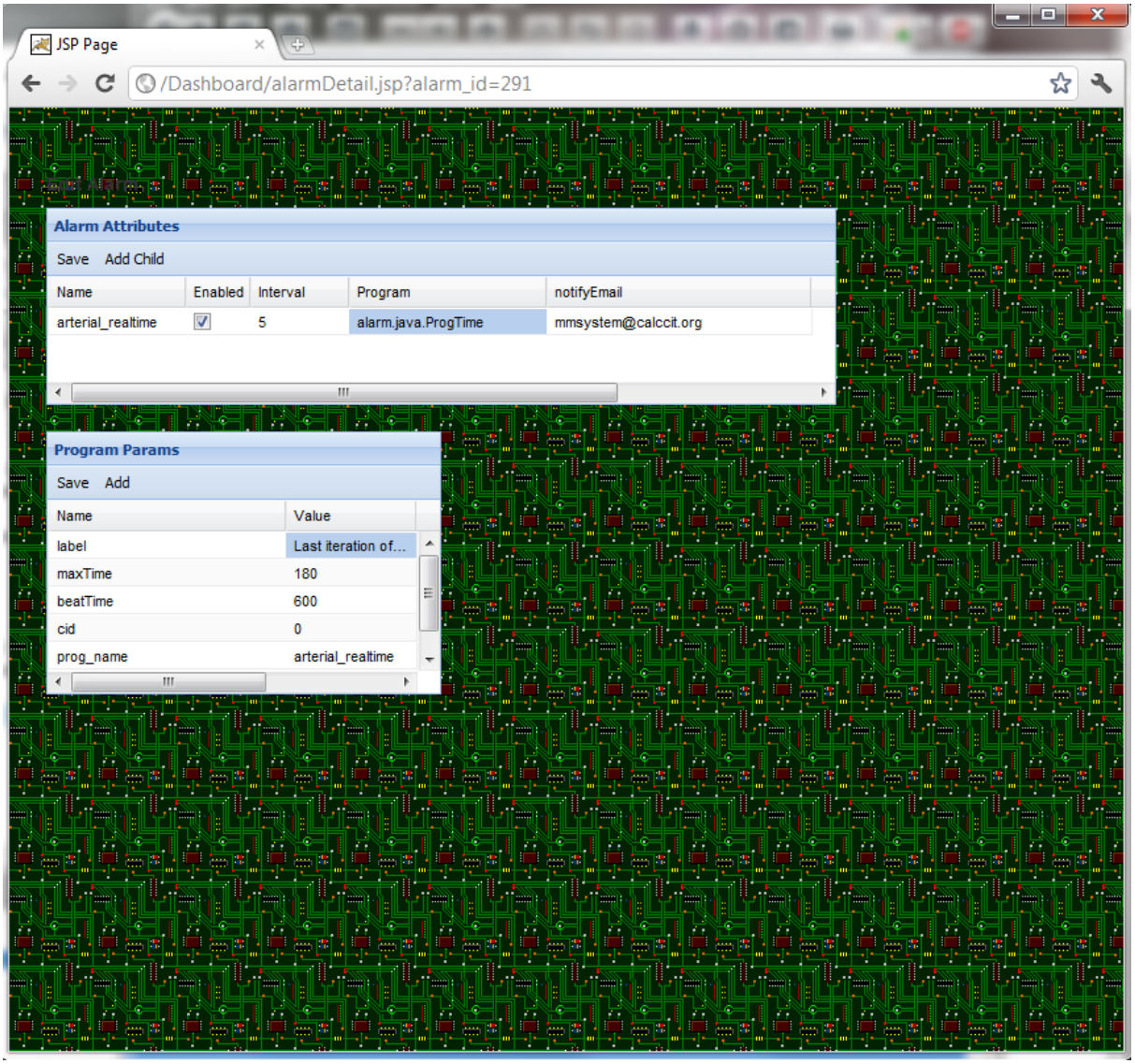


Figure 10.2. Screenshot of the Mobile Millennium dashboard alarm configuration screen.

The configuration screen consists of two windows. The first window named Alarm Attributes consists of 5 columns:

- Name - A descriptive name that is used to identify the alarm.
- Enabled - Indicates whether or not the alarm should run. This is particularly useful for when researchers know that particular service will be down for a while and

hence do not need to receive alarms for it.

- Interval - This is how often the alarm will be checked in minutes.
- Program - This specifies the type of alarm.
- notifyEmail - This is who will receive email in the event that the alarm is triggered.

The second window named Program Params is used to provide specific details about the alarm. The details are specific to the type of alarm being used.

In order to give researchers a lot of flexibility for alerts, the system supports 6 different types of alarms, as shown in Figure 10.3.

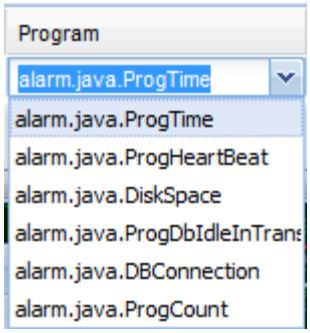


Figure 10.3. Screenshot of the Mobile Millennium dashboard alarms list.

ProgTime, ProgHeartBeat, and ProgCount are the most used. They allow researchers to alert based on how long something takes, how often a particular application accesses the system, and how many times an application performs a specific action within a given time period.

6.10.3. Monitoring Database

The monitoring database is a PostgreSQL database that runs on the monitoring server (mmmon). It contains the alarm configuration information that is entered through the dashboard, all of the monitoring information that is provided by the Mobile Millennium software via the monitor core class, and the state of all of the alarms as determined by the back-end.

Every single piece of monitoring information that is retained is stored in the database. This makes it possible for researchers to write database queries to get historical information about how well their application has been running. Furthermore, if they wish, they can even use software to graph this information to gain an overall historical perspective.

6.10.4. Back-end

The back-end is a combination of a Perl script and Java classes that run on the monitoring server (mmmon). The purpose of the back-end is to determine whether or not an alarm needs to be triggered and to send out the appropriate notifications if one was. The Perl script is scheduled to run every minute. When it runs, the script sets up the appropriate environment variables and then

runs the Java classes. The Java classes connect to the monitoring database, get the alarm configuration information, and then read the appropriate monitor core class tables to determine whether or not a problem has occurred. After the Java classes have determined an alarm's state, it writes the state to the monitoring database so that it can be displayed by the dashboard.

The Java classes can be thought of as plugins. Each alarm type has a Java class associated with it. If a researcher wishes to add a new type of alarm to the system, they would add a new Java class to the back-end that would be responsible for handling that alarm type.

6.10.5. Conclusion

The Mobile Millennium Monitoring System proved to be an invaluable tool for several reasons. First it greatly reduced system downtime by immediately alerting researchers to any problems with system when they occurred. Second, it made manual checks of the system unnecessary, thus allowing researchers to devote time to other important tasks. Lastly, the monitoring system provided a central information repository that researchers could use gain historical perspective about how well various software components were running.

One improvement that could be made with the monitoring system is to replace both the front-end and back-end with Nagios. Nagios is an open source monitoring tool that provides a web-based graphical display and an easy to use plugin mechanism for adding alarms. It provides several advantages over the current system. First it allows plugins to be written in any programming language. Second it automatically provides graphs to easily gain historical perspective. Lastly it provides a lot of options for notifications. One of the biggest problems with the current system is that in the event there is a problem, the system can send out an excessive number of email notifications. Nagios can be easily configured to only send out notifications when they are truly necessary.

7. System Modules

7.1. Introduction

The purpose of this chapter is to briefly explain the structure and function of modules in the *Mobile Millennium* system. Module names are capitalized and appear in **BOLD**. Class, field and method names are in `typewriter` font. Static methods are referred to in the form `ClassName. methodName` and instance methods are referred to using the method name (i.e. simply `methodName`).

7.2. ALARM

The **ALARM** module is used to alert MM system administrators in the event of a software or hardware problem. The main class for controlling the alarm system is `AlarmDispatcher`. `AlarmDispatcher` keeps an up-to-date list alarms and their statuses. When an error message is received, it sends an e-mail message to the systems team containing information regarding the error. The list of recipients can be specified per alarm.

Each alarm is expressed as an instance of the `Alarm` class. Each `Alarm` contains fields to store its status, the program with which it is associated, e-mail addresses to which to send notifications, and a list of child alarms. Each `Alarm` can spawn child alarms or be a child itself. This is useful for grouping related alarms, and making them more specific.

ALARM also contains a group of `Action` classes which are used to execute tasks when appropriate, such as sending an e-mail or running a command. `Alarm` objects have a field for action. Alarms may be based on different types of conditions such as checking a numerical or Boolean value, verifying that a connection is valid, or listening to a program's heartbeat.

7.3. ARTERIAL

The **ARTERIAL** module contains the software implementation of the arterial traffic models, a

significant component of the MM system. It is very complex, and a detailed description is outside the scope of this chapter. The Java source consists of 5 packages:

- `arterial_core:` Contains the basic elements of the model expressed as abstract classes such as `ArterialNetwork`, `ArterialLink`, etc. Also contains interfaces used by the model (e.g. `LinkDensityDistribution`) and a file read/write utility.
- `chmm`: Contains classes specific to the CHMM model, including subclasses of the network elements in arterial_core.
- `density_model`: Contains classes for the density model, including a `LearningAlgorithm` class that varies model parameters to improve accuracy.
- `independent_link_tt_model`: Contains the model for computing travel times on individual links in real-time, assuming links are independent. A Gaussian distribution of congestion and travel time is computed for each link.
- `isttt`: Contains classes for the ISTTT model, including network elements, probability distributions, and route-based travel times.

The Independent Link Travel Time model is the current arterial algorithm employed by the system to provide real-time traffic information. However, it has both real-time and historical components.

7.4. CABSPOTTING

CABSPOTTING is a feed that obtains and stores location information collected by San Francisco taxi cabs. The information is stored on an external server, and the feed pulls data through a query for a specific time interval of data. Upon receiving a response, the data is parsed and inserted into the database. The **CABSPOTTING** module is fairly old code; it does not use the database classes available in the **CORE** module.

Instead, it has its own database accessor class `cabsAccessor` and its own `Exception` handling classes. These, and the `cabsMain` class that does all the work, are in a single package called `rawFeed`. The `cabsMain` class extends Java's `TimerTask` and runs every 60 seconds to retrieve and store the latest data from cabspotting.org.

7.5. CORE

As its name suggests, the **CORE** module contains classes used by almost every other module in the MM system. The purpose of **CORE** is to provide various functions which are commonly used and which would be redundant if programmed for every module. For example, almost every part of the MM system must access the database, and **CORE** provides classes to do this that can be used anywhere in the code. In addition to reducing code size and complexity, **CORE** also improves compatibility and portability. It also contains third-party .JAR libraries which are widely used such as PostgreSQL JDBC tools.

7.5.1. Database

CORE provides classes capable of reading and writing to the database through the `DatabaseReader` and `DatabaseWriter` classes, each of which are subclasses of the abstract `Database` class. These classes provide a simple, uniform manner for accessing the database for all modules of the system. The `Database` classes form a JDBC connection by setting parameters such as host, port, name, user and password based on the properties of the program that instantiates them. Thus, a successful connection can often be programmed without specifying any parameters directly. Additionally, the `Database` classes allow programmers to create the own prepared SQL statements, execute these statements, and manage the result set that is returned.

7.5.2. Exceptions

The `Exceptions` class provides useful static methods for analyzing exceptions, extending those already available in Java. For example, Java does not provide an easy way to store an exception's stack trace in a `String`, but `Exceptions` has a method to do it. There are also methods that can analyze an exception to help diagnose its root cause, e.g. the `isDuplicatePK` method will return a Boolean value indicating whether or not a database exception was caused by attempting to insert a row with a duplicate primary key.

7.5.3. Geometry

The MM system contains a great amount of location-based information, most commonly stored as a point (mostly WGS84 longitude and latitude) or a sequence of points. **CORE** offers a couple of classes to represent them. The `Coordinate` class stores a single point and has methods for computing the distance between points depending on the spatial reference being used (plane, sphere, ellipsoid).

The `GeoMultiLine` represents a sequence of points (which when connected by lines are often called a line string). It stores a list of `Coordinate` objects, and it provides functions for analyzing and manipulating the geometry of the line string. Both this and `Coordinate` are used by the `Database` classes when inserting spatial data.

7.5.4. Monitor

The `Monitor` class provides logging, debugging, error display and monitoring for the entire system. `Monitor` is mainly used for sending messages to the console, like Java's `System.out.print` method, but it offers specific messaging functions for debug, error reporting and general information that can be individually enabled/disabled.

`Monitor` is a singleton class that gets instantiated by every program that runs in the system. It keeps track of things like the program name, IP address, current environment, etc. The `Database` classes use this information to determine the connection parameters without needing

the programmer to specifying them.

`Monitor` also connects to a special monitoring database and can record messages there. It is possible to record metrics such as count and duration or send a "heartbeat" to indicate a program is running properly. These features are often used by the alarm system; after a certain length of time without a heartbeat or if a count or duration exceeds a threshold, an alarm will occur (see the **ALARM** section).

7.5.5. Time

Programming tasks involving time and date can be challenging, particularly when different timezones are involved. **CORE**'s `Time` class extends Java's `GregorianCalendar` class and makes operations involving time much easier, and is used across the MM system.

Most `Time` objects are created using the default constructor which sets the time fields (`YEAR`, `DAY_ OF_ MONTH`, `SECOND`, etc.) to the current Berkeley time based on a call to `System.currentTimeMillis`. There arealso static methods that will create a new `Time` object by parsing a `String` for a certain application such as `Time.newTimeFromGoogleXMLFeed`. The instance methods in `Time` are used for time arithmetic (`this.add` which adds an integer number to the specified field) and accessing the fields, converting to `String`, etc.

7.6. DASHBOARD

DASHBOARD is a web application that provides a GUI front end for the **ALARM** module. Through a series of JSP pages, users can create new alarms, view current alarm status on a per program basis, edit existing alarm parameters such as thresholds and e-mail lists, and enable or disable alarms.

DASHBOARD has only a single class on the server side: `AlarmSchema`. `AlarmSchema` calls the appropriate method(s) in **ALARM**'s `AlarmDao` class based on the user's input.

7.7. DEV_ENV

DEV_ENV contains a set of classes that perform miscellaneous functions. They are organized into the following packages:

7.7.1. geo

This package contains a single class, `CCS83Point`, which represents a location in the California Coordinate System of 1983. CCS83 divides California into six zones and expresses coordinates as a planar Easting and Northing pair in meters or feet. This class was used by the now defunct California Highway Patrol incident feed which reported traffic incidents (e.g. accidents) using CCS83 for location. `CCS83Point` converts CCS83 coordinates into the

familiar decimal degrees longitude and latitude format using a formula found in a Caltrans document entitled “The California Coordinate System" by Vincent J. Sincek”. It can also perform the reverse transformation. There is also a `HashMap` which maps each CHP communication center with thezone in which it is situated.

7.7.2. process

There is only a single class in this package called `RunWindow` which defines the times when MM system processes will run. The system generally shuts down from midnight to 5:00 AM in order to facilitate database maintenance, with an extended period on the first Saturday of each month. The `RunWindow.shouldRun` method returns a Boolean which indicates whether a process should be running based on the current time. Modules can call this method to determine whether they should shut down.

7.7.3. tcphelper

`TCPTransaction`, the only class in this package, transmits a byte array to a given URL. This was used for sending arterial model output directly to NAVTEQ, as they requested data in byte arrays which they unpacked into a datum object.

7.7.4. util.encrypt

As the package name implies, it contains tools for encrypting (and decrypting) data. It employs the DES (Data Encryption Standard) classes in Java to encrypt raw radar data as it is not permitted to be stored in a readable form.

7.8. DIVA

DIVA is the latest version of the visualizer. **DIVA** stands for Dynamic Information Visualization Application, and its function is to pull data from the database, transform it into objects containing location and attribute information, and display those objects on a map in a web application. For example, users can see the current output of the highway model overlaid on Google Maps tiles with each highway link colored according to estimated speed.

Unlike most other modules in the MM system, **DIVA** is not run as a standard Java application in under the MM Manager. Rather, it is a web application that runs in the Apache Tomcat servlet container and is accessed through a web browser. **DIVA** is composed of both server-side Java and client-side JSP and JavaScript.

DIVA uses the OpenLayers JavaScript mapping engine which provides an easy way to display geographic data. However, additional JavaScript is needed in order to provide a working GUI and process responses from the server. AJAX data marshaling between client and server are accomplished using a third-party library called DWR (Direct Web Remoting).

Requests for data are handled on the server by the `Diva` class. It passes the name of the

requested layer to the `VectorFeatureFactory` which returns an instance of the feature class for the layer. All feature classes implement the `VectorFeature` interface and extend the abstract `VectorFeatureBase` class. A feature represents anything that is shown on the map, such as PeMS speed measurements or arterial model links (which are respectively represented by classes `Pems30secFiltered` and `ArterialDynamicGaussian`.

These classes have a `getFeatures` method which returns an array of the features based on the request parameters supplied by `Diva` such as geographical bounding box or network. This method submits a query to the database and parses the result set into features having attribute and location information. The features are loaded into a `VectorLayer` object and `Diva` sends it to the client through DWR.

7.9. The Highway Model

The highway model is divided into three modules:

7.9.1. HIGHWAY

The **HIGHWAY** module contains the Kalman filter used to estimate highway traffic speeds. Kalman filtering is an algorithm for estimating the true value of something that has been measured over time (traffic speed in this case). It works by making a prediction based on past measurements, and then estimating the true value by computing a weighted (based on uncertainty) average of the prediction and the current measurement.

It also contains the manager of the entire highway model, `ModelManager`, which acts as a timer for the model processes and is used to start and stop the model.

7.9.2. HIGHWAYCOMMON

This module contains primarily classes representing the data types of the model input and output, such as `VelocityLinkRecord` and `TravelTimeRouteRecord.`

7.9.3. HIGHWAYFLOWMODEL

The highway flow model is contained in this module. It uses the Model Graph classes found in **NETCONFIG** and estimates traffic flow over the network over time, modeling phenomena such as “shockwave” speed and density.

7.10. MACHINE_STAT_TRACKER

The *Mobile Millennium* depends heavily on its computer hardware and ensuring that computers are operating properly at all times is paramount. The **MACHINE_STAT_TRACKER** module continuously (every minute) logs CPU, memory and disk usage onall MM machines. It uses third-party software called Sigar to extract relevant information from computers. **MACHINE_STAT_TRACKER** is composed of the following packages:

- `core`: Contains classes found in the **CORE** module such as `DatabaseWriter`, `Time`, and `Monitor` which have been stripped of all functions not used by **MACHINE_STAT_TRACKER**.
- `machine`: Contains the lone class `MachineStatTracker` which is the main class; it extends Java's `TimerTask` and executes the statistic logging process every 60 seconds.
- `machine.data`: Contains classes `CPUData`, `DiskData` and `MemoryData` which store data for CPU, disk space and memory use respectively. For example, `DiskData` has fields for mount name, bytes available and bytesused and stores that information in memory before it is sent to the monitoring database.
- `machine.monitor`: Contains classes `CPUMonitor`, `DiskMonitor`, `MemoryMonitor` which interact with `core.Monitor` to store the machine statistics. These classes contain `collectData` methods which retrieveusage statistics by calling Sigar and then call `core.Monitor` to store the data in the monitoring database.

By keeping track of hardware performance, the MM project team is able to diagnose problems easily and test ways of improving performance.

7.11. MM_MANAGER

The **MM_MANAGER** is used to manage and control the various programs running within the system. Unlike most of the system, it is written in Scala and not Java, and it is contained within a single class called `Main` in package `mm_manager`. **MM_MANAGER** allows users to see what programs are currently running, start and stop programs, and view log files through the command line. It also connects to the MM code repository and can download and compile code.

The commands are fairly simple and require very few arguments. Available commands include (Arguments surrounded by < > are always required and those surrounded by [ ] may be optional depending on the program):

- `mm help` to print help documentation.
- `mm log <PROGRAM>[DB_ENV] [NID] [MAIN_FUNCTION_NAME]` to view the whole desired log file.
- `mm logtail <PROGRAM>[DB_ENV] [NID] [MAIN_FUNCTION_NAME]` to view the desired log file as it is written.
- `mm restart <PROGRAM>[DB_ENV] [NID] [MAIN_FUNCTION_NAME]`

to restart a program.

- `mm run <PROGRAM>[DB_ENV] [NID] [MAIN_FUNCTION_NAME]` to run a program in the foreground.
- `mm start <PROGRAM>[DB_ENV] [NID] [MAIN_FUNCTION_NAME]` to run a program in the background.
- `mm startall` to start all programs that were stopped by the last `stopall`.
- `mm status` to show all of the programs currently running.
- `mm stop <PROGRAM>[DB_ENV] [NID] [MAIN_FUNCTION_NAME]` to stop a program.
- `mm stopall` to stop all running programs. Nothing else can be started or run until a `startall` command is given.
- `mm update` to update code from SVN repository, and compile it. Note that this command works on the entire SVN tree and compiles everything.

Where:

`mm`	=	accesses the **MM_MANAGER**
`PROGRAM`	=	program name
`DB_ENV`	=	database environment
`NID`	=	network ID (for running models)
`MAIN_FUNCTION_NAME`	=	name of main function

7.12. NETCONFIG

NETCONFIG is a very large module that is essential to the highway and arterial traffic models. It contains Java code representations of the road network and all its tangible and abstract components as a "Model Graph". The Model Graph represents all of the interconnected elements which the model needs to run, including their geometry through the `Coordinate` and `GeoMultiLine` provided by **CORE**.

Network objects are at the top of the hierarchy and are composed of `ModelGraphLink` and `ModelGraphNode` objects. The links represent roads, and nodes are points where links intersect, and both objects contain as fields many `HashMap`s which describe their relationships. E.g. a `ModelGraphNode` object contains maps of all links entering and exiting the node (according to direction of travel). Links and nodes can be grouped to form a `Route`: a possible way to traverse the network.

The Model Graph links and nodes are composed of links and nodes from the NAVTEQ NAVSTREETS database, and there are analogous classes representing them and their relationships as well. There are also objects which describe the network in terms of traffic flow as either `TrafficFlowSources` or `TrafficFlowSinks` and these are mapped to the links and nodes of the network. There are also objects representing sensors such as PeMS and toll tag readers, and there is class called `VTL` for the virtual trip lines used by the model to process GPS data.

7.13. PATH_INFERENCE

PATH_INFERENCE is written in Java and Scala. It implements a complex mathematical algorithm that estimates the path traveled between two GPS points. In an arterial road environment, there can be multiple possible routes from one point to another and analysis require a certain route to be assumed. The shortest path is usually correct, but this is not always true and there may be multiple paths of equal length possible.

The **PATH_INFERENCE** module is used for various GPS probe data sources, mostly notably

CABSPOTTING. **PATH_INFERENCE** interfaces with the Model Graph from **NETCONFIG**.

7.14. PEMS

The **PEMS** module encompasses all functions related to the acquisition, filtering, and storage of data from the Performance Measurement System (PeMS). California's PeMS is a statewide system of vehicle detector stations (VDS) where induction loop sensors are embedded in each lane of the road. Each VDS provides the count of vehicles passing over each lane's sensor (termed flow, expressed as vehicles per hour) and the proportion of time a vehicle was over each lanes sensor (occupancy). Though valuable, raw PeMS data contains a significant amount of erroneous and missing values due to sensor malfunctions. Furthermore, traffic speed cannot be directly measured; it must be estimated using the following formula:

$$v(t) = g(t) \times \frac{c(t)}{o(t) \times T}$$

Where:

$v(t)$ = Average speed during period T
$g(t)$ = Effective vehicle length (also known as g-factor)
$c(t)$ = Number of vehicles that passed over the sensor during period T
$o(t)$ = Proportion of time the detector sensed a vehicle present during period T
T = Time period under consideration

The **PEMS** module retrieves raw PeMS data from the Caltrans server and filters it. The PeMS filter replaces null or erroneous values with better estimates and improves speed estimation accuracy through using g-factors calculated per lane varying over time. Effective vehicle length (also known as g-factor) which must be assumed/estimated, introducing error into the estimation since vehicle length varies over time as the mix of vehicles changes (e.g. there are more large trucks on the road compared to cars during off-peak hours) and with the location of the sensor (even between lanes on the same road).

PEMS has three packages which are explained below.

7.14.1. rawFeed

The `rawFeed` package is concerned with retrieving PeMS data from Caltrans' server via FTP and storing it in the database. The feed is run from a class simply called `main` which runs a `main` method. The first task of `main` is to determine the environment in which the feed is being run (e.g. live, dev, etc.) and set accordingly the FTP access account credentials to be used, the

MM database to connect with, and the list files which will be requested. Both 30 second and aggregated 5 minute data are available, divided into separate files for each interval per Caltrans district.

Once this is complete, the program will execute the `main.do_pems` method every 30 seconds until killed. The connection to the PeMS server is accomplished through the `pemsFTPClient` class which has a single public method `pemsFTPClient.getFiles`. Upon a successful connection to the server[4], `pemsFTPClient.download` is called for each file required. The method locates the needed file on the server and downloads it, storing it in memory.

Each line of the file is parsed into either a `DatumRaw5Min` or `DatumRaw30Sec` object depending on the type of file. Both of these classes have a `store` method which validates and then inserts the datum, returning a Boolean to indicate success or failure.

7.14.2. filterParameters

This package contains code used to compute the parameters for the PeMS filter, most notably the g-factors. The `pemsFilterParameters` class computes g-factors for each sensor at each VDS on a weekly basis. The output of `pemsFilterParameters` is stored in a `gFactor` object which has the following fields:

- `date`: When parameters were computed.
- `pems_id`: ID number of the VDS.
- `statusForSensor`: Boolean array indicating which sensors are functioning.
- `gFactor`: Array of g-factors (effective vehicle lengths) for each sensor.
- `maxFlowForSensor`: Maximum flow value recorded for each sensor.
- `noLargeValues`: Number of times a “large” flow value is recorded at each sensor.
- `date_based_start`: Start date of raw PeMS data on which the computed g-factors are based.
- `date_based_end`: End date of raw PeMS data on which the computed g-factors are based.

The `pems_writer` class accesses the database to store filtered data and the updated g-factors and related parameters. It also retrieves all static information related to PeMS VDS locations (e.g. ID number, number of lanes, speed limit, etc.) and stores it in a `pems_prop` object which is kept in memory and used throughout the parameter calculation process.

7.14.3. filter

The purpose of the filter is to compute flow, speed, confidence and standard deviation using the

[4]In the event of a failure an `Exception` is thrown, and the feed completes its iteration and tries again in 30 seconds.

parameters computed by the `filterParameters` package on raw PeMS data as a continuously running process. Its main class is `ProcessorMain` which starts the program, initializes its database accessors, and provides high level methods such as `processData` and `writeFilteredDatum`. There is also a `HistoricalProcessorMain` which performs the computations on old data to backfill gaps when necessary.

The computations of flow, speed, confidence and standard deviation for each `FilteredDatum` are performed by separate classes:

- `FlowComputer`: This class computes an "adjusted flow" by replacing invalid values estimates based on the surrounding sensors and adjusts valid ones based on the estimated free flow speed at that VDS (uses code from the `SpeedComputer` and `Validator` classes).
- `SpeedComputer`: For validated raw data and g-factors (effective vehicle length), this class computes speed from flow and occupancy.
- `ConfidenceComputer`: Computes a confidence index based on the number of functioning detector loops at the VDS.
- `DeviationComputer`: Computes the speed standard deviation of each raw datum.

The `Validator` class contains a large set of constants which are also used the computer classes described above. It has only a single public method, `sensorTest`, which determines whether the flows and occupancies reported by a VDS are valid by performing a series of tests which compare the raw data against known thresholds for acceptability.

The `gFactor` class actually contains fields other than just the g-factors (and the time intervals on which they are based) for each sensor (lane) at a VDS; it also contains statuses and flow validity thresholds for each sensor. It is merely a container for these values; they are computed in the `filterParameters` package.

8. Visualization

8.1. Introduction

Traffic information is inextricably linked to spatial information, and it is infinitely easier for people to understand when presented in a spatial context, particularly as a familiar map. Very early in the *Mobile Millennium* project, the ability to display data on a map was identified as a crucial component of the system, and personnel were added to the team specifically to fulfill data visualization requirements.

Some specific benefits of a data mapping application —referred to often throughout the project and herein as "the visualizer"— include:

- Allowing researchers to inspect the output of traffic models in real time.
- Providing a platform for demonstrating the *Mobile Millennium* system to external parties, including the public.
- Permitting users to visually compare MM outputs against other data sources.

Satisfying these needs was crucial in order for the *Mobile Millennium* project be successful. This article describes the evolution of the MM visualizer over the course of the project, from early design concepts to the different incarnations that were eventually produced. The various design decisions made along the way are described, and the functions of the visualizer in its various incarnations are explored.

8.2. VIZ: The First Mobile Millennium Visualizer

8.2.1. Design Strategy

The development of the visualizer began in September 2008 under the name **VIZ**. It was around this time that highway and arterial traffic models began writing their output to an early version of the MM database, and a way to view the output in a meaningful way, such as on a map, became

a pressing need.

Both traffic models require a network of roads in order to run. Constructing these networks, composed of links[5] and nodes[6], was an early challenge of the project. Stored in the database was a copy of NAVTEQ's NAVSTREETS data for California, a very detailed and extensive geospatial dataset representing the entire road network.

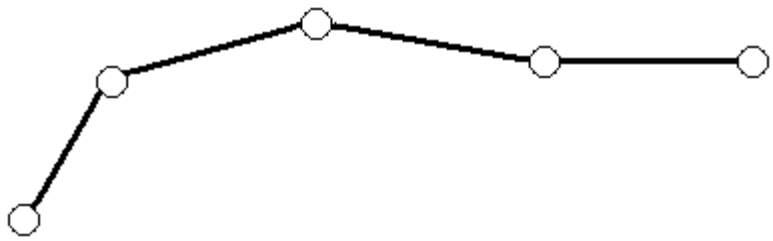


Figure 2.1. A link is an ordered series of two or more points that represent the geometry of a road segment, geometrically known as a "linestring".

Early attempts were made to run the models using a network simply constructed from a subset of the NAVSTREETS dataset, but this approach proved to be unsuitable. Instead, researchers created their own system of links and nodes by modifying the original NAVSTREETS data (e.g. fusing two links together, splitting bidirectional links into two unidirectional ones) and designing networks using the modified links and nodes. This task was extremely difficult without a way to see the actual network overlaid on a map. Free and commercial GIS (Geographical Information System) software were tried, but proved to be ultimately insufficient. Researchers needed mapping software that could convert the modified NAVSTREETS links and nodes into objects meaningful to the models, and display the network fused with model output.

Once the need to develop an in-house visualizer was established, design meetings were held between researchers and developers to determine what features **VIZ** should have. First and foremost, the **VIZ** needed to be easily accessible by team members regardless of their location or the computer they were using, particularly in the case of demonstrations at conferences, meeting, media events, etc. Therefore, it was decided that **VIZ** should be a web application, requiring no software downloading or installation.

[5]A link is an ordered series of two or more points that represent the geometry of a road segment, geometrically known as a "linestring"
[6]A node, aka junction is a point where one or more links intersect.

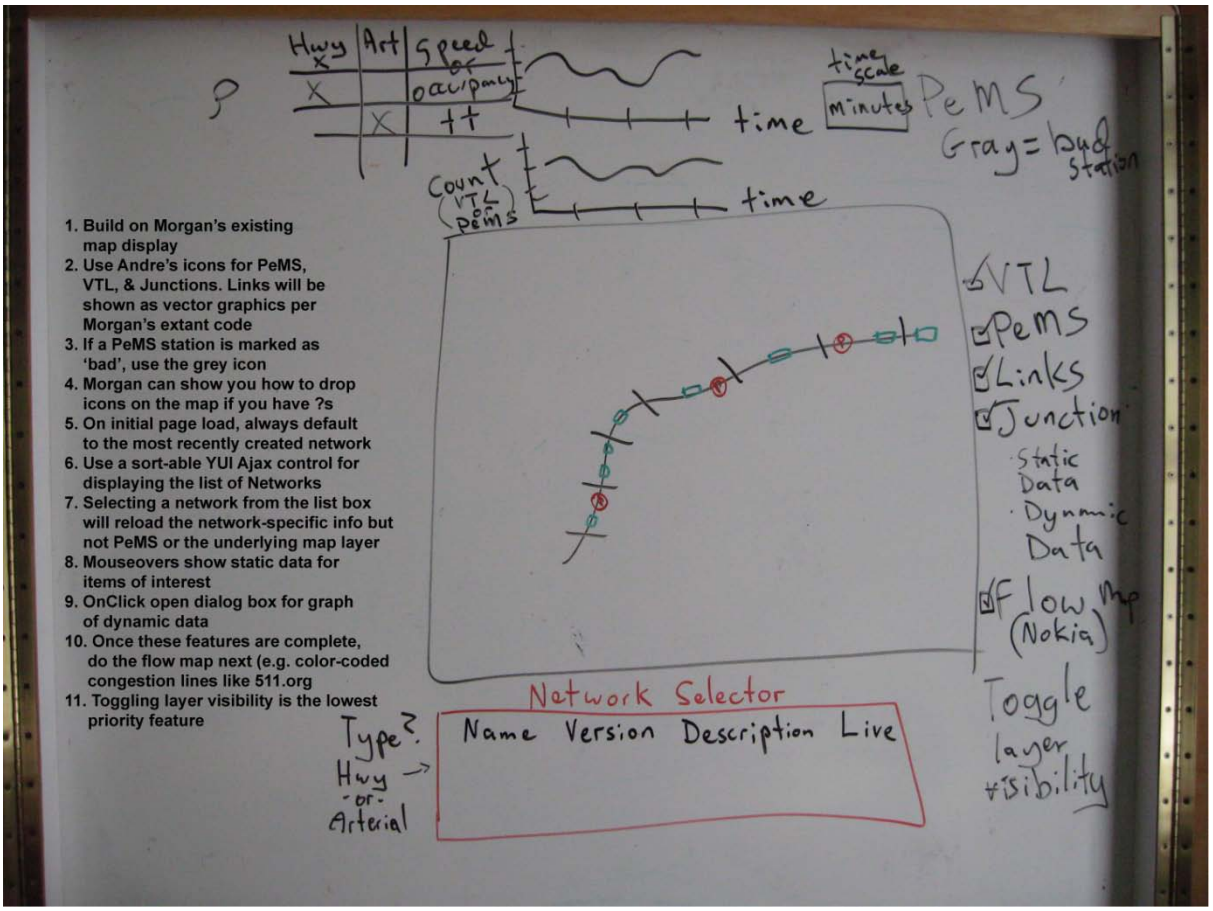


Figure 2.2. Early VIZ design.

The other requirements the team identified were focused on what data sources would initially be included, how features would appear on the map, and what controls would be available to users through the GUI. The product requirements were sorted by priority and from this a development plan was agreed upon. The main requirements included:

- PeMS station locations must be displayed as points. The color of the point should indicate both the current speed and whether or not the station is active/valid. Clicking on any PeMS station would display additional information about that station and the road on which it was situated.
- Roads should be plotted as "model links" as opposed to simple modified NAVSTREETS links, and they should, where applicable, be fused with highway or arterial model output. The links should be colored to give an indication of traffic speed and when clicked on provide additional details about the road and the model's current state (e.g. how recent is the speed estimate, confidence value, etc.). Model links should be aggregated or subdivided depending on the situation[7].

[7]At the time, the arterial model computed estimates based on combined model links called "superlinks". Conversely, the highway model computed estimates for fractions of a model link.

- Virtual Trip Lines (VTLs) should be plotted on the map and colored according to speed.
- Networks and other data sources should be selectable from a GUI.
- The links and junctions of a network should be plotted as separate layers.

In the early phase of the *Mobile Millennium* project, PATH was assisted by external contractors with experience in building real-time traffic data systems. The contractors provided advice regarding options for designing and implementing such a system, and they also wrote some simple code that demonstrated the concept as a simple web application running as a Tomcat servlet[8].

The most complicated part of **VIZ** is the mapping engine itself: the code that generates a map with which the user can interact, on which data is plotted. Developing this from scratch was not deemed practical, so OpenLayers was chosen instead. OpenLayers is a widely used, open source JavaScript mapping engine for web pages. It provides JS classes and methods for creating a map widget for a web page that can display vector features (and some other data types) on top of map tiles from virtually any source. However, additional programming is required to control the map widget and convert user data to a format that OpenLayers can use.

A model link divided into equal length segments for which the highway model computed individual speed estimates was called a "discretized link".

[8]Apache Tomcat is a servlet container that provides an HTTP web server environment in which Java code can run.

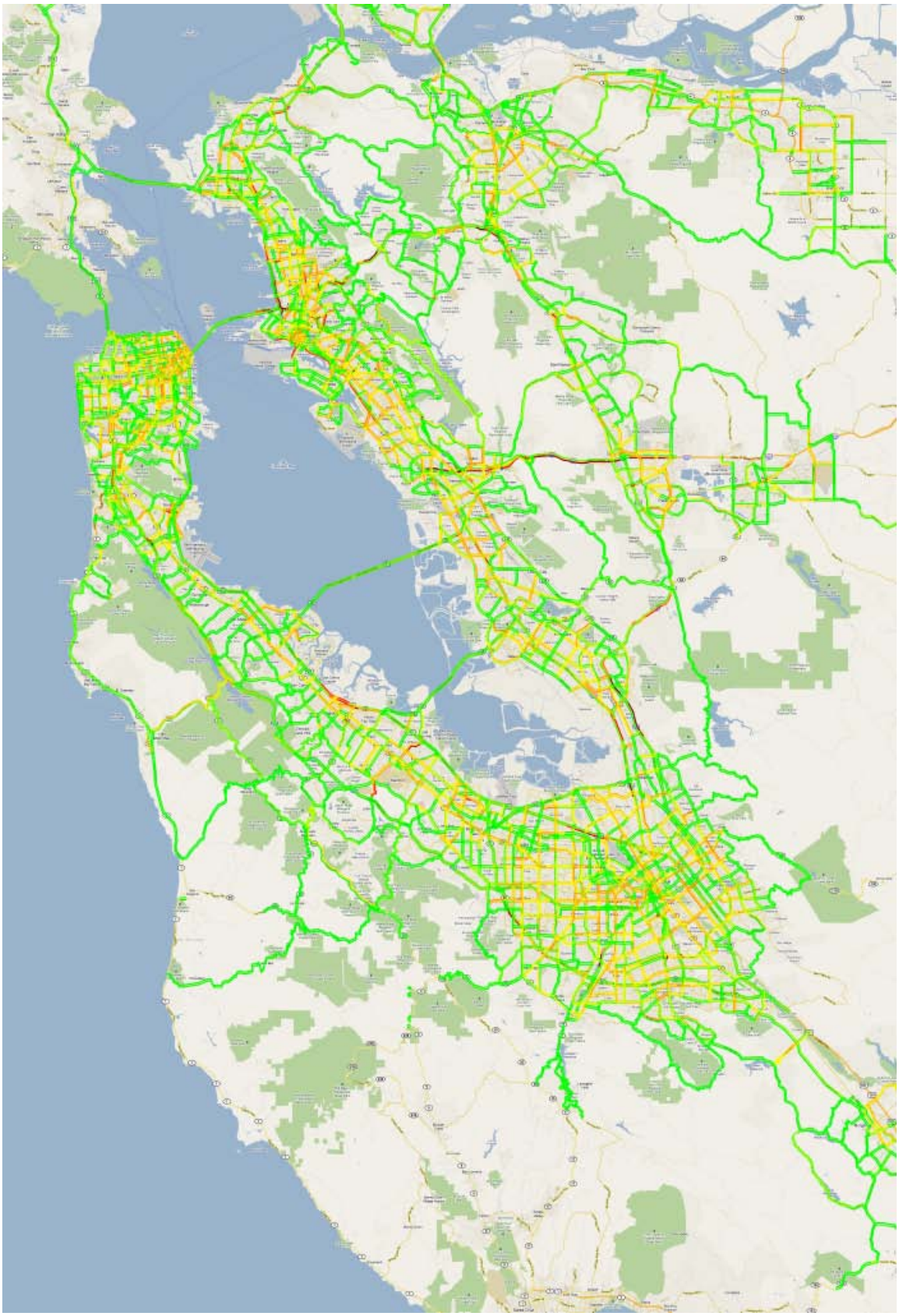


Figure 2.3. Mosaic of Bay Area highways and arterials from VIZ.

Getting the user data into a web page requires interaction between the client and the server. Web applications typically use AJAX[9] to accomplish this. AJAX programming can be challenging, so a third-party library called DWR (Direct Web Remoting) was employed. DWR automatically generates JavaScript that allows the web browser to call Java code and marshals data from the server to the client so that it is accessible through JavaScript.

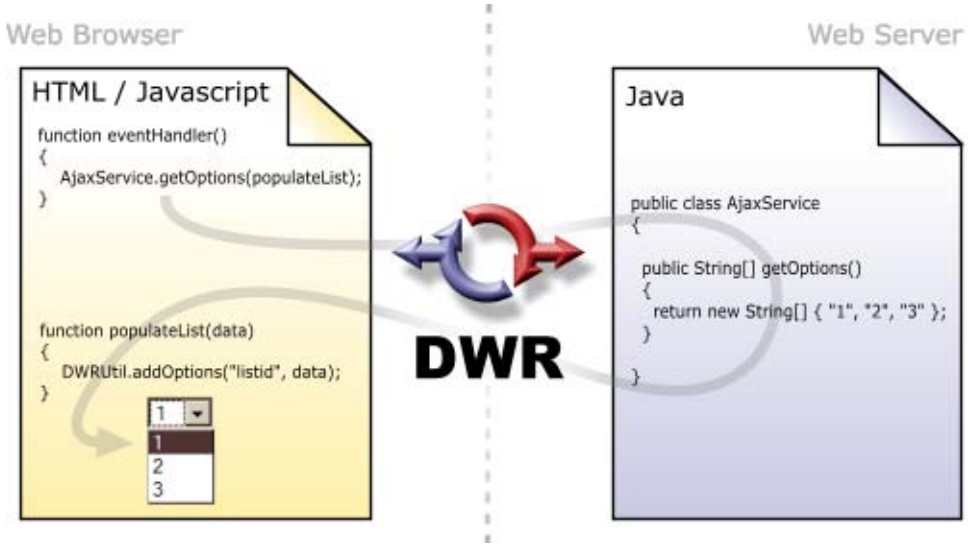


Figure 2.4. AJAX through Direct Web Remoting.

On the server side, Java code was written to accept requests for map layers from the client(s), retrieve the relevant data, format it into layers, and send it back to the client.

In order to understand how **VIZ** works; it is best to consider it as two separate pieces of software, the server and the client, that interact through AJAX. The following sections provide a detailed description of the **VIZ** software on the server and client sides. Although **VIZ** has been replaced with a new incarnation, **DIVA** (discussed later in this chapter), much of it still applies. The following subsections describe the architecture of **VIZ**.

8.2.2. Server Code

The main class in **VIZ** is `GeoDataBean` which is responsible for accepting requests from the client for map layer data, creating the layer and sending it back to the client. Through DWR, the client calls `getDrawLayer` and supplies the name of the requested layer and geographic bounds of the map view. The `GeoDataBean` calls its `getFeatures` method which forms and executes an SQL query, processes the result set and returns an array of features that can be

[9]Asynchronous JavaScript and XML

loaded into a`VectorLayer`.

The `VectorLayer` object contains an `ArrayList` of features, the name of the layer, and the names of attributes for the feature type. The `getDrawLayer` method in `GeoDataBean` returns the `VectorLayer` to the client through DWR. There is also a method in `GeoDataBean` called `getNetworkSelections` which returns the list of layers available from the system in a `NetworkSelections` object.

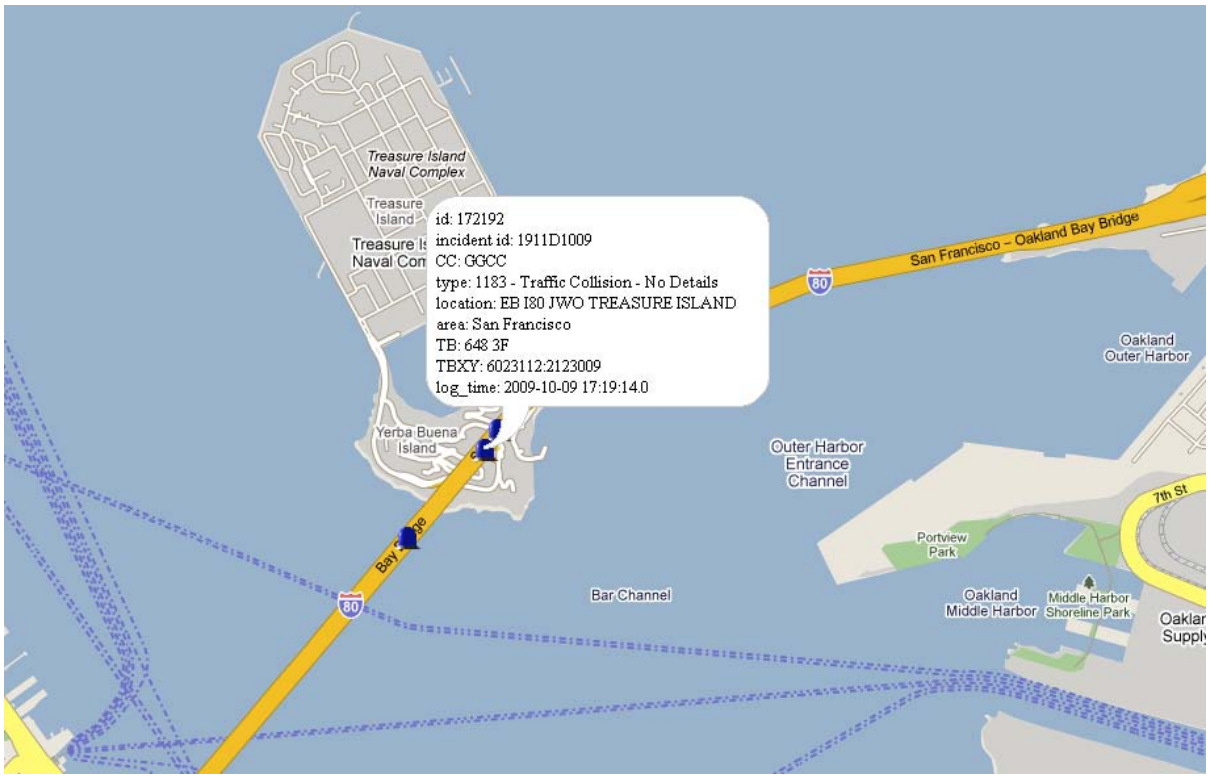


Figure 2.5. VIZ showing traffic incidents reported by California Highway Patrol.

Feature classes represent an object that will be displayed on the map, such as a road link or PeMS vehicle detector station. All feature classes are subclasses of `FeatureBase` and implement the `OverlayFeature` interface. `FeatureBase` is an object that contains a set of `GeoPoint`s describing its geometry and location and a map of attribute names and values. `FeatureBase` offers basic methods for accessing and manipulating this information.

The `OverlayFeature` interface requires methods for defining the attributes and geometry of a feature, as well as a `getSQL` for retrieving the SQL query for a particular feature. These methods are called in `GeoDataBean`'s `getFeatures` method for the specific type of feature being requested.

The `GeoDataBean` uses the `OverlayFeatureFactory` to get the correct feature type. The name of the requested layer is passed to the `OverlayFactory` and is parsed. For example, if

the layer name is “37_SanFrancisco_arterial_links” the factory will return an `ArterialLink` object, and the `VectorLayer` will be populated with `ArterialLink`s.

Here are some of the features available in **VIZ**:

- `ArterialLink`: Links from arterial networks colored according to estimated speed. Green indicates free flow and turns yellow, red and then black with decreasing speed. Contains other attribute information such as the link ID, estimate timestamp, confidence, etc.
- `HighwayLink`: Highway links similar to the `ArterialLink` class described above.
- `DiscretizedHighwayLink`: A `DiscretizedHighwayLink` is a `HighwayLink` that has been further divided into “sublinks” giving speed estimates of a higher spatial resolution.
- `CabDensity`: Shows the amount of taxi cab data (Cabspotting) per link.
- `Incident`: Shows recent traffic incidents reported by the California Highway Patrol. Users can click the icons to get information about the incident.
- `PemsStation`: A point showing the location of a PeMS vehicle detector station. The point is colored according to the most recent traffic speed, and users can click the point to see more information.
- `RadarDetector`: Displays radar speed detectors in a manner similar to `PemsStation`.
- `Vtl`: Shows the locations of Virtual Trip Lines and colors them according to their most recently reported speeds.
- `VtlDensity`: Shows the VTLs, but colors them according to the number of records.

The geometry of each feature is stored as a list of one or more `GeoPoints`, which are a basic 2D point class with some functions for spherical geometry. Some feature classes will have methods to adjust their geometry to improve the appearance of the feature on the map. For example, at a wide zoom road links adjacent to each other will be overlaid on top of each other due to their relative proximity. Therefore, it is necessary to implement a `spread` method which will artificially shift the vectors apart based on the zoom level.

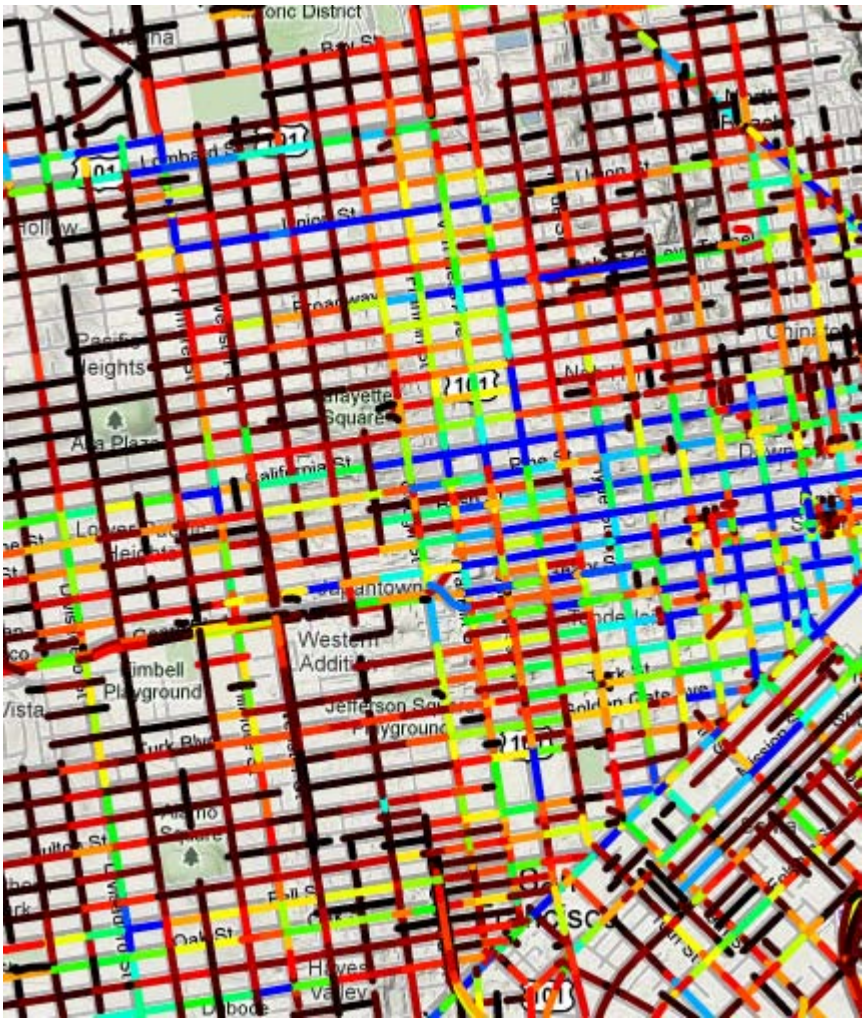


Figure 2.6. VIZ showing density of taxi cab data in San Francisco indicated by link color. Black/red indicates few records, green/blue indicates many records.

8.2.3. Client

The client side consists of JSP web pages and several JavaScript files, the most important of those being `OpenLayers.js`. OpenLayers is a free, open source mapping engine for web pages. However, additional programming is required to connect to the server and translate its response into features OpenLayers can use.

The structure of the **VIZ** client code is not very organized and difficult to read and modify. This was a major factor that motivated the redesign of **VIZ** into **DIVA**. Almost all of the necessary functions were contained in a single file called `MapMain.js`. Here are its most important functions:

- `initMap`: Calls OpenLayers functions to create the map on the page and add controls depending on the version of **VIZ**.

- `initOverlays`: Examines which layers have been selected in the GUI and sends requests to the server for those layers.
- `handleGeoData`: This is the main function of the client which accepts the response from the server, translates it into features and layers, and draws them on the map.
- `onFeatureUnselect`: Upon clicking on a feature, this function generates a popup containing its attribute information.
- `initSelector`: This function initializes the GUI by sending a request to the server for a list of available layers and then displays the list in the GUI with checkboxes the user can use to select them.

In addition to these, there are a multitude of functions used for creating and deleting layers and popups, and for managing the map. Some functions were moved to other JavaScript files:

- `Discretized.js`: Contains special functions for drawing discretized highway links, which are normal highway links divided into smaller ones each with an individual speed.
- `Routing.js`: Contains functions that support routing. The user can select the start and end points by clicking the map, send the coordinates to the server and plot the resulting route.
- `Styling.js`: Contains methods that affect the appearance of features, such as coloring features based on speed or some other attribute and setting size and other qualities.
- `Time.js`: Contains functions that deal with time, mainly converting between numerical time stamps and readable strings, but also determining if the system should be running based on time of day.
- `Togglers.js`: Contains functions that toggle the values of Boolean variables. This is mainly used for controlling the GUI.

JSP pages contained the map and its GUI controls. The `index.jsp` page forced users to enter a user name and password in order to access the application. `Auth.jsp` checked the password entered by the user against a correct one on the server, and then, if correct, forwarded the user to either `mapmain.jsp` (the internal version) or `citris.jsp` (for the CITRIS Tech Museum) depending on the user name entered.

8.2.4. Results

The first major milestone was to provide a working visualizer in time for the 2008 Intelligent Transport Systems World Congress held November 16–30 in New York, NY. This event was the first public demonstration of *Mobile Millennium* system; arrangements were made for 20 cars equipped with Nokia phones running the MM client to drive a loop in Manhattan during the conference. The models would run in real-time and the results would be shown live on a display at the *Mobile Millennium* booth using **VIZ**.

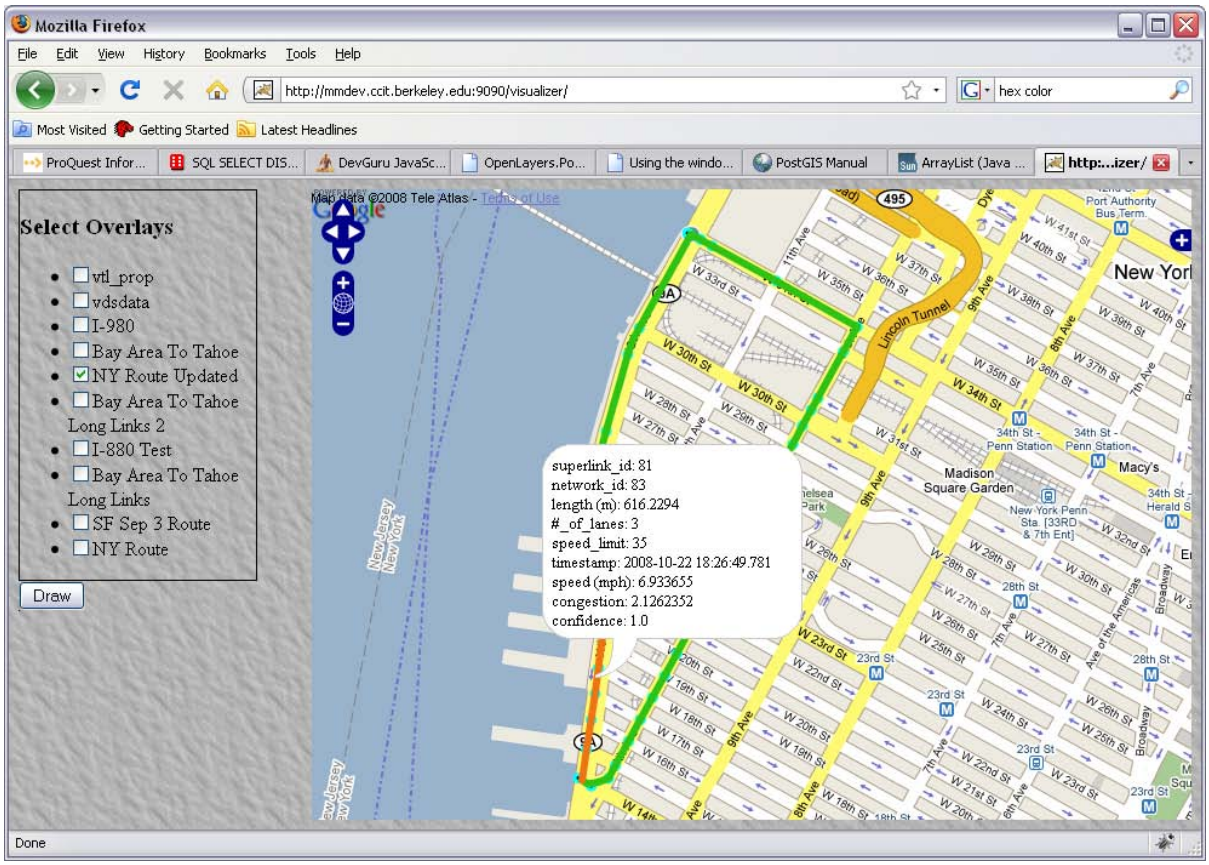


Figure 2.7. Screenshot of VIZ during the 2008 ITS World Congress in New York, NY. Arterial model is running and shows congestion on the orange link.

This demonstration was very successful. As the probes made their way around the road loop, the *Mobile Millennium* exhibit showed the results of the arterial model in real-time, with links changing color from green to red when congestion was detected. This was well-received by the news media and other observers. Simultaneously, the systems team in Berkeley used **VIZ** to monitor the experiment. This demonstrated the advantage of the web application design choice; it permitted users in multiple locations to access the software easily.

After this milestone, **VIZ** development continued. More feature types representing various data types were added, as were special features based on requests from researchers. A significant amount of time was spent improving the **VIZ** speed and reliability, particularly as it pertained to loading and processing data from the database efficiently. By improving the load time, it was possible to load and reload data over a large area to capture changing traffic conditions.

Over the course of the project, **VIZ** was an integral part of several successful experiments and demonstrations using real drivers by showing results in real time. **VIZ** also accelerated traffic model development since researchers could adjust their models based on the output they saw through **VIZ** and identify and fix bugs. For example, as links were colored based on speed it was easy to visually identify links with abnormally high or low speed estimate and investigate those

links. **VIZ** was an essential part of the *Mobile Millennium* system, both as a research tool for model development and as a public relations tool through demonstrations and images provided for promotional materials and academic papers.

8.3. CITRIS VIZ

In March 2009 was the opening of the new headquarters of CITRIS (Center for Information Technology Research in the Interest of Society) in Sutardja Dai Hall on the UC Berkeley campus. The new headquarters includes a "Tech Museum" which showcases various UC projects. As *Mobile Millennium* was a very high profile project, the museum was to include an interactive MM exhibit.

In addition to a download station where the public could sign up and download the MM phone client, the exhibit was to include a live traffic map that could be manipulated by the through a touch sensitive device. During the first quarter of 2009, **VIZ** development focused on finding a physical platform for the exhibit and adapting the **VIZ** code to meet the needs of the exhibit.

The concept for the CITRIS **VIZ** was a display that included a large map showing live traffic conditions as links colored according to speed. The user can pan the map and interact with other controls using their finger. The map should refresh automatically. Occupying a smaller portion of the screen would be an interactive multimedia presentation. The CITRIS **VIZ** is not separate from **VIZ**; it is merely a different mode within **VIZ** that loads a different web page.

The project team, including CITRIS personnel designing the museum, evaluated a few different platforms for the touch screen. This process included a visit to Microsoft to evaluate the Microsoft Surface. The Surface is an advanced, multi-touch operated table display aimed at commercial applications. However, the high cost (approximately $15000) and the fact that a completely new visualization application would have to be written from scratch within a short time frame made this solution unfeasible.

Figure 3.1. The Panasonic TY-TP50P10S touch sensor mounted on a monitor. Image source: http://www.plazmy.pl/img/more_expimg27.gif

Instead, a simpler type of touch sensor was chosen. The Panasonic TY-TP50P10S is a touch sensor that attaches to the Panasonic TH-50PF11 50-inch plasma display and enables a single touch control when connected to a computer running the device software. It creates an infrared plane around the display that detects the presence and position of a finger or wand and reflects its

motion through the behavior of the mouse cursor. Straight out of the box, the TY-TP50P10S proved to be an effective controller when connected to a PC running the **VIZ** application. Adjustments to the driver software settings further improved its accuracy and performance.

The touch sensor hardware chosen made it possible to use much of the code from **VIZ** with minimal modifications, although the CITRIS **VIZ** had a few stark differences from the internal **VIZ**. The highway and arterial roads are expressed as instances of `CitrisHighwayLink` and `CitrisArterialLink` which are subclasses of the respective classes used in the internal **VIZ**. Their main difference is that they contain no attribute information and their color values are computed on the server instead of the client to avoid the need to pass any attributes such as "speed" which would otherwise be used in the color calculation on the client. Minimizing the size of the data sent to the client to maximize performance is the reason for doing this; each feature contains only an array of points for the geometry and a hexadecimal color string.

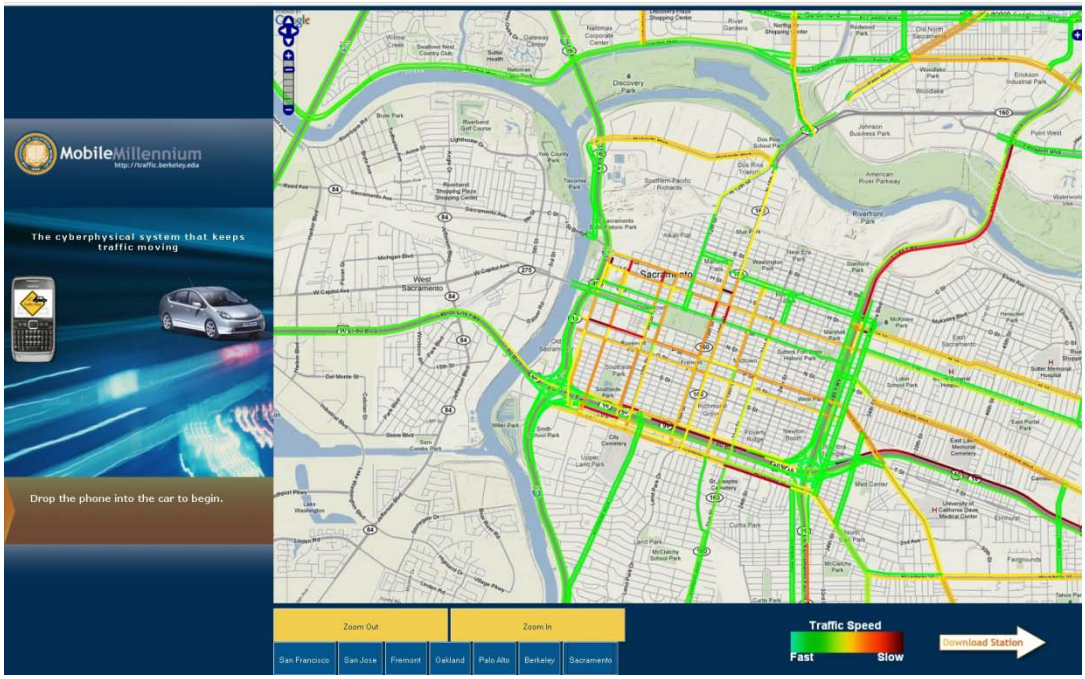


Figure 3.2. The CITRIS VIZ shows arterial and highway traffic conditions estimated by the models in Sacramento.

Most of the customization occurred on the client side. Unlike the internal **VIZ**, the CITRIS version does not require the user to select which layers to load; a layer for a Bay Area highways and a layer for each county's arterial model are the only possible layers, and they are loaded automatically every time the map is moved. In order to avoid trying to load too much data at once, the zoom level is used to determine which layers are loaded (i.e. if the area viewed is too large, arterials and possibly highways will be omitted from the request sent to the server). A message appears telling users to zoom in if they want to see more roads.

Figure 3.3. The *Mobile Millennium* exhibit at the CITRIS Tech Museum, shown during a visit from Valerie Pecresse, the Secretary of Higher Education in France. The CITRIS VIZ is mounted on the wall on the left.

Another special feature of the CITRIS **VIZ** was an interactive slide show that appeared on the left pane of the display which described the benefits of *Mobile Millennium*. Users were invited to perform various actions using the touchscreen (e.g. dragging an image of a mobile phone onto an image of a car) to advance the presentation which also included music and a voice-over. This was created in collaboration with another unit on campus.

The CITRIS **VIZ** and touch sensor software run on a Mac Mini computer tucked carefully out of view. As the computer was not especially powerful, a significant amount of time was invested in optimizing the CITRIS **VIZ** to improve speed and reliability. This mainly involved minimizing the size of the data transferred from the server and improving the way CITRIS **VIZ** handled unexpected events such as timeouts when waiting for a layer that never comes.

In the two years (of this writing) since its release, the CITRIS visualizer has served as a very successful public relations tool for the project. It is a very popular exhibit and is often demonstrated for visiting dignitaries and news media.

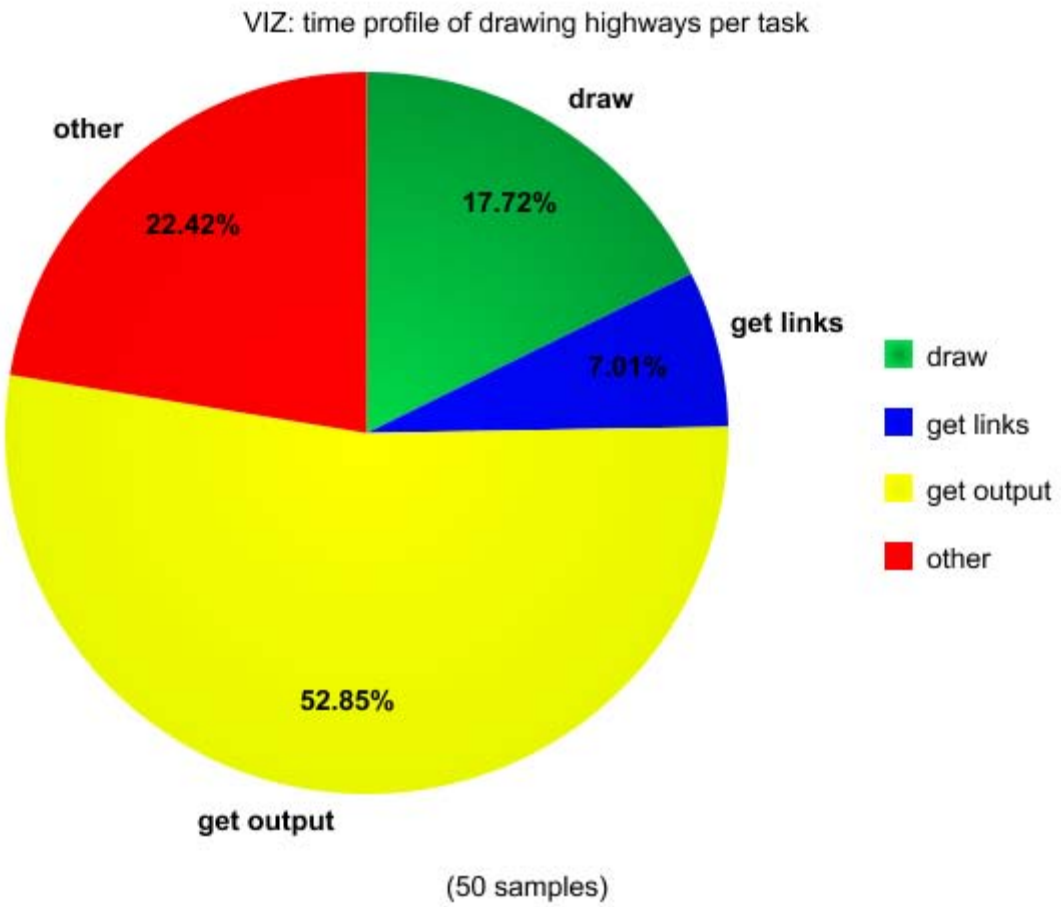


Figure 3.4. The breakdown of time per task when requesting a layer from the server and drawing it on the map. In this case, over half the time was spent on "get output", i.e. obtaining the speeds from the highway model and fusing them with the links.

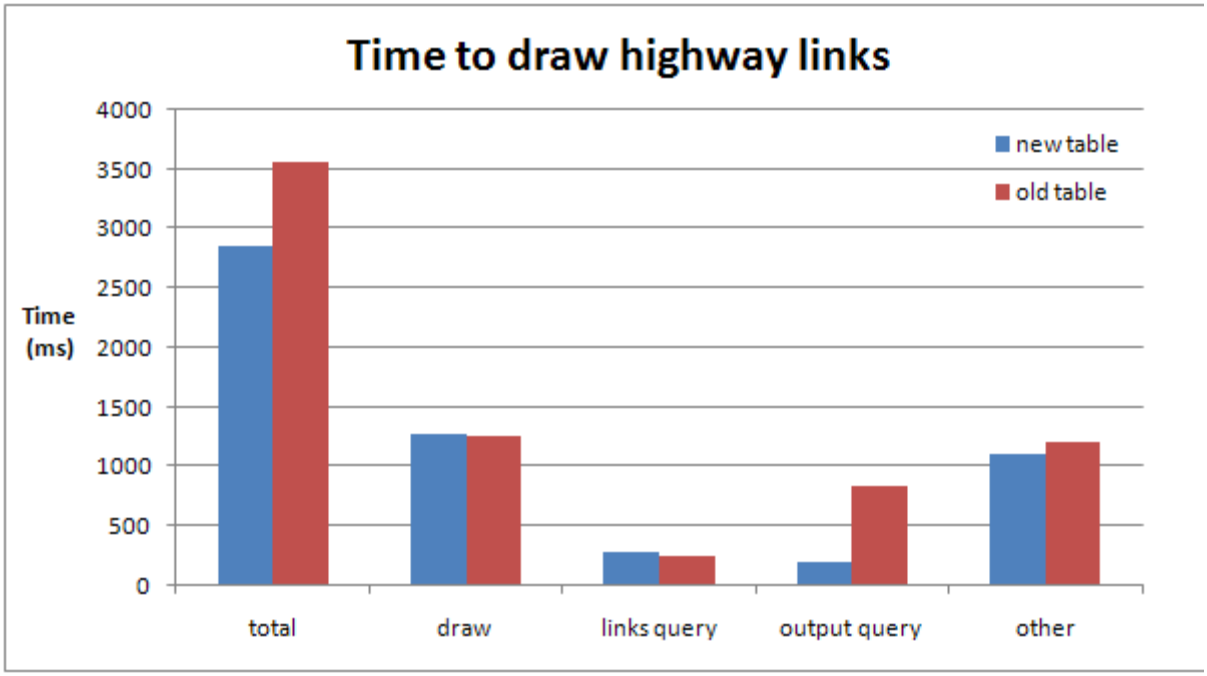


Figure 3.5. This graph shows how a change to the highway model output table improved the speed of VIZ.

8.4. Special Applications

8.4.1. *Mobile Millennium* Website

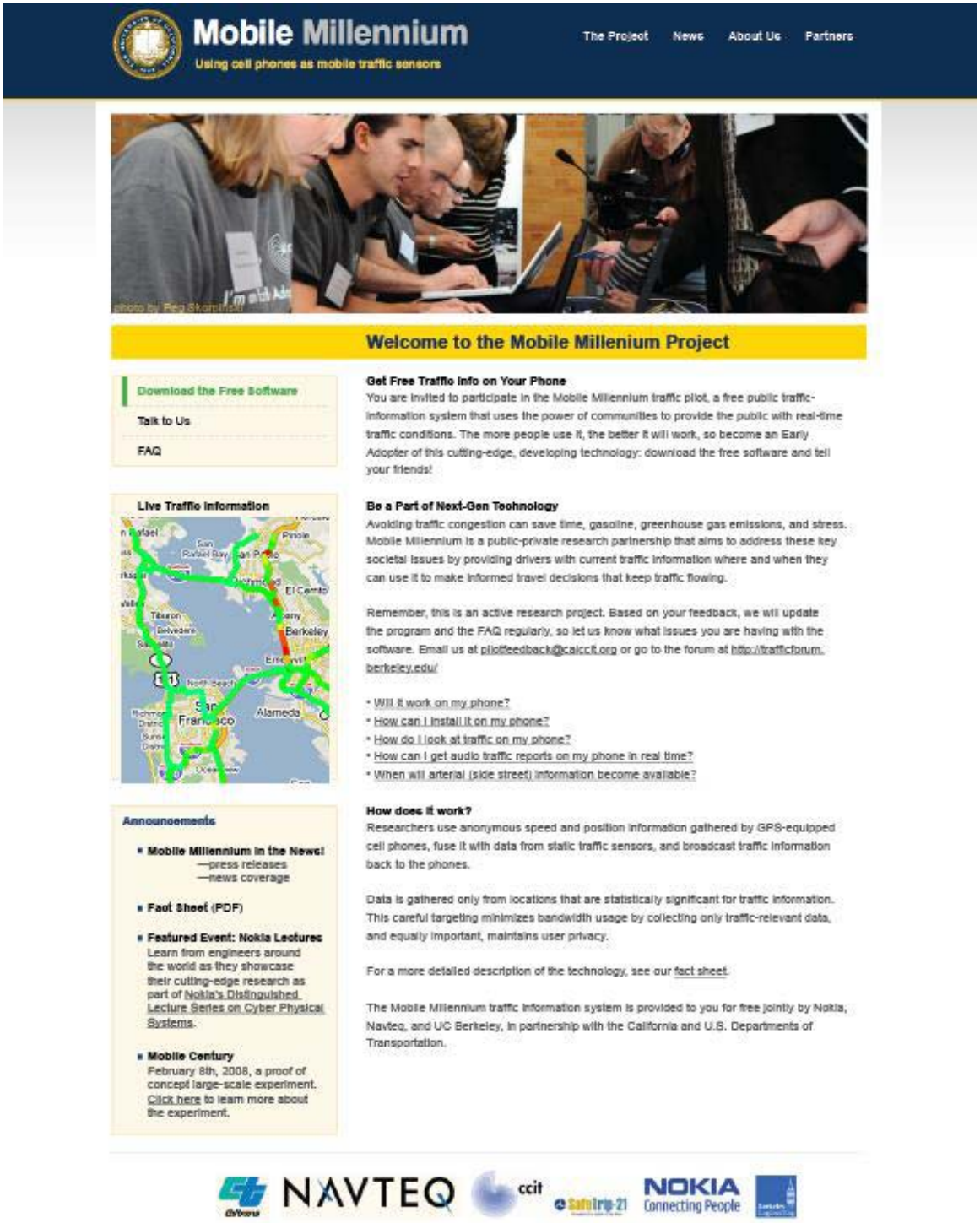


Figure 4.1. The *Mobile Millennium* with live traffic conditions.

The **VIZ** was also incorporated into the official *Mobile Millennium* website. As of this writing, the website is no longer online, but during the project its front page featured a live traffic map image of San Francisco and the surrounding areas (highway model only). A new traffic map was produced every minute by a computer running a special version of **VIZ**.

The computer had **VIZ** running in a browser window. **VIZ** would refresh every minute and a special process would take a screenshot of the browser window, trim the image down to the desired dimensions for the website map, and save it to disk. The machine also ran Apache HTTP Server which hosted the map image. The server hosting the MM website pulled a copy of the map image every time a new one was produced and displayed on the web page.

8.4.2. AASHTO

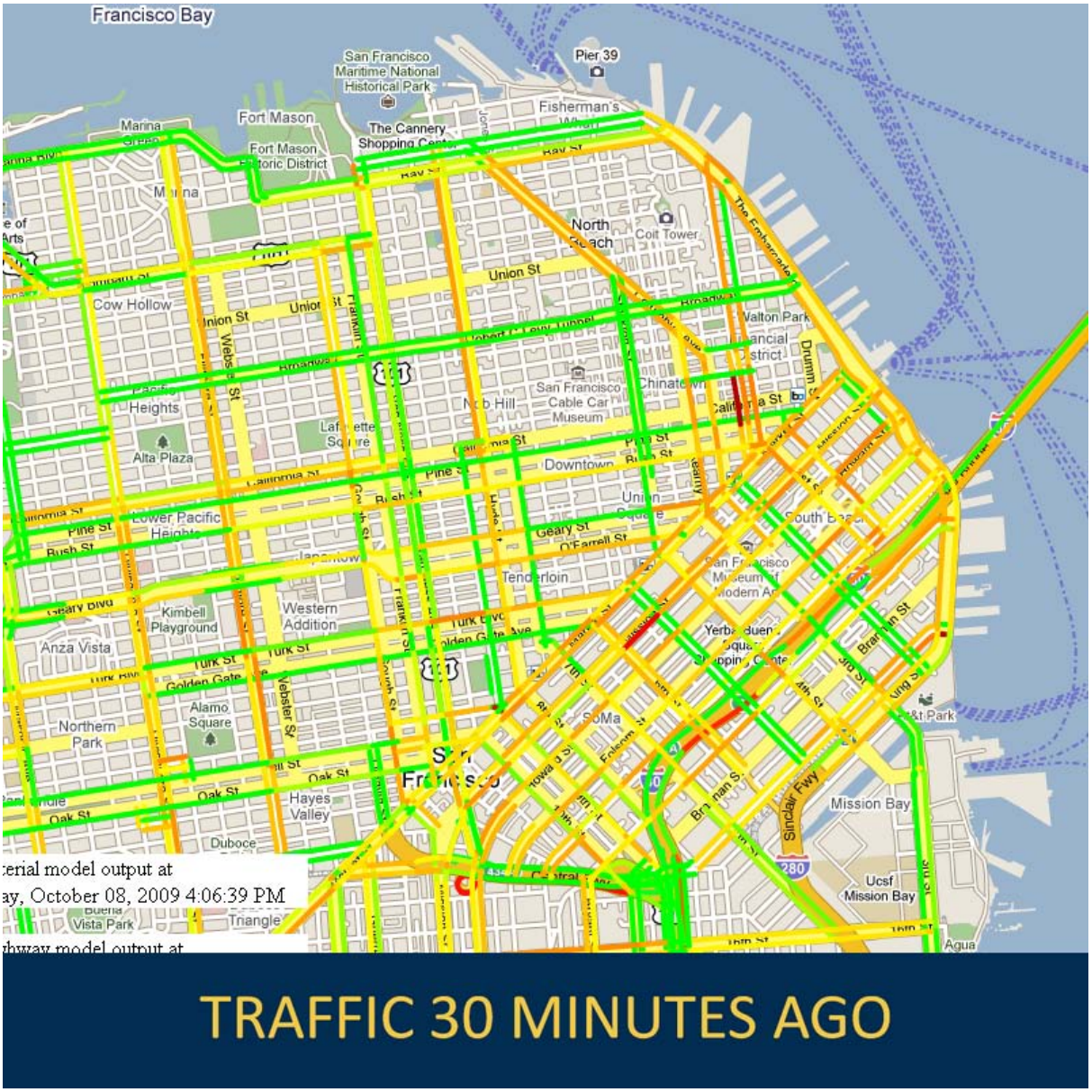


Figure 4.2. A frame from the AASHTO animation.

Based on the success of the screen capture process described above, a more ambitious demonstration was planned for 2009 Annual Meeting of AASHTO[10] in Palm Desert, CA. PATH attended this meeting and brought its CITRIS **VIZ** display. However, a second monitor showed a live animation of the arterial traffic conditions in San Francisco over the past 30 minutes.

This was accomplished by using the same screenshot function used to generate maps for the MM

[10]American Association of State Highway and Transportation Officials

website. A program would take a screenshot every 60 seconds of the **VIZ** showing the arterial model output in San Francisco and keep a running list of the last 30 screenshots. Every time a new screenshot was taken, the 30 screenshots were combined into an animation showing the arterials over the past 30 minutes. This all happened back at PATH on a machine dedicated for that purpose. The computer at the AASHTO meeting pulled the image every minute from PATH's server.

8.4.3. Routing Research

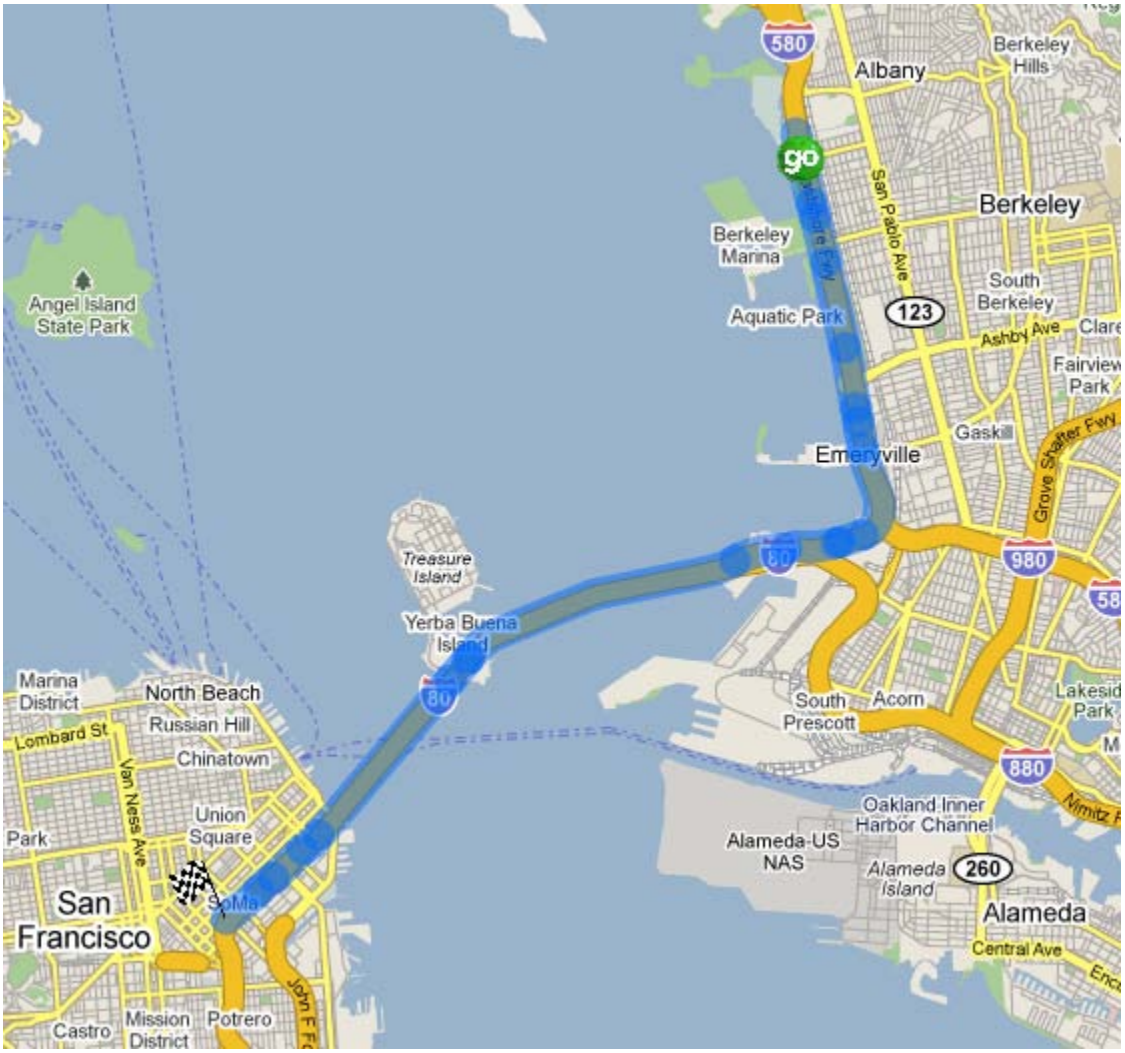


Figure 4.3. VIZ's routing application.

Routing is a very promising research topic, and **VIZ** was adapted to support the efforts of the *Mobile Millennium* in this area. Researchers developed a routing algorithm that was incorporated into **VIZ**. By clicking the map, the user selected a starting point and destination. Their coordinates are sent to the server which sends back a layer containing links of the route.

8.4.4. Drifter Project

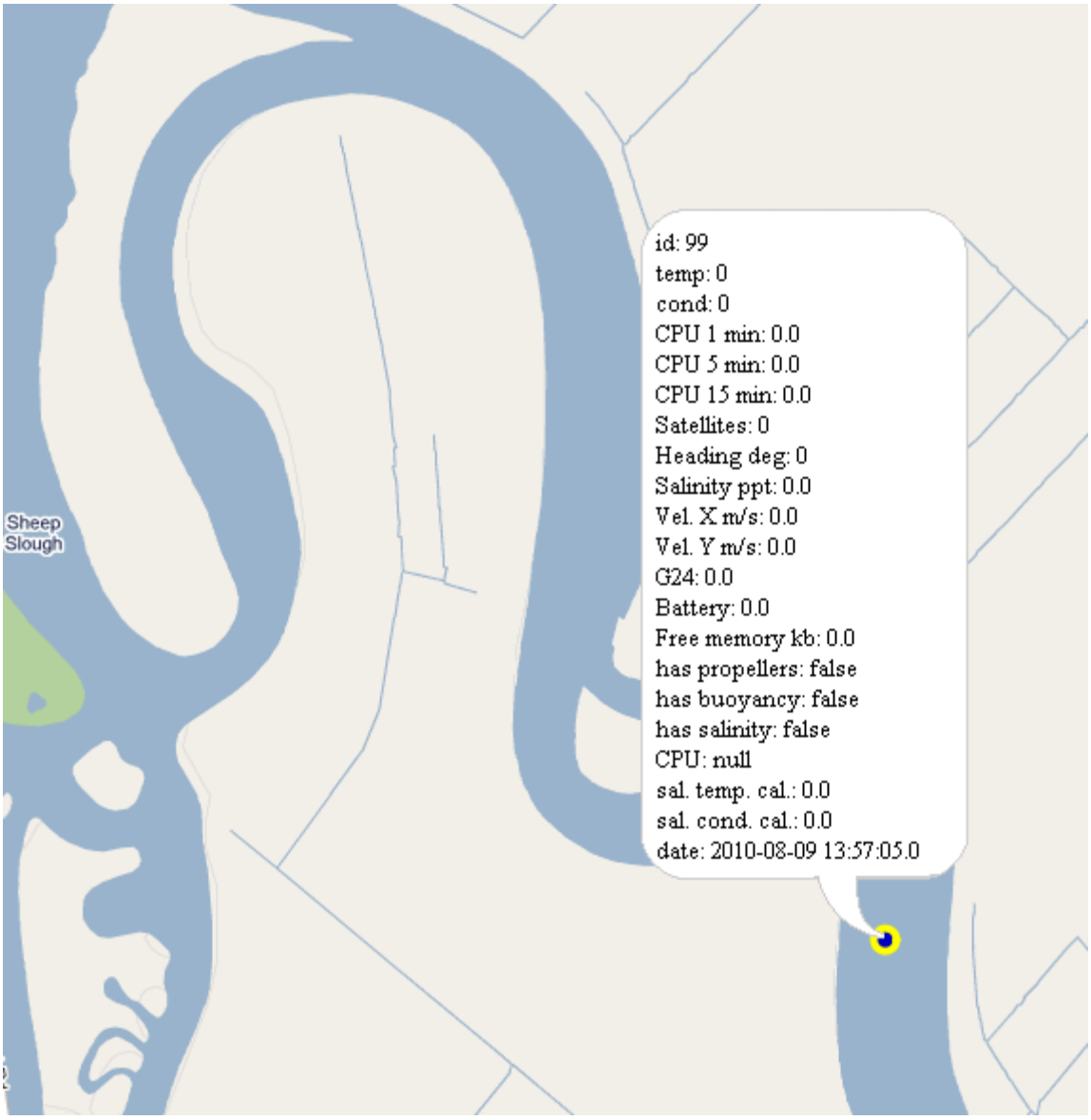


Figure 4.4. Location and status of a Drifter robot shown in FLO_VIZ.

The Floating Sensor Network project involves mobile, portable, floating sensors known as Drifters that are used to model water quality and flow. The project was a joint effort between Lagrangian Sensor Systems Laboratory (LSSL) at UC Berkeley, Lawrence Berkeley National Laboratories (LBNL), the National Science Foundation, the California Bay Delta Authority, and the California Department of Water Resources. PATH provided system support

FLO_VIZ was a version of **VIZ** developed to support this project. As the Drifters report their location and status in real time, **FLO_VIZ** was very useful for monitoring live experiments in the Sacramento-San Joaquin River Delta. **FLO_VIZ** also displayed the results of the model in

2D cells. **FLO_VIZ**'s functions have been implemented in **DIVA**.

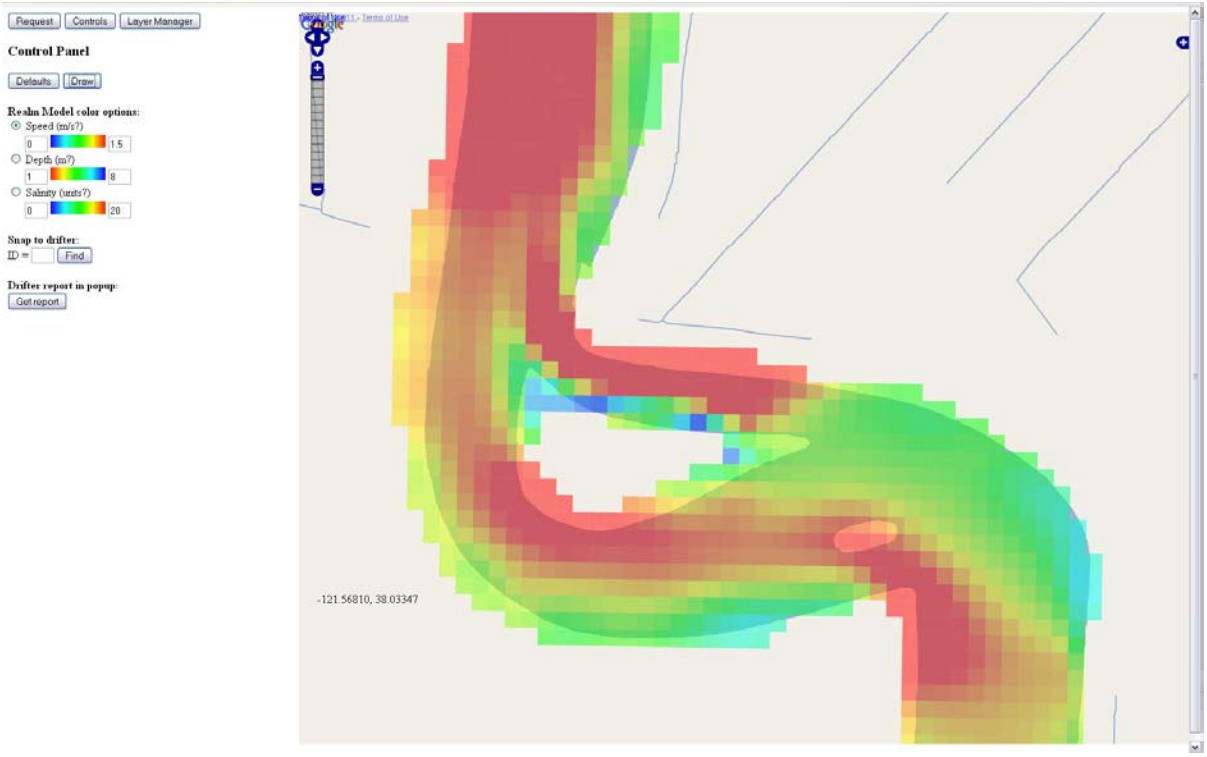


Figure 4.5. Realm flow model shown in FLO_VIZ.

8.4.5. ClearSky Project

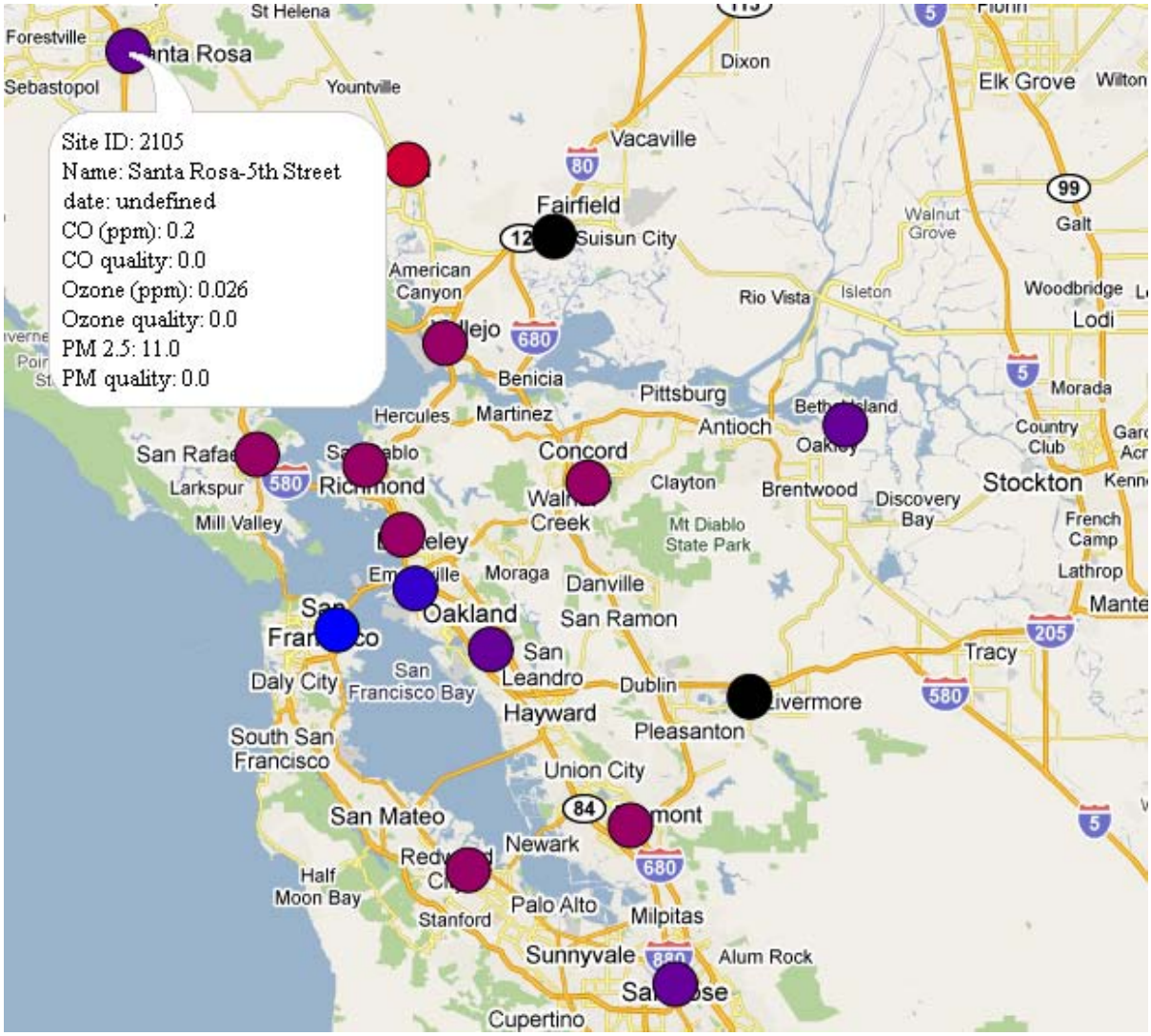


Figure 4.6. Bay Area air quality sensors shown in SKY_VIZ.

ClearSky was the name of a collaboration between PATH, UC Berkeley's School of Public Health, Nokia, and the Department of Civil and Environmental Engineering. The concept of ClearSky could be described succinctly as "*Mobile Millennium* for air quality" since it involved using data from multiple sources–including stationary and new mobile sensors– to power modeling of air quality.

The team developed prototype for a small, inexpensive sensor that detects air pollutant concentrations and can be attached to vehicles. Models for vehicular emissions and pollution dispersion were developed that could use this data in conjunction with existing air quality and weather data sources.

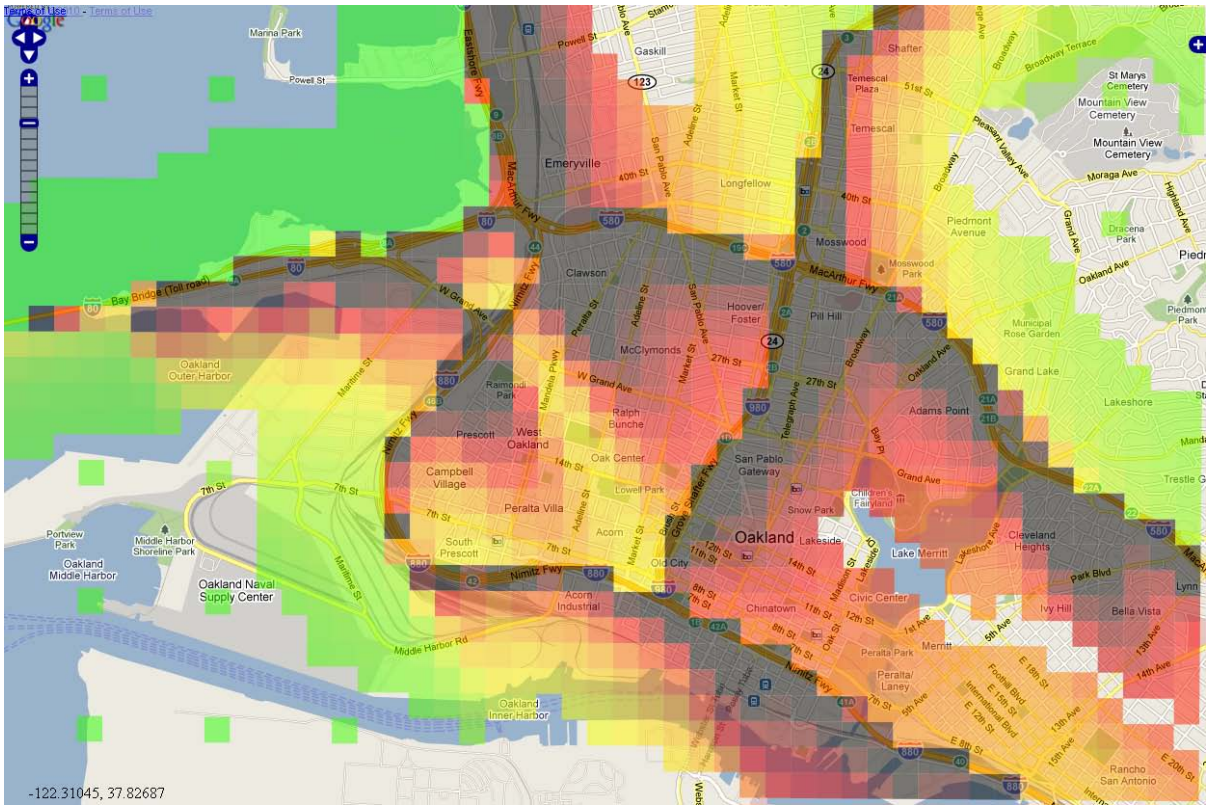


Figure 4.7. Air pollution dispersion model shown in SKY_VIZ.

As with *Mobile Millennium*, there was a pressing need to visualize the data in the system, both to accelerate development and demonstrate the system. **SKY_VIZ** was a web application based on the original **VIZ** that connected to the database, constructed features and displayed the on the map. The available layers included stationary sensors from the California Air Resources Board, Google Weather, roadway emissions modeled on highway links, and a pollution dispersion model with two dimensional cell input. These features were incorporated into **DIVA**.

8.5. DIVA: The Second Mobile Millennium Visualizer

A major problem with **VIZ** is that it suffered from a lack of long term planning. Functions and features were added in response to immediate needs, and over time the code became very difficult to read and edit. Development of **DIVA**, PATH's Dynamic Information Visualization Application, began in 2010 was driven by the desire to apply the lessons learned during the development of **VIZ** to a completely new visualizer.

At the same time, many of the other parts of the system were undergoing similar redesigns. The creation of the **CORE** module with its common classes for database access and monitoring made writing working code significantly easier. The database itself was also maturing; standards for table design and data storage were implemented. Feeds and models became more streamlined and modular. **DIVA** was written to take advantage of the ongoing system development.

8.5.1. Server

The most glaring issue on the server side of **VIZ** was that the `GeoDataBean` class and its subclasses for special applications had become extremely complicated and disorganized. The `getFeatures` method was used to return features for all features, each of which had their own intricacies which `getFeatures` handled through a large number of if-else blocks.

In **DIVA**, the `GeoDataBean` classes have been replaced by the single `Diva` class. `Diva` sits at the top of the server hierarchy and is the only class that receives requests from the client. The `getVectorLayer` methodstill calls a factory to get the correct feature type based on the layer name, but the `getFeatures` methods have been instead moved to the individual feature classes so there is no longer a "one size fits all" approach.

DIVA also combines the traffic **VIZ** and its **SKY_VIZ** and **FLO_VIZ** relatives into a single application. This reduces the number of visualizers to maintain to one rather than three. The core parts of the visualizer can be used for different applications and require only a single update.

As *Mobile Millennium* evolved from a traffic model to a platform for collecting, storing and fusing various types of data (weather, air quality, etc.) it made sense to make a visualizer that could support any type of data in the database. The feature classes are more carefully organized into packages and subpackages based on their application.

Here is a list of all packages in **DIVA** and their functions:

- `diva`: Contains the `Diva` class whose purpose is to accept and parse all requests from the client and return the appropriate data. Mainly these are requests for `VectorLayer`s, but the client also requests a list of available layers from `Diva`.
- `diva.citris`: Contains all classes related to the CITRIS **DIVA**, including features (`CitrisArterial` and `CitrisHighway`) and layer and point classes.
- `diva.features`: `VectorFeature` is an interface implemented by all feature classes, and `VectorFeatureBase` is the superclass of all features. `VectorFeatureFactory` returns an instance of the appropriate class based on the layer name it receives.
- `diva.features.airmodels`: Classes representing air quality and emission models from the ClearSky project.
- `diva.features.airsensors`: Classes representing mobile and stationary air quality sensors used by the ClearSky project.
- `diva.features.modelgraph`: Used for showing model graph networks on the map without any model output attached.
- `diva.features.trafficmodels`: Shows the output of the highway and arterial models fused with model graph links.
- `diva.features.trafficsensors`: Mobile and stationary traffic sensors such as radar speed detectors, PeMS and Cabspotting are represented by the classes in this package.

- `diva.features.watermodels`: The Floating Sensor Network models flow, depth and salinity and the `Realm` class is used to show it on the map.
- `diva.features.watersensors`: Shows the location of the Drifter water sensor robots.
- `diva.features.weather`: Classes representing weather information feeds such as `GoogleWeatherFeed`.
- `diva.geo`: Contains all geometry classes used by **DIVA**, mainly `GeoPoint` but other types could be possible (e.g. polygon).
- `diva.gui`: The layer selection tool in the **DIVA** web GUI is populated using the `AvailableLayers` class which accesses the database on a certain machine, determines what layers are available, and returns the list to the client
- `diva.layers`: The `VectorLayer` class is what is sent to the client and contains an array of features.

Each feature class works more or less the same. It represents a feature with location and attribute information that will be plotted on the map. Every feature has a `getFeatures` method which returns an `ArrayList` of its own type. Each class stores the SQL statements needed to retrieve the necessary data from the database and process it into features based on whatever parameters have been supplied.

8.5.2. Client

The organization of the client side of **VIZ** had problems similar to the server side. The `MapMain.js` became unwieldy after ballooning to 1000 lines, and so an effort was undertaken to divide its functions into several files:

- `AutoRefresh.js`: Manages the timer used for automatically refreshing the map.
- `CitrisCode.js`: Contains all functions applicable to the CITRIS **DIVA**: generating the map, handling touch input, requesting and drawing layers, etc.
- `FeatureSelection.js`: Code that is executed when a feature is selected by the user such as changing its color or displaying a popup bubble.
- `GuiCore.js`: Methods used for initializing and controlling the GUI that are widely applicable. Does not include populating the layer selector tool which is done by the specific files described below.
- `GuiClearSky.js`: GUI related methods for the ClearSky **DIVA**.
- `GuiDrifter.js`: GUI related methods for the Drifter **DIVA**.
- `GuiTrafficInternal.js`: GUI related methods for the traffic version.
- `ImportantVars.js`: Contains variables which are essential to the map functioning properly that describe its current state, such as a list of current features or layers.
- `Init.js`: Initializes the map and GUI for all versions.
- `MapManagement.js`: Functions used to control the map and add or remove

layers, features or other objects on the map.

- `RequestLayers.js`: Calls `Diva.getVectorLayer` for each layer the user has requested.
- `ResponseHandler.js`: Accepts the layers received from the server, converts them into OpenLayers compatible objects and plots them on the map.
- `StylingCore.js`: Contains feature styling functions used by all versions of **DIVA**, most notably the `getStyle` which colors a feature by calling a styling function suited to that feature type.
- `StylingClearSky.js`: Styling functions for ClearSky feature types.
- `StylingDrifter.js`: Styling functions for Drifter feature types.
- `StylingTrafficInternal.js`: Styling functions for traffic feature types.

The web pages for the different **DIVA** versions are also part of the client side. The `index.jsp` page presents the user with links to each version's `.jsp` page which in turn calls its own initialization function. This improved organization of the visualizer client code has reduced its size and complexity.

8.6. Conclusion

In all its incarnations, the visualizer was an essential part of the *Mobile Millennium* project. Without it, there would have been no way to easily observe the inputs and outputs of the system in real-time; this was invaluable to researchers during model development. It was also an indispensable tool for capturing the attention of the media, industry and the public, as it was included in many project demonstration events.

Development began in 2008 and was driven based on the immediate needs of researchers and the goal of a live demonstration at the ITS World Congress. After accomplishing these objectives, the **VIZ** was converted into an interactive traffic mapping display for the opening of the new CITRIS headquarters. Development continued in order to add more functions and support new research initiatives.

Other versions of the visualizer were developed for special functions and even to provide visualization for other projects based on its record of success. However, **VIZ** suffered from a lack of planning and design due to constantly evolving needs of the project. Eventually, the code became difficult to maintain and modify. This prompted a complete redesign of the **VIZ** known as the Dynamic Information Visualization Application or **DIVA** for short. **DIVA** is now a stable application that can be expanded to display data on a map for various applications and remains a valuable part of the *Mobile Millennium* system to this day.

9. IBM Traffic Prediction Tools

9.1. Introduction

The *Mobile Millennium* system has proven to be an effective means for acquiring, storing, processing and presenting real-time and historical data for traffic modeling. An exciting advantage of the *Mobile Millennium* project is that it can be adapted easily to support other research endeavors. A prime example of this is the collaboration between IBM and CCIT to develop a personalized commuter forecast system called “Smarter Traveler”.

IBM’s Traffic Prediction Tool, or TPT, is a traffic forecasting algorithm that has been integrated with IBM’s Smarter Traveler program. The aim of the Smarter Traveler is to provide drivers with predictions through mobile and web platforms that will enable them to avoid congestion before they begin their trip. By making informed travel decisions, commuters using Smarter Traveler will save time and energy, but other road users will benefit as well as traffic loads are more balanced over the entire road network due to informed drivers avoiding congested roads. Additionally, transportation agencies and urban planners will be able to analyze traffic patterns and TPT predictions, allowing them to design and manage transportation systems to maximize efficiency.

To be effective over a large area and serve a significant number of customers, traffic forecasting efforts such as TPT require large quantities of traffic data from different sources. Though there are publicly available data sources, PATH is able to provide high quality filtered data, including years worth of archived data.

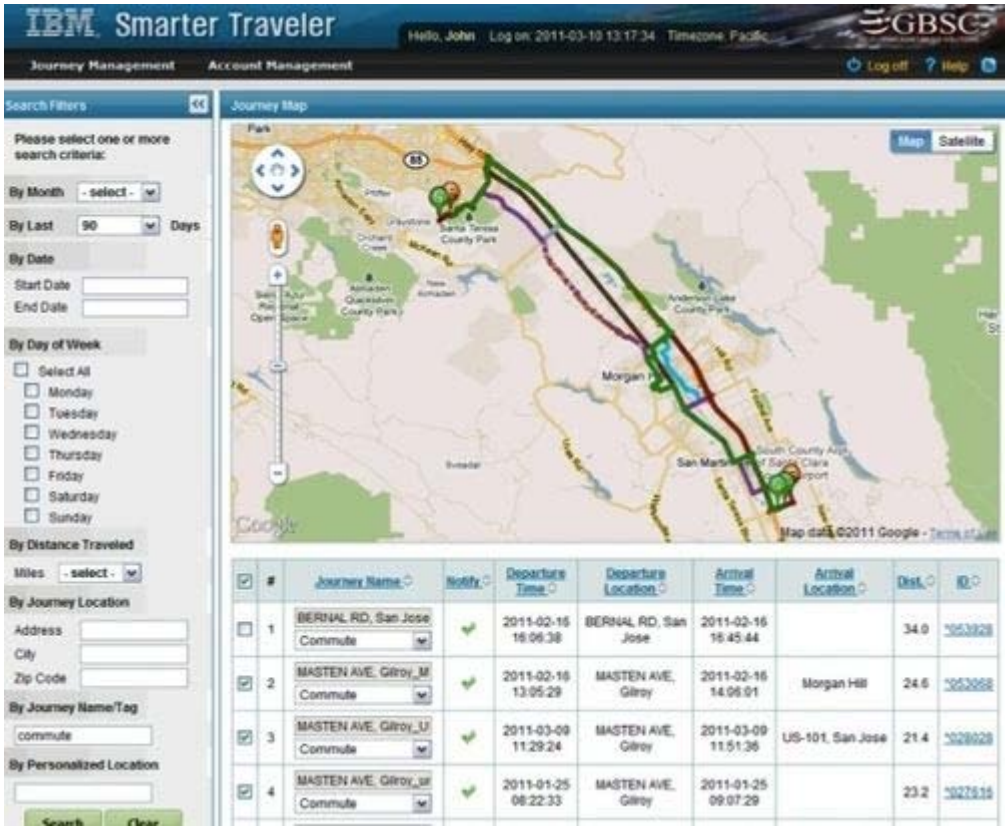


Figure 1.1. Screen shot of Smarter Traveler client.

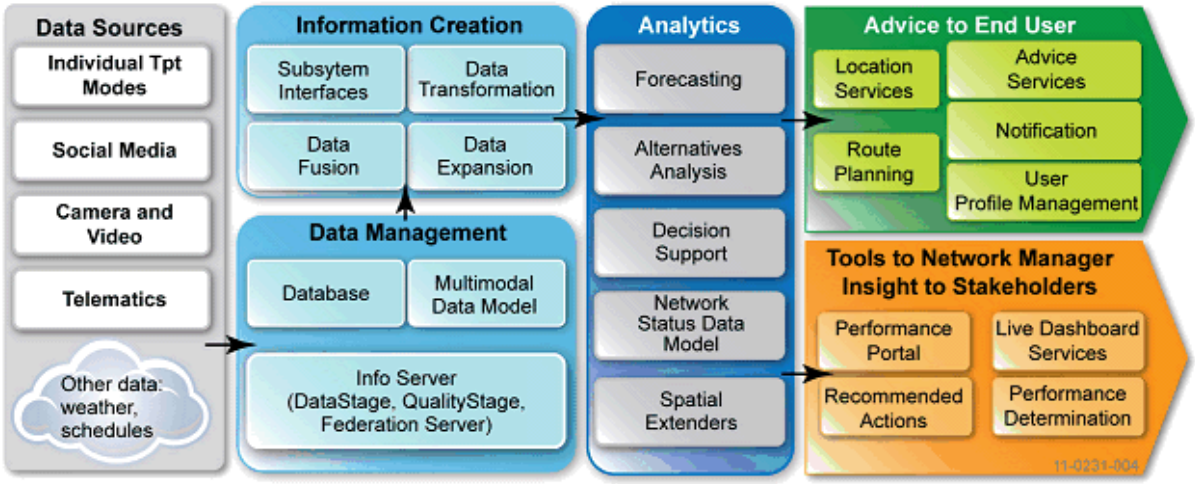


Figure 1.2. Data from various sources are consumed by information creation and data analytics tools prior to becoming custom output for Smarter Traveler end users.

This collaboration was proposed in an amendment to Research Technical Agreement #65A0348 entitled "Collaboration with IBM on Multi-Sourced Traffic Information". The responsibilities of each party were roughly defined as follows: -

- IBM: Develop Smarter Traveler application, including underlying algorithms, software platform, user interface and integration mechanisms.
- PATH: Provide data (primarily filtered PeMS data) to support IBM's development efforts, and integrate TPT forecasts into PATH's system.

Establishing access to PATH's filtered PeMS data was the first major milestone of this project. Weekly teleconferences were held throughout the project, and the first discussions focused on developing a constant, real-time feed of PATH's filtered 30 second PeMS data to IBM. Once this was accomplished, IBM's research engineers used two months worth of PeMS data to initialize and calibrate the TPT tool. Additionally, PATH provided archived incident information collected from the California Highway Patrol's public feed to IBM researchers to further their development efforts.

During TPT calibration and integration with the Smarter Traveler system, PATH and IBM engineers worked together to maintain and refine data feeds. PATH provided support to IBM researchers, addressing questions about the content of the data provided, correcting errors, and verifying results. Concurrently, IBM also provided PATH with the format and samples of TPT data so that PATH could establish a feed for pulling and storing prediction data from IBM. This input feed was fully integrated into the *Mobile Millennium* system and database.

TPT data was collected over a period of a couple months. A week's worth of data was selected for analysis by PATH researchers. PATH examined the accuracy of the predictions in a few different ways, and found the results to be very promising. This article describes the work completed during the course of this project and discusses its successes, challenges and opportunities for future work.

9.2. Filtered PeMS Feed

The Performance Measurement System (PeMS) provides live flow and occupancy measurements from induction loop sensors embedded in the roadway. Since it was deployed statewide in 2002, PeMS has proven to be an effective, low-cost means of measuring the efficiency of California's highway system. The flow and occupancy measurements can be converted into an estimated speed using the following formula:

$$v(t) = g(t) \times \frac{c(t)}{o(t) \times T}$$

Where:

$v(t)$ = Average speed during period T
$g(t)$ = Effective vehicle length (also known as g-factor)
$c(t)$ = Number of vehicles that passed over the sensor during period T
$o(t)$ = Proportion of time the detector sensed a vehicle present during period T
T = Time period under consideration

However, the speed values obtained from PeMS are not entirely accurate. In order to convert the raw data to speed, an average vehicle length must be assumed which introduces error into the estimation since vehicle length varies over time as the mix of vehicles changes (e.g. there are more large trucks on the road compared to cars during off-peak hours) and with the location of the sensor (even between lanes on the same road). Furthermore, the sensors themselves are known to occasionally return grossly erroneous or incomplete data, and error variance increases during free flow hours.

A major part of the *Mobile Millennium* project was the development of a PeMS filter which was effective at removing erroneous values, filling in data gaps and computing more appropriate g-factors for each detector. This filtered data is very valuable for traffic modeling and forecasting purposes, and was sought by IBM to complement other data sources used by the TPT.

The first step was to design and implement a continuous feed of filtered 30 second PeMS data from PATH to IBM so that IBM researchers could evaluate the data. In late 2010, both teams began weekly discussions to determine the specifications and terms for data exchange. It was determined that IBM should send regular requests for data to PATH's server. IBM's system sends an HTTP GET request to the PATH server machine. The server parses the request, queries the database for the specified epoch or interval, and returns the data as an XML document.

The filtered PeMS feed to IBM does not run on the main *Mobile Millennium* system. Rather than increasing the load on **mmdev**, it was decided that a separate machine should be used. The server, called **ibmpems**, was purposed exclusively to provide data to IBM in response to their requests. The *Mobile Millennium* management tool was installed, and the PeMS feed and filter code was downloaded from the SVN repository, compiled and executed. The database installed was identical to that found on the main machines, albeit with empty tables except for those relevant to PeMS.

The feed accepts a few different types of requests based on the values of supplied parameters. The possible parameters are:

- **single**: A Boolean parameter which specifies whether records for a single epoch will be returned (as opposed to records for a range of time).
- **start**: A timestamp in the YYYY-MM-DD_ hh-mm-ss format representing the start of requested interval or requested single epoch.
- **end**: A timestamp in the YYYY-MM-DD_ hh-mm-ss format representing the end of requested interval.

Depending on which parameters are supplied in the request, different types of datasets are returned. Table 1 describes the available modes. Requests that do not conform to the modes in the table are rejected; any parameters supplied other than those listed above are ignored.

single	**start**	**end**	**Effect**
true	null	null	Returns the records for the most recent epoch only.
true	timestamp	null	Returns the records for the epoch specified by the **start** parameter only.
false	timestamp	null	Returns all records from the **start** time up to and including the most recent epoch.
false	timestamp	timestamp	Returns all records from the **start** time up to and including the **end** time.

Table 1. Different modes for requesting filtered PeMS data from PATH based on parameters.

The different modes allow for data to be collected constantly through requests for the most recent records, or in batches (though a limit of 6 hours of data was imposed to avoid overloading the system). Based on the type of request and the parameter values, the program forms an SQL query to retrieve the requested data from the **ibmpems** database. The result set is translated into an agreed upon XML format shown below (only one record is shown, an ellipsis (…) is shown in place of the other < R> elements:

```
<PemsFiltered30sData>
  <R>
    <D>2011-05-23 09:55:30.0 </D>
    <I>311974</I>
    <F>3823.0</F>
    <V>61.826088</V>
  </R>
  ...
</PemsFiltered30sData>
```

$Where$:

$R =$ Record

$D =$ Date

$I =$ ID number of the VDS

$F =$ Flow (vehicles per hour)

$V =$ Velocity (miles per hour)

This feed is fairly simple and worked well for the most of the duration of the project. In cases where the feed failed (due to software or hardware problems on either the PATH or IBM side), the missing data was either requested from the feed in batches, or manually extracted from the database and uploaded to IBM via FTP. However, most of the problems encountered were caused by a malfunction of PATH's raw PeMS feed, often due an error on the PeMS FTP server from which PATH pulls. In such cases, it was often not possible to recover the missing raw data and filter it resulting in permanent gaps of hours or days. Such incidents were fortunately rare enough that it did not significantly hinder IBM's research; the feed established by PATH proved to be a reliable source of input data for the TPT.

9.3. CHP Data

To supplement the filtered PeMS data and provide additional information to IBM's research team, PATH also provided several months worth of archived traffic incident logs from the California Highway Patrol. CHP provides live traffic incident information tothe public and news media through a basic website (http://cad.chp.ca.gov) and XML feed (http://media.chp.ca.gov/sa_ xml/sa.xml). Below is an XML example of a single incident reported by CHP.


```
< Log ID="0136D0607">
    < LogTime> "5/17/2011 3:59:23 PM"< /LogTime>
   < LogType> "1182 - Traffic Collision - No Injuries"</ LogType>
   < Location> "NB SR99 JSO 1ST AV"< /Location>
   < Area> "Bakersfield"< /Area>
   < ThomasBrothers> "200 5D"< /ThomasBrothers>
   < TBXY> "6191518:2463701"< /TBXY>
    < LogDetails>
     < details>
       < DetailTime> " 4:33PM"< /DetailTime>
       < IncidentDetail> "1039 J S TO RESPOND TO HIGH"< /IncidentDetail>
     < /details>
     < details>
       < DetailTime> " 4:32PM"< /DetailTime>
```

```
        < IncidentDetail> "INFO FOR CT 30 FT OF PERIM CHAIN LINK FRNCE DOWN"< /IncidentDetail>
      < /details>
      < details>
        < DetailTime> " 4:31PM"< /DetailTime>
        < IncidentDetail> "IGNITION PUNCHED - APPEARS GTA"< /IncidentDetail>
      < /details>
      < details>
        < DetailTime> " 4:27PM"< /DetailTime>
        < IncidentDetail> "1039 J S ETA 20"< /IncidentDetail>
      < /details>


      < details>
        < DetailTime> " 4:13PM"< /DetailTime>
        < IncidentDetail> "DPD SGT IS 97 REQ ETA"< /IncidentDetail>
      < /details>
      < details>
          < DetailTime> " 3:59PM"< /DetailTime>
        < IncidentDetail> "VEH VS TREE"< /IncidentDetail>
      < /details>
  < /LogDetails>
< /Log>
```


As part of the *Mobile Millennium* project, PATH established a feed to pull this data from CHP and store it in the database. Each incident record describes the type of incident (a minor collision in the example above), the time it occurred, and a description of the location. The location is expressed in a few elements:

- <Location> : A description of the address using street names and helpful abbreviations. In the above example, "NB SR99 JSO 1ST AV" means "northbound State Road 99, just South of 1st Avenue" (in Bakersfield, as indicated by the < Area> element).
- <ThomasBrothers> : A page number and grid coordinates of the incident location in the Thomas Guide, a commonly used atlas.

- <TBXY> : A set of Cartesian coordinates that are presumably a more precise extension of the Thomas Guide coordinates and that happen to (usually) line up with the California Coordinate System of 1983 (CCS83).

Unfortunately, none of these location elements are easy for computers and/or most people to understand. Fortunately, the *Mobile Millennium* system employs an algorithm to convert the <TBXY> coordinates into the familiar decimal degrees longitude and latitude format using a formula found in a Caltrans document entitled "The California Coordinate System" by Vincent J. Sincek"[11]. This was the only manipulation of data performed.

Many incident records, including the given example, include additional details as the situation changes over time. These extra details are useful because researchers can use it to estimate the duration of an incident, particularly when the detail is explicit that the incident is over with (e.g. "ROAD OPEN", "RESUME NORMAL TRAFFIC", etc.).

IBM researchers used CHP data to support the development of TPT; the data offered additional insight into peculiar traffic behavior, allowing the researchers to adjust the algorithm. Efforts were made to automate the analysis of the CHP logs in real time, but this was not successfully implemented due to the complexity involved and lack of necessity. For this reason, the development of a requisite real-time feed of (geographically transformed) CHP incident data from PATH to IBM was not completed.

9.4. TPT Input Feed

Once the filtered PeMS feed had been deployed, tested and stabilized; IBM collected several weeks worth of data. After a period of algorithm refinement, IBM made their prediction data available on their servers. PATH developed a feed to pull this data and store it in the *Mobile Millennium* database.

The TPT generates estimates every 5 minutes. The dataset for a single epoch consists of speed (mph) and flow (vehicles/hour) predictions at 5, 10, 15, 30, 45 and 60 minutes into the future for each PeMS VDS. XML was chosen as the data transfer medium. The PATH system requests prediction data every 5 minutes, and the IBM TPT server would respond with an XML document containing predictions for each VDS. Below is an example of the response received from IBM. As the entire response is quite large, only forecasts for VDS 400001 are included; ellipses (…) have replaced the <ForecastData> elements for other VDS locations.

[11]The conversion was not successful for all of CHP data centers; in some cases the the converted coordinates were offset from their correct position and required an additional transformation.

```
<TPToolTransaction>
  <TrafficResponse>
    <Date> 2010-11-30T22:36:50.574+0000</Date>
    <TransactionId> 1290719400000</TransactionId>
    <Status> OK</Status>
    <Forecast>
      <Date> 2010-11-30T22:36:50.574+0000</Date>
      <TransactionId> 1290719400000</TransactionId>
      <ForecastId> 1290719400000</ForecastId>
      <ForecastType> 1</ForecastType>
      <ForecastDate> 2010-11-25T21:15:00.000+0000</ForecastDate>
      <WDay> 4< /WDay>
      <Period> 255< /Period>
      <ForecastData>
          <LinkID> 400001</LinkID>
          <Speed> 63.19046</Speed>
          <Volume> 2144.9019</Volume>
      </ForecastData>
      . . .
    </Forecast>
    <Forecast>
      <Date> 2010-11-30T22:36:50.578+0000</Date>
      <TransactionId> 1290719400000</TransactionId>
      <ForecastId> 1290719400000</ForecastId>
      <ForecastType> 2< /ForecastType>
      <ForecastDate> 2010-11-25T21:20:00.000+0000</ForecastDate>
      <WDay> 4</WDay>
      <Period> 256< /Period>
      <ForecastData>
          <LinkID> 400001</LinkID>
          <Speed> 63.11071</Speed>
          <Volume> 2074.6997</Volume>
      </ForecastData>
      . . .
      < Forecast>
      <Date> 2010-11-30T22:36:50.581+0000</Date>
      <TransactionId> 1290719400000</TransactionId>
      <ForecastId> 1290719400000</ForecastId>
      <ForecastType> 3</ForecastType>
      <ForecastDate> 2010-11-25T21:25:00.000+0000</ForecastDate>
      <WDay> 4</WDay>
      <Period> 257</Period>
      <ForecastData>
          <LinkID> 400001</LinkID>
          <Speed> 63.35995</Speed>
          <Volume> 2017.1316</Volume>
      </ForecastData>
      . . .
    </Forecast>
    <Forecast>
      <Date> 2010-11-30T22:36:50.584+0000</Date>
      <TransactionId> 1290719400000</TransactionId>
      <ForecastId> 1290719400000</ForecastId>
      <ForecastType> 6</ForecastType>
      <ForecastDate> 2010-11-25T21:40:00.000+0000</ForecastDate>
      <WDay> 4</WDay>
      <Period> 260</Period>
      <ForecastData>
          <LinkID> 400001</LinkID>
          <Speed> 62.58852</Speed>
          <Volume> 2195.759</Volume>
          < /ForecastData>
          . . .
      </Forecast>
      <Forecast>
          <Date> 2010-11-30T22:36:50.588+0000</Date>
          <TransactionId> 1290719400000</TransactionId>
          <ForecastId> 1290719400000</ForecastId>
          <ForecastType> 9< /ForecastType>
```

```
        <ForecastDate> 2010-11-25T21:55:00.000+0000</ForecastDate>
        <WDay> 4< /WDay>
        <Period> 263</Period>
        <ForecastData>
          <LinkID> 400001</LinkID>
          <Speed> 63.06185</Speed>
          <Volume> 1768.372</Volume>
        </ForecastData>
        . . .
      </Forecast>
      <Forecast>
        <Date> 2010-11-30T22:36:50.591+0000</Date>
        <TransactionId> 1290719400000</TransactionId>
        <ForecastId> 1290719400000</ForecastId>
        <ForecastType> 12</ForecastType>
        <ForecastDate> 2010-11-25T22:10:00.000+0000</ForecastDate>
        <WDay> 4</WDay>
        <Period> 266</Period>
        <ForecastData>
          <LinkID> 400001</LinkID>
          <Speed> 63.738174</Speed>
          <Volume> 1562.9463</Volume>
        </ForecastData>
        . . .
      </Forecast>
    </TrafficResponse>
  </TPToolTransaction>
```


Column Name	Description	Example
date	Timestamp in local Berkeley time	2011-05-17 16:10:00
fk_ pems_ id	PeMS VDS identifier	402104
speed_ 5min	Estimated speed 5 minutes into future, mph	60.7932
vol_ 5min	Estimated volume 5 minutes into future, vehicles/hour	3251.49
speed_ 10min	Estimated speed 10 minutes into future, mph	60.7341
vol_ 10min	Estimated volume 10 minutes into future, vehicles/hour	3331.83
speed_ 15min	Estimated speed 15 minutes into future, mph	60.4084
vol_ 15min	Estimated volume 15 minutes into future, vehicles/hour	3483.57
speed_ 30min	Estimated speed 30 minutes into future, mph	58.4332
vol_ 30min	Estimated volume 30 minutes into future, vehicles/hour	3581.23
speed_ 45min	Estimated speed 45 minutes into future, mph	60.2771
vol_ 45min	Estimated volume 45 minutes into future, vehicles/hour	3659.52
speed_ 60min	Estimated speed 60 minutes into future, mph	60.2769
vol_ 60min	Estimated volume 60 minutes into future, vehicles/hour	3582.09

Table 2. Storage format for TPT data in table ibm_ tpt.raw. Note: the speed and volume values are stored as they are received from IBM; the number of decimal places is not a reflection of estimate precision.

The TPT input feed was written in Java as part of the *Mobile Millennium* software, module name **IBMTPT**. It runs on the **mmdev** system and executes every 5 minutes to pull the most recent

data from the IBM server. The IBM server requires a single parameter: a timestamp in the YYYYMMDDhhmmss format representing the epoch at which predictions are generated. E.g. if 20110425084500 is supplied, then predictions 5, 10, 15, 30, 45 and 60 minutes into the future from 08:45:00 on April 25, 2011are returned. If a value where $minutes \bmod 5 \neq 0$ is supplied, the minutes are rounded down to the last whole multiple of 5.

Upon receiving the TPT response, the **IBMTPT** feed parses it into an XML document and then extracts the information into a Java object. The data is then inserted into the ibm_ tpt.raw table. The data is stored by timestamp and PeMS VDS as shown inTable 2. The feed proved to be a reliable means of collecting data; fortunately there were very few outages since a function for backfilling gaps in the dataset was not implemented.

9.5. Results

The IBM TPT predictions were collected over several weeks and stored in the PATH's MM database. The week of April 23-29, 2011 was chosen for analysis as a representative portion of the entire dataset which contained few gaps and errors. TPT predictions were produced every 5 minutes; at each epoch there were predictions for 5, 10, 15, 30, 45 and 60 minutes into the future. These predictions were compared with PATH's filtered PeMS data; the 30 second data were averaged into 5 minute intervals corresponding to the predictions. There were 628 PeMS detector locations (each known as a VDS) considered in this study.

There were three methods of analysis employed for both speed and flow: 0.analysiscounter

- Mean absolute error between TPT prediction and filtered PeMS for all VDS and 5 minute intervals according to each prediction type (i.e. 5 minutes, 10 minutes, etc. into the future).
- Error magnitude based on time of day for each prediction type, averaged for all VDS.
- Error magnitude based on time of day for each prediction type, for selected individual VDS.

9.5.1. Total error

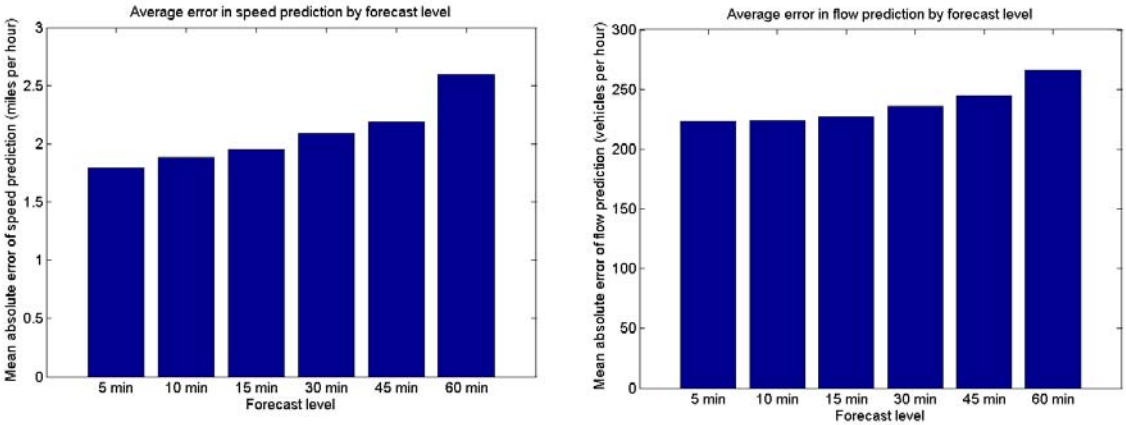


Figure 5.1. Average speed and flow errors by prediction level.

For each prediction type, the mean absolute error (MAE) was computed for the entire data set. This metric gives a simple overview of the accuracy of the predictions. For a set of n estimations

$$MAE = \frac{\sum_{i=1}^{n} |\, x_i - \hat{x}_i|}{n}$$

where x isthe true value (the filtered PeMS measurement in this case) and $\hat{x}$ is the predicted value. Figure 5.1 shows the results of this computation.

Unsurprisingly, the error increases as the predictions are farther into the future. However, the relative increase in error is small; speed errors ranged from 1.8 to 2.6 miles per hour, and flow errors ranged from approximately 220 to 265 vehicles per hour. For both speed and flow, these are very low error levels. Over a long period of time, the TPT predictions are very accurate.

9.5.2. Error over time, all VDS

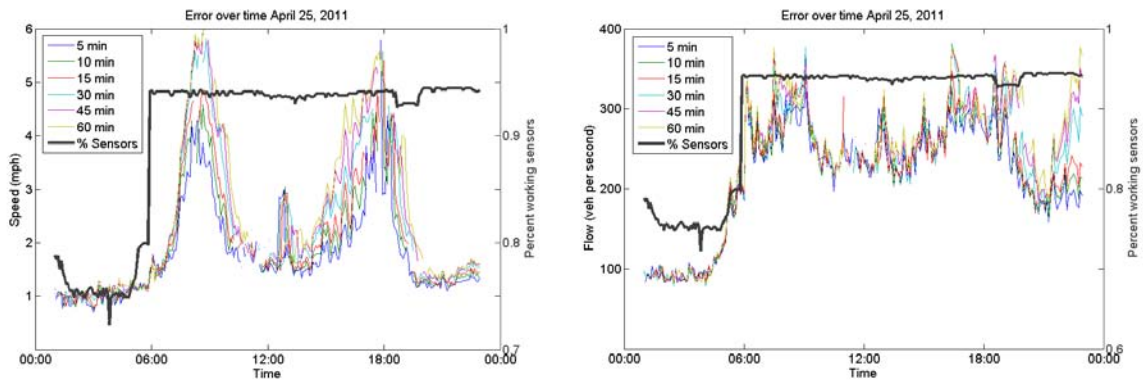


Figure 5.2. Speed and Flow error on April 25, 2011. Each line represents a different prediction level. The thicker solid line indicates the percentage of "healthy" sensors at the corresponding moment in time (and corresponds to the right y-axis on the plot).

The results presented in the previous section are promising but provide only a cursory look at the TPT accuracy. Traffic flow and speed vary with time, particularly the time of day due to commuters, so it makes sense to examine the error behavior over the course of the day. The figures below show the speed and flow errors of the TPT predictions on April 25, 2011 (the results for other days were similar).

As expected, the error is lowest during evening and midday hours when traffic volumes are lower and more predictable. The speed error plot in particular shows prominent error peaks during the morning and evening commutes (with a small peak during lunch hour); the magnitude of speed error ranges from 1 to 6 mph. The variation in flow error is less pronounced, ranging from 100 to 350 vehicles per hour. Both of these error magnitudes are low enough to be useful in traffic analysis and forecasting (e.g. identifying bottlenecks).

9.5.3. Individual VDS error over time

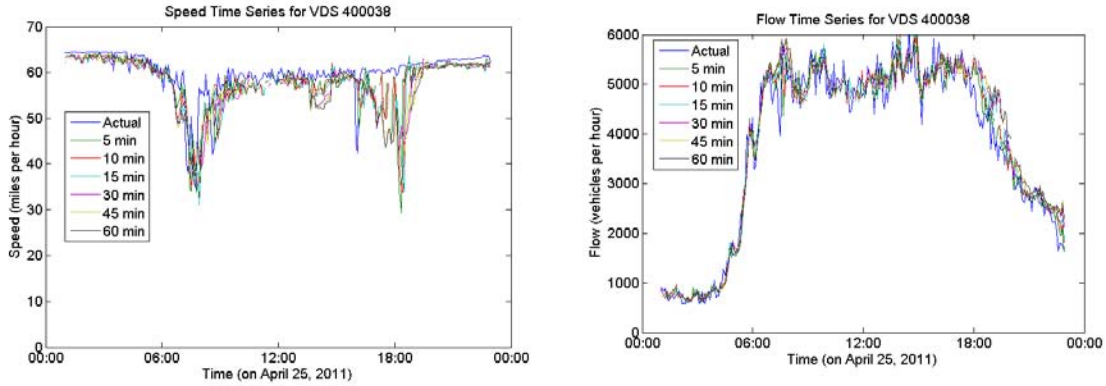


Figure 5.3. Speed and Flow values for VDS 400038. Each line represents a different prediction level.

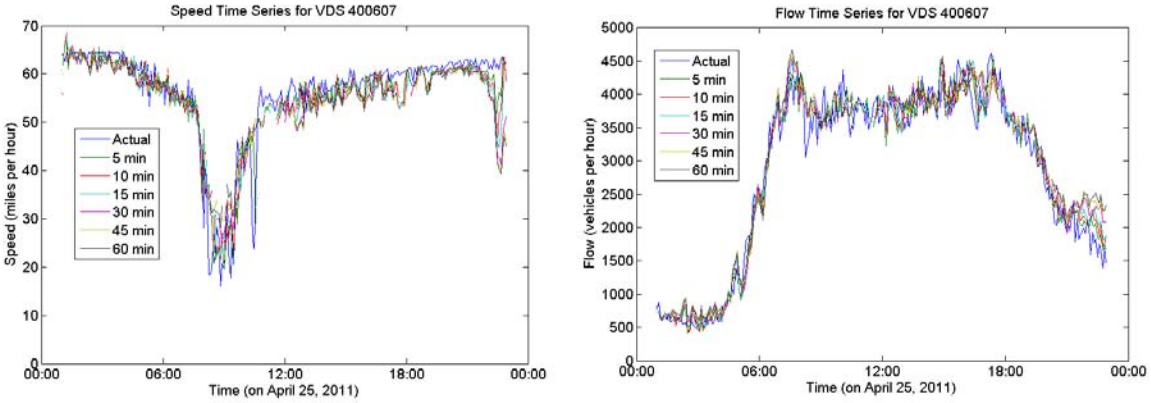


Figure 5.4. Speed and Flow values for VDS 400607. Each line represents a different prediction level.

This section provides a look at the accuracy of the TPT predictions at individual VDS locations 400038 and 400607 which are located on I-880 near Hayward and Oakland respectively. The figures below show the speed and flow predictions and the filtered PeMS measurements (denoted on the figures as “actual"). Note that the prediction data are plotted at the time of the predicted value (e.g. a point for 45 minute prediction that appears at 9:00 on the graph was computed at 8:15).

In each case, for both speed and flow, the predictions line up very closely with the blue PeMS line plot. They also line up with each other well; earlier predictions tend to agree with ones made later (which have the advantage of more recent information).There is one notable exception where the predictions diverge from the filtered PeMS: at VDS location 400038 during the evening commute the TPT predicted a drop in speed due to congestion, but the filtered PeMS shows a fairly constant speed.

9.6. Conclusion

The collaboration between PATH and IBM to develop personalized commuter forecast system, which included predicting traffic conditions up to one hour into the future demonstrated the effectiveness of the *Mobile Millennium* as a means for acquiring, storing, processing and presenting real-time and historical traffic data that can be adapted for a variety of traffic modeling purposes.

As raw PeMS data includes a significant number of gaps, gross errors and approximations which reduce PeMS’s accuracy and utility, PATH researchers developed a powerful filter for addressing these shortcomings and smoothing the data. While this proven valuable over the course of the *Mobile Millennium* project, the collaboration with IBM has further demonstrated its value and that of the *Mobile Millennium* system.

A feed was established to provide filtered PeMS data to IBM based on their specifications. This feed was seamlessly integrated into the *Mobile Millennium* system and fulfilled its intended role. PATH also provided archived traffic incident data from the California Highway Patrol to supplement PeMS and other sources in the course of IBM's research and development. However, a constant feed of CHP data was not deemed necessary.

IBM provided PATH with a feed of TPT forecasts over several weeks. PATH integrated this new data source into the *Mobile Millennium* system and evaluated the predictions. PATH compared the predicted speed and flow values from TPT to the filtered PeMS measurements using a few different techniques. The accuracy of the TPT was impressive; the total error in speed was less than 3 mph. These promising results merit further testing and development of the TPT.

As a result of this collaboration, IBM was able to implement a powerful traffic prediction algorithm with its Smarter Traveler program on California highways. PATH's *Mobile Millennium* system made it possible to acquire large amounts of unique and useful data in real time, and the predictions returned by IBM were seamlessly integrated into MM as a new data source. Both organizations accomplished their objectives and established a productive relationship that will hopefully continue to yield benefits for the public and transportation community.

10. High Performance Computing

This chapter describes two successive efforts to utlize High Performance Computing (HPC), also called Cloud Computing, in the context of Mobile Millennium. Specifically addressed are issues involved with parallelization of two key algorithms that together form a cornerstone of the Mobile Millennium system. The first algorithm, called path inference and map matching (or just path inference, for short), is what makes usable non-VTL based GPS probe data. The second algorithm, called expectation maximization (EM), is used to estimate travel times on network links.

The first effort discussed here is that of parallelizing the path inference algorithm. This effort revealed a bottleneck between the database and the computing cluster. Steps to mitigate or remove this bottleneck are explained. The second effort is that of parallelizing the EM algorithm. Substantial implementation optimizations were required to achieve acceptable performance.

10.1. Background

10.1.1. Path Inference Algorithm

The path inference algorithm is a probabilistic framework to recover trajectories and road positions from low resolution probes. It is a pre-processing step for all streams of non-VTL GPS data coming from Mobile Millennium partners. The algorithm takes as input the raw data and maps them onto the road using a machine learning algorithm. The output is application-specific: trajectories for high-frequency reconstructions, map-matched points for density algorithms, path segments for travel time algorithms. These outputs are then stored in a database for subsequent use. This algorithm accomodates various inputs and is robust to a wide range of sampling frequencies and computational power. Since it has to handle massive amounts of data, it is an ideal target for parallelization.

Internally, the path inference algorithm accomplishes the following steps:

- projects a raw GPS point onto candidate points on the road network
- connects the candidate points between two time steps by candidate paths (valid paths on the road network)
- applies a machine learning algorithm that weighs all the possible sequences of paths and points and returns the most probable one.

10.1.2. Link Travel Time Estimation

In order to estimate traffic conditions with a limited amount of input data, methods for learning traffic patterns for a network are also necessary. The data collected are used to compute a historical model of traffic conditions for each link, for each time interval, and for each day of the week.

The historical data needs to be organized in order to be processed effectively:

- Each day is split into time intervals that share a common general behavior from day-to-day.
- The road network is represented as a directed graph composed of a set of links. Each link corresponds to a road between two intersections in the road network.
- The input data consists of a starting point, an end point and a travel time.

The goal is to infer how congested the links are in an arterial road network (V, E), where V are the vertices (road intersections) and E are the links (streets). For each link $e \in E$, where n is the total number of links in the network, the algorithm outputs the time it takes to traverse the link as a probability distribution. To make the inference problem tractable, we model the link traversal times for each link e as independent Gamma distributions with parameters θ_e (as shorthand, we let θ_e represent the two values that parametrize a Gamma distribution).

The algorithm inputs are the road network (V, E), as well as the observed trajectories of GPS-equipped vehicles Y (where m is the number of trajectories in the input). Each observation Y_i describes the i-th trajectory's travel time and path (which consists of one or more road links) as inferred by the path inference processing stage described above. Physical properties of the road network, such as speed limits and link lengths, are also taken into account.

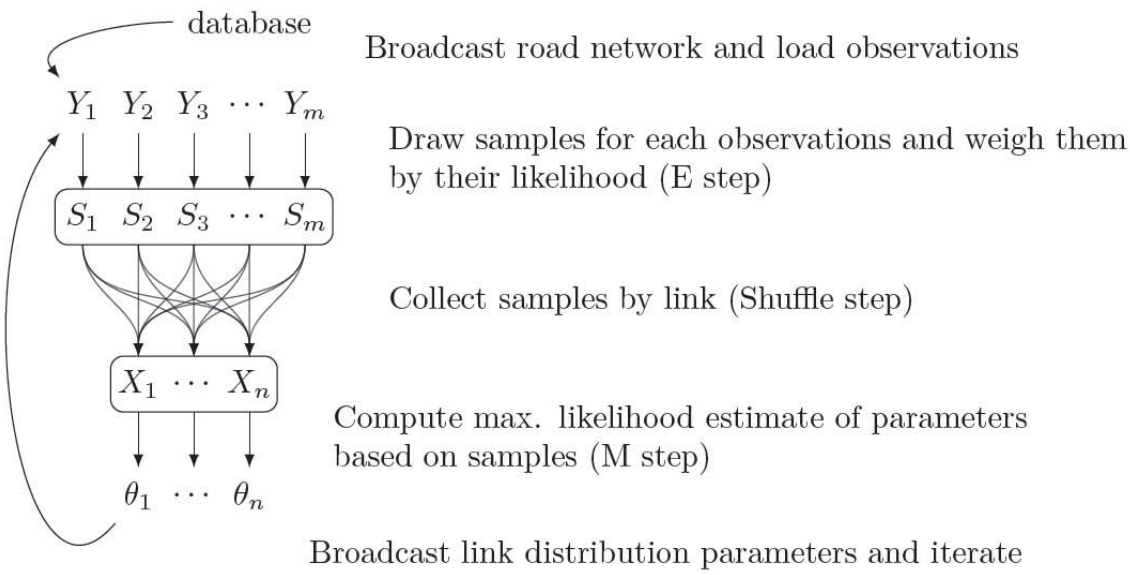


Figure 1.1. Data flow in the importance sampling EM algorithm for traffic estimation. This is common to a number of other machine learning algorithm as discussed in Section 6.

Because we do not directly observe per-link travel times (only the end-to-end time for paths of links), we use the expectation maximization (EM) algorithm to infer the expected per-link travel times (Figure 1.1). In the E-step, we *generate* per-link travel time samples from whole trajectories — specifically, for each trajectory Y_i we produce a set of samples $S_i = \{(s_{e_i}, w_{e_i})\}_{e_i \in Y_i}$ by randomly dividing Y_i's travel time among its constituent links (producing a travel time s_{e_i} for each edge $e_i \in Y_i$), and we assign a weight w_{e_i} as the likelihood of travel time s_{e_i} according to e's current travel time distribution θ_e. In the shuffle step, we regroup the samples in the S_i's to go from being grouped in a per-trajectory basis to being grouped on a per-link basis — so now each link e has samples $X_e = \{(s_{e_i}, w_{e_i})\}_{i \in \{1 \ldots m\}}$. In the M-step we recompute the parameters θ_e to fit link e's travel time distribution to the samples X_e. Finally, we iterate through the E, shuffle and M steps until convergence.

Conceptually, parallelization of this algorithm is straightforward, either on a per-trajectory (E-step) or per-link (M-step) basis. However, implementing this algorithm on the cloud posed unexpected challenges.

10.2. High Performance Computing Systems (HPC)

High Performance Computing, or Cloud Computing, is a technology that consists of using computer clusters and supercomputers in order to solve some computationally intensive problems. This technology is widely used in universities and industries since it provides highly available (24/7) infrastructure to process heavy computations.

A computer cluster used in HPC consists of a group of computers linked to each other in order to behave as a single system. The computers are interconnected via a fast network and each one contains homogeneous hardware and software. The main objective of HPC clusters is to use the processing power of multiple nodes in parallel. This parallelization requires communication between the nodes while processing tasks if the tasks are dependent.

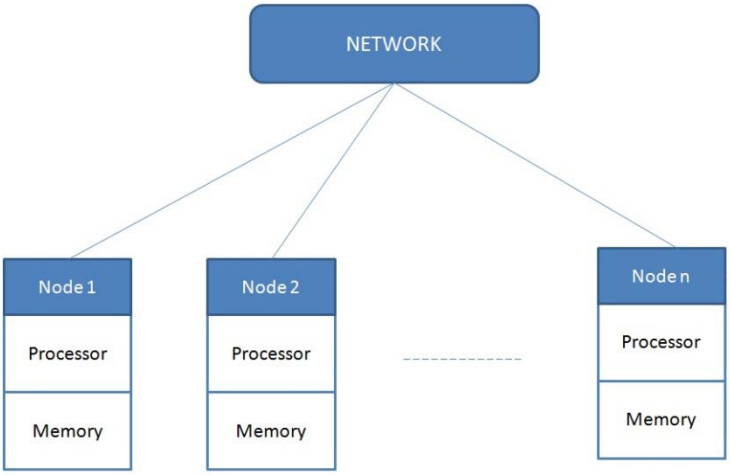


Figure 2.1. Structure of a HPC Cluster

The algorithms described in this chapter were implemented on several clusters having different properties in order to compare the performances of the different systems. Each system had different characteristics and advantages.

10.2.1. NERSC Carver Cluster

This cluster is located at the National Energy Research Scientific Computing Center (NERSC) at the Lawrence Berkeley National Laboratory in Berkeley. NERSC is a world leader in providing high performance scientific computing facilities for research and is sponsored by the Office of Science in the U.S Department of Energy.

The NERSC machines were available for the implementation of the path inference project. The

carver cluster consists of 400 computing nodes (3,200 cores total).

- Advantages: The main advantage of using NERSC is the high number of available nodes.
- Disadvantages: Being a highly used cluster, the NERSC Cluster was sometimes in maintenance and it took longer to get assigned some due to the high number of requests.

10.2.2. UCB CITRIS Cluster

The CITRIS cluster is located in the Center for Information Technology Research in the Interest of Society (CITRIS) on the UC Berkeley Campus. This cluster is available for information technology researches that has potential to improve the quality of life in California. The CITRIS cluster is used by only four research groups and hence it is fast to obtain the requested nodes.

The CITRIS cluster consists of 33 server nodes. 1 login node and 32 computing nodes (256 cores in total). The path inference project was implemented on this cluster.

- Advantages: This cluster of smaller size was used mainly for debugging and testing. The support team was very efficient and it was possible to install new programs very easily (using modules). The request of nodes was very fast since initially we were one of only four main research groups working on this cluster.
- Disadvantages: The cluster contains only 32 computing nodes, which is quite small for a simulation with a large amount of data.

10.2.3. Amazon EC2

Amazon EC2 is a commercial cluster on which the PIP was first implemented. The Amazon Elastic Compute Cloud is a web service from Amazon that allows you to acquire re-sizable computational capacity in the Amazon's cloud.

- Advantages: There are a lot of features available since it is a commercial product. The Amazon S3 storage system is really useful for the implementation of a database on a node. The number of nodes available is nearly unlimited since you pay for each node requested.
- Disadvantages: The system is less flexible; you have to use the tools offered by Amazon.

10.2.4. Cluster Specifications

The main clusters on which the path inference project was implemented were the NERSC Carver Cluster and the UCB CITRIS Cluster. Both of them were managed with a shell command line and using modules in order to make available different programs such as gcc for example. Here is a table describing the characteristics of both clusters:

	Computing Nodes	Login Node	Type of Processors per node	Memory per node	Home Directory Quota	Scratch or Project Quota	Shared Parallel File System
NERSC	400	6	2 Quad-core Intel Nehalem 2.67 GHz	24 GB DDR3	40 GB	20 TB	GPFS
CITRIS	32	1	2 Quad-core Intel Nehalem 2.7 GHz	24 GB DDR3	10 GB	500 GB	GPFS

Table 3. Cluster Specifications

10.3. Parallelization Approach

In this chapter, the primary source of non-VTL GPS-based data was from a taxi fleet in San Francisco. In order to do the processing of the data efficiently, the data is separated into samples by time interval and each sample of data is processed in parallel on several computers in the cluster.

The structure of the project is described in Figure 3.1. The main elements and steps are:

- The cars (taxis) driving around San Francisco send their GPS information which are written in the database with the following fields (Starting Point, End Point, Travel Time, Time of measurement)
- On the cluster, we have a master/slave structure where the master determines which samples of data will be executed by which slaves. Each slave runs the EM Algorithm on the sample of data it was assigned and writes its results in the database
- The database contains the new measurements from the Taxi feeds and the outputs of the EM Algorithm.
- The visualizer produces a live representation of the traffic estimation

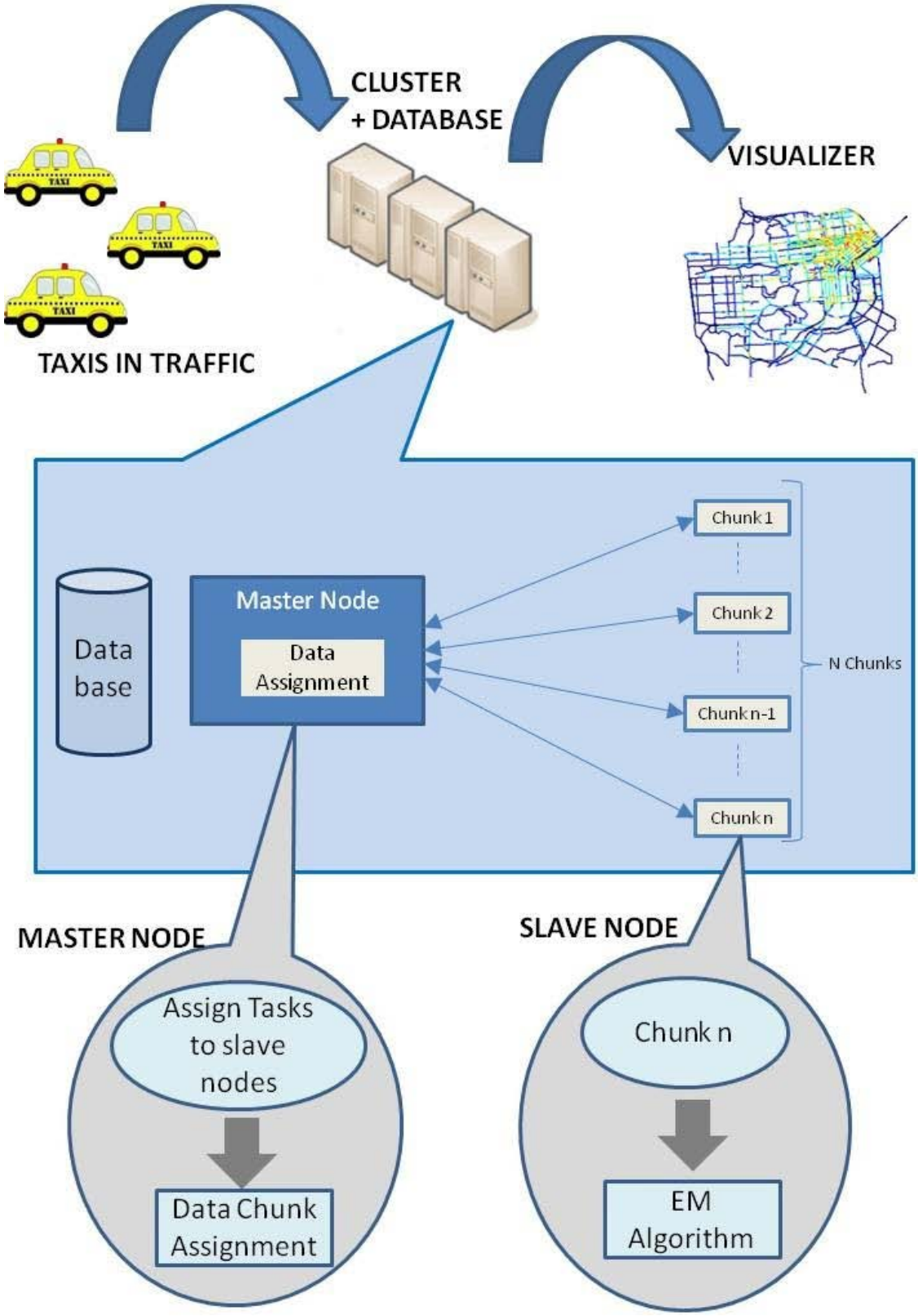


Figure 3.1. Structure of the Path Inference Project

Infrastructure on the cluster

Master/Slave structure

The path inference project requires a master/slave structure. The master separates the data into samples assigned to each slave. The master/slave structure has been implemented on the cluster so that the tests of slaves run in parallel.

The infrastructure on the cluster consists of:

- One master node taking care of splitting the data into samples and distributing the job among the available slave nodes.
- N slave nodes each running the Expectation-Maximization algorithm on its assigned sample.
- Database on which the new live data from the taxis will be written as well as the output of the EM algorithm from the slave nodes.

The simulation also requires the following properties:

- Data from live feeds must be stored in the database. These data will be used by the slaves in their EM run.
- A visualizer must have access to the database in order to produce a live map of the traffic estimation and forecast.

The whole infrastructure is represented in the Figure 3.2.

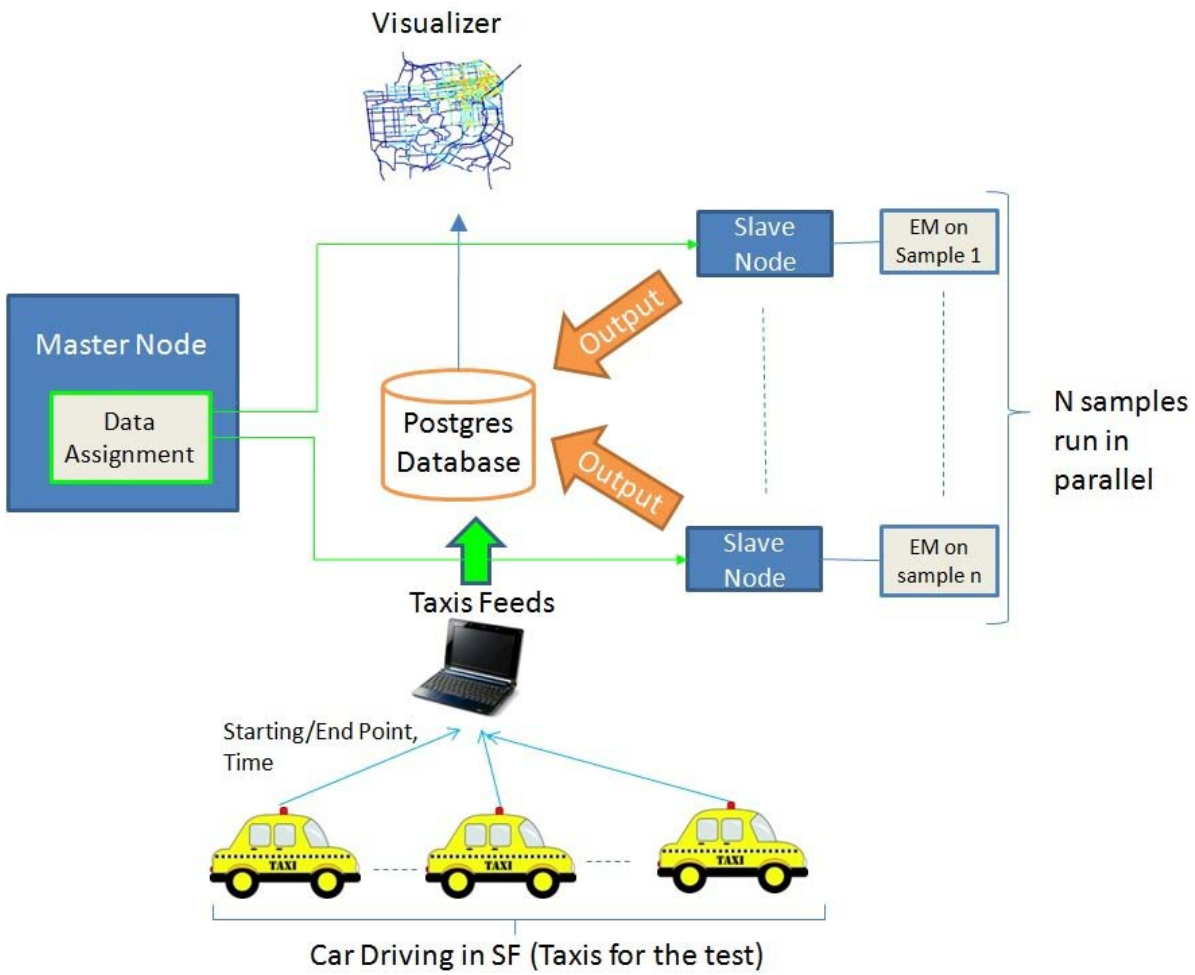


Figure 3.2. Structure on the cluster

The advantages of using the cluster for the path inference project are that we can run simultaneously an analysis on several samples. The tasks are split among the available nodes and this would reduce the processing time by a factor N (corresponding to the number of slaves available). The structure of the EM algorithm fits well to a deployment on a cluster since each EM task is independent.

Database Challenges

The path inference project requires the use of a database in order to store the GPS measurements as well as the outputs of each slave. Hence, the database needs to be accessed by three main components:

- Slave Nodes: reading new data measurements and writing the output of the EM algorithm
- GPS feed: writing new data measurements
- Visualizer: reading the output of the simulation to produce a live map of the

results

The database needs to be a relational database since we want to be able to use some psql queries in order to retrieve some specific data from a specific area at a determined time. The database used is a postgreSQL database; this database of PATH contains all the data and tables needed for a simulation of the path inference project.

The main reason a relational database was used instead of a NoSQL database such as Cassandra is because we need to retrieve input data from a specific time interval on a specific geography location. These kinds of queries are easily implemented in postgreSQL and effectively processed in a relational database. Moreover, the project was using other project classes that were already implemented so as to use a postgreSQL relational database. The postgreSQL database provides a uniform storage mechanism standardized across all the projects implemented in the Mobile Millennium System.

The main challenge was to find the most efficient way to access the new data measurement as well as having a relational database accessible from the visualizer and the live data feeds. Two options were available:

- Database directly on the cluster
- Database in the Mobile Millennium System

The most efficient structure for the database was to have the database on the cluster so as to avoid risk of a bottleneck in the ssh tunnel. Although the migration of the database to the cluster was a time consuming process, once completed substantial improvementes in processing performance were obtained.

Mesos/Spark

The master/slave structure was implemented using the Mesos/Spark framework. Mesos is a cluster management platform and Spark is a cluster computing framework built on top of the Mesos platform. Mesos and Spark were developed at UC Berkeley by the RadLab team. [12]

Mesos

Mesos is a management platform able to run cluster-based projects using different distributed applications (frameworks) such as Hadoop, MPI and Spark as well as several instances of the same framework on the same cluster. Mesos uses a fine-grained resource sharing method. This allows Mesos to organize how each different framework running on the cluster is accessing shared resources. Each framework takes turns to access the shared resources. Mesos decides how many resources to provide to each framework. The framework will then determine which computation to run on the available resources.

Mesos also has several features useful for the implementation of the path inference project:

- The master node is replicated using ZooKeeper in order to have a fault-tolerant

system

- There is a Web User Interface to keep track of the status of the different nodes
- The tasks are isolated using Linux containers, hence each task has its virtual environment with its own processes.

Here is a schema explaining the structure of Mesos:

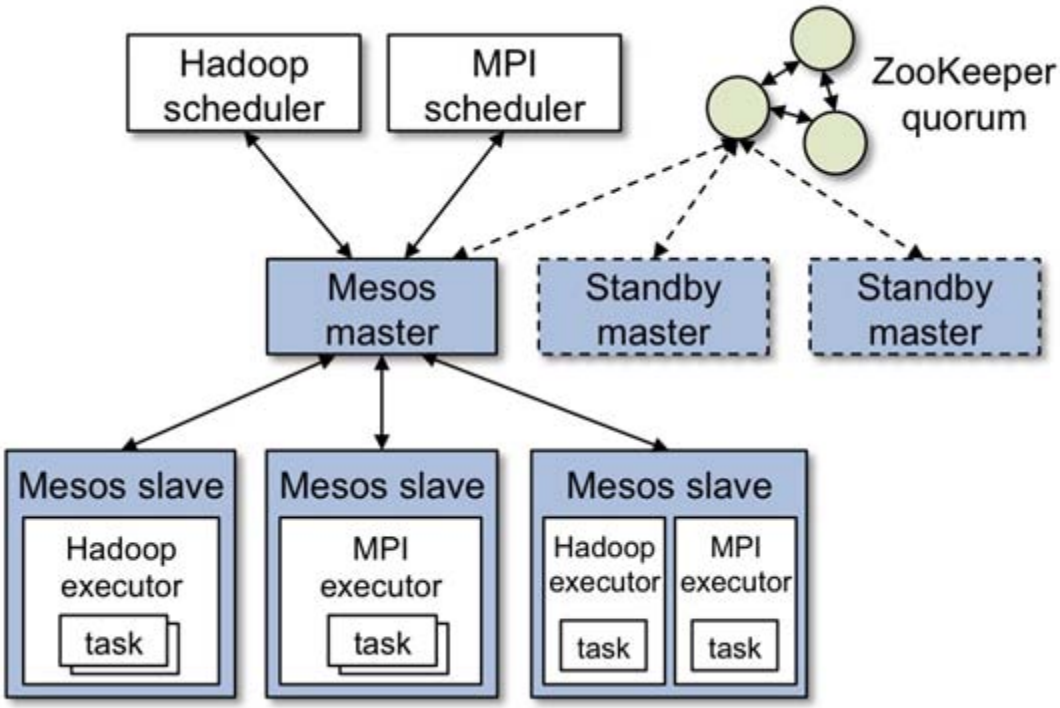


Figure 3.3. Mesos Structure [13]

The main component of Mesos is the Mesos master, which is a master daemon that manages the jobs running on slave daemons. The Mesos master is replicated using ZooKeeper in order to be a fault-tolerant (in Figure 3.3, each duplicate is called a standby master).

Each slave daemon runs on a different assigned node in the cluster. The master manages the sharing of resources among the different frameworks. Each framework running on top of Mesos has its own scheduler and executor. The scheduler takes care of getting the resources needed for the job and the executor runs the framework's tasks on the slave nodes.

Spark

Spark is a cluster computing framework implemented in Scala. Spark manages its own execution and scheduling. The main specificities of Spark are:

- The types of shared variables are limited to read-only variables and accumulators. An accumulator is a value that can be updated using an operation such as an addition. Spark limits the type of shared variables to those easily implemented in a distributed

system.

- Parallel loops over distributed datasets are possible with the condition that the distributed dataset can be reused among different parallel loops (not the case in Hadoop, for example). A series of iterations could hence work on the same distributed dataset.

The second property is extremely useful for this effort. Indeed, for each iteration of the Expectation-Maximization Algorithm, the dataset processed needs to be available in the loop. In other cluster computing frameworks such as Hadoop, each iteration is computed as a MapReduce job but this means that each job must reload the data from the disk. This behavior is not very efficient and Spark proposes a good alternative. It consists of using read-only resilient distributed datasets (RDD). An RDD is a collection of objects partitioned across machines, and can be rebuilt if a partition is lost. This partitioning allows the users to store an RDD across several machines and reuse it in multiple parallel jobs (Map-Reduce types). The Spark/Mesos structure fits well to the path inference project type consisting of running the EM algorithm on a shared dataset in several parallel jobs. The use of Spark in this case presents some improvements in terms of performance (10x better than Hadoop for classical iterative machine learning jobs [349]).

Deployment

The deployment of the path inference project on the cluster consists of two main steps:

- Deployment of the Master/Slave structure: The mpirun tool is used to deploy the structure. The first step is to retrieve the IP of each node requested for the jobs. The first node is used as a master node whereas the rest of the nodes are the slave nodes which will run the spark jobs. The IP and hostname of the nodes are retrieved and written into configuration files. The mpirun call will then run a script which will start the master on its corresponding node as well as the slaves in parallel.
- Parallelization of the jobs: In order to run the jobs in parallel on each slave, the parallelization tools implemented in Spark are be used. The intervals are first created; they will be used to split the data in different samples. Each task associated to a sample is then launched in parallel among the slave nodes using the spark function *parallelize()*. The jobs are now distributed among the available slaves after being assigned by the Mesos allocation module.

Database settings

Two options for the database implementation were used on the cluster. The database directly on NERSC is used for the NERSC implementation and the database dev in the Mobile Millenium System is used for the CITRIS implementation.

Implementation on NERSC

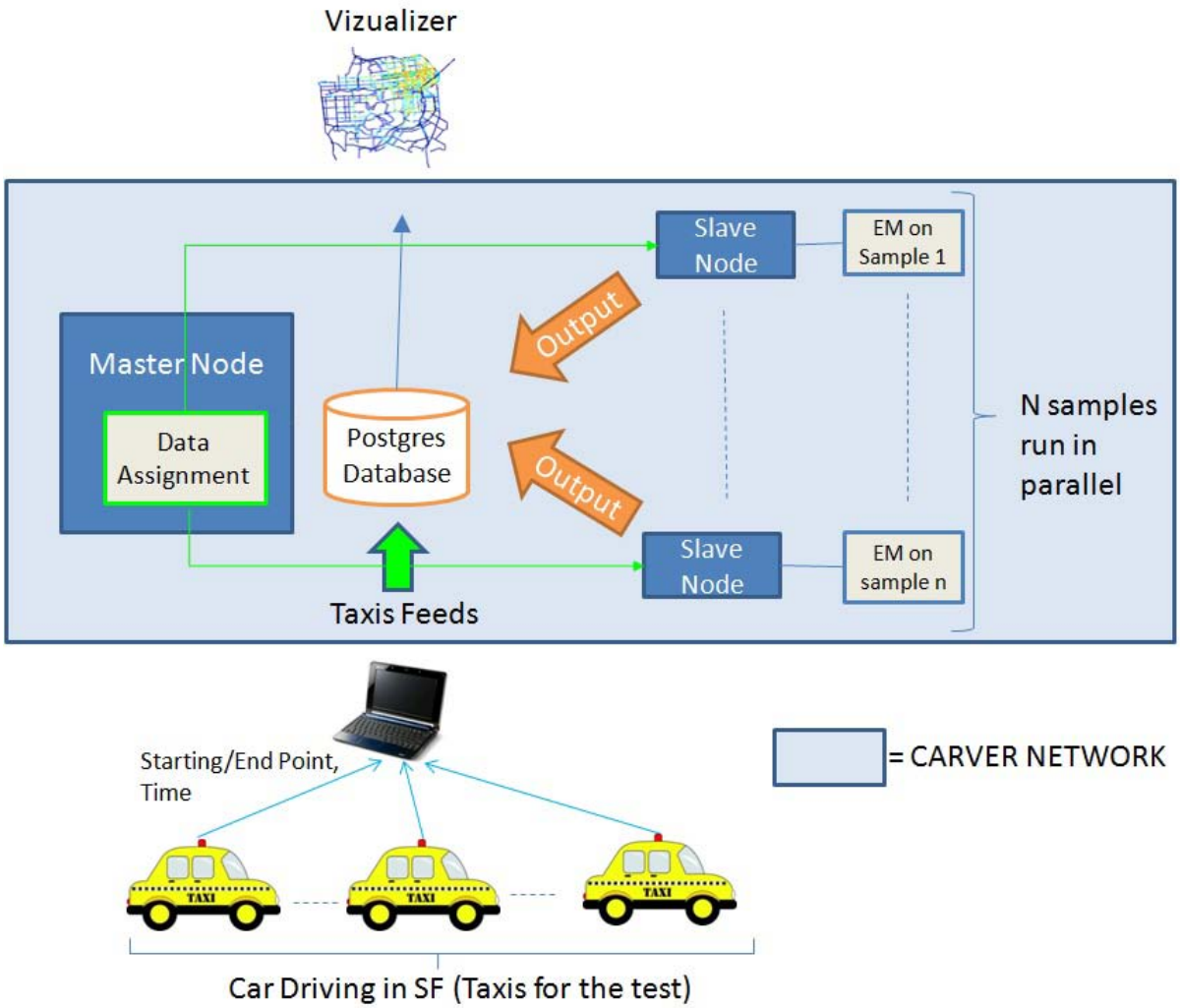


Figure 3.4. Structure of the Path Inference Project on NERSC

Implementation on CITRIS

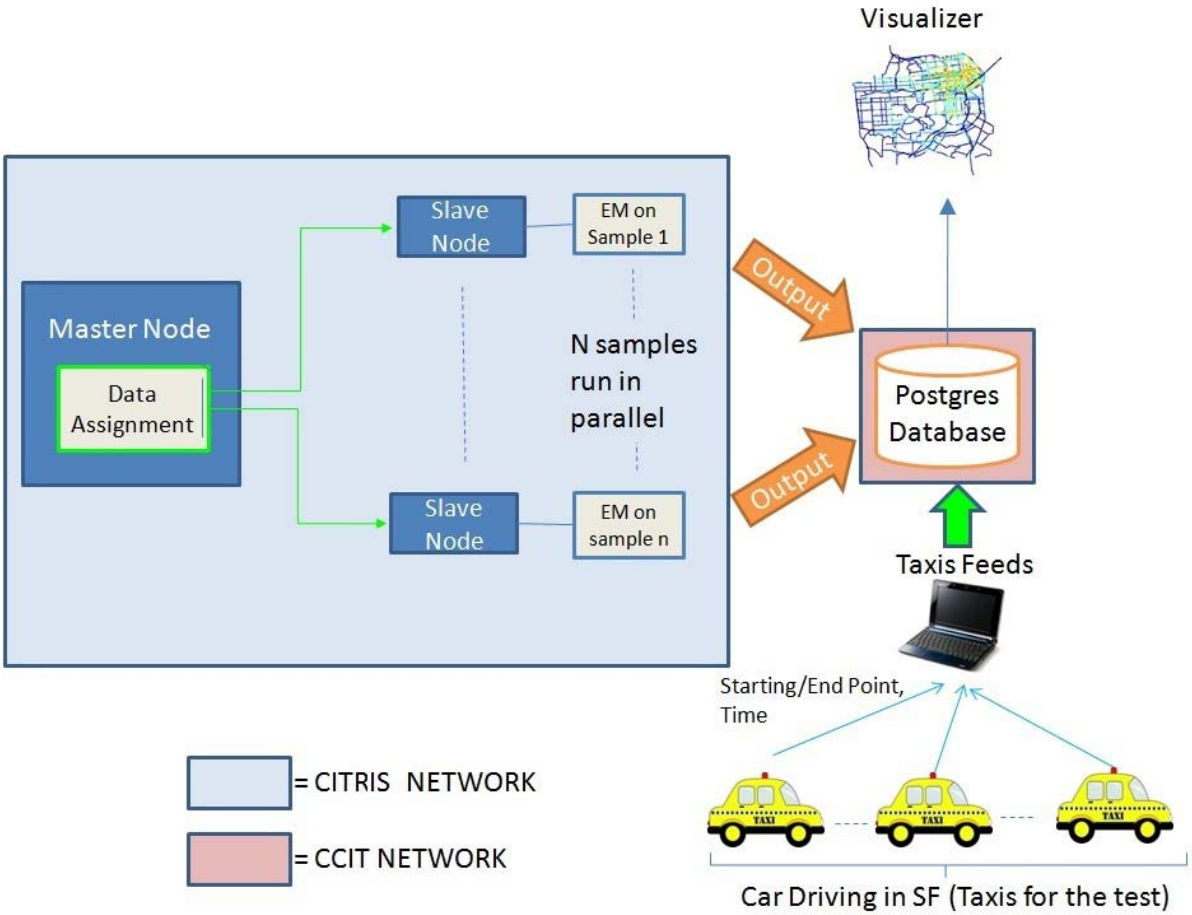


Figure 3.5. Structure of the Path Inference Project on CITRIS

10.4. First Parallelization Effort

In this first parallelization effort, the path inference algorithm was tested to show the time performance improvement made possible with a cluster implementation. The path inference algorithm was implemented on the NERSC and CITRIS clusters and the tests were run on both of them. The main objective of implementing the system on CITRIS and on NERSC was to be able to compare the performances of both clusters and to test the advantages of having a database on the cluster (NERSC) or outside the cluster (CITRIS).

In order to produce a pure comparison of the database structures, the implementation on NERSC was also modified so that it could be tested in two cases: (1) where it reads and writes on the local database on NERSC, and (2) where it reads and writes on the dev database in the Mobile Millenium System.

10.4.1. Performance Testing

In order to see the difference in computing time, we ran the same amount of data available on different configurations using a different number of slaves. The non-parallelized implementation corresponds to one slave running the whole dataset.

The values observed are:

- Total Job Time: Total time taken to finish all the jobs on all the slaves
- Time for finding data: Total time needed to filter the data corresponding to the interval assigned to each slave
- Time for computing the trajectory (TrajectoryTime): Total Time taken to compute the trajectories by each slaves
- Number of Analyzed Points
- Processing Throughput (Tput) = NumberOfAnalyzedPoints/TrajectoryTime
- System Throughput (Tput) = NumberOfAnalyzedPoints/TotalJobTime

The different tests were run using one, three or six slaves. The number of slaves could of course be increased in the future when a special allocation on the cluster is assigned to our project. Each slave had 20GB of memory available for its task.

10.4.2. Results

Cluster Name	NERSC writing with one slave on the portal-auth database on NERSC						
Time Interval of the dataset	Total Job Time [s]	Time for finding data [s]	Trajectory Time [s]	Chunk length [s]	# of analyzed points	Processing Tput [pts / s]	System Tput [pts / s]
1h30	153.10	4.99	4.31	5400	37517	8700.60	245.04
3h	524.76	65.55	34.90	10800	79813	2287.10	152.09
6h	644.33	184.55	210.03	21600	163917	780.45	254.39
9h	1490.21	722.75	531.87	32400	200846	377.62	134.78

Table 4. Test of the Path Inference project: one slave on NERSC writing on the NERSC DB

Cluster Name	NERSC writing with 3 slaves on the portal-auth database on NERSC						
Time Interval of the dataset	Total Job Time [s]	Time for finding data [s]	Trajectory Time [s]	Chunk length [s]	# of analyzed points	Processing Tput [pts / s]	System Tput [pts / s]
1h30	207.95	9.52	6.37	1800	37517	5887.79	180.40
3h	315.17	14.37	9.57	3600	79813	8339.92	253.23
6h	631.56	47.47	31.76	7200	163917	5161.12	259.54
9h	596.83	91.24	54.33	10800	248032	4563.37	415.58

Table 5. Test of the Path Inference project: 3 slaves on NERSC writing on the NERSC DB

Cluster Name	NERSC writing with six slaves on the portal-auth database on NERSC						
Time Interval of the dataset	Total Job Time [s]	Time for finding data [s]	Trajectory Time [s]	Chunk length [s]	# of analyzed points	Processing Tput [pts / s]	System Tput [pts / s]
1h30	133.16	10.64	10.15	900	37517	3695.53	281.75
3h	326.86	16.05	14.27	1800	79813	5591.49	244.17
6h	474.49	27.43	22.64	3600	163917	7239.19	345.46
9h	497.08	41.38	36.28	5400	248032	6837.36	498.98
24h	396.65	49.50	40.72	14400	333425	8189.04	840.60

Table 6. Test of the Path Inference project: six slaves on NERSC writing on the local database on NERSC

Cluster Name	CITRIS writing with six slaves on the dev database						
Time Interval of the dataset	Total Job Time [s]	Time for finding data [s]	Trajectory Time [s]	Chunk length [s]	# of analyzed points	Processing Tput [pts / s]	System Tput [pts / s]
1h30	107.45	8.79	9.35	900	36061	3856.38	335.61
3h	235.40	13.92	14.30	1800	73994	5175.49	314.34
6h	1945.02	81.59	186.93	3600	144529	773.16	74.31
9h	4037.36	855.33	1037.29	5400	198809	191.66	49.24

Table 7. Test of the Path Inference project: six slaves on CITRIS writing on the dev database

Cluster Name	NERSC writing on the portal-auth database on NERSC					
# of Slaves and DB Used	3 slaves writing on dev			3 slaves writing on NERSC		
Time Interval of the dataset	1h30	3h	6h	1h30	3h	6h
Total Job Time [s]	1673.16	1155.95	1420.63	207.95	315.17	631.56
Time for finding data [s]	7.25	22.13	56.30	9.52	14.37	47.47
Trajectory Time [s]	6.40	9.73	64.64	6.37	9.57	31.76
Chunk length [s]	1800	3600	7200	1800	3600	7200
# of points analyzed	37517	79813	163917	37517	79813	163917
Processing Tput [pts / s]	5863.55	8204.46	2536.00	5887.79	8339.92	5161.11
System Tput [pts / s]	22.42	69.04	115.38	180.40	253.23	259.54

Table 8. Test of the Path Inference project on NERSC writing on the CCIT database or local database on NERSC

10.4.3. Analysis of the tests

The two main values from the tests are the Processing Throughput and the System Throughput, as shown in Tables 4, 5, and 6.

The processing throughput represents the pure efficiency of the processing. It indicates the relationship between the number of points analyzed and the time needed to compute the trajectories related to these points.

The system throughput represents the efficiency of the whole implementation. It shows the relationship between the number of points analyzed and the time needed to process the whole task, including the communication time, the reading and writing time as well as the time needed for memory management (garbage collection, for example).

The following graph represents the evolution of the processing throughput on NERSC using the local database on NERSC. Four measures were taken with a total data set covering 1h30, 3h, 6h and 9h of the simulation representing respectively 37517, 79813, 163917 and 248032 points analyzed. A test with six slaves running on a data set covering 24h was also launched, analyzing a total of 333425 points.

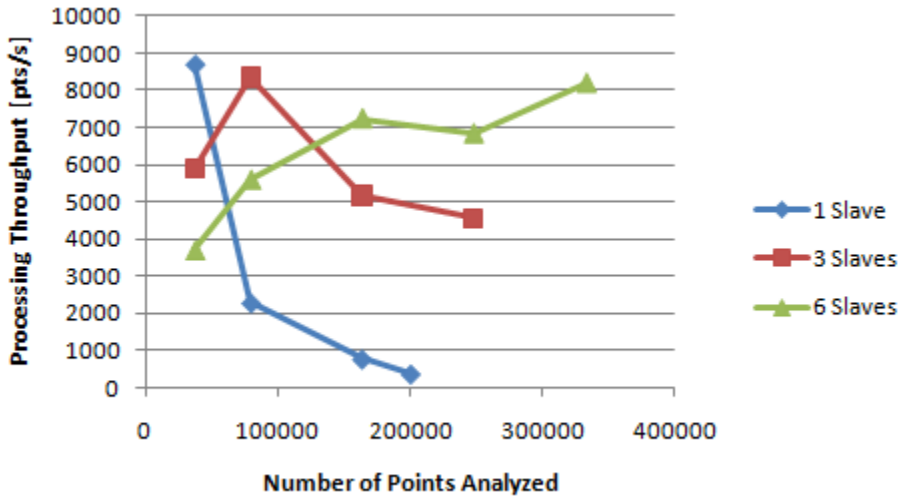


Figure 4.1. Graph of the Processing Throughput on NERSC using the local database on NERSC

The throughput shown in Figure 4.1 is higher for large data samples when having more slaves in parallel (green line). On the other hand, the throughput is higher when having one slave on a small dataset (1h30). This can be explained by the fact that each task first needs to create a cache of the 20 best paths between each pair of points in the input data. For a small dataset, the time needed for caching this data in comparison to the real trajectory computing time is large enough to “hide” the benefits of the parallelization of the tasks.

The next graph represents the evolution of the system throughput on NERSC using the local

database on NERSC, which is the structure we wanted to implement initially. Four measures were taken with a total data set covering 1h30, 3h, 6h and 9h of simulation, representing respectively 37517, 79813, 163917 and 248032 points analyzed. A test with six slaves running on a data set covering 24h was also launched, analyzing a total of 333425 points.

We can observe that the throughput of the whole system is much better with a higher number of slaves running in parallel. Again, for a small dataset such as 1h30, the system throughput is better with one slave than with three slaves. Again, this can be explained by the fact that the time needed to cache the best paths as well as the time needed to setup the parallelization of the tasks is high enough to "hide" the advantage of the parallelization.

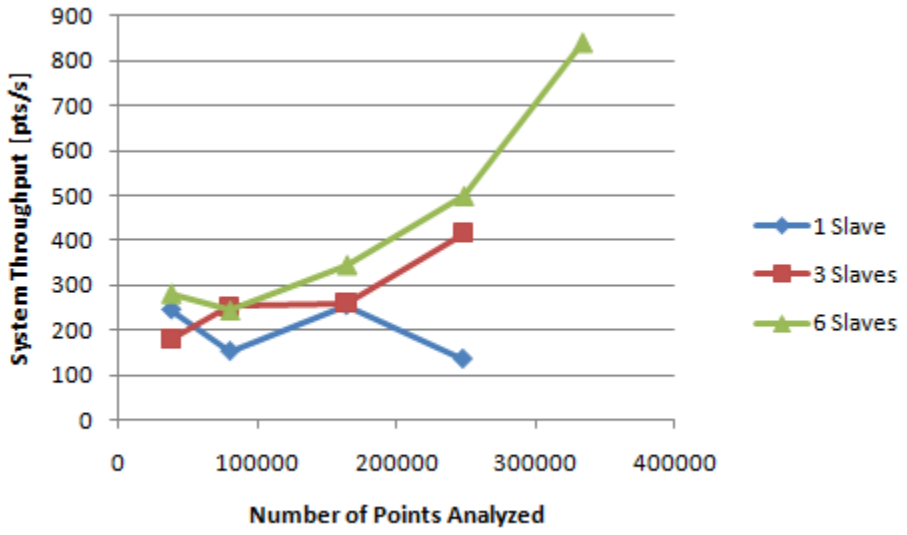


Figure 4.2. Graph of the System Throughput on NERSC using the local database on NERSC

These graphs highlight the advantage of the parallelization of the tasks. The performance with six slaves instead of one slave (a non-parallelized implementation) is improved by a factor of nearly two in terms of the whole system performance. This improvement can even be bigger when tested on a larger scale by running the job on a larger number of nodes in order to have more slaves in parallel as well as using a larger dataset.

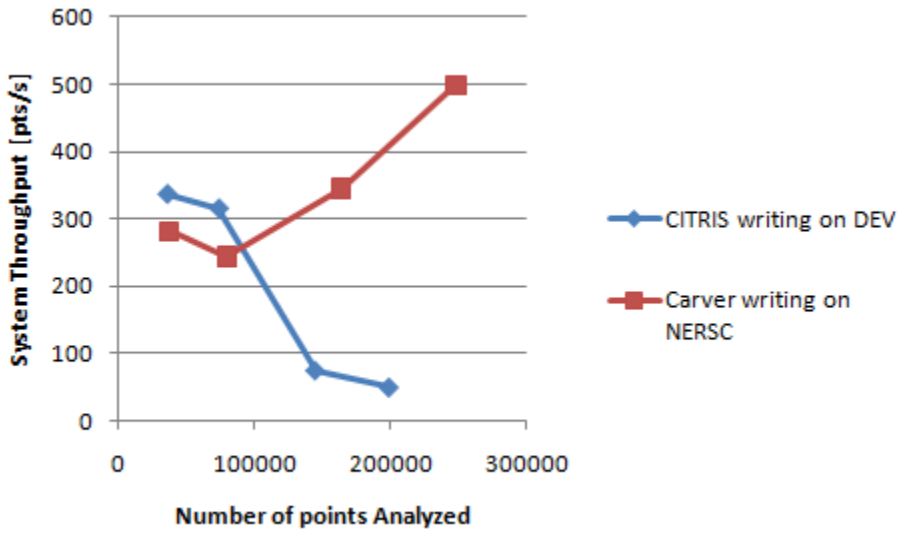


Figure 4.3. Graph comparing the System Throughput on NERSC (DB on the cluster) and CITRIS (DB out of the cluster network)

The second observation that can be made from Table 6 and 7 is to compare the performances of the implementation on CITRIS and NERSC. CITRIS was writing on the dev database whereas NERSC was writing in the database directly on the cluster (See Figure 4.3). We can see that the performances are much better with the database directly on the cluster when a large dataset is used. The performances of the CITRIS cluster are good with a dataset over an interval smaller than 3h. This good performances can be explained by the fact that the ssh tunnel is not yet a bottleneck for this quantity of data. Indeed, we can see that the CITRIS cluster is less effective with a large dataset than NERSC is.

Another observation that can be made from the tests is displayed in Table 8. This table presents a comparison of the performance on NERSC when having the database directly on NERSC (on the cluster) or in the Mobile Millenium System. This test produces a comparison of the "pure" performance of each database. (See Figure 4.4). The system throughput is the value analyzed on this graph since this is the throughput affected by the modification of the database implementation. Indeed, the system throughput shows the relationship between the number of points analyzed and the time needed to process the whole job, including reading/writing in the database. The System Throughput is calculated having three slaves in parallel running tasks on a dataset of 1h30, 3h and 6h.

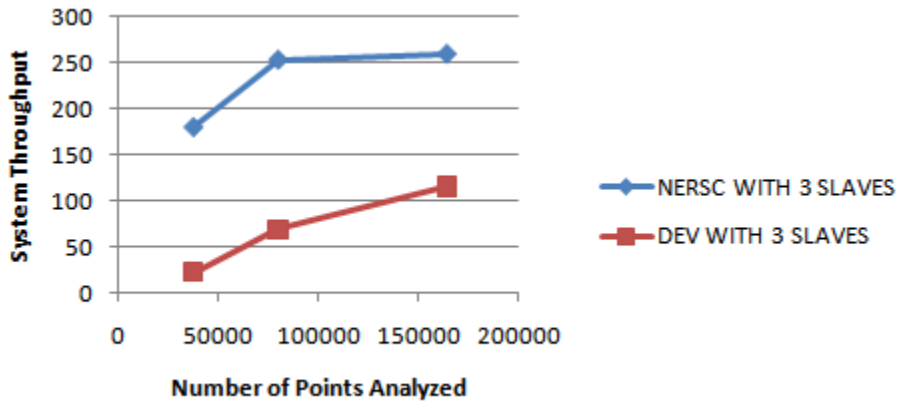


Figure 4.4. Graph of the System Throughput on NERSC using different DBs

The system throughput is improved by a factor of four to eight depending on the size of the dataset. This improvement seems reasonable; the ssh tunnel used for accessing the MS database in the Mobile Millenium System becomes a bottleneck as the amount of data to be read or written is increased. The dev database on the CCIT network is also less stable than the one on the cluster since it is used by several projects, hence its performance is variable depending on the number of accesses requested at the time. In general, all these test results are influenced by the actual usage of the cluster nodes as well as the database access from other projects but it provides a good indication of the behavior in the cluster in order to verify if the parallelization is working efficiently.

10.4.4. Summary of the tests

These tests demonstrated the advantages of parallelizing the path inference algorithm. The behavior of the system is logical and the improvements in terms of performance are more obvious when working on larger data sets with a larger number of slaves.

The advantages of the implementation of the database directly on the cluster was also highlighted with these tests. The performance was significantly improved with the database on NERSC since the latency due to an ssh tunnel was avoided and the communication time between the nodes and the database was reduced.

10.5. Second Parallelization Effort

In the second parallelization effort, the EM algorithm was implemented on the Amazon EC2 and NERSC clusters and tests were run on both of them. Although the EM algorithm can readily be expressed using MapReduce, Dryad or other cluster computing frameworks, we found that simply expressing the algorithm in a parallel manner was not enough to achieve either good performance or scalability. To obtain more acceptable performance, three substantial optimizations were necessary: (1) in-memory computation, (2) efficient broadcast of large objects, and (3) optimized access to the application's storage system.

10.5.1. In-Memory Computation

We chose to implement our EM algorithm using Spark [348], a cluster computing framework developed at Berkeley for iterative applications. Spark offers several benefits. First, it provides a high-level programming model (map, reduce, and other operations using a language-integrated syntax similar to DryadLINQ [347]), saving substantial development time over lower-level programming models like MPI. Second, Spark programs are written in Scala [25], a high-level language for the Java VM, which allowed us to integrate with the mostly-Java codebase of *Mobile Millennium*.

Much like Twister [136] and Piccolo [274], Spark was designed to support iterative algorithms more efficiently than pure data flow frameworks like MapReduce and Dryad by letting users cache data in memory on the worker nodes between iterations. Iterative applications implemented in MapReduce, Dryad, or other pure data flow frameworks have to run as a series of distinct jobs, each of which reads state from disk and writes it back out to disk, incurring significant overhead due to I/O and object serialization. In contrast, Spark lets users build in-memory "distributed datasets" storing Scala objects (e.g., the road network or the vehicle location observations) and reuse them efficiently across iterations.

Our results validate the benefit of in-memory storage for iterative applications: we see a 2.8× speedup from caching the location observations in memory across iterations (Section 10.5.4). Nonetheless, we found that simply having in-memory storage facilities available was insufficient to achieve good performance in a complex application like traffic estimation.

One of the main challenges we encountered was *efficient utilization* of memory. Unlike simpler machine learning applications that cache and operate on large numeric vectors, our application cached data structures representing paths traveled by vehicles or sets of links parameters. When we first implemented these data structures using idiomatic Java constructs, such as linked lists and hashtables, we quickly exhausted the memory on our machines, consuming more than 4× the size of the raw data on disk. This happens because the standard data structures, especially pointer-based ones, incur considerable storage overhead per item. For example, in a Java LinkedList, each entry costs 24 bytes (for an object header and pointers to other entries) [249], whereas the values we stored in these lists were often 4-byte `int`s or `float`s. With this much overhead, running a learning algorithm in memory can be much costlier than anticipated; indeed, our first attempts ran very slowly because they were constantly garbage-collecting.

Solution and Lessons Learned: We improved our memory utilization by switching to array-backed data structures where possible for the objects we wanted to cache, losing some convenience in the process. One difficult part of the problem was simply recognizing the cause of the bloat: Java (and Scala) programmers are typically unaware of the overhead of simple collection types.

We believe that designers of in-memory computing frameworks can do a lot to help users utilize memory efficiently. In particular, it would be useful for frameworks to provide libraries that expose an idiomatic collection interface but pack data efficiently, as well as tools for pinpointing sources of overhead. We are developing both types of tools for Spark.

10.5.2. Broadcast of Large Parameter Vectors

Many parallel machine learning algorithms need to broadcast data to the worker nodes, either at the beginning of the job or on each iteration. For example, in our traffic estimation algorithm, we broadcast the road network to all the nodes at the start of the job.

In the simple machine learning algorithms commonly evaluated in the systems literature, such as *k*-means and logistic regression, these broadcasts are small (hundreds of bytes each). In our application, they were considerably larger: about 38 MB for the road network of the Bay Area. We have seen even larger parameter vectors in other applications: the spam classifier in [318] had a parameter vector hundreds of MB in size, with a feature for each of several million terms. Furthermore, this vector needed to be re-broadcast after each iteration.

Initially, our application performed poorly because we packaged the parameter vectors needed with each task (partition of a job) sent to the cluster, which was the default behavior in Spark. The master node's bandwidth became a bottleneck, capping the rate at which tasks could be launched and limiting scalability.

Solution and Lessons Learned: To mitigate the problem of large parameter vectors, we added an abstraction called *broadcast variables* to Spark, which allows a programmer to send a piece of data to each slave only once [348]. To the programmer, broadcast variables look like wrapper objects around a value, with a `get()` method that can be called to obtain the value. The variable's value is written once to a distributed file system, from which it is read once by each node the first time that a task on the node calls `get()`. We illustrate the syntax for broadcast variables in Figure 5.1.

```
net = //read road network

observations.map(
  ob => process(ob, net)
).reduce(...)
```

a) Original

```
net = //read road network
bv = spark.broadcast(net)

observations.map(
  ob => process(ob, bv.get())
).reduce(...)
```

b) With Broadcast Variables

Figure 5.1. Example Spark syntax showing how to use broadcast variables to wrap a large object (`net`) that should only be sent to each node once.

As we show in Section 10.5.4, broadcast variables improved our performance of our data loading phase by about 4.6×, leading to an speedup of 1.6× for the overall application.

For larger parameter vectors, such as the O(100 MB) vector in the spam classification job above,

even reading the data once per node from a distributed file system is a bottleneck. This led us to implement more efficient broadcast methods in Spark, such as a BitTorrent-like mechanism [99]. Because many real-world machine learning applications have large parameter vectors (with features for each word in a language, each link in a graph, etc), we believe that efficient broadcast primitives will be essential to support them.

10.5.3. Access to On-Site Storage System

One of the more surprising bottlenecks we encountered was access to our application's storage system. *Mobile Millennium* uses a PostgreSQL database to host information shared throughout our pipeline, including the road network and the observations received over time. We chose PostgreSQL for its combination of reliability, convenience, and support for geographic data through PostGIS [23]. While PostgreSQL had served our on-site pipeline without problems, it performed very poorly under the access pattern of our parallel implementation. Our application initially spent more than 75% of its time waiting on the database.

The problem behind this slowdown was the *burstiness* of the access pattern of the parallel application. The database had to service a burst of hundreds of clients reading slices of the observation data when the application started, as well as a similar burst of writes when we finished computing. The total volume of data we read and wrote was not large—about 800 MB per job—so it should have been within the means of a single server. However, the contention for disk access between the simultaneous queries slowed them down dramatically.

Solution and Lessons Learned: We ultimately worked around the problem by exporting the data to a Hadoop file system (HDFS) instance on EC2. Though this approach removed the bottleneck, it also created management challenges, as we must manually keep HDFS consistent with the database. Specifically, *Mobile Millennium* receives new data from taxi cabs every few minutes, so we would prefer a solution that lets us access this data from the cloud as it arrives.

We believe that database implementers can do a lot to support the bursty patterns of requests from workers in parallel applications. Neither the amount of data we read nor the amount of data we wrote was beyond the means of a single server, but the bursty access pattern was simply not well-supported by the engine. Enabling developers to use the same storage system for both on-site and cloud applications would be a key step in making the cloud an accessible platform for parallel data processing, and given the near-ubiquitous use of RDBMSes in existing applications, they are a natural place to start.

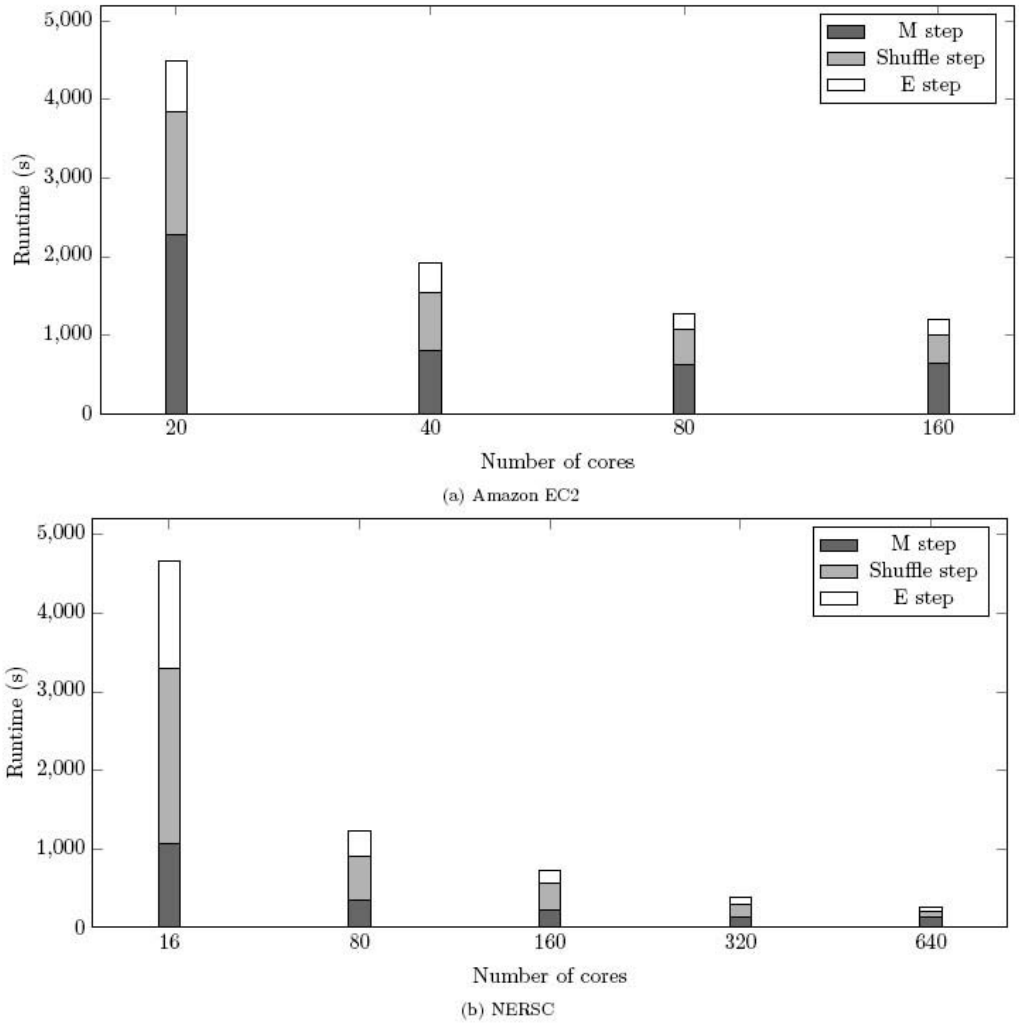


Figure 5.2. Running time experiments on different clusters.

10.5.4. Performance Evaluation

We now evaluate how much the cloud implementation and its associated memory, broadcast and storage optimizations (Section 5) helped with scaling the *Mobile Millennium* EM traffic estimation algorithm.

We originally designed and prototyped the algorithm in the Python programming language. Then, we reimplemented the algorithm in Scala, to integrate it with the rest of the *Mobile Millennium* system. However, working on a single computer quickly revealed its limitations: the code would take 40 minutes per iteration to process a single 30 minute time interval in the data! Moreover, the amount of memory available on a single machine would limit the number of

observations considered, as well as the number of samples generated. Distributing the computation across machines provides a twofold advantage: each machine can process computations in parallel, and the overall amount of memory is much greater. To understand how restricted the single-machine deployment was, we could only generate 10 samples for each observation in the E-step in order for the computation to stay within the machine's 10 GB limit. Because the single-node implementation cannot generate enough E-step samples to create a good approximation of the travel time distribution, the accuracy of the algorithm was limited as well. By contrast, using a 20-node cluster and letting the E-step generate 300 samples per observation, the computation was an order of magnitude faster, and the accuracy of the predicted models increased significantly.

Scaling. First, we evaluated how the runtime performance of the EM job could improve with an increasing number of nodes/cores. The job was to learn the historical traffic estimate for the San Francisco downtown for a half-hour time-slice. This data included 259,215 observed trajectories, and the network consisted of 15,398 road links. We ran the experiment on two cloud platforms: the first was using Amazon EC2 `m1.large` instances with 2 cores per node, and the second was a cloud managed by NERSC with 4 cores per node. Figure 5.2 shows near-linear scaling on EC2 until 80–160 cores. Figure 5.2 shows near-linear scaling for all the NERSC experiments. The limiting factor for EC2 seems to have been erratic network connectivity. In particular, some tasks were lost due to repeated connection timeouts.

Individual optimizations. We evaluated the effects of the individual optimizations discussed in Section 5. This time, we ran the experiments on a 50-node Amazon EC2 cluster of `m1.large` instances, and used a data set consisting of 45×10^6 observations split into 800 subtasks.

With respect to the data loading (Section 3) we looked at three configurations: (a) connecting to the main database of *Mobile Millennium* which stores the master copy of the data, (b) connecting to a cloud-hosted version of the *Mobile Millennium* DB, and (c) caching data from the main DB to the cloud's HDFS. Table 9 shows the throughput for loading data under each configuration, and shows that our final solution (c) shows a three orders of magnitude improvement over the original implementation using (a).[12]

Configuration	Throughput
Connection to main DB	239 records/sec
Connection to cloud-hosted DB	6,400 records/sec
Main DB data cached in HDFS	213,000 records/sec

Table 9. Data loading throughput.

[12]Although we do not report the extraction and preprocessing overhead for (c), this initial cost is amortized over the number of repetitions we perform for the experiments.

Configuration	Load time	E step	Shuffling	M step
Baseline	468	437	774	936
Single core	4073	6276	18578	7550
No cache	0	2382	2600	835
No broadcast	2148	442	740	1018

Table 10. Comparing runtimes with different settings: a baseline configuration (all optimizations turned on), single core (same experiment, run on a single thread), no cache and no broadcast

To evaluate the benefit of in-memory computation (Section 1), we compared the run times of the EM job without caching (i.e., reloading data from HDFS on each iteration) and with in-memory caching (Table 10). Without caching, the runtime was 5,800 seconds. With caching, the runtime was reduced to 2,100 seconds, providing a nearly 3× speedup. Most of this improvement comes from reducing the runtime of the E-step and the shuffle step since they read the cached observations. The M-step does not improve because it reads newly-generated per-link samples (which have to be regenerated on each iteration as per Section 2), and the current implementation of shuffle writes its outputs to disk to help with fault tolerance.

Finally, we explore the benefit of broadcasting parameters (Section 2). A copy of the road network graph must be available to every worker node as it loads and parses the observation data, so broadcast is crucial. To this end, we evaluated how long it took to load 45 million observations over a 50-node cluster when (1) a copy of the road network graph is bundled with each task and (2) the network graph is broadcast ahead of time. The network graph for the Bay Area was 38 MB, and it took 8 minutes to parse the observations using a broadcast network graph — by contrast, the loading time was 4.5 times longer without broadcasting.

10.6. Related Work

There has recently been great interest in running sophisticated machine learning applications in the cloud. Chu et al. showed that MapReduce can express a broad class of parallel machine learning algorithms, and that it provides substantial speedups on multicore machines [101]. However, as we discussed in this chapter, these algorithms encounter performance challenges when we want to scale beyond a single machine and run them on a public cloud. The main remedies to these challenges involve exploiting data locality and reducing network communication between nodes.

In the systems literature, Twister, Spark, HaLoop and Piccolo provide MapReduce-like

programming models for iterative computations using techniques such as in-memory storage [136,348,82,274]. GraphLab and Pregel also store data in memory, but provide a message-passing model for graph computations [234,235]. While these systems enable substantial speedups, we found that issues other than in-memory storage, such as broadcast of large parameter vectors, also posed challenges in our application. In contrast with the simple benchmarks that are commonly employed, we highlight these challenges in the context of a complex real-world application.

Recent work in large-scale machine learning has addressed some of the algorithmic issues in scaling applications to the cloud. McDonald et al. [237] discuss distributed training strategies over MapReduce where data is partitioned across nodes, and nodes perform local gradient descent before averaging their model parameters between iterations. Other studies about distributed dual averaging optimization methods [134] and distributed EM [338] explored the network bandwidth savings of optimization algorithms that restrict the communication topology of the worker nodes.

Our experiences scaling up some key *Mobile Millennium* traffic estimation algorithms to run on the cloud presented issues that we believe will also apply to other complex machine learning applications. Our work affirmed the value of in-memory computation for iterative algorithms, but also highlighted three challenges that have been less studied in the systems literature: efficient memory utilization, broadcast of large parameter vectors, and integration with off-cloud storage systems. All three factors were crucial for performance. Our experiences with *Mobile Millennium* have already influenced the design of the Spark framework.

Part III

Mobile Millennium Highway Model

11. The Mobile Millennium Highway Model

A detailed explanation of the *Mobile Millennium* Highway Model begins here and spans the following three chapters. The first iteration of this model was showcased in the *Mobile Century* experiment and is explained briefly in the *Mobile Century* Final Report [269]. Substantial practical implementation improvements, theoretical refinements, and extensions were made during the course of the *Mobile Millennium* Project. Detailed explanations of key innovations span Chapters 12 though 14.

Prior to this work, there was no known velocity model for traffic. All such models were based on count and mass conservation. Velocity models are important because it is not yet possible to directly reconstruct counts from vehicle probes with the amount of data available today. The velocity model described here has been proven consistent with mass conservation, and represents a breakthrough for traffic information systems. Due to specific features of the model, the estimation procedure could not rely on traditional filtering techniques. As a result, the estimation procedure relies on *ensemble Kalman filtering* (EnKF), which bypasses challenges of the model essential to capturing traffic well, but are not a serious problem for the mathematical treatment of the estimation.

This chapter introduces the framework for the *Mobile Millennium* Highway Model in the context of *Distributed parameters systems* (DPS)—a class of systems that arise naturally in large-scale civil and environmental engineering problems. A distributed parameter system is one whose spatial variation of its infinite dimensional state (distributed parameters) plays an important role in the evolution of the system in time. Thus, a distributed parameter system can be used to model contaminants propagating in rivers and estuaries, air quality and pollution dispersion in urban areas, the behavior of structures under wind or seismic loads, and traffic congestion on roadways, to name a few examples. A common mathematical representation of a distributed parameter system is in the form of a *partial differential equation* (PDE).

11.1. Formulations of Distributed Parameters Systems

11.1.1. The forward problem

A simple formulation to study distributed parameters systems, known as the *direct* or *forward problem*, is illustrated in Figure 1.1. First, an abstraction of the physical world must be constructed, typically in the form of a mathematical equation which describes the system's evolution. Second, the system's initial condition, boundary conditions, and model parameters must be defined. Finally, the model is solved or simulated forward in time, so that the behavior of the system can be studied and analyzed.

In Figure 1.1, arrows show how information flows between the physical world, the distributed parameter system abstraction, and the computational model simulations. In particular, information from the physical world enters the model simulations only through the mathematical model.

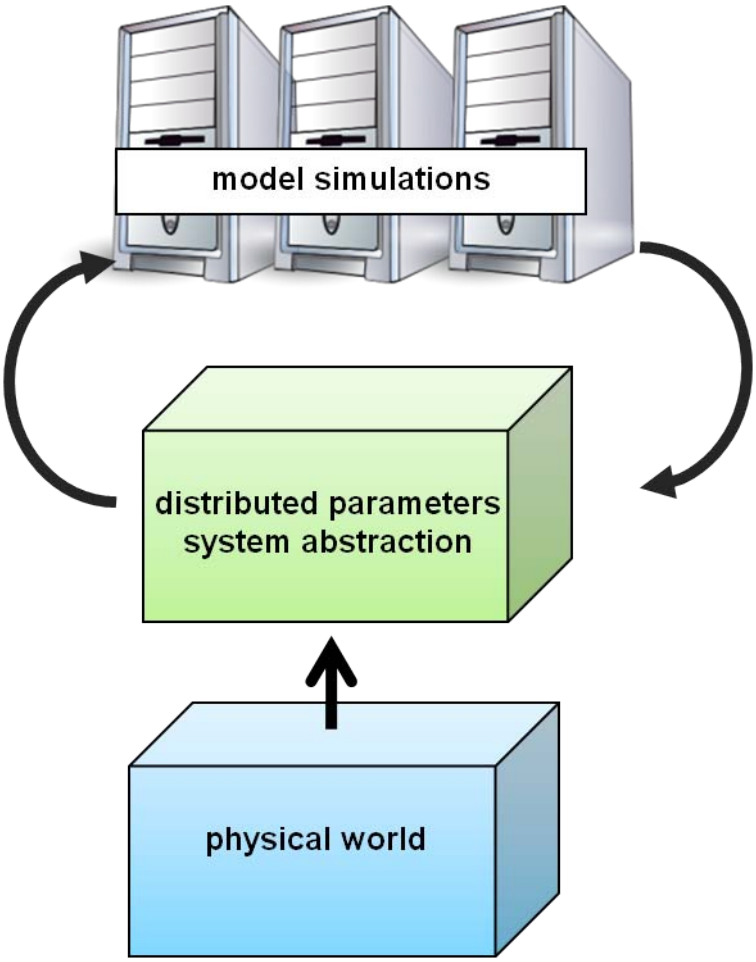


Figure 1.1. The forward problem.

This process has two fundamental limitations which prevent a direct matching between events

occurring in the simulated environment and that which occurs in reality.

- The mathematical model is only an abstraction of reality in which the physical world is approximated and simplified. Thus the model contains some error inherent to the abstraction. This modeling error can be difficult to quantify, and therefore the accuracy of the model may be hard to determine.
- Often, the initial condition, boundary conditions, and model parameters are only known approximately, if at all, which adds to the uncertainty of the model.

11.1.2. The estimation problem

To address the limitations of the forward problem formulation described above, a related *estimation problem* can also be studied. In the estimation problem, the mathematical model is augmented with additional information from the physical world in the form of data from sensors. This process is illustrated in Figure 1.2. Like the forward problem, the physical world is again used to build the mathematical abstraction in the form of a partial differential equation. Moreover, it is also used to generate observations from the physical world through sensor data. The process of combining the model and the data is known as estimation. This process is also referred to as parameter estimation or *inverse modeling* when the goal is to estimate parameters in the system. When the objective is to estimate the state of the system, it is called state estimation or *data assimilation* [225].

Estimation algorithms can be described according to how they incorporate new sensor data into the estimate of the system. When the estimation problem is solved *online*, the algorithm uses sensor data piece by piece as it becomes available, without the need for all of the data at once. On the other hand, an *off-line* or *batch* algorithm requires all measurements at the same time. In practice, online algorithms may achieve estimates more quickly since only a portion of the data is needed at any time, but perhaps at the cost of improved accuracy achieved by batch algorithms.

Estimation algorithms can also be characterized according to the time constraints under which they operate. A *real–time* algorithm has strict deadlines on the timing of the computation, while an algorithm which is not real–time does not. The timing deadline is often a function of the rate at which the physical system evolves, so that information produced by a real–time algorithm can be used to control the physical world before it becomes outdated or obsolete. When this is achieved, the computation infrastructure and the physical world become tightly coupled, creating a *cyber–physical system*.

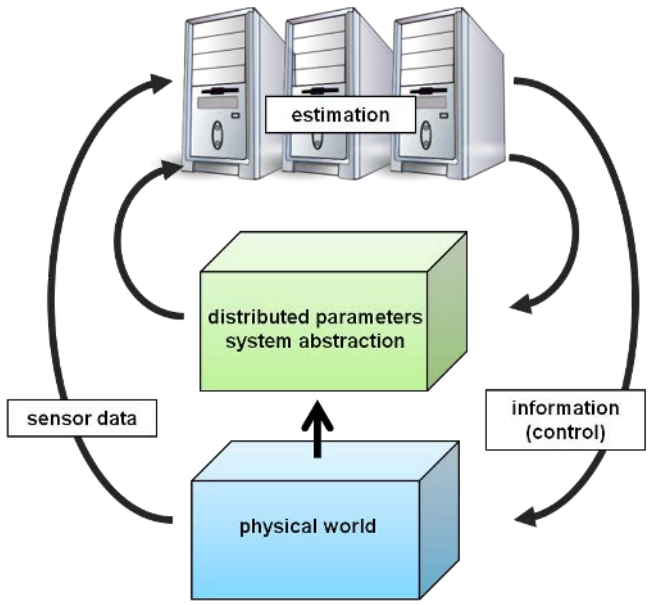


Figure 1.2. The estimation problem, with a feedback loop to the physical world.

One of the fundamental challenges for estimation problems for distributed parameters systems is the acquisition of sensor data, precisely because the system is distributed in space. In the context of traffic monitoring for highways, data acquisition is typically achieved through the placement of dedicated sensing infrastructure deployed in the pavement, such as an inductive loop detector, or sensors adjacent to the infrastructure such as video, radar or RFID. Due to the expense of installation and maintenance, it is difficult to achieve sensor coverage at a global scale.

11.2. Velocity estimation from GPS-equipped mobile devices

Motivated both by the availability of GPS data from mobile devices, and by the need for increased accuracy demanded by mobile Internet services, the goal of this chapter is to outline a framework in which to understand a real-time estimation algorithm for monitoring traffic using velocity data from mobile devices. In this framework a model is constructed to describe the evolution of a velocity field $v(x,t)$ on a highway segment $x \in [0, L]$, which is a distributed parameter system. Vehicles labeled by $i \in \mathbb{N}$ travel along the highway with trajectories $x_i(t)$, and measure the velocity $v(x_i(t), t)$ along their trajectories.

These discrete measurements are then combined with the velocity evolution model, and together they are used to reconstruct or estimate the function $v(x,t)$, in real–time, using an online Ensemble Kalman filtering framework. Fig. 2.1 illustrates the process: the evolution of the velocity field $v(x,t)$ can be depicted as a surface, which is to be reconstructed. A subset of the vehicles is sampled along their trajectories. For illustration purposes in the figure, four vehicles are sampled at time $t = t_m$, which produces four points on the $v(x,t)$ surface which can be used by the algorithm to reconstruct the surface.

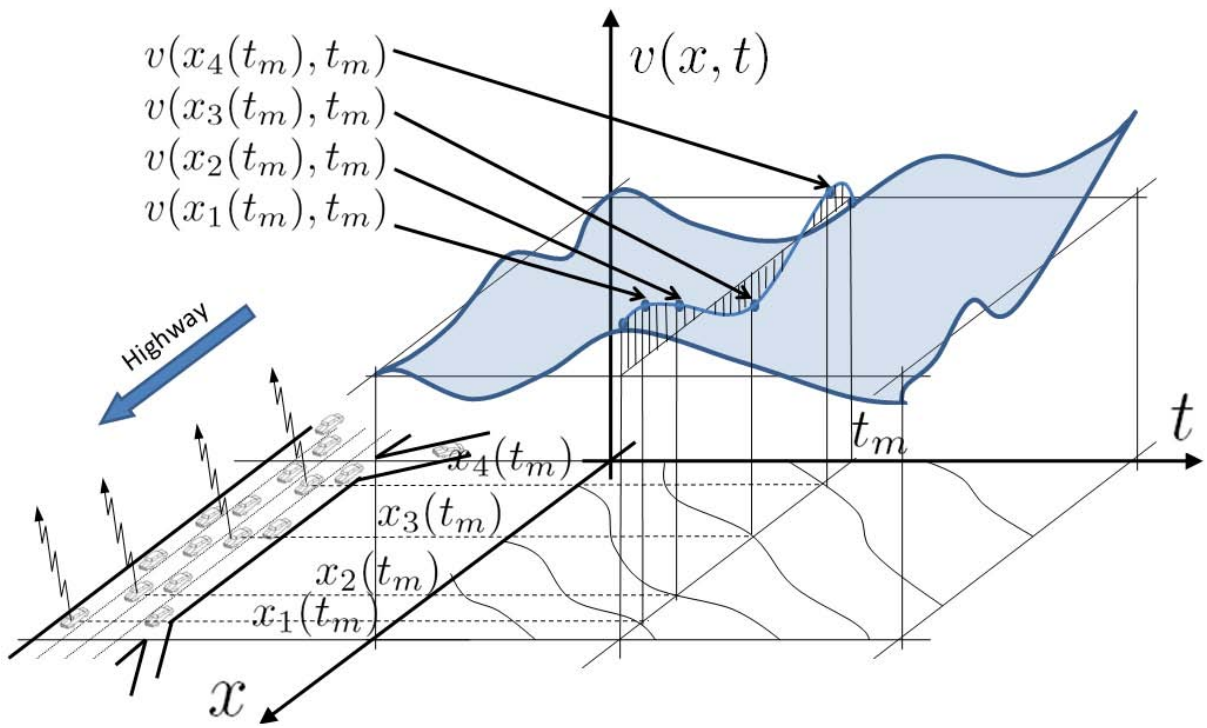


Figure 2.1. Illustration of the distributed velocity field $v(x,t)$ to be reconstructed from GPS samples. Four samples $v_i(x_i(t),t)$ are shown at $t = t_m$, from vehicles i transmitting their data (indicated by up-arrows above the vehicles).

When the surface $v(x, t_m)$ is estimated using measurements up to time $t_1 = t_m$, $v(x, t_m)$ is known as a filtered estimate. Two related problems include prediction and smoothing. If the surface $v(x, t_m)$ is estimated using measurements up to time $t_2 < t_m$, the resulting estimate is known as a prediction. Finally, when the surface $v(x, t_m)$ is estimated using measurements up to time $t_3 > t_m$, the result is known as a smoothed estimate.

To address one important issue in sensing with mobile devices, we consider velocity measurements obtained through a privacy-aware architecture introduced by Nokia, called *virtual trip lines* (VTLs) [190]. VTLs are virtual geographic markers which act as triggers for mobile sensing, and therefore can be viewed as a spatial sampling strategy. The main constraint VTLs place on traffic estimation is that it is not possible to track vehicles (full vehicle trajectories are never disclosed), or identify measurements as belonging to the same vehicle [190].

11.3. Related work

11.3.1. Traffic flow theory

The origins of traffic flow theory dates back to the 1950's with the pioneering works of Lighthill and Whitham [226], and independently Richards [285], who proposed a macroscopic model of traffic based on conservation of vehicles. The model, now known as the *Lighthill–Whitham–Richards* (LWR) PDE, is a nonlinear hyperbolic conservation law. The main

mathematical challenge to the LWR PDE, and more generally systems of conservation laws, is the development of discontinuities (shocks), which can occur in finite time even from smooth initial conditions. The existence and uniqueness of the *Cauchy problem* (i.e. an initial value problem) for conservation laws on infinite domain are achieved through a suitable entropy condition introduced in the seminal works of Oleinik [53] and Kruzkov [215] (See also the existence result of Glimm [164]).

The introduction of a boundary condition for systems of conservation laws was first treated by Bardos, Leroux, and Nedelec [69]. The well-posedness of a scalar *initial boundary value problem* for a conservation law with a convex flux function was addressed by Le Floch [149] and recently by Frankowska [151]. The recent work of Strub and Bayen [309] instantiates the well-posedness of the initial boundary value problem for the LWR PDE explicitly.

Soon after the introduction of the scalar LWR PDE, higher-order traffic models were introduced in an attempt to reconcile some of the deficiencies of first-order models. These models augment the mass conservation equation with a momentum equation, the most notable being the model of Payne [270]. However this model had several deficiencies, including characteristics moving faster than the average velocity of traffic, and vehicles moving backward, as pointed out by del Castillo [90], and in particular Daganzo's "Requiem for second-order fluid approximations of traffic flow" [123]. These problems were addressed and the models were "resurrected" independently by Aw and Rascle [62] and Zhang [352], leading to the class of *Aw–Rascle–Zhang* (ARZ) models.

A class of phase transition models was introduced by Colombo [112], which combines a scalar conservation law in free flow with a 2×2 system of conservation laws in congestion. The global well posedness of phase transition models such as [112] was given by [113]. This model was later extended by Blandin et. al. [76] to simplify the phase transition analysis, and is described in detail in Chapter 23.

In the discrete domain, the *Cell Transmission Model* was introduced by Daganzo [126,127] as a mass conserving traffic model consistent with the LWR PDE using a Godunov discretization [166,220]. Papageorgiou [265] introduced a discrete model which is a modified version of the discretized Payne model.

11.3.2. Traffic estimation

The process of recursively estimating traffic conditions using a traffic flow model and experimental data begins with the 1970's with the early work of Szeto and Gazis [314]. Using a mass conservation equation with flow measurements at the segment edges, they applied an *extended Kalman filter* (EKF) to estimate the traffic density in the Lincoln Tunnel in New York City. The extended Kalman filter is a widely used extension of the recursive minimal variance estimator known as the Kalman filter [208]. For state estimation on nonlinear systems, the model equation and observation equation are linearized to fit the framework of Kalman filtering, resulting in a suboptimal filter.

In the early 1980s, a modified version of Payne model was used for a variety of estimation and

control problems, in particular through the work of Papageorgiou and his collaborators [118,265,331,332]. In [118], Cremer and Papageorgiou introduced the parameter estimation problem with experimental data on this model, and in [265], extended Kalman filtering is applied to this model for state estimation. The work of Wang et al [331] details the simultaneous solution of the state and parameter estimation problem, again with extended Kalman filtering. The article [331] also provides a concise review of related estimation problems appearing in traffic. The interested reader should also see [332], which provides some results of EKF on the modified Payne model as implemented an experimental test bed known as the Renaissance system.

A key ingredient of these works [118,265,331,332] is the differentiability of the numerical scheme employed for the second order model of traffic used, which is a feature the first-order CTM does not possess. The early work of Szeto and Gazis [314], and later Gazis and Liu [158] circumvent this issue for first-order models by directly observing the flows at ends of the road segments, which enables the application of extended Kalman filtering.

Sun, Munoz, and Horowitz [312] treat the nonlinearity of the CTM by recognizing it can be transformed into a switching state space model, which enables the use of a set of linear equations to describe the state evolution for the distinct flow regimes on the highway (e.g. highway is in free-flow or congestion). The density state estimation problem is then solved with a *mixture Kalman filter* for the purpose of ramp metering. In [180,182], specific modes of the dynamics presented in [312] are used to incorporate Lagrangian velocity trajectories into an extension of the CTM, called the *Switched Mode Model* (SMM), using mixture Kalman filtering.

Recently, the cell transmission model has been used in state estimation problems through increasingly advanced nonlinear filters, including *unscented Kalman filtering* (UKF) [204] in the work of Mihaylova, Boel, Hegyi [240], particle filtering by Mihaylova and Boel [239], Mihaylova, Boel, Hegyi [240], and Sau et al [292]. In [240], the particle filter is shown to perform better than UKF, but has a higher computational cost. Implementation of particle filtering techniques on high dimensional systems (several thousand states or more), remains an open challenge due to inherent scalability challenges for particle filters [306]. Other treatments of traffic estimation include adjoint-based control and data assimilation [200,201].

11.4. Outline of Highway Model Description

The following three chapters together describe the *Mobile Millennium* Highway Model in detail. Chapter 12 derives the LWR PDE, explains its important mathematical properties, and summarizes basic mathematical tools thus laying the groundwork for development of a velocity evolution equation consistent with the LWR PDE. It is shown that significant mathematical challenges are introduced by the formation of shocks in the density profile, at which point classical solutions to the LWR PDE no longer exist. The introduction of more general weak solutions allows for shocks, but the uniqueness of solutions can only be guaranteed in the presence of an additional entropy condition.

The LWR PDE serves as the basis for the velocity evolution equations, consistent with hydrodynamic theories that are derived in Chapter 13. Two separate models are introduced [339,340].

- The first model, known as the *Lighthill-Whitham-Richards for velocity* (LWR–v), is a velocity–based partial differential equation with weak solutions consistent with the classical LWR PDE for the Greenshields flux function. For general flux functions, this equivalence is proved not to hold, which is a negative result.
- The second model, known as the *Cell Transmission Model for velocity* (CTM–v), is a discrete evolution equation derived from a Godunov discretization scheme applied to an integral form of the LWR PDE. Its consistency with the Godunov discretization scheme for the LWR PDE, also known as the *Cell Transmission Model*, is ensured by equivalence of the Riemann solvers used in the numerical scheme.

A crucial addition to the original *Mobile Century* Highway Model is the extension of CTM-v to networks. This is accomplished by using a generalized Riemann solver at vertices in the network that is consistent, by construction, with the density Riemann solvers of Coclite, Garavello, and Piccoli [109] and Daganzo [127].

In Chapter 14, the estimation problem is posed in state space form and solved with ensemble Kalman filtering [339,341].

- Using the CTM–v, we pose the state estimation problem as a non-linear non-differentiable dynamical system with a linear observation operator. By using the velocity as the state instead of density (as would be the case for the CTM), we avoid the need to linearize a nonlinear observation operator. The recursive velocity state estimation problem is then solved using *ensemble Kalman filtering* (EnKF).
- We prove the non-differentiability of LWR PDE with a Godunov discretization (also the CTM), around model states which generate a standing shock wave. This fact prevents direct application of the widely popular nonlinear extension of Kalman filtering, known as extended Kalman filtering, to these models.

Chapter 14 concludes by revisiting the dataset from *Mobile Century*, but processed with the refined *Mobile Millennium* Highway Model.

12. LWR PDE

The main objective of this chapter is to describe a mathematical model of traffic evolution that expresses conservation of vehicles, known as the *Lighthill-Whitham-Richards* (LWR) *partial differential equation* (PDE) [226,285], and to review the important mathematical attributes of this model. Two main ideas explored in this chapter are as follows.

- **Strong boundary conditions**. By transforming the classical statement of weak boundary conditions for scalar conservation laws applicable to the LWR PDE into mutually exclusive conditions, we derive an explicit statement for boundary conditions for the LWR PDE on a finite domain to be applied in the strong sense.
- **Non-differentiability of the discretized LWR PDE around arbitrary model states**. We prove that the LWR PDE is not differentiable around model states resulting in a standing shock wave. This result is the main motivation for developing filtering techniques that do not suffer from the same shortcomings as the extended Kalman filter, in particular the ensemble Kalman filter.

The chapter is organized as follows. In Section 1 we recall the derivation of the LWR PDE as an integral equation expressing conservation of vehicles on a stretch of roadway, and note that when the density is smooth, it yields the well-known LWR PDE. We also show non-smooth solutions satisfy the integral form of the equation when the *Rankine-Hugoniot* jump condition is satisfied. In Section 2, we treat non-smooth solutions to the PDE by considering a weak formulation of the partial differential equation, and a suitable entropy condition to guarantee uniqueness of solutions. Proper formulation of the weak boundary conditions and the derivation of the strong boundary conditions are given. In Section 3, we introduce the Riemann problem and its solutions to provide further clarity on the results in Section 2. Numerical discretization of the LWR PDE using a Godunov scheme and its relation to the Riemann problem is presented in Section 1. We also show that this discretization yields a discrete time discrete space evolution equation for density, which cannot be linearized around states resulting in a standing shock wave.

12.1. Derivation of a mass conservation law for traffic

In this section, we derive the well-known Lighthill–Whitham–Richards partial differential

equation [226,285]. Let $\rho(x,t)$ be the vehicle density (the number of vehicles per unit length) at the point x in space and t in time, and let $Q(\cdot)$ be the flux (number of vehicles per unit time) as a function of the density. The flux function $Q(\cdot)$ is defined in an interval $[0,\rho_{\max}]$, where $\rho_{\max}$ is the maximal density, sometimes referred to as "jam density". The total number of vehicles on a segment between two points x_1 and x_2 is given by $\int_{x_1}^{x_2}\rho(x,t)dx$. Assuming vehicles do not appear or disappear within the segment, we have:

$$\frac{d}{dt}\int_{x_1}^{x_2}\rho(x,t)dx = Q(\rho(x_1,t)) - Q(\rho(x_2,t)) \tag{1}$$

$$= -Q(\rho(x,t))|_{x_1}^{x_2}$$

$$= -\int_{x_1}^{x_2}\frac{\partial}{\partial x}Q(\rho(x,t))dx \tag{2}$$

Equation (1) can be understood in the following way. Consider a segment of roadway shown in Figure 1.1, with vehicles entering from the left and exiting to the right. The change in the number of vehicles in the segment over time is just the difference between the number vehicles which entered at x_1, given by $Q(\rho(x_1,t))$ and the number that leave at x_2, given by $Q(\rho(x_2,t))$.

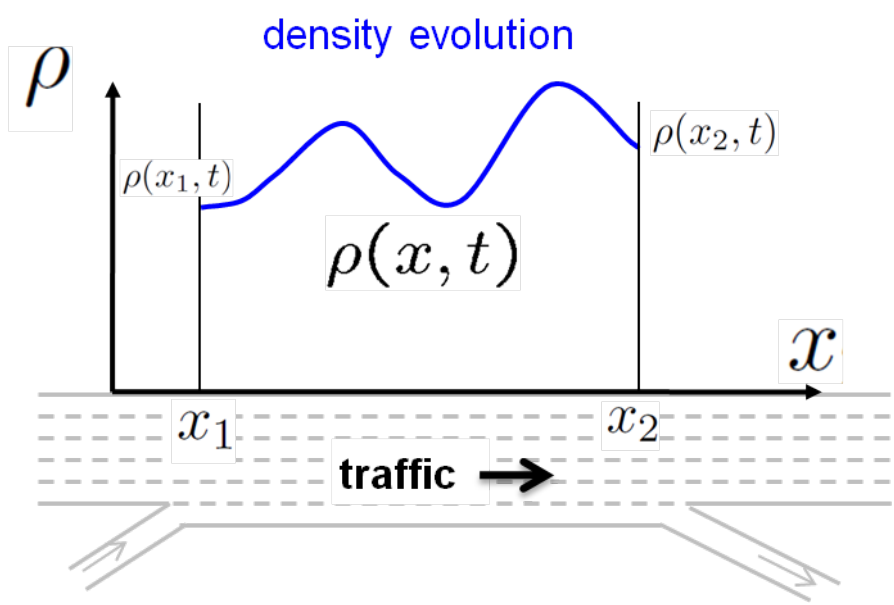


Figure 1.1. The LWR PDE describes the evolution of density on the roadway.

When $\rho(x,t)$ is smooth[13], (2) can be rewritten as

[13]This turns out to be a critical assumption, since often the density profile contains discontinuities such as shocks.

$$\int_{x_1}^{x_2} \left(\frac{\partial \rho(x,t)}{\partial t} + \frac{\partial Q(\rho(x,t))}{\partial x} \right) dx = 0 \tag{3}$$

Since (3) holds for any x_1 and x_2, we obtain the seminal LWR PDE model [226,285]:

$$\frac{\partial \rho(x,t)}{\partial t} + \frac{\partial Q(\rho(x,t))}{\partial x} = 0 \quad (x,t) \in (-\infty, +\infty) \times (0,T) \tag{4}$$

$$\rho(x,0) = \rho_0(x) \quad x \in (-\infty, +\infty) \tag{5}$$

which is the macroscopic traffic flow model expressing conservation of vehicles along an infinite stretch of roadway from time $t = 0$ through $t = T$, augmented with the initial condition ρ_0.

For traffic applications, the flux function $Q(\cdot)$ is generally assumed to be concave and piecewise C^1. This function may be approximated by strictly concave C^2 flux functions with superlinear growth to fit the framework of [69] and [149], which is used to define existence and uniqueness properties of scalar conservation laws (such as the LWR PDE) on a finite domain. In the transportation engineering community, the flux function $Q(\cdot)$ is also known as the *fundamental diagram*.

The fundamental assumption of the LWR PDE is that the average vehicle velocity can be defined in terms of the density alone. With this assumption, we introduce the velocity function $V(\cdot)$ of the density in $[0, \rho_{max}]$. Then the flux function reads:

$$Q(\rho) = \rho\, V(\rho) \tag{6}$$

Solutions to the LWR PDE can be constructed through the method of characteristics. For this, one needs to transform the partial differential equation into a system of ordinary differential equations along curves $(x(z), t(z))$. We seek solutions on the curves of the form

$$\frac{d\rho(x(z),t(z))}{dz} = F\big(x(z), t(z), \rho(x(z),t(z))\big) \tag{7}$$

where $F(\cdot,\cdot,\cdot)$ is a function to be determined. Applying the chain rule to the left side of equation (7) yields

$$\frac{d\rho(x(z),t(z))}{dz} = \frac{\partial \rho}{\partial x}\frac{dx}{dz} + \frac{\partial \rho}{\partial t}\frac{dt}{dz} \tag{8}$$

Note that for smooth functions $Q(\cdot)$, the LWR PDE can be written in quasi-linear form:

$$\frac{\partial \rho(x,t)}{\partial t} + Q'(\rho(x,t))\frac{\partial \rho(x,t)}{\partial x} = 0 \tag{9}$$

If we let $\frac{dx}{dz} = Q'(\rho(x(z),t(z)))$ and $\frac{dt}{dz} = 1$, and substitute into equation (8) we have

$$\frac{d\rho(x(z),t(z))}{dz} = \frac{\partial\rho(x(z),t(z))}{\partial x} Q'(\rho(x(z),t(z))) + \frac{\partial\rho(x(z),t(z))}{\partial t} = 0 \tag{10}$$

where the second equality is given by equation (9). This means the solution $\rho(x(z),t(z))$ is constant on the characteristics. Moreover, solving $\frac{dt}{dz} = 1$, with the initial condition $t(0) = 0$ yields $t = z$. Therefore $\frac{dx}{dz} = \frac{dx}{dt} = Q'\big(\rho(x(z),t(z))\big)$. Thus, we obtain three well-known and important properties of the LWR PDE:

- The density is constant along characteristic curves.
- The speed of the characteristics is given by the slope of the flux function $Q(\cdot)$.
- Because the density is constant along the characteristic curves, the speed of each characteristic curve is a constant.

More details about the method of characteristics can be found in [139]. With these three properties, we can now point out the crux of all mathematical difficulties which arise when solving the LWR PDE. Even from smooth initial conditions, shocks may develop in finite time, and classical (smooth) solutions to the PDE may not exist.

For the purpose of illustration, we consider the flux function given by $Q(\rho) = \rho - \rho^2$ where $\rho \in [0,1]$. The speed of the characteristic curves is positive for $\rho \in \left[0, \frac{1}{2}\right)$, and negative for $\rho \in \left(\frac{1}{2}, 1\right]$. If the initial condition is specified such that upstream characteristic curves have positive velocity, and downstream characteristic curves have negative velocity, the characteristic curves may intersect, yielding a point where the solution is discontinuous.

Moreover, discontinuous solutions satisfy the integral form of the conservation law [224], as we now show. First, apply the integral form of the conservation law (1) over a small interval $x \in [x_1, x_1 + \Delta x]$ inspace and $t \in [t_1, t_1 + \Delta t]$ in time:

$$\int_{x_1}^{x_1+\Delta x} \rho\,(x, t_1 + \Delta t)dx - \int_{x_1}^{x_1+\Delta x} \rho\,(x, t_1)dx$$

Then let s denote the speed of a shock which exists over the full interval, which connects the two states ρ^- and ρ^+ on the left and right sides of the shock, respectively (Figure 1.2) and choose $(\Delta x, \Delta t)$ sufficiently small so that ρ^-, ρ^+, and s can be viewed as a constant. Then we have

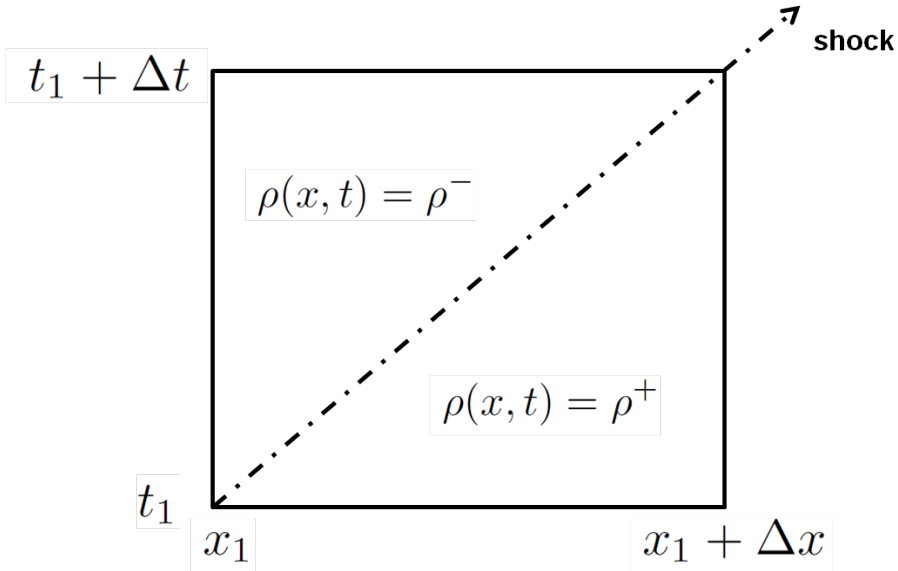


Figure 1.2. A shock traveling at speed s connects the states ρ^- and ρ^+ over a small interval $x \in [x_1, x_1 + \Delta x]$ in space and $t \in [t_1, t_1 + \Delta t]$ in time.

$$\Delta x \rho^- - \Delta x \rho^+ = \Delta t Q(\rho^-) - \Delta t Q(\rho^+) + O(\Delta t^2) \tag{11}$$

where $O(\Delta t^2)$ accounts for the small variation in the fluxes at the boundaries. Since the shock speed over this small interval is given as $s = \frac{\Delta x}{\Delta t}$, we can substitute $\Delta x = s\Delta t$ in (11), divide by Δt, and take the limit as $\Delta t \to 0$, yielding

$$s(\rho^- - \rho^+) = Q(\rho^-) - Q(\rho^+)$$

After rearranging terms, this leads to the *Rankine–Hugoniot* jump condition for the speed of the shock:

$$s = \frac{Q(\rho^+) - Q(\rho^-)}{\rho^+ - \rho^-} \tag{12}$$

The important result is that if (12) is satisfied, then the discontinuity satisfies the integral form of the LWR PDE (1).

To reconcile the fact that discontinuous solutions to the LWR PDE (4) can arise in finite time, even from smooth initial data, and that discontinuous solutions satisfying the Rankine–Hugoniot jump condition (12) also satisfy the integral form of the conservation law (1), we must consider a more general class of solutions to the LWR PDE known as *weak* solutions. We describe this next.

12.2. Weak solutions and the entropy condition

We begin by assuming temporarily that $\rho(x,t)$ is smooth, and satisfies the LWR PDE (4) and (5). Then we introduce a smooth test function with compact support $\varphi(x,t) \in C_c^1((-\infty,\infty) \times [0,T))$. Since the LWR PDE is satisfied, we have

$$\int_{-\infty}^{\infty}\int_0^T \left(\frac{\partial \rho(x,t)}{\partial t} + \frac{\partial Q(\rho(x,t))}{\partial x}\right)\varphi(x,t)dtdx = 0 \tag{13}$$

Now by applying integration of parts, we can rewrite (13) so that the derivatives appear only on the smooth test function:

$$\begin{aligned}\int_{-\infty}^{\infty}\int_0^T \left(\frac{\partial \varphi(x,t)}{\partial t}\rho(x,t) + \frac{\partial \varphi(x,t)}{\partial x}Q(\rho(x,t))\right)dtdx \\ = -\int_{-\infty}^{\infty}\rho_0\,(x)\varphi(x,0)dx\end{aligned} \tag{14}$$

Note that only the initial condition appears, due to the compact support of φ, and that (14) no longer requires a smooth density profile. A solution satisfying (14) is known as a *weak solution* to the LWR PDE, and can be shown to be equivalent to the integral form (1).

An unfortunate result of the weak formulation is that, due to the possibility of discontinuities, the solution is no longer unique (as will be discussed further in Example 8 in Section 3).

One approach to isolate a unique solution is to consider an entropy function of the density, with the property that the entropy is conserved when the density profile is smooth, and that the entropy either increases or decreases due to discontinuities in the density profile. Thus the entropy acts as an indicator of discontinuities, and can be used to isolate a unique solution. This approach leads to the following definition.

Definition 1. (*Weak entropy solution* [*80*,*215*]). A weak entropy solution $\rho(\cdot,\cdot)$ of (4) and (5) is defined as follows:

$$\int_{-\infty}^{\infty}\int_0^T \left(|\rho(x,t)-k|\,\frac{\partial}{\partial t}\varphi(x,t) + \operatorname{sgn}(\rho(x,t)-k)\,(Q(\rho(x,t)) - Q(k))\,\frac{\partial}{\partial x}\varphi(x,t)\right)dxdt$$

where $\operatorname{sgn}(x) = 1$ for $x > 0$, $\operatorname{sgn}(x) = -1$ for $x < 0$, and $\operatorname{sgn}(x) = 0$ for $x = 0$.

In practice, it is often easier to use an equivalent entropy condition which explicitly defines admissible shocks. For a smooth concave flux function $Q(\cdot)$ a discontinuous solution connecting two states ρ^- and ρ^+ propagating at speed s satisfies the Lax entropy condition if:

$$Q'(\rho^+) \leq s \leq Q'(\rho^-) \tag{15}$$

When the LWR PDE is specified on a finite domain:

$$\frac{\partial \rho(x,t)}{\partial t} + \frac{\partial Q(\rho(x,t))}{\partial x} = 0 \quad (x,t) \in (0,L) \times (0,T) \tag{16}$$

special consideration must be made at the boundaries to ensure well posed problem, as given by the next two definitions.

Definition 2. (*Left weak boundary condition - concave flux function* [*69,309*]). For a general flux function $Q(\cdot)$, the proper weak description of the left boundary condition for (16) in terms of the trace of the solution $\rho(0,t)$ and the left boundary data $\rho_l(t)$ is as follows:

$$\begin{aligned} \sup_{k \in D(\rho(0,t),\rho_l(t))} & \big(\operatorname{sgn}(\rho(0,t) - \rho_l(t))(Q(\rho(0,t)) - Q(k))\big) \\ & = 0 \quad \text{for } a.e.\, t > 0 \end{aligned} \tag{17}$$

where $D(x,y) = [\inf(x,y), \sup(x,y)]$.

Definition 3. (*Right weak boundary condition - concave flux function* [*69,309*]). For a general flux function $Q(\cdot)$, the proper weak description of the right boundary condition for (16) in terms of the trace ofthe solution $\rho(L,t)$ and the right boundary condition $\rho_r(t)$ is as follows:

$$\begin{aligned} \inf_{k \in D(\rho(L,t),\rho_r(t))} & \big(\operatorname{sgn}(\rho(L,t) - \rho_r(t))(Q(\rho(L,t)) - Q(k))\big) \\ & = 0 \quad \text{for } a.e.\, t > 0 \end{aligned} \tag{18}$$

where $D(x,y) = [\inf(x,y), \sup(x,y)]$.

It was proposed in [149] to write boundary conditions in such a way that the entropy solution to equation (16) exists and is unique, for in domain bounded on the left and unbounded on the right. For a strictly convex continuously differentiable flux function under sufficient regularity of the boundary data $\rho_l(\cdot)$ and $\rho_r(\cdot)$, an equivalent formulation of (17) and (18) can be obtained. In [151], it is shown that continuity of the boundary data is sufficient for an equivalent formulation. In our case, this formulation reads for the left boundary:

$$\begin{aligned} &\text{for } a.e.\, t > 0, \\ &\begin{cases} & \rho(0,t) = \rho_l(t) \\ \text{xor} & Q'(\rho(0,t)) \leq 0 \text{ and } Q'(\rho_l(t)) \leq 0 \text{ and } \rho(0,t) \neq \rho_l(t) \\ \text{xor} & Q'(\rho(0,t)) \leq 0 \text{ and } Q'(\rho_l(t)) > 0 \text{ and } Q(\rho(0,t)) \leq Q(\rho_l(t)) \end{cases} \end{aligned} \tag{19}$$

and for the right boundary:

$$\text{for } a.e.\, t > 0,$$
$$\begin{cases} & \rho(L,t) = \rho_r(t) \\ \text{xor} & Q'(\rho(L,t)) \geq 0 \text{ and } Q'(\rho_r(t)) \geq 0 \text{ and } \rho(L,t) \neq \rho_r(t) \\ \text{xor} & Q'(\rho(L,t)) \geq 0 \text{ and } Q'(\rho_r(t)) < 0 \text{ and } Q(\rho(L,t)) \leq Q(\rho_r(t)) \end{cases} \tag{20}$$

where $\rho_l(\cdot)$ and $\rho_r(\cdot)$ are functions of $C^0(0,T)$.

The preceding equations (19) and (20) are a description of cases for which (17) and (18) are satisfied. Note the description is slightly different from [309] in that the sets defined on each line of (19) and (20) are mutually exclusive. For example, first line of (19) corresponds to the case when the trace of the solution $\rho(0,t)$ takes the value of the boundary data $\rho_l(t)$, which is analogous to a prescription of the boundary condition in the strong sense. The second line and third lines correspond to cases which satisfy (17), but where the value of the trace does not take the value prescribed at the boundary.

We now expand on the first line of equations (19) – (20) in order to state explicitly the set of the boundary data, trace pairs for which the boundary data is prescribed in the strong sense.

Lemma 4. (*Strong boundary conditions - concave flux*). For a strictly concave flux function $Q(\cdot)$, the cases for strong boundary conditions read as follows: for $a.e.\, t > 0$,

$$\rho(0,t) = \rho_l(t) \text{ iff}$$
$$\begin{cases} & Q'(\rho(0,t)) \geq 0 \text{ and } Q'(\rho_l(t)) \geq 0 \\ \text{xor} & Q'(\rho(0,t)) \leq 0 \text{ and } Q'(\rho_l(t)) \leq 0 \text{ and } \rho(0,t) = \rho_l(t) \\ \text{xor} & Q'(\rho(0,t)) \leq 0 \text{ and } Q'(\rho_l(t)) > 0 \text{ and } Q\big(\rho(0,t)\big) > Q(\rho_l(t)) \end{cases} \tag{21}$$

and for $a.e.\, t \geq 0$,

$$\rho(L,t) = \rho_r(t) \text{ iff}$$
$$\begin{cases} & Q'(\rho(L,t)) \leq 0 \text{ and } Q'(\rho_r(t)) \leq 0 \\ \text{xor} & Q'(\rho(L,t)) \geq 0 \text{ and } Q'(\rho_r(t)) \geq 0 \text{ and } \rho(L,t) = \rho_r(t) \\ \text{xor} & Q'(\rho(L,t)) \geq 0 \text{ and } Q'(\rho_r(t)) < 0 \text{ and } Q\big(\rho(L,t)\big) > Q(\rho_r(t)) \end{cases} \tag{22}$$

Proof. We prove the case of the left boundary condition for a concave flux and note a similar argument holds for the right boundary and in the case of convex flux functions. Beginning with the statement of weak boundary conditions, (19) we can write: for $a.e.\, t > 0$,

$$\rho(0,t) \neq \rho_l(t) \text{ iff}$$
$$\begin{cases} & Q'(\rho(0,t)) \leq 0 \text{ and } Q'(\rho_l(t)) \leq 0 \text{ and } \rho(0,t) \neq \rho_l(t) \\ \text{xor} & Q'(\rho(0,t)) \leq 0 \text{ and } Q'(\rho_l(t)) > 0 \text{ and } Q(\rho(0,t)) \leq Q(\rho_l(t)) \end{cases}$$

If we are not in one of these two cases, then by taking their complement, we must have either

$$\begin{cases} & Q'(\rho(0,t)) \geq 0 \text{ and } Q'(\rho_l(t)) \geq 0 \\ \text{xor} & Q'(\rho(0,t)) \leq 0 \text{ and } Q'(\rho_l(t)) \leq 0 \text{ and } \rho(0,t) = \rho_l(t) \\ \text{xor} & Q'(\rho(0,t)) \leq 0 \text{ and } Q'(\rho_l(t)) > 0 \text{ and } Q\big(\rho(0,t)\big) > Q(\rho_l(t)) \\ \text{xor} & Q'(\rho(0,t)) > 0 \text{ and } Q'(\rho_l(t)) < 0 \end{cases} \tag{23}$$

For the fourth line in (23), for $a.e.\, t > 0$ we will have $Q'(\rho(0,t)) = 0$, so it is removed and the conditions for strong left boundary conditions are obtained. □

12.3. The Riemann problem

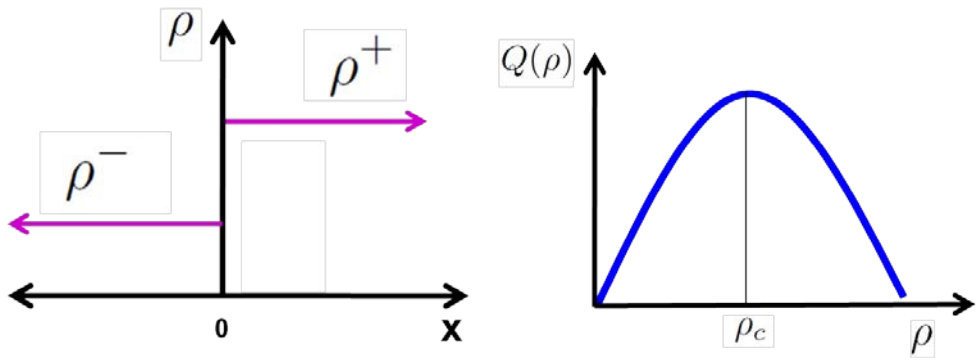
In order to provide clarity on the weak entropy solution to the LWR PDE on a bounded domain, we now introduce the Riemann problem, which is the Cauchy problem equation (4) with a Heaviside initial condition. By tracking the evolution of the characteristic curves on this problem, we can easily recover the weak boundary conditions (19) and (20). Later, it will be used again for numerical discretization of the PDE, and for its extension to networks.

The proof of a global solution to (4)–(5) by successive local solutions of Riemann problems is due to [122]. We now proceed to explain case-by-case all of the behaviors of the LWR PDE which can arise from the Riemann problem.

Let the initial data for equation (4) be given by (Figure 3.1(a)):

$$\rho_0(x) = \begin{cases} \rho^- & \text{if } x < 0 \\ \rho^+ & \text{if } x > 0 \end{cases} \tag{24}$$

and let us again consider a flux function of the form $Q(\rho) = \rho - \rho^2$ (Figure 3.1(b)). Note that the flux function is increasing for all $\rho \in [0, \rho_c)$, and decreasing for all $\rho \in (\rho_c, \rho_{\max}]$, where $\rho_c = \frac{1}{2}$ is the critical density and $\rho_{\max} = 1$ is the maximal density.



(a) (b)

Figure 3.1. (a) Initial data for the Riemann problem; (b) quadratic flux function.

12.3.1. Riemann solver

The weak entropy Riemann solver for the Riemann problem (4) and (24) is given by

- If $\rho^+ > \rho^-$,

$$\rho(x,t) = \begin{cases} \rho^- & \text{if } \frac{x}{t} < s \\ \rho^+ & \text{if } \frac{x}{t} > s \end{cases} \tag{25}$$

where $s = \frac{Q(\rho^+)-Q(\rho^-)}{\rho^+-\rho^-}$.

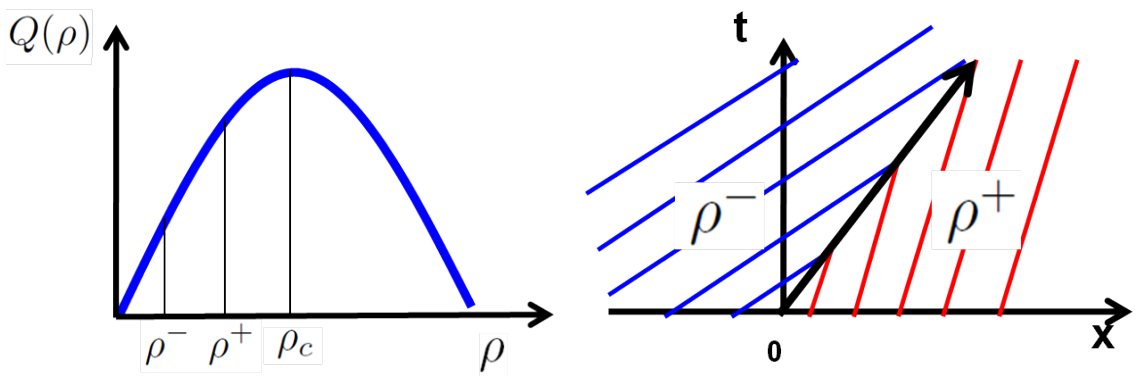
- If $\rho^+ < \rho^-$,

$$\rho(x,t) = \begin{cases} \rho^- & \text{if } \frac{x}{t} < Q'(\rho^-) \\ (Q')^{-1}(\frac{x}{t}) & \text{if } Q'(\rho^-) < \frac{x}{t} < Q'(\rho^+) \\ \rho^+ & \text{if } \frac{x}{t} > Q'(\rho^+) \end{cases} \tag{26}$$

We now proceed case-by-case through the solutions of the Riemann problem.

Example 5. (*Small shock moving forward*). Let the initial data be defined such that $\rho^- \leq \rho^+ \leq \rho_c$ (Figure 3.2(a)). Then the speeds of the characteristics are ordered by $Q'(\rho^-) \geq Q'(\rho^+) \geq 0$. Since the characteristics on the left move faster than the characteristics on the right, they intersect and a small shock wave is formed (Figure 3.2(b)). The speed of the shock is given by

the Rankine-Hugoniot relation $s = \frac{Q(\rho^+)-Q(\rho^-)}{\rho^+-\rho^-} \geq 0$, and so the shock travels forward. Moreover, since $Q'(\rho^-) \geq s \geq Q'(\rho^+)$, the shock is entropy admissible.



(a) (b)

Figure 3.2. Riemann problem solved by a small shock wave moving forward. (a) initial data; (b) evolution of the characteristic curves.

Example 6. (*Rarefaction wave moving forward*). Let the initial data be defined such that $\rho^+ \leq \rho^- \leq \rho_c$ (Figure 3.3(a)). Then the speeds of the characteristics are ordered by $Q'(\rho^+) \geq Q'(\rho^-) \geq 0$. Now the characteristics on the right move faster than the characteristics on the left, and an envelope appears (solid gray area in Figure 3.3(b)) through which no characteristic curves emanating from the initial condition pass through. Because the weak form of the LWR PDE allows for discontinuities, a forward-moving shock is a weak admissible solution (Figure 3.3(c)). Moreover, a self-similar solution in the form of a rarefaction wave moving forward (26) is also a solution (Figure 3.3(d)), which does not require a discontinuity. Note, however, that only the rarefaction wave satisfies the entropy condition, since for the shock we have $Q'(\rho^-) \ngeq s \ngeq Q'(\rho^+)$.

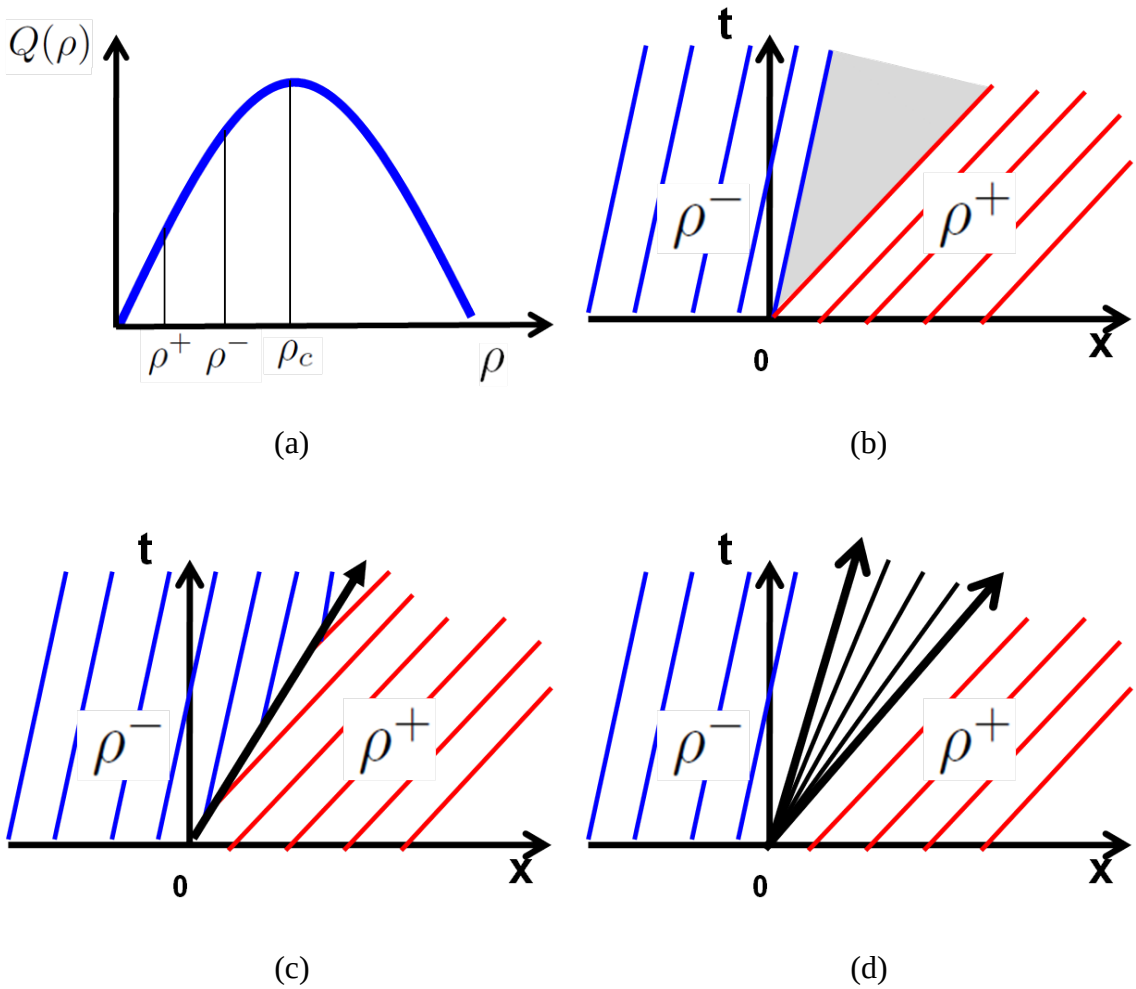
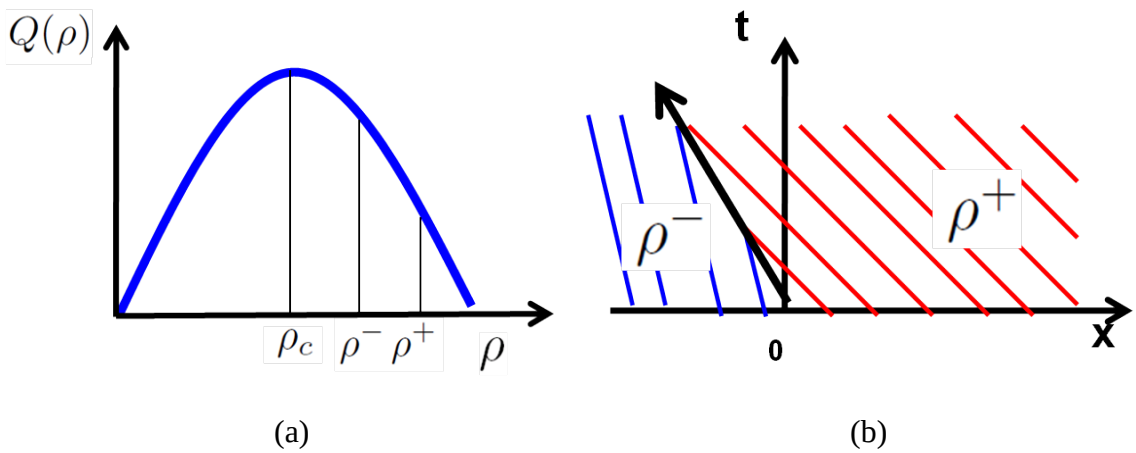


Figure 3.3. Riemann problem solved by a rarefaction wave moving forward. (a) initial data; (b) an envelope appears which is not determined by characteristic curves emanating from the initial data; (c) a weak (but not entropy admissible) evolution of the characteristic curves; (d) entropy admissible evolution of the characteristic curves.

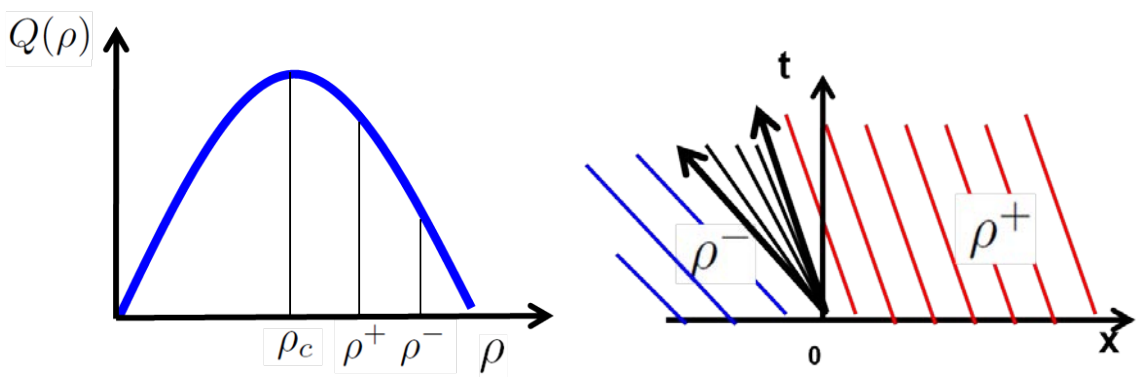
Example 7. (*Small shock moving back*). Let the initial data be defined such that $\rho_c \leq \rho^- \leq \rho^+$ (Figure 3.4(a)). Then the speeds of the characteristics are ordered by $0 \geq Q'(\rho^-) \geq Q'(\rho^+)$. Since the characteristics on the left move slower than the characteristics on the right, they intersect and a small shock wave is formed (Figure 3.4(b)). The speed of the shock is given by the Rankine-Hugoniot relation $s = \frac{Q(\rho^+)-Q(\rho^-)}{\rho^+-\rho^-} \leq 0$, and so the shock travels back. Moreover, since $Q'(\rho^-) \geq s \geq Q'(\rho^+)$, the shock is entropy admissible.



(a) (b)

Figure 3.4. Riemann problem solved by a small shock wave moving back. (a) initial data; (b) evolution of the characteristic curves.

Example 8. (*Rarefaction wave moving back*). Let the initial data be defined such that $\rho_c \leq \rho^+ \leq \rho^-$ (Figure 3.5(a)). Then the speeds of the characteristics are ordered by $0 \geq Q'(\rho^+) \geq Q'(\rho^-)$. Now the characteristics on the left move faster than the characteristics on the right, and again an envelope appears through which no characteristic curves emanating from the initial condition pass through (Similar to Example 6). Since the entropy condition prevents the formation of the shock, a rarefaction wave is formed (Figure 3.5(b)).



(a) (b)

Figure 3.5. Riemann problem solved by a rarefaction wave moving back. (a) initial data; (b) evolution of the characteristic curves.

Example 9. (*Rarefaction wave moving forward and back*). Let the initial data be defined such that $\rho^+ \leq \rho_c \leq \rho^-$ (Figure 3.6(a)). Then the speeds of the characteristics are ordered by $Q'(\rho^+) \geq 0 \geq Q'(\rho^-)$. Now the characteristics on the left move left, and the characteristics on the right side move right, so again and envelope appears through which no characteristic curves emanating from the initial condition pass through. Since the entropy condition prevents the formation of the shock, a rarefaction wave is formed (Figure 3.6(b)).

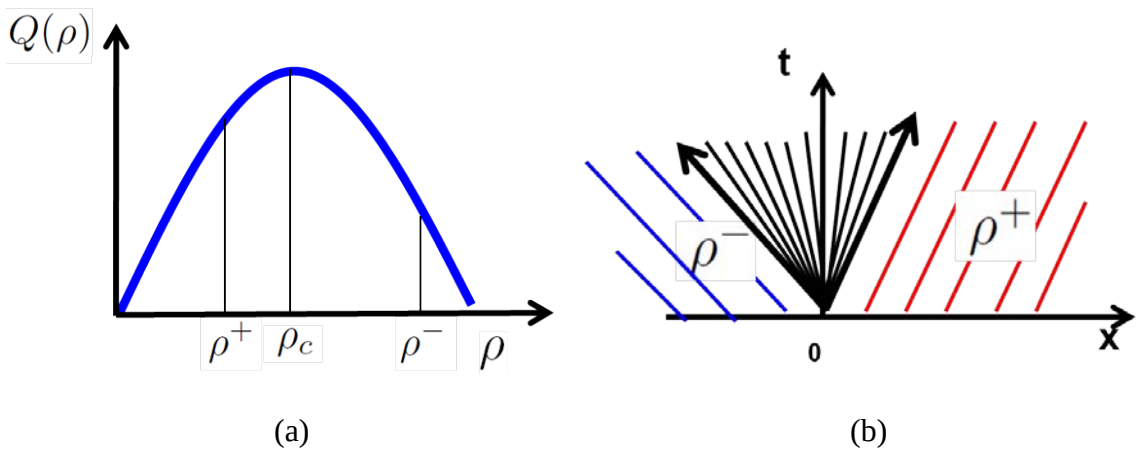


Figure 3.6. Riemann problem solved by a rarefaction wave moving back and forward. (a) initial data; (b) evolution of the characteristic curves.

Example 10. (*Big shock wave moving back*). Let the initial data be defined such that $\rho^- \leq \rho_c \leq \rho^+$, and additionally $Q(\rho^+) < Q(\rho^-)$ (Figure 3.7(a)). Then the speeds of the characteristics are ordered by $Q'(\rho^-) \geq Q'(\rho^+) \geq 0$. Since the characteristics on the left move right, and the characteristics on the right move left, they intersect and a big shock wave is formed (Figure 3.7(b)). The speed of the shock is given by the Rankine-Hugoniot relation $s = \frac{Q(\rho^+)-Q(\rho^-)}{\rho^+-\rho^-} \geq 0$, and since the numerator is negative and the denominator is positive, the shock travels back. Moreover, since $Q'(\rho^-) \geq s \geq Q'(\rho^+)$, the shock is entropy admissible.

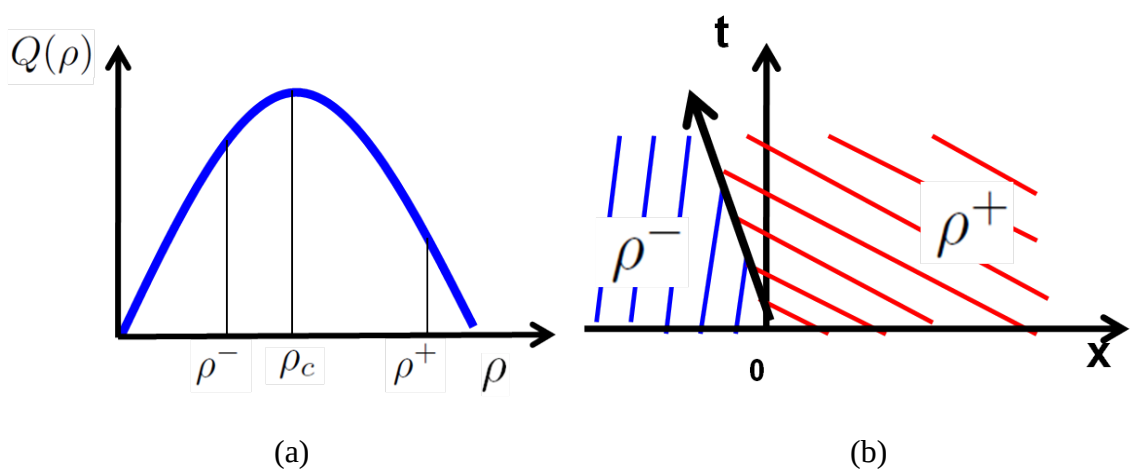
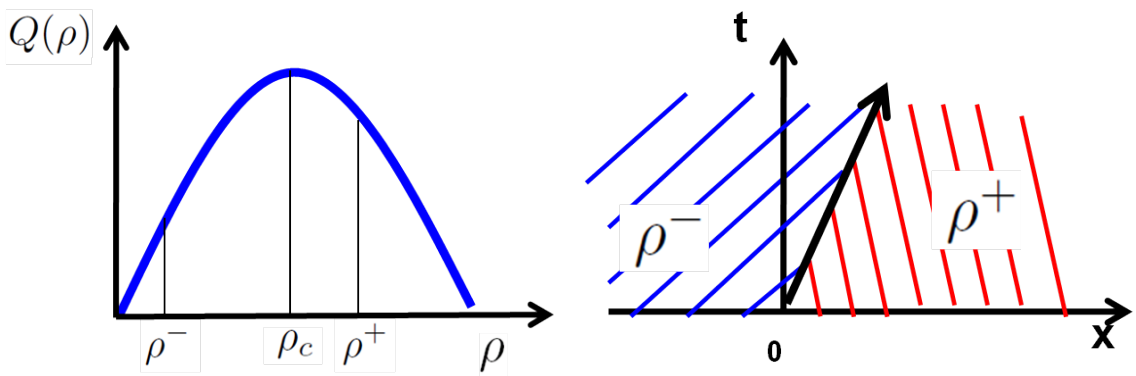


Figure 3.7. Riemann problem solved by a big shock moving back. (a) initial data; (b) evolution of the characteristic curves.

Example 11. (*Big shock wave moving forward*). Let the initial data be defined such that $\rho^- \leq \rho_c \leq \rho^+$, and additionally $Q(\rho^+) > Q(\rho^-)$ (Figure 3.8(a)). Then the speeds of the characteristics are ordered by $Q'(\rho^-) \geq Q'(\rho^+) \geq 0$. Since the characteristics on the left move right, and the characteristics on the right move left, they intersect and a big shock wave is formed (Figure 3.8(b)). The speed of the shock is given by the Rankine-Hugoniot relation $s =$

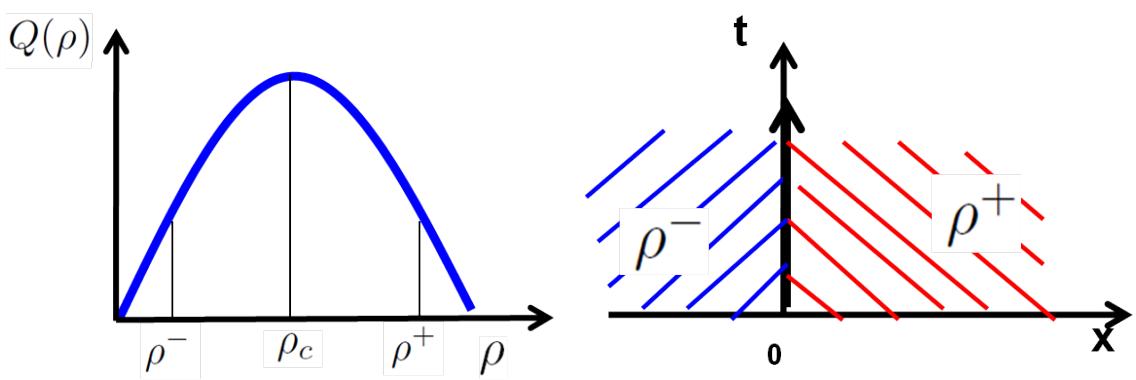
$\frac{Q(\rho^+)-Q(\rho^-)}{\rho^+-\rho^-} \geq 0$, and since now both the numerator and denominator are positive, the shock travels forward. Moreover, since $Q'(\rho^-) \geq s \geq Q'(\rho^+)$, the shock is entropy admissible.



(a) (b)

Figure 3.8. Riemann problem solved by a big shock moving forward. (a) initial data; (b) evolution of the characteristic curves.

Example 12. (*Big stationary shock wave*). Let the initial data be defined such that $\rho^- \leq \rho_c \leq \rho^+$, and additionally $Q(\rho^+) = Q(\rho^-)$ (Figure 3.9(a)). Then the speeds of the characteristics are ordered by $Q'(\rho^-) \geq Q'(\rho^+) \geq q0$. Since the characteristics on the left move right, and the characteristics on the right move left, they intersect and a big shock wave is formed (Figure 3.9(b)). The speed of the shock is given by the Rankine-Hugoniot relation$s = \frac{Q(\rho^+)-Q(\rho^-)}{\rho^+-\rho^-} \geq 0$. Because the numerator is zero, the shock wave is stationary. Moreover, since $Q'(\rho^-) \geq s \geq Q'(\rho^+)$, the shock is entropy admissible.



(a) (b)

Figure 3.9. Riemann problem solved by a big stationary shock. (a) initial data; (b) evolution of the characteristic curves.

It is easy to verify that the previous examples are exhaustive on the set of initial conditions for

the Riemann problem. The solutions are shown graphically in Figure 3.10. Note that the entropy condition only allows shocks above the line $Q'(\rho^+) = Q'(\rho^-)$, and so solutions below the line form rarefaction waves. The curve corresponding to the big stationary shock is defined by $Q(\rho^+) = Q(\rho^-)$, and its shape depends on the exact flux function used.

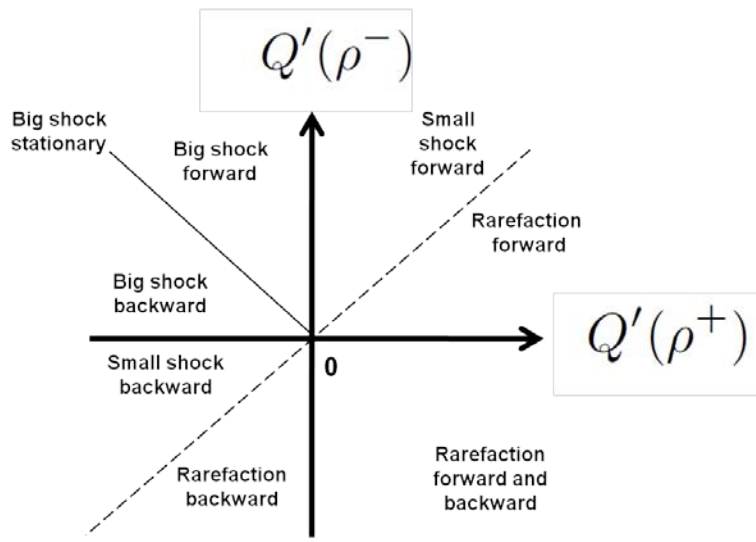


Figure 3.10. Summary of the various Riemann problem solutions as a function of the speed of the characteristics of the initial data.

12.3.2. Weak boundary conditions revisited

With the solution to the Riemann problem now defined, it is possible to verify the statement of weak boundary conditions given by (19) and (20), through an interpretation of the Riemann solver. We now proceed to show this for the left boundary condition. Consider the Riemann problem (4) with the following initial data defined by (4):

$$\rho_0(x) = \begin{cases} \rho_l(0) & \text{if } x < 0 \\ \rho(0,0) & \text{if } x > 0 \end{cases} \tag{27}$$

where $\rho_l(0)$ is the left boundary condition we would like to apply (but which may not hold) and $\rho(0,0)$ is the trace of the solution approaching the left boundary. In order to determine if the boundary data applies on the domain, we only need to solve the Riemann problem and identify if the characteristic curves associated with $\rho_l(0)$ cross the boundary $x = 0$ for $t > 0$. If so, then the boundary data will carry into the domain, and the boundary condition will hold in the strong sense. Looking back at the previous Riemann problem, it is clear this is true when the solution to the Riemann problem is a forward moving rarefaction wave or a forward moving shock wave. This corresponds to the first line of the left weak boundary condition (19).

On the other hand, if the Riemann problem results in a rarefaction wave moving back or a shock wave moving back, then the boundary condition will not hold unless the boundary data and the trace have the same value (the initial condition for the Riemann problem is a single constant value). In the context of a boundary control problem, in this pathological case one must choose

the boundary control to be the unique value dictated by the trace, and any perturbation in the boundary control value would cause the solution to be implemented in the weak sense again.

When the solution to the Riemann problem yields a big shock wave moving backward, it is covered by the third line of (19), while a small shock wave or a rarefaction wave moving backward corresponds to the second line of (19). Finally, the solution to the Riemann problem resulting in a rarefaction wave with characteristics pointing both forward and back corresponds to the second line of (19), since for $t > 0$, $Q'(\rho(0,t)) = 0$.

12.4. Numerical discretization

12.4.1. Godunov Scheme

We now describe the Godunov discretization scheme used to numerically approximate weak entropy solutions to the LWR PDE. We discretize the time and space domains by introducing a discrete time step ΔT, indexed by $n \in \{0, \cdots, n_{max}\}$ and a discrete space step Δx, indexed by $i \in \{0, \cdots, i_{max}\}$. Let us integrate equation (1) over a single timestep, yielding:

$$\begin{aligned}\int_{x_{i-1/2}}^{x_{i+1/2}} \rho\,(x, t_{n+1})dx - \int_{x_{i-1/2}}^{x_{i+1/2}} \rho\,(x, t_n)dx \\ = \int_{t_n}^{t_{n+1}} Q\,(\rho(x_{i-1/2}, t))dt - \int_{t_n}^{t_{n+1}} Q\,(\rho(x_{i+1/2}, t))dt\end{aligned} \tag{28}$$

We introduce the variables ρ_i^n and Q_i^n as follows:

$$\rho_i^n \approx \frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_{i+1/2}} \rho\,(x, t_n)dx \tag{29}$$

$$Q_i^n \approx \frac{1}{\Delta T}\int_{t_n}^{t_{n+1}} Q\,(\rho(x_i, t))dt \tag{30}$$

where ρ_i^n approximates the average density in the ith cell at time t_n and Q_i^n approximates the average flux at x_i over the time interval $[t_n, t_{n+1}]$. Then after substituting into (28) and rearranging terms, we obtain

$$\rho_i^{n+1} = \rho_i^n - \frac{\Delta T}{\Delta x}\left(Q_{i+1/2}^n - Q_{i-1/2}^n\right) \tag{31}$$

which is the basis of the Godunov discretization scheme.

In particular, the Godunov discretization scheme is as follows.

- Approximate the function $\rho(x,t)$ with a piecewise constant function $\bar{\rho}(x,t)$,

where $\bar{\rho}(x,t)$ is constant in each cell. Then

$$\rho_i^n = \frac{1}{\Delta x}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \bar{\rho}\,(x,t_n)dx = \bar{\rho}(x_i,t_n)$$

- Solve the Riemann problems at the cell boundaries $x_{i+1/2}$ and $x_{i-1/2}$. Let $G(\rho_i^n,\rho_{i+1}^n)$ denote the flux at $x_{i+1/2}$ when the Riemann problem is solved between the two states ρ_i^n and ρ_{i+1}^n. Because $\bar{\rho}(x,t)$ is piecewise constant, the Riemann problem can be solved exactly, and the flux $G(\rho_1,\rho_2)$ is given as (Figure 4.1):

$$G(\rho_1,\rho_2) = \begin{cases} Q(\rho_2) & \text{if } \rho_c \le \rho_2 \le \rho_1 \\ Q(\rho_c) & \text{if } \rho_2 \le \rho_c \le \rho_1 \\ Q(\rho_1) & \text{if } \rho_2 \le \rho_1 \le \rho_c \\ \min\,(Q(\rho_1),Q(\rho_2)) & \text{if } \rho_1 \le \rho_2 \end{cases} \tag{32}$$

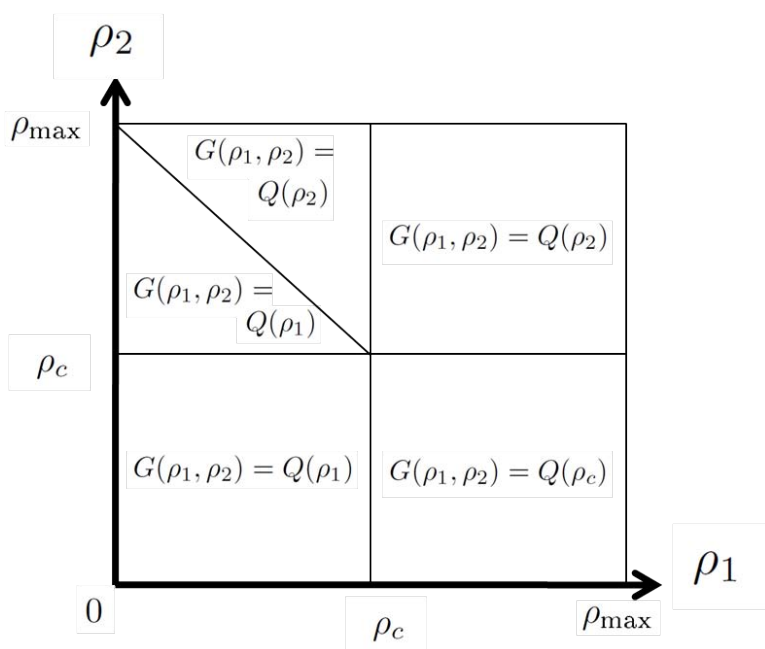


Figure 4.1. Graphical representation of the numerical flux function equation (32) as a function of $\boldsymbol{\rho_1}$ and $\boldsymbol{\rho_2}$. Note that the line connecting $(\boldsymbol{\rho_c},\boldsymbol{\rho_c})$ and $(\mathbf{0},\boldsymbol{\rho}_{\mathbf{max}})$ may have different shapes depending on the flux function $\boldsymbol{Q}(\cdot)$.

This yields

$$Q_{i+1/2}^n = \frac{1}{\Delta T}\int_{t_n}^{t_{n+1}} Q\,(\bar{\rho}(x_{i+1/2},t))dt = G(\rho_i^n,\rho_{i+1}^n)$$

- Compute the density at the next time step according to (31).

In practice, the scheme is implemented as the nonlinear discrete evolution equation:

$$\rho_i^{n+1} = \rho_i^n - \frac{\Delta T}{\Delta x}\left(G(\rho_i^n, \rho_{i+1}^n) - G(\rho_{i-1}^n, \rho_i^n)\right) \tag{33}$$

In order to ensure numerical stability, the time and space steps are coupled by the CFL condition [224]: $|\alpha_{max}|\Delta t \leq \Delta x$ where α_{max} denotes the maximal characteristic speed. This restriction guarantees that the solution of the Riemann problem at each cell boundary is independent of the Riemann problems at adjacent boundaries.

This discrete model is derived independently by Daganzo [126,127] as a macroscopic traffic model consistent with the LWR PDE, commonly referred to as the *Cell Transmission Model* in the transportation engineering community.

12.4.2. A note on linearization

Later in this discussion we address the traffic velocity estimation problem using a velocity evolution equation consistent with the discretized LWR PDE. Because of the nonlinearity of the hyperbolic conservation law for density, standard linear estimation techniques such as the Kalman filter cannot be used. Moreover, we now show that the LWR PDE discretized with the Godunov discretization scheme cannot be linearized around an arbitrary model state. This unfortunate fact prevents the use of extended Kalman filtering for traffic estimation when using this model (the Cell Transmission Model). The set of states under which the model cannot be linearized corresponds to the case when the Riemann problem is solved with a stationary shock.

Theorem 13. Let $Q(\rho)$ be a smooth C^1 concave flux function with $\rho \in [0, \rho_{\max}]$, with a maximum obtained at ρ_c. Let $\rho^n = \left[\rho_0^n, \cdots, \rho_{i_{\max}}^n\right]$ denote the vector of states at time n. The LWR PDE is discretized according to the Godunov scheme (32) and (33). The discrete model can be linearized if and only if no standing shockwaves are formed.

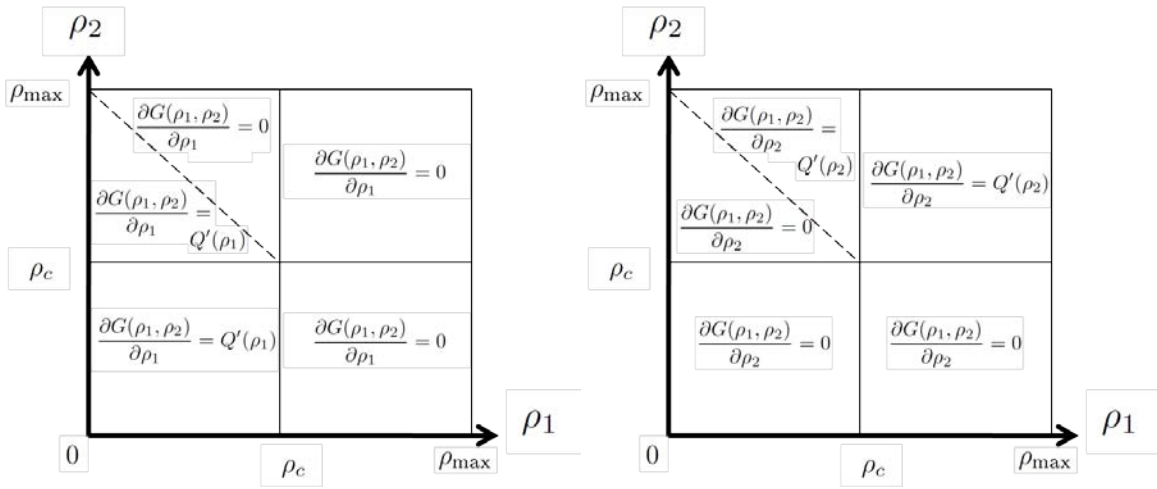
Proof. It is easy to see from (33) that the model can be linearized if and only if the function $G(\cdot,\cdot)$ is differentiable. We determine the differentiability of (32) by an exhaustive computation of the partial derivatives $G(\cdot,\cdot)$ with respect to each of the inputs. Note that $G(\cdot,\cdot)$ is continuous for all $(\rho_1, \rho_2) \in [0, \rho_{\max}] \times [0, \rho_{\max}]$, so it remains to check if the partial derivatives exist and are continuous.

We begin by computing $\frac{\partial G(\rho_1,\rho_2)}{\partial \rho_1}$.

- For $\rho_1, \rho_2 > \rho_c$, we have $\frac{\partial G(\rho_1,\rho_2)}{\partial \rho_1} = 0$. This corresponds to either a small shock or a rarefaction wave moving backward, and $G(\rho_1, \rho_2)$ depends only on ρ_2.

- For $\rho_1 > \rho_c > \rho_2$, we have $\frac{\partial G(\rho_1,\rho_2)}{\partial \rho_1} = 0$. This corresponds to a rarefaction wave with characteristics moving forward and backward, and $G(\rho_1, \rho_2)$ is a constant.
- For $\rho_1 > \rho_2 = \rho_c$, $G(\rho_1, \rho_2)$ is again independent of ρ_1, and so $\frac{\partial G(\rho_1,\rho_2)}{\partial \rho_1} = 0$.
- For $\rho_1, \rho_2 < \rho_c$, we have $\frac{\partial G(\rho_1,\rho_2)}{\partial \rho_1} = Q'(\rho_1)$, which corresponds to either a small shock or a rarefaction wave moving forward.
- For $\rho_2 < \rho_1 = \rho_c$, we have $\lim_{\rho_1 \to \rho_c^+} \frac{\partial G(\rho_1,\rho_2)}{\partial \rho_1} = 0$, and $\lim_{\rho_1 \to \rho_c^-} \frac{\partial G(\rho_1,\rho_2)}{\partial \rho_1} = \lim_{\rho_1 \to \rho_c^-} Q'(\rho_1) = 0$, since $Q(\cdot)$ is maximized at ρ_c, so the partial derivative is continuous.
- The case when $\rho_1 < \rho_c < \rho_2$ corresponds to a big shock, so we must further specify the ordering of the flux. Let $Q(\rho_2) > Q(\rho_1)$, so the shock moves forward. In this case, $\frac{\partial G(\rho_1,\rho_2)}{\partial o_1} = Q'(\rho_1)$.
- When $\rho_1 < \rho_2 = \rho_c$ and $Q(\rho_2) > Q(\rho_1)$, we have $\frac{\partial G(\rho_1,\rho_2)}{\partial \rho_1} = Q'(\rho_1)$ and the solution is again a forward moving shock.
- When $\rho_1 < \rho_c < \rho_2$ and $Q(\rho_2) < Q(\rho_1)$, we have $\frac{\partial G(\rho_1,\rho_2)}{\partial \rho_1} = 0$ since $G(\rho_1, \rho_2)$ depends only on ρ_2. The solution is a big shock moving back.
- When $\rho_1 = \rho_c < \rho_2$ and $Q(\rho_2) < Q(\rho_1)$, we have $\frac{\partial G(\rho_1,\rho_2)}{\partial \rho_1} = 0$ in the solution is again a shock moving back.
- The last case is when $\rho_1 < \rho_c < \rho_2$ and $Q(\rho_2) = Q(\rho_1)$. For a given ρ_2, let ρ^* be defined such that $\rho^* < \rho_c < \rho_2$ and $Q(\rho^*) = Q(\rho_2)$. Then $\lim_{\rho_1 \to \rho^{*+}} \frac{\partial G(\rho_1,\rho_2)}{\partial \rho_1} = 0$, and $\lim_{\rho_1 \to \rho^{*-}} \frac{\partial G(\rho_1,\rho_2)}{\partial \rho_1} = Q'(\rho_1) > 0$ from the concavity of the flux function. Thus, solution is not differentiable around this point, which corresponds to a standing shock wave.

The proof for $\frac{\partial G(\rho_1,\rho_2)}{\partial \rho_2} = 0$ proceeds similarly, and is also differentiable everywhere except around states corresponding to a standing shock wave. The computation of the partial derivatives are summarized in Figure 4.2. □



(a) (b)

Figure 4.2. Summary of the computation of (a) $\frac{\partial G(\rho_1,\rho_2)}{\partial \rho_1}$; (b) $\frac{\partial G(\rho_1,\rho_2)}{\partial \rho_2}$. Note that the line connecting (ρ_c, ρ_c) and $(0, \rho_{\max})$ may have different shapes depending on the flux function $Q(\cdot)$, and represents the set for which $G(\cdot,\cdot)$ is not differentiable.

Direct application of extended Kalman filtering [326], or implicit switching between linearized regimes [315] will not be applicable around these points. Fortunately, in practical traffic estimation problems, the state around which the model is linearized will rarely result in a stationary shock; the numerical values of the flux almost always cause the shock to move slightly forward or back. Nevertheless, this still creates a theoretical challenge which has not yet been addressed in a traffic estimation literature.

13. Derivation of a velocity evolution equation

Motivated by the problem of estimating traffic conditions using only velocity measurements from mobile phones, this chapter focuses on the development of a mathematical model for velocity evolution consistent with the LWR PDE. Advancements and discoveries described in this chapter are as follows.

- **Derivation of a velocity partial differential equation**. When the relationship between velocity and density is affine (as is the case for the Greenshields flux function), we derive a new velocity conservation law consistent with the weak form of the LWR PDE, called the *LWR–v* PDE.
- **Limitation of the velocity partial differential equation**. For general invertible velocity functions that are not affine, we show that there is no equivalent velocity conservation law. This is a negative result.
- **Derivation of a discrete velocity evolution equation**. For general, nonlinear invertible velocity functions, we derive a numerical approximation to the integral form of the LWR PDE (1), which describes velocity evolution on a discrete domain and overcomes the above limitation. We call this model the *Cell Transmission Model for velocity* (CTM–v), due to similarities with the CTM model.
- **Extension to networks**. We discuss how to extend the velocity evolution equation to networks of roads by using a generalized Riemann solver consistent with [109,127].

The chapter begins with an introduction of several velocity functions which have been historically used in the LWR PDE, in Section 1. In Section 2, a conservation law for velocity consistent with the LWR PDE is derived for the Greenshields velocity function. We prove for general invertible velocity functions that this equivalence cannot be achieved. In Section 3 we circumvent this issue by developing a discrete time discrete space velocity evolution equation consistent with the discretized LWR PDE. The model is extended to networks in Section 4.

13.1. Velocity functions

In order to obtain a velocity of local evolution equation consistent with density, we require the

velocity function used in the LWR PDE (6) to be invertible. The algebraic expression of the velocity function is a modeling choice, and it is typically constructed to fit experimental data.

Introduced in 1935, one of the earliest velocity functions considered is the Greenshields [174] affine velocity function:

$$v = V_G(\rho) = v_{max}(1 - \rho/\rho_{max})$$

where v_{max} is the maximum (freeflow) velocity, and $\rho_{\max}$ is the maximum (jam) density. This model remains a useful mathematical model because of its algebraic simplicity, despite disagreements with observed traffic data. Since it expresses a linear relationship between speed and density, it is clearly invertible as:

$$\rho = V_G^{-1}(v) = \rho_{max}(1 - v/v_{max}) \tag{34}$$

The widely used Daganzo–Newell velocity function assumes a constant velocity in free-flow and a hyperbolic velocity in congestion:

$$v = V_{\mathrm{DN}}(\rho) = \begin{cases} v_{max} & \text{if } \rho \leq \rho_c \\ -w_f\left(1 - \dfrac{\rho_{max}}{\rho}\right) & \text{otherwise} \end{cases}$$

where v_{max}, ρ_{max}, ρ_c and w_f are respectively the maximum velocity, maximum density, critical density at which the flow transitions from free-flow to congested, and the backwards propagating wave speed, respectively. Because the Daganzo-Newell velocity function is not strictly monotonic in freeflow, it cannot be inverted.

In order to use the Daganzo-Newell model in a velocity setting, we approximate it by the Smulders velocity function [305], with a linear expression in free-flow and a hyperbolic expression in congestion:

$$v = V_{\mathrm{S}}(\rho) = \begin{cases} v_{max}\left(1 - \dfrac{\rho}{\rho_{max}}\right) & \text{if } \rho \leq \rho_c \\ -w_f\left(1 - \dfrac{\rho_{max}}{\rho}\right) & \text{otherwise} \end{cases}$$

For continuity of the flux at the critical density ρ_c, the additional relation $\frac{\rho_c}{\rho_{max}} = \frac{w_f}{v_{max}}$ must be satisfied.

The Smulders velocity function can be inverted to obtain the velocity as a function of density:

$$\rho = V_S^{-1}(v) = \begin{cases} \rho_{max}\left(1 - \dfrac{v}{v_{max}}\right) & \text{if } v \geq v_c \\ \rho_{max}\left(\dfrac{1}{1 + \dfrac{v}{w_f}}\right) & \text{otherwise} \end{cases} \tag{35}$$

where v_c is the critical velocity: $v_c = V(\rho_c)$. This Smulders velocity function yields a quadratic-linear flux function as illustrated in Figure 1.1.

Unless noted otherwise, we assume that the velocity function is invertible throughout the remainder of this discussion.

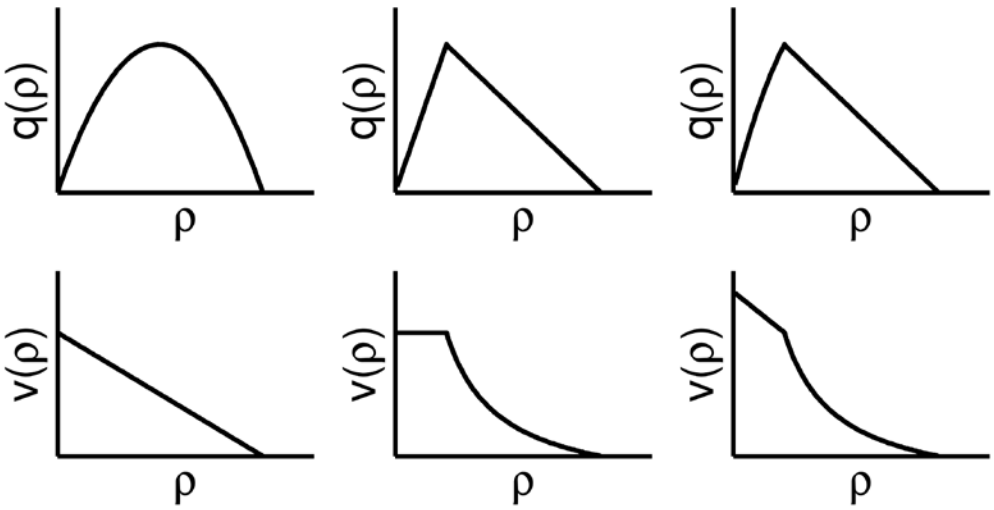


Figure 1.1. Fundamental diagrams (top row) and velocity functions (bottom row) for Greenshields (left), Daganzo-Newell (center), and Smulders (right).

13.2. Derivation of a velocity PDE in conservative form for the Greenshields flux function

In this section, we derive a velocity PDE in conservative form for the Greenshields flux and we show that for other C^1 velocity functions, there is no velocity transport equation equivalent to the LWR equation. The important result shown here is that unless the velocity function is affine (i.e., the Greenshields case), there will not be equivalence between weak solutions to the derived velocity PDE and the weak solutions of the density PDE written in terms of the velocity.

First, we introduce the notion of a weak velocity solution to the LWR PDE. Assuming that the velocity function is invertible with inverse $V^{-1}(\cdot)$, the PDE (16) in weak form for $\rho(\cdot,\cdot)$ is equivalent to the following formulation for $v(\cdot,\cdot)$:

$$\int_0^L \int_0^T \left(V^{-1}(v(x,t)) \frac{\partial \varphi}{\partial t}(x,t) + Q(V^{-1}(v(x,t))) \frac{\partial \varphi}{\partial x}(x,t) \right) dxdt$$

In order to use existing numerical analysis schemes for the PDE we want to obtain, we would like to transform the weak formulation (36) into the following conservation law for velocity with initial condition $v_0(\cdot)$:

$$\begin{cases} \frac{\partial}{\partial t} v(x,t) + \frac{\partial}{\partial x} R(v(x,t)) = 0 \\ v(x,0) = v_0(x) \end{cases} \tag{37}$$

By analogy with the classical LWR equation, the velocity PDE (37) is called LWR-v PDE. Because the flux function $R(v)$ in the velocity conservation law (37) is convex, the weak boundary conditions are given as follows:

Definition 14. *Weak boundary conditions - convex flux function [69,149]* For a convex flux function $R(\cdot)$, the weak formulation of boundary conditions reads:

$$\text{for } a.e.\, t > 0,$$
$$\begin{cases} & v(0,t) = v_l(t) \\ \text{xor} & R'(v(0,t)) \leq 0 \text{ and } R'(v_l(t)) \leq 0 \text{ and } v(0,t) \neq v_l(t) \\ \text{xor} & R'(v(0,t)) \leq 0 \text{ and } R'(v_l(t)) > 0 \text{ and } R(v(0,t)) \geq R(v_l(t)) \end{cases}$$

and

$$\text{for } a.e.\, t > 0,$$
$$\begin{cases} & v(L,t) = v_r(t) \\ \text{xor} & R'(v(L,t)) \geq 0 \text{ and } R'(v_r(t)) \geq 0 \text{ and } v(L,t) \neq v_r(t) \\ \text{xor} & R'(v(L,t)) \geq 0 \text{ and } R'(v_r(t)) < 0 \text{ and } R(v(L,t)) \geq R(v_r(t)) \end{cases}$$

where $v_l(\cdot)$, $v_r(\cdot)$ are functions of $C^0(0,T)$. The functions $v_l(\cdot)$ and $v_r(\cdot)$ are the strong boundary conditions one wants to apply at the left and the right boundaries.

We can now state the main result of this section, which defines the velocity functions for which a velocity evolution PDE in conservative form can be constructed.

Theorem 15.

For a velocity function piecewise analytic in $[0, \rho_{max}]$, the velocity PDE in weak form (36) is equivalent to system (37) if and only if the velocity function is affine (Greenshields case).

Proof. The proof proceeds in two steps. Beginning with equation (36) instantiated for the

Greenshields velocity function $V_G(\cdot)$ defined by (34), we show that the conservation equation obtained is the one from system (37). Substitution of the explicit expression of V_G^{-1} in (36) yields:

$$\int_0^L \int_0^T \rho_{max} \frac{\partial}{\partial t}\varphi(x,t)dxdt - \int_0^L \int_0^T \frac{\rho_{max}}{v_{max}} v(x,t) \frac{\partial}{\partial t}\varphi(x,t)dxdt + \int_0^L \int_0^T Q_G\left(\rho_{max} - \frac{\rho_{max}}{v_{max}} v(x,t)\right) \frac{\partial}{\partial x}\varphi(x,t)dxdt - \int_0^L \frac{\rho_{max}}{v_{max}} v_0(x)\, \varphi(x,0)dx + \int_0^L \rho_{max}\, \varphi(x,0)dx = 0$$

where $Q_G(\rho) = \rho\, V_G(\rho)$. Since $\varphi \in C_c^2([0,L] \times [0,T))$ the first term equals $-\int_0^L \rho_{max}\, \varphi(x,0)dx$ and cancels with the last term. Multiplication by $-\frac{v_{max}}{\rho_{max}}$ gives:

$$\int_0^L \int_0^T v\,(x,t) \frac{\partial}{\partial t}\varphi(x,t)dxdt + \int_0^L v_0\,(x)\, \varphi(x,0)dx - \int_0^L \int_0^T \frac{v_{max}}{\rho_{max}} Q_G\left(\rho_{max} - \frac{\rho_{max}}{v_{max}} v(x,t)\right) \frac{\partial}{\partial x}\varphi(x,t)dxdt = 0$$

which means that v is a weak solution of the PDE:

$$\frac{\partial}{\partial t}v(x,t) + \frac{\partial}{\partial x}(R_G(v(x,t))) = 0$$

with the initial condition $v(x,0) = v_0(x)$, and the velocity flux function

$$R_G(v) = -\frac{v_{max}}{\rho_{max}} Q_G(V_G^{-1}(v)) = v^2 - v_{max}\, v$$

This completes the first part of the proof. Now, we show that the Rankine-Hugoniot jump condition [139,224] is not conserved in the transformation from (16) to (37) for the general case, which means that the equivalence is not obtained for general flux functions.

First, note that a necessary condition to have equivalence between the LWR PDE (16) and the LWR-v PDE (37) is to have the same characteristic speeds for a state ρ in (16) and for the state $V(\rho)$ in (37). This yields $Q'(V^{-1}(v)) = R'(v)$. Integrating this relation between any states (ρ_1, v_1) and (ρ_2, v_2) we obtain:

$$\int_{v_1}^{v_2} Q'\,(V^{-1}(v))dv = \int_{v_1}^{v_2} R'\,(v)dv$$

Using the variable change $v = V(\rho)$, we obtain:

$$\int_{\rho_1}^{\rho_2} Q'(\rho)\, V'(\rho) d\rho = \int_{v_1}^{v_2} R'(v) dv \tag{38}$$

Next, at a discontinuity, the Rankine-Hugoniot jump condition [139,224] reads:

$$\frac{Q(\rho_2) - Q(\rho_1)}{\rho_2 - \rho_1} = \frac{R(v_2) - R(v_1)}{v_2 - v_1} \tag{39}$$

which we can rewrite as:

$$\int_{v_1}^{v_2} R'(v) dv = \frac{v_2 - v_1}{\rho_2 - \rho_1} \int_{\rho_1}^{\rho_2} Q'(\rho) d\rho \tag{40}$$

If we substitute equality (38) into equation (40) we obtain:

$$\int_{\rho_1}^{\rho_2} Q'(\rho)\, V'(\rho) d\rho = \frac{V(\rho_2) - V(\rho_1)}{\rho_2 - \rho_1} \int_{\rho_1}^{\rho_2} Q'(\rho) d\rho$$

which translates to:

$$\begin{aligned} &\int_{\rho_1}^{\rho_2} V'(\rho)\,(V(\rho) + \rho\, V'(\rho)) d\rho \\ &\qquad = \left(\frac{1}{\rho_2 - \rho_1} \int_{\rho_1}^{\rho_2} V'(\rho) d\rho\right) \left(\int_{\rho_1}^{\rho_2} (V(\rho) + \rho\, V'(\rho)) d\rho\right) \end{aligned} \tag{41}$$

If we define the function G_{ρ_1} in $[\rho_1, \rho_i]$ by $G_{\rho_1}(\rho_2) = \frac{1}{\rho_2 - \rho_1} \int_{\rho_1}^{\rho_2} V'(\rho) d\rho$, on intervals on which V is smooth, we can write:

$$\begin{aligned} &V'(\rho_2)\,(V(\rho_2) + \rho_2\, V'(\rho_2)) \\ &\qquad = G'_{\rho_1}(\rho_2)\,(\rho_2\, V(\rho_2) - \rho_1\, V(\rho_1)) + G_{\rho_1}(\rho_2)\,(V(\rho_2) + \rho_2\, V'(\rho_2)) \end{aligned} \tag{42}$$

Given the expression of G_{ρ_1}, if we differentiate $(\rho_2 - \rho_1)\, G_{\rho_1}(\rho_2)$ w.r.t ρ_2 we obtain for all ρ_2 in $[\rho_1, \rho_i]$:

$$((\rho_2 - \rho_1)\, G_{\rho_1}(\rho_2))' = G_{\rho_1}(\rho_2) + (\rho_2 - \rho_1)\, G'_{\rho_1}(\rho_2) = V'(\rho_2)$$

Thus if we factor $V(\rho_2) + \rho_2\, V'(\rho_2)$ in the first and last term of (42) and if we replace $G_{\rho_1}(\rho_2) - V'(\rho_2)$ by $-(\rho_2 - \rho_1)\, G'_{\rho_1}(\rho_2)$ we obtain:

$$G'_{\rho_1}(\rho_2)\,((\rho_2\, V(\rho_2) - \rho_1\, V(\rho_1)) - (\rho_2 - \rho_1)\,(V(\rho_2) + \rho_2\, V'(\rho_2))) = 0 \tag{43}$$

The second term in the product can be written as $Z(\rho_1, \rho_2) = Q(\rho_2) - Q(\rho_1) - (\rho_2 - \rho_1)\, Q'(\rho_2)$. So either $Q(\cdot)$ is affine and $Z(\rho_1, \rho_2)$ is zero, either Q is strictly concave or

strictly convex and $Z(\rho_1, \rho_2)$ is different from zero, and the first term of (43) must be zero. If the first term in (43) is zero, it means that V is of the form $V(\rho) = a\,\rho + b$. If the second term is zero, it means that V is of the form $V(\rho) = \frac{a}{\rho} + b$. So we obtain a necessary condition that V must be piecewise affine or hyperbolic.

If there exists a point $\rho_i \in [0, \rho_{max}]$ s.t. V has a different algebraic expression for $\rho > \rho_i$ and $\rho < \rho_i$, simple algebra shows that the equality of the Rankine-Hugoniot speeds (39) does not hold in general. Therefore V is either of the form $a\,\rho + b$ in $[0, \rho_{max}]$, or $\frac{a}{\rho} + b$ in $[0, \rho_{max}]$. The second possibility is excluded by assumption on V (unbounded speed as ρ goes to zero). □

Thus for more realistic traffic models with nonlinear velocity functions, it is not possible to derive a PDE model for velocity in conservation form (37).

13.3. Numerical approximation of the velocity evolution equation

Since a partial differential equation for velocity consistent with the LWR PDE does not exist for arbitrary invertible velocity functions, we instead return to the integral form of the LWR PDE (1) to perform the variable change. Then we will derive a Godunov scheme for velocity to approximate the solution.

Following the same procedure as in Section 1, we discretize the time and space domains by introducing a discrete time step ΔT, indexed by $n \in \{0, \cdots, n_{max}\}$ and a discrete space step Δx, indexed by $i \in \{0, \cdots, i_{max}\}$. Let us integrate equation (1) over a single timestep, and apply the variable change $\rho(x,t) = V^{-1}(v(x,t))$:

$$\begin{aligned}\int_{x_{i-1/2}}^{x_{i+1/2}} V^{-1}\left(v(x,t_{n+1})\right)dx - \int_{x_{i-1/2}}^{x_{i+1/2}} V^{-1}\left(v(x,t_n)\right)dx \\ = \int_{t_n}^{t_{n+1}} Q\left(V^{-1}(v(x_{i-1/2},t))\right)dt - \int_{t_n}^{t_{n+1}} Q\left(V^{-1}(v(x_{i+1/2},t))\right)dt\end{aligned} \tag{44}$$

Define the piecewise constant function $\bar{v}(x,t) := V^{-1}(\bar{\rho}(x,t))$ where $\bar{\rho}(x,t)$, is a piecewise constant approximation of $\rho(x,t)$ and $\bar{\rho}(x,t)$ is a constant in each cell. We introduce the variable v_i^n and recall the definition of Q_i^n as follows:

$$v_i^n = \frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_{i+1/2}} \bar{v}\,(x,t_n)dx \tag{45}$$

$$Q_i^n \approx \frac{1}{\Delta T}\int_{t_n}^{t_{n+1}} Q\left(V^{-1}(v(x_i,t))\right)dt \tag{46}$$

where v_i^n approximates the average velocity in the ith cell at time t_n and Q_i^n approximates the average flux at x_i over the time interval $[t_n, t_{n+1}]$. Substituting into (44) and rearranging, we obtain

$$V^{-1}(v_i^{n+1}) = V^{-1}(v_i^n) - \frac{\Delta T}{\Delta x}\left(Q_{i+1/2}^n - Q_{i-1/2}^n\right) \tag{47}$$

which is the basis of the Godunov discretization scheme for velocity evolution. The full algorithm is as follows.

1. Approximate the function $v(x,t)$ with a piecewise constant function $\bar{v}(x,t)$, where $\bar{v}(x,t)$ is constant in each cell.
2. Solve the Riemann problems at the cell boundaries $x_{i+1/2}$ and $x_{i-1/2}$. Let $\tilde{G}(v_i^n, v_{i+1}^n)$ denote the flux at $x_{i+1/2}$ when the Riemann problem is solved between the two states v_i^n and v_{i+1}^n. For consistency with the density evolution, we require $\tilde{G}(v_i^n, v_{i+1}^n) = G(V^{-1}(v_i^n), V^{-1}(v_{i+1}^n))$ where $G(\rho_1, \rho_2))$ is given by (32), and solves the Riemann problem exactly. Moreover, let us define $\tilde{Q}(v) := Q(V^{-1}(v))$, so by substitution into (32), we obtain

$$\tilde{G}(v_1, v_2) = \begin{cases} \tilde{Q}(v_2) & \text{if } V^{-1}(\rho_c) \le V^{-1}(v_2) \le V^{-1}(v_1) \\ \tilde{Q}(v_c) & \text{if } V^{-1}(v_2) \le V^{-1}(v_c) \le V^{-1}(v_1) \\ \tilde{Q}(v_1) & \text{if } V^{-1}(v_2) \le V^{-1}(v_1) \le V^{-1}(v_c) \\ \min\left(\tilde{Q}(v_1), \tilde{Q}(v_2)\right) & \text{if } V^{-1}(v_1) \le V^{-1}(v_2) \end{cases} \tag{48}$$

Note that if $\rho_1 \le \rho_2$, with $v_1 = V(\rho_1)$ and $v_2 = V(\rho_2)$, then $v_1 \ge v_2$ when $V(\cdot)$ is monotonically decreasing (which is typically the case for traffic applications). Then $\tilde{G}(v_i^n, v_{i+1}^n)$ is given by:

$$\tilde{G}(v_1, v_2) = \begin{cases} \tilde{Q}(v_2) & \text{if } v_c \ge v_2 \ge v_1 \\ \tilde{Q}(v_c) & \text{if } v_2 \ge v_c \ge v_1 \\ \tilde{Q}(v_1) & \text{if } v_2 \ge v_1 \ge v_c \\ \min\left(\tilde{Q}(v_1), \tilde{Q}(v_2)\right) & \text{if } v_1 \ge v_2 \end{cases} \tag{49}$$

This yields

$$Q_{i+1/2}^n = \frac{1}{\Delta T}\int_{t_n}^{t_{n+1}} Q\left(V^{-1}(\bar{v}(x_{i+1/2}, t))\right)dt = \tilde{G}(v_i^n, v_{i+1}^n)$$

3. Compute the density at the next timestep according to (47).

In practice, the scheme is implemented as the nonlinear discrete evolution equation:

$$v_i^{n+1} = V\left(V^{-1}(v_i^n) - \frac{\Delta T}{\Delta x}\left(\tilde{G}(v_i^n, v_{i+1}^n) - \tilde{G}(v_{i-1}^n, v_i^n)\right)\right) \tag{50}$$

which we call the *Cell Transmission Model for velocity* CTM-v.

Example 16. (*Smulders model)* After evaluation of the function (35), equation (49) reduces to:

$$\tilde{G}(v_1, v_2) = \begin{cases} v_2 \rho_{max} \left(\dfrac{1}{1 + \dfrac{v_2}{w_f}} \right) & \text{if } v_c \geq v_2 \geq v_1 \\ v_c \rho_{max} \left(1 - \dfrac{v_c}{v_{max}} \right) & \text{if } v_2 \geq v_c \geq v_1 \\ v_1 \rho_{max} \left(1 - \dfrac{v_1}{v_{max}} \right) & \text{if } v_2 \geq v_1 \geq v_c \\ \min\left(V_S^{-1}(v_1)v_1, V_S^{-1}(v_2)v_2\right) & \text{if } v_1 \geq v_2 \end{cases} \tag{51}$$

We choose not to simplify the last line in (51) due to the piecewise analytical expression of function $V_S^{-1}(\cdot)$.

We note that the evolution of the velocity field at each discrete point on an edge except at the boundary points v_0^n and $v_{i_{max}}^n$ is well defined by (50) and (51). At these boundaries, the equations

$$v_0^{n+1} = V\left(V^{-1}(v_0^n) - \frac{\Delta T}{\Delta x}\left(\tilde{G}(v_0^n, v_1^n) - \tilde{G}(v_{-1}^n, v_0^n) \right) \right) \tag{52}$$

$$v_{i_{max}}^{n+1} = V\left(V^{-1}\left(v_{i_{max}}^n\right) - \frac{\Delta T}{\Delta x}\left(\tilde{G}\left(v_{i_{max}}^n, v_{i_{max}+1}^n\right) - \tilde{G}\left(v_{i_{max}-1}^n, v_{i_{max}}^n\right) \right) \right) \tag{53}$$

contain references to the ghost sells v_{-1}^n and $v_{i_{max}+1}^n$, which are points which do not lie in the physical domain. The values of v_{-1}^n and $v_{i_{max}+1}^n$ are given by the prescribed boundary conditions to be imposed on the left and right side of the domain respectively. Note that these boundary values do not always affect the physical domain because of the nonlinear operator (51), which causes the boundary conditions to be implemented in the weak sense (See Section 2).

13.4. Extension of the model to networks

13.4.1. Network model and edge boundary conditions at junctions

We now describe the extension of the velocity evolution equation to networks. On each edge, the velocity field evolves according to (50), with an important modification in the computation of the points at the boundary. Instead of implementing ghost points, it is natural to require the left and right boundary conditions to be a function of upstream and downstream edges, so that the velocity field can be evolved across the network.

We model the highway transportation network as a directed graph consisting of vertices $\nu \in \mathcal{V}$ and edges $e \in \mathcal{E}$. Let L_e be the length of edge e. The spatial and temporal variables are $x \in [0, L_e]$, and $t \in [0, +\infty)$ respectively. In order to model traffic flow across the network, we define a junction $j \in \mathcal{J}$ as a tuple $\left(\nu_j, I_j, \mathcal{O}_j\right) \subseteq \mathcal{V} \times \mathcal{E} \times \mathcal{E}$, consisting of a single vertex $\nu_j \in \mathcal{V}$,

a set of incoming edges indexed by $e_{\text{in}} \in \mathcal{I}_j$, and a set of outgoing edges indexed by $e_{\text{out}} \in \mathcal{O}_j$.

In general, extending the velocity model to handle these networks is challenging because one must prescribe a unique solution to the velocity Riemann problem on junctions, where multiple road segments merge or diverge. In general, even with mass conservation and a natural extension of the entropy condition, a unique solution to the Riemann problem in the density domain is not guaranteed, and therefore a unique solution in the velocity domain is not guaranteed either.

To illustrate the problem, let us consider the density Riemann problem for a junction with one incoming edge ($\mathcal{I} = \{1\}$) and to outgoing edges ($\mathcal{O} = \{2,3\}$), with initial conditions given by $\rho_1(x,0) = \rho_{\max}$ and $\rho_2(x,0) = \rho_3(x,0) = 0$, shown in Figure 1. Let us also assume that the flux functions on each edge are the same.

An infinite number of solutions exist which satisfy the LWR PDE on each edge. For example, Figure 1 shows one such solution, where no vehicles pass through the vertex. The solution is given by $\rho_1(x,0) = \rho_{max}$ and $\rho_2(x,0) = \rho_3(x,0) = 0$. Because the data is piecewise constant on each edge, of the LWR PDE is trivially satisfied.

Other solutions can be constructed with vehicles passing through the vertex, shown in Figure 1(c) and 1(d). In Figure 1(c), the maximal number of vehicles that can be sent from the incoming edge are sent, but all vehicles are received by edge two, and none are received by edge three. In this case, the solution satisfying the LWR PDE on each edge is given by

$$\rho_1(x,t) = \begin{cases} \rho_{\max} & \text{if } x < (Q')^{-1}(\rho_{\max})t, \\ (Q')^{-1}(\frac{x}{t}) & \text{otherwise} \end{cases}$$

$$\rho_2(x,t) = \begin{cases} (Q')^{-1}(\frac{x}{t}) & \text{if } x < (Q')^{-1}(0)t, \\ 0 & \text{otherwise} \end{cases}$$

$$\rho_3(x,t) = 0$$

Figure 1 shows the opposite scenario, where all vehicles are sent to edge three, and none are sent to edge two (the solutions of $\rho_2(x,t)$ and $\rho_3(x,t)$ are interchanged).

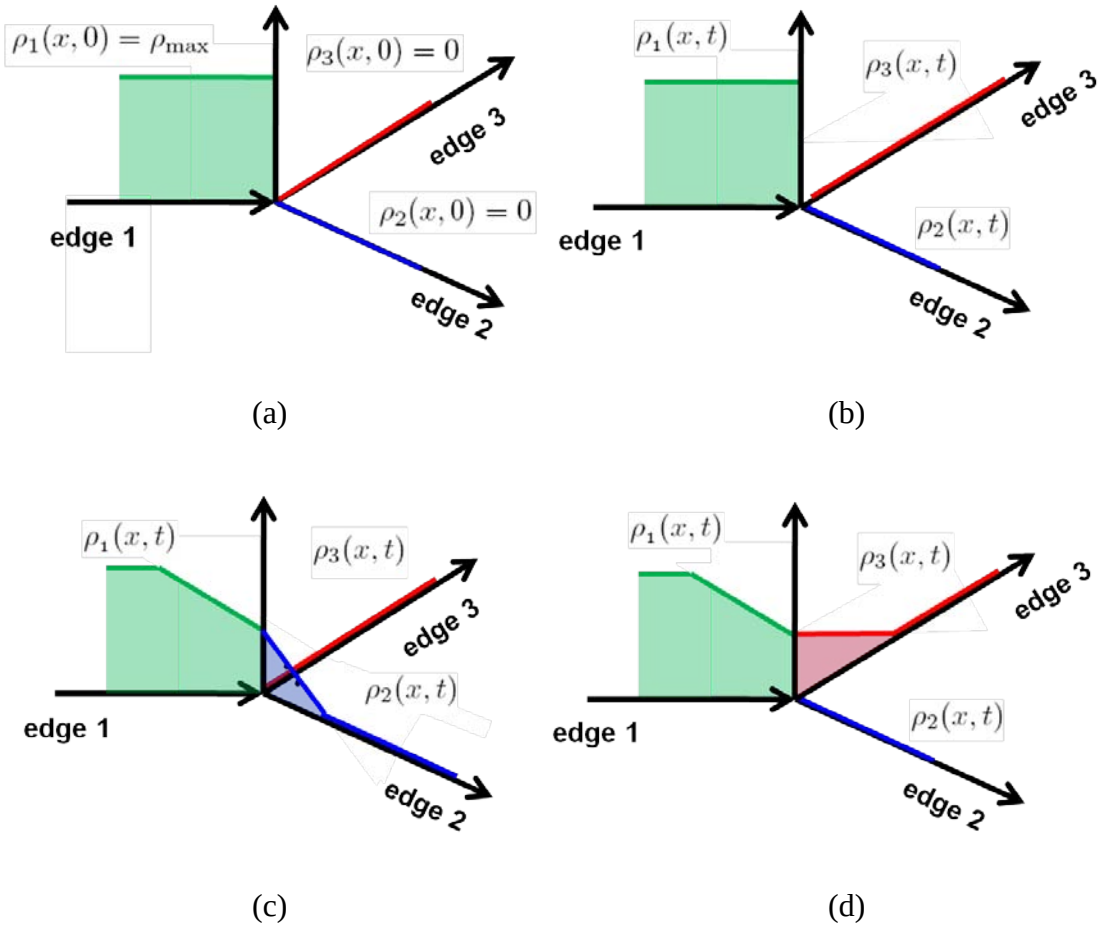


Figure 4.1. Riemann problem for a diverge (a) initial condition; (b) a solution with no vehicles crossing the junction; (c) a solution with all vehicles received by edge 2; (d) a solution with all vehicles received by edge 3.

To resolve this nonuniqueness, several Riemann solvers in the density domain have been proposed in the literature. The first solver is due to Holden and Risebro [194], which maximizes a strictly convex function of the individual edge fluxes into and out of the junction, subject to mass conservation. Because the mass conservation constraints are linear, and the objective function is strictly convex, the solution to the Riemann problem is unique. In the context of the discretized LWR PDE (cell transmission model), Daganzo [127] provides a unique evolution of density where the total flux is maximized across the junction, subject to flow allocation parameters which encode the proportion of flow from an incoming edge associated to an outgoing edge. For diverge problems, a unique solution is constructed with the aid of additional priority parameters which specify preference for flows from upstream edges when a downstream link comes congested. Coclite, Garavello, and Piccoli [109] formalize this Riemann solver in the continuous domain and show existence of a global solution using wave front tracking [80]. A Riemann solver with internal state dynamics was also proposed by Labacque [219], and a multilane solver was introduced by Herty and Klar in [185] to better model traffic flow through intersections. The incremental transfer principle introduced by Daganzo [125] is a also Riemann solver which can be used to model exit and high occupancy vehicle lanes, yielding more realistic flow dynamics at intersections. The interested reader is referred to the recent book [155] for a

detailed mathematical treatment of traffic flows on networks.

In what follows, we summarize the Riemann solver [109,127], which will be the basis of our velocity model extension to networks.

We look for unique description of the evolution of the velocity dynamics at the junctions. Following the conditions for uniqueness of [155], we present three physically motivated restrictions on the dynamics, namely *(i)* conservation of vehicles across the junction, *(ii)* vehicles follow a set route across the junction, which define how the traffic flux from edges into the junction are routed to the outgoing edges *(iii)* traffic flow across the junction is maximized. Conditions *(i)* and *(ii)* imply that for the edge boundaries at the junction, boundary conditions must hold in the strong sense. This creates an upper bound on the flows on each edge into and out of the junction, which can be computed. By transforming these conditions into the velocity domain, the velocity evolution at the junctions can be determined by solving a linear programming problem.

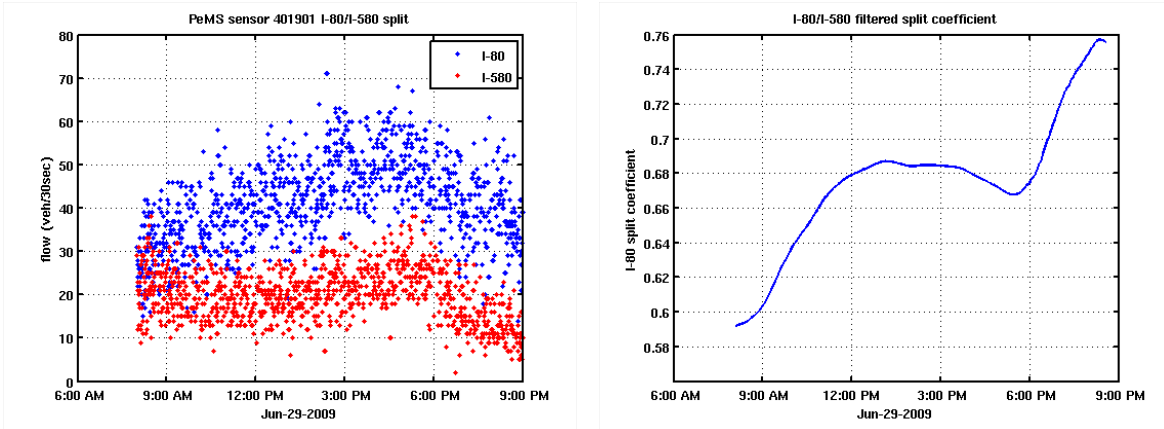
Physical constraints

Consider a junction j with $|\mathcal{I}_j|$ incoming edges and $|\mathcal{O}_j|$ outgoing edges. First, we assume that the junction has no storage capacity, so all vehicles which enter the junction must also exit the junction. Conservation of the number of vehicles across the junction gives rise to the constraint that the total flux into the junction must equal the total flux out of the junction:

$$\sum_{e_{\text{in}} \in \mathcal{I}_j} \tilde{Q}_{e_{\text{in}}} \left(v_{e_{\text{in}}} \left(L_{e_{\text{in}}}, t \right) \right) = \sum_{e_{\text{out}} \in \mathcal{O}_j} \tilde{Q}_{e_{\text{out}}} \left(v_{e_{\text{out}}}(0, t) \right) \tag{54}$$

Next, we assume that the total volume of traffic entering from an incoming edge is distributed amongst the outgoing edges according to an allocation parameter $\alpha_{j,e_{\text{in}},e_{\text{out}}}(t) \geq 0$. The allocation matrix $A_j \in [0,1]^{|\mathcal{O}_j| \times |\mathcal{I}_j|}$, where $A_j(e_{\text{out}}, e_{\text{in}}) = \alpha_{j,e_{\text{out}},e_{\text{in}}}$, encodes the aggregate routing information of the traffic across the junction. That is, for all vehicles entering the junction j on edge e_{in}, $\alpha_{j,e_{\text{out}},e_{\text{in}}}$ denotes the proportion of vehicles which will exit the junction through edge e_{out}. This proportion can be determined empirically using historical origin-destination tables, or by analyzing the volumes of data collected near the junction (See Figure 55). Because the vertex has no storage capacity, the sum of allocated flows from a fixed incoming link across all outgoing flows must be equal to one:

$$\sum_{e_{\text{out}} \in \mathcal{O}_j} \alpha_{e_{\text{out}},e_{\text{in}}} = 1 \tag{55}$$



(a) (b)

Figure 4.2. (a) Vehicle flows for the I80 – I580 diverge near Berkeley, California, obtained from the PeMS system; (b) filtered time-varying allocation parameter for flow to I80.

Note that constraints *(i)* and *(ii)* combined imply $A_j \tilde{Q}_{e_{\text{in}}} = \tilde{Q}_{e_{\text{out}}}$. If we view the exiting flows from the incoming edges of the junction as a boundary condition for an outgoing edge, then the physical constraint $\sum_{e_{\text{in}} \in \mathcal{I}_j} \alpha_{e_{\text{out}}, e_{\text{in}}} \tilde{Q}_{e_{\text{in}}} = \tilde{Q}_{e_{\text{out}}}$ for each e_{out} can be interpreted as a requirement that strong boundary conditions must be imposed on e_{out}. But strong boundary conditions (*i.e.* equality) cannot always be imposed for an arbitrary pair $\left(\sum_{e_{\text{in}} \in \mathcal{I}_j} \alpha_{e_{\text{out}}, e_{\text{in}}} \tilde{Q}_{e_{\text{in}}}, \tilde{Q}_{e_{\text{out}}}\right)$, so the statement of strong boundary conditions ((21) and (22) for a concave flux) provides upper bounds on the admissible incoming and admissible outgoing fluxes over which the flow is maximized (constraint (*iii*)). The maximum incoming admissible flux into the junction from edge e_{in} given a desired velocity $v_{e_{\text{in}}}$ to be prescribed in the strong sense is denoted by $\gamma_{e_{\text{in}}}^{max}(v_{e_{\text{in}}})$ (resp. $\delta_{e_{\text{in}}}^{\max}(\rho_{e_{\text{in}}})$ for a given density). Similarly, the maximum outgoing admissible flux out of the junction from edge e_{out} given a desired velocity $v_{e_{\text{out}}}$ to be prescribed in the strong sense is denoted by $\gamma_{e_{\text{out}}}^{\max}(v_{e_{\text{out}}})$ (resp. $\delta_{e_{\text{in}}}^{\max}(\rho_{e_{\text{in}}})$ for a given density).

Thus the three conditions give rise to the following linear program for the fluxes (denoted by the vector dummy variable $\xi \in \mathbb{R}^{|\mathcal{I}|}$) on the incoming edges e_{in} for junction j:

$$\begin{aligned} \textbf{\textit{maximize}} \quad & 1^T \xi \\ \textbf{\textit{subject to}} \quad & A_j \xi \leq \gamma_{\mathcal{O}_j}^{max} \\ & \mathbf{0} \leq \xi \leq \gamma_{\mathcal{I}_j}^{max} \end{aligned} \tag{56}$$

where $\gamma_{\mathcal{I}_j}^{max} := \left(\gamma_{e_{\text{in},1}}^{max}, \cdots, \gamma_{e_{\text{in},|\mathcal{I}_j|}}^{max}\right)$, $\gamma_{\mathcal{O}_j}^{max} := \left(\gamma_{e_{\text{out},1}}^{max}, \cdots, \gamma_{e_{\text{out},|\mathcal{O}_j|}}^{max}\right)$ are the upper bounds on the fluxes on the edges entering and exiting the junction, to be computed subsequently. With the optimal solution to (56), denoted by $\xi_{e_{\text{in}}}^*$, the terms $\tilde{G}_{e_{\text{in}}}\left(v_{i_{max}}^n, v_{i_{max}+1}^n\right)$ and $\tilde{G}_{e_{\text{out}}}(v_{-1}^n, v_0^n)$ in the CTM-v (52) and (53) are given by:

$$\tilde{G}_{e_{\text{in}}}\left(v_{i_{max}}^{n}, v_{i_{max}+1}^{n}\right) = \xi_{e_{\text{in}}}^{*} \tag{57}$$

$$\tilde{G}_{e_{\text{out}}}(v_{-1}^{n}, v_{0}^{n}) = \sum_{e_{\text{in}} \in \mathcal{I}_j} \alpha_{e_{\text{out}}, e_{\text{in}}} \xi_{e_{\text{in}}}^{*} \tag{58}$$

We note that the solution to this linear program is not always unique. In fact, for some instantiations of A_j, the gradient of the objective function may be normal to a facet of the constraint set polytope, in which case all feasible points on the facet will obtain the same objective value. This can be resolved with a technical condition on the coefficients A_j [109,155], to explicitly prevent this nonuniqueness. However, when multiple links merge into a single link, additional priority constraints (detailed in [109,127]) must be added to resolve the nonuniqueness, in which case the optimization problem becomes an integer program. Regardless, these optimization problems are small (typically only a few variables and less than 10 constraints) and can be solved quickly, even by brute force.

Computation of the maximum admissible flux

First we introduce a function $\tau(\cdot)$, used to describe the domain for which we obtain admissible fluxes $Q(\cdot)$. For a continuous strictly concave C^0 flux function with $Q(0) = Q(\rho_{\max})$, the mapping from flux $Q(\rho)$ to ρ is double valued, with one value above and one value below the critical value ρ_c. For a given ρ, $\tau(\rho)$ is the map which produces the alternate ρ for the same flux. The function is expressed as follows:

$$Q(\tau(\rho)) = Q(\rho) \quad \forall\, \rho \in [0, \rho_{\max}]$$

$$\tau(\rho) \neq \rho \quad \forall \rho \in [0, \rho_{\max}] \backslash \{\rho_c\}$$

Given that $Q(\cdot)$ is in $C^0([0, \rho_{max}])$, strictly increasing in $[0, \rho_c)$ and strictly decreasing in $(\rho_c, \rho\ \ ]$ the following holds:

$$0 \leq \rho \leq \rho_c \Leftrightarrow \rho_c \leq \tau(\rho) \leq \rho_{\max}$$

We now define the upper bounds on the flux entering the junction from each incoming edge, and the flux leaving the junction on each outgoing edge. More precisely, for each incoming and outgoing link, we seek to find the upper bound on the admissible flux entering (resp. leaving) the link such that strong boundary conditions are imposed on the boundaries for all edges at the vertex. First we derive these admissible fluxes $\delta_{e_{\text{out}}}(\cdot)$ (resp. $\delta_{e_{\text{in}}}(\cdot)$) in terms of the trace of the density $\rho_{e_{\text{out}}}(0, t)$ (resp. $\rho_{e_{\text{in}}}(L, t)$), then apply the velocity inversion to arrive at admissible fluxes $\gamma_{e_{\text{out}}}(\cdot)$ (resp. $\gamma_{e_{\text{in}}}(\cdot)$) in terms of the trace of the velocity $v_{e_{\text{out}}}(0, t)$ (resp. $v_{e_{\text{in}}}(L, t)$).

For a strictly concave flux $Q(\cdot)$ with a maximum obtained at the critical value ρ_c we categorize the values of $\rho(0,\cdot)$ and $\rho_l(\cdot)$ for which (54) holds:

$$\text{for } a.e.\, t > 0, \rho(0,t) = \rho_l(t) \text{ iff}$$
$$\begin{cases} \quad\quad \rho(0,t) \in [0,\rho_c] \text{ and } \rho_l(t) \in [0,\rho_c] \\ \text{xor} \quad \rho(0,t) \in (\rho_c,\rho_{\max}] \text{ and } \rho_l(t) \in [0,\tau(\rho(0,t))) \cap \{\rho(0,t)\} \end{cases} \tag{59}$$

Recalling that incoming admissible fluxes are the set of fluxes corresponding to boundary data for the outgoing links which can be imposed in the strong sense, we can define the set of incoming admissible fluxes on an outgoing edge in two parts:
For $\rho_{e_{\text{out}}}(0,t) \in [0,\rho_{c,e_{\text{out}}}]$:

$$\delta_{e_{\text{out}}}\left(\rho_{e_{\text{out}}}(0,t)\right) \in \Pi_{e\ \text{out}}\left(\rho_{e_{\text{out}}}(0,t)\right) := \left\{\hat{Q}: \exists \hat{\rho} \in \left[0,\rho_{c,e_{\text{out}}}\right];\ \hat{Q} = Q(\hat{\rho})\right\} \tag{60}$$

where $\rho_{c,e_{\text{out}}}$ is the critical density on the edge e_{out}.
For $\rho_{e_{\text{out}}}(0,t) \in [\rho_{c,e_{\text{out}}},\rho_{\max,e_{\text{out}}}]$:

$$\delta_{e_{\text{out}}}\left(\rho_{e_{\text{out}}}(0,t)\right) \in \Pi_{e\ \text{out}}\left(\rho_{e_{\text{out}}}(0,t)\right) := \\ \left\{\hat{Q}: \exists \hat{\rho} \in \left\{\rho_{e_{\text{out}}}(0,t)\right\} \cup \left[0,\tau\left(\rho_{e_{\text{out}}}(0,t)\right)\right); \hat{Q} = Q(\hat{\rho})\ \right\} \tag{61}$$

Similarly, (55) can be rewritten in terms of outgoing admissible fluxes for incoming edges as:
For $\rho_{e_{\text{in}}}(L_{e_{\text{in}}},t) \in [0,\rho_{c,e_{\text{in}}}]$:

$$\delta_{e_{\text{in}}}\left(\rho_{e_{\text{in}}}\left(L_{e_{\text{in}}},t\right)\right) \in \Pi_{e_{\text{in}}}\left(\rho_{e_{\text{in}}}\left(L_{e_{\text{in}}},t\right)\right) := \\ \left\{\hat{Q}: \exists \hat{\rho} \in \{\rho_{e_{\text{in}}}(L_{e_{\text{in}}},t)\} \cup \left(\tau(\rho_{e_{\text{in}}}(L_{e_{\text{in}}},t),\rho_{\max,e_{\text{in}}}\right]; \hat{Q} = Q(\hat{\rho})\right\} \tag{62}$$

where $\rho_{\max,e_{\text{in}}}$ is the maximum density on the edge e_{in}.
For $\rho_{e_{\text{in}}}(L_{e_{\text{in}}},t) \in [\rho_{c,e_{\text{in}}},\rho_{\max,e_{\text{in}}}]$:

$$\delta_{e_{\text{in}}}\left(\rho_{e_{\text{in}}}\left(L_{e_{\text{in}}},t\right)\right) \in \Pi_{e_{\text{in}}}\left(\rho_{e_{\text{in}}}\left(L_{e_{\text{in}}},t\right)\right) := \\ \left\{\hat{Q}: \exists \hat{\rho} \in \left[\rho_{c,e_{\text{in}}},\rho_{\max,e_{\text{in}}}\right];\ \hat{Q} = Q(\hat{\rho})\right\} \tag{63}$$

If the admissible flux is maximized, and written in terms of velocity, we obtain:

$$\gamma_{e_{\text{out}}}^{\max}\left(v_{e_{\text{out}}}(0,t)\right) = \begin{cases} \tilde{Q}(v_{c,e_{\text{out}}}) & \text{if } v_{e_{\text{out}}}(0,t) \in \left[v_{c,e_{\text{out}}}, v_{\max,e_{\text{out}}}\right] \\ \tilde{Q}\left(v_{e_{\text{out}}}(0,t)\right) & \text{if } v_{e_{\text{out}}}(0,t) \in \left[0, v_{c,e_{\text{out}}}\right] \end{cases}$$

and

$$\gamma_{e_{\text{in}}}^{\max}\left(v_{e_{\text{in}}}(L_{e_{\text{in}}},t)\right) = \begin{cases} \tilde{Q}\left(v_{e_{\text{in}}}(L_{e_{\text{in}}},t)\right) & \text{if } v_{e_{\text{in}}}(L_{e_{\text{in}}},t) \in \left[v_{c,e_{\text{in}}}, v_{\max,e_{\text{in}}}\right] \\ \tilde{Q}\left(v_{c,e_{\text{in}}}\right) & \text{if } v_{e_{\text{in}}}(L_{e_{\text{in}}},t) \in \left[0, v_{c,e_{\text{in}}}\right] \end{cases}$$

which are the upper bounds used in (56).

Example 17. *Maximum admissible flux - Smulders model* The maximum outgoing admissible flux is given as:

$$\gamma_{e_{\text{out}}}^{\max}\left(v_{e_{\text{out}}}(0,t)\right) = \begin{cases} \rho_{max}\left(1-\dfrac{v_{c,e_{out}}}{v_{\max}}\right)v_{c,e_{\text{out}}} \\ \qquad \text{if } v_{e_{\text{out}}}(0,t) \in \left[v_{c,e_{\text{out}}}, v_{\max,e_{\text{out}}}\right] \\ \rho_{max}\left(\dfrac{1}{1+\dfrac{v_{e_{out}}(0,t)}{w_f}}\right)v_{e_{\text{out}}}(0,t) \\ \qquad \text{if } v_{e_{\text{out}}}(0,t) \in \left[0, v_{c,e_{\text{out}}}\right] \end{cases} \tag{64}$$

and the maximum incoming admissible flux is given as:

$$\gamma_{e_{\text{in}}}^{\max}\left(v_{e_{\text{in}}}(L_{e_{\text{in}}},t)\right) = \begin{cases} \rho_{max}\left(1-\dfrac{v_{e_{in}}(L_{e_{in}},t)}{v_{\max}}\right)v_{e_{\text{in}}}(L_{e_{\text{in}}},t) \\ \qquad \text{if } v_{e_{\text{in}}}(L_{e_{\text{in}}},t) \in \left[v_{c,e_{\text{in}}}, v_{\max,e_{\text{in}}}\right] \\ \rho_{max}\left(\dfrac{1}{1+\dfrac{v_{c,e_{in}}}{w_f}}\right)v_{c,e_{\text{in}}} \\ \qquad \text{if } v_{e_{\text{in}}}(L_{e_{\text{in}}},t) \in \left[0, v_{c,e_{\text{in}}}\right] \end{cases} \tag{65}$$

13.4.2. Discrete CTM-v network algorithm

The *CTM-v network algorithm* is obtained by sequentially applying the *CTM-v* scheme on each link of the network and solving the junction conditions as presented in the previous section, which includes solving the LP (56) posed earlier. The algorithm as illustrated in Figure 4.3.

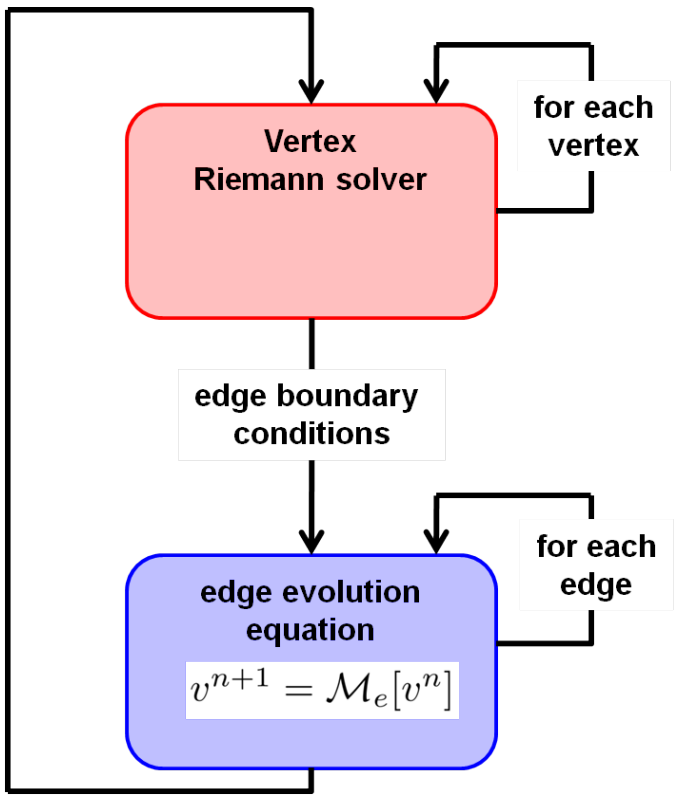


Figure 4.3. The discrete network velocity evolution equation proceeds in two steps. First, the Riemann problem at each vertex is solved to determine the strong boundary conditions for each edge. Then, for each edge the velocity is evolved according to (50).

The network is thus marched in time and consists in a large scale discrete dynamical system which can be used for data assimilation and inverse modeling. Given the velocity field at each discrete point $i \in \{0, \cdots, i_{max}\}$ on all edges of the network

$$v^n := \left[v^n_{0,e_0}, \cdots, v^n_{i_{\max},e_0}, \cdots, v^n_{0,e_{|\mathcal{E}|}}, \cdots, v^n_{i_{\max},e_{|\mathcal{E}|}}\right] \tag{66}$$

the velocity at time $t_{n+1}\Delta T$ is given by:

$$v^{n+1} = \mathcal{M}(v^n, \theta^n) \tag{67}$$

where θ^n represents the parameters of the velocity function on each link $(v_{\max,v_c,w_f})$, flow allocation and priority parameters at the junctions, and boundary data at the network boundaries, and $\mathcal{M}(\cdot,\cdot)$ denotes the following update algorithm:

1. For all junctions $j \in \mathcal{J}$:
   (a) Compute $\gamma^n_{i_{max},e_{\text{in}}}\left(v^n_{i_{max},e_{\text{in}}}\right)$ $\forall e_{\text{in}} \in \mathcal{I}_j$, and $\gamma^n_{0,e_{\text{out}}}\left(v^n_{0,e_{\text{out}}}\right)$ $\forall e_{\text{out}} \in \mathcal{O}_j$ using (64) and (65).
   (b) Solve the LP (56) for ξ^*, and update $\tilde{G}_{e_{\text{in}}}\left(v^n_{i_{max}}, v^n_{i_{nax}+1}\right)$ and $\tilde{G}_{e_{\text{out}}}(v^n_{-1}, v^n_0)$ through (57) and (58).
2. For all edges $e \in \mathcal{E}$: Compute $v^{n+1}_{i,e}$ $\forall i \in \{1, \cdots, i_{\max,e}\}$ according to the CTM-v (50), (52), and (53).

14. Velocity estimation

This chapter completes the explanation of the *Mobile Millennium* Highway Model. Herein, an estimator is built to reconstruct the evolution of the velocity field on the highway, given velocity measurements from GPS devices such as mobile phones. Data obtained from the one-day field experiment, *Mobile Century*, is revisited, and assimilated with the refined *Mobile Millennium* Highway Model. The key points in this chapter are as follows.

- **State space formulation**. We pose the velocity estimation problem in state space form. The resulting system has a nonlinear and nondifferentiable evolution equation, but contains a linear observation equation. This is an improvement over a nonlinear nondifferentiable evolution equation with a nonlinear observation equation when the discretized LWR PDE is used directly as an evolution equation.
- **Solution with ensemble Kalman filtering**. The resulting state estimation problem is solved using ensemble Kalman filtering, representing the first application of the technique for traffic monitoring.
- **Assessment on experimental data**. We assess the performance of the estimation algorithm on experimental data collected from the *Mobile Century* experiment. A prototype version of the velocity estimation algorithm ran live during the experiment, broadcasting results in real–time to monitors of the experiment.

The chapter is organized as follows. In Section 1 we pose the estimation problem in state space form; we describe the ensemble Kalman filtering technique and we compare it to extended Kalman filtering. We describe a mechanism for sampling GPS data from mobile phones in a privacy aware environment using virtual trip lines in Section 1, and describe a 100 vehicle field experiment known as *Mobile Century*. Finally, we conclude the chapter with experimental velocity estimation results from the *Mobile Century* experiment in Section 2.

14.1. Development of a recursive velocity estimation algorithm

14.1.1. State–space model

Before we begin with the estimation problem using the CTM–v model derived in Chapter 13, let

us consider a traffic estimation problem in a more general form, in order to characterize the errors in the CTM–v model. We start by introducing a true state vector $\tilde{z}^n$, with dimension two times the number of vehicles in the transportation network at time n. In this vector, half of the elements correspond to the true positions of the individual vehicles, while the other half of the elements correspond to the velocities of the vehicles. This vector evolves according to

$$\tilde{z}^{n+1} = \widetilde{\mathcal{M}}\left(\tilde{z}^n, \tilde{\theta}^n, \tilde{\eta}^n\right) \tag{68}$$

where $\widetilde{\mathcal{M}}(\cdot,\cdot,\cdot)$ represents a function which maps the true previous position and velocity of all vehicles $\tilde{z}^n$ to their true position and velocity at the next timestep $\tilde{z}^{n+1}$, with the help of parameters such as driver behavior, vehicle performance characteristics, road geometry, etc. represented by $\tilde{\theta}^n$, and some stochastic input given by $\tilde{\eta}^n$. If the model of the true evolution $\widetilde{\mathcal{M}}(\cdot,\cdot,\cdot)$, the parameters $\tilde{\theta}^n$, and the stochastic input $\tilde{\eta}^n$ were known, we could completely characterize the traffic evolution throughout the network for all time by evolving (68) forward. Unfortunately, each of the preceding components are unknown in practice. Thus, $\widetilde{\mathcal{M}}(\cdot,\cdot,\cdot)$ is an abstraction which represents the true, error-free model of traffic evolution, but which we cannot instantiate due to its unknown form and inputs.

The true state vector representing all vehicle positions and velocities is related to the average traffic velocity vector in each discrete segment on the network v^n, which is the vector we are interested in estimating, by:

$$v^{n+1} = P(\tilde{z}^{n+1})$$

where $P(\cdot)$ is an averaging operator which computes the average traffic velocity in each discrete segment from the individual vehicles' velocities in each discrete segment. Note then that the average velocity can be computed according to

$$v^{n+1} = P\left(\widetilde{\mathcal{M}}\left(\tilde{z}^n, \tilde{\theta}^n, \tilde{\eta}^n\right)\right)$$

Unfortunately, since $\widetilde{\mathcal{M}}(\cdot,\cdot,\cdot)$ and its inputs are unknown, we need another evolution equation which is known to approximate this model. We will use the network velocity evolution algorithm $\mathcal{M}(\cdot,\cdot)$, given in Section 2. This algorithm consists of the following steps. For each vertex in the network, a linear program is solved such that strong boundary conditions are imposed on the incoming and outgoing edges of the junction. Next, the velocity field is updated according to the numerical scheme outlined earlier (which is nonlinear and non-differentiable). Then our approximate model is derived as follows:

$$
\begin{aligned}
v^{n+1} &= P\left(\widetilde{\mathcal{M}}\left(\tilde{z}^n, \tilde{\theta}^n, \tilde{\eta}^n\right)\right) \\
&= \left[P\left(\widetilde{\mathcal{M}}\left(\tilde{z}^n, \tilde{\theta}^n, \tilde{\eta}^n\right)\right) - \mathcal{M}(P(\tilde{z}^n), \theta^n)\right] + \mathcal{M}(P(\tilde{z}^n), \theta^n) \\
&= \mathcal{M}(P(\tilde{z}^n), \theta^n) + \hat{\eta}^n \\
&= \mathcal{M}(v^n, \theta^n) + \hat{\eta}^n
\end{aligned}
\tag{69}
$$

The term θ^n represents the model parameters, and $\hat{\eta}^n$ represents the error due to the use of the approximate model in place of the unknown true model. Interestingly, $\mathcal{M}(\cdot,\cdot)$ is derived from a conservation law, and the principle of conservation of vehicles has no error. Instead, $\hat{\eta}^n$ appears through (i) the fundamental assumption of the LWR PDE, which is that velocity can be described as a function of density only, (ii) the nonuniqueness of solutions to conservation laws, and the choice of an entropy condition to isolate a unique solution, and (iii) the nonuniqueness of generalized Riemann solvers at junctions. It is not caused by uncertain boundary data or model parameters, which we treat next.

In practice, boundary data and model parameters θ^n are uncertain. If we choose to replace the true θ^n with an estimate $\bar{\theta}^n$, then (69) is modified by:

$$
\begin{aligned}
v^{n+1} &= \mathcal{M}(v^n, \theta^n) + \hat{\eta}^n \\
&= [\mathcal{M}(v^n, \theta^n) + \hat{\eta}^n - \mathcal{M}(v^n, \bar{\theta}^n)] + \mathcal{M}(v^n, \bar{\theta}^n) \\
&= \mathcal{M}(v^n, \bar{\theta}^n) + \eta^n
\end{aligned}
\tag{70}
$$

Now the term η^n in (70) represents both the errors in the approximate model, and the errors caused by the incorrect model parameters $\bar{\theta}$. Unfortunately the preceding derivation of (70) shows that the true model $\mathcal{M}(\cdot,\cdot,\cdot)$ is needed to compute the statistics of these errors. This problem can be addressed in part through the aid of a more accurate but computationally more intensive model, see for example [206,207], or by estimating the errors, for example by $\eta^n \sim (0, \boldsymbol{Q}^n)$, a zero–mean, white state noise with covariance $\boldsymbol{Q}^n$. The latter is the approach we will use for estimation of velocity fields from mobile phone data. The zero mean assumption and white noise assumption are introduced to simplify the presentation of various Kalman filtering algorithms, and can be relaxed with suitable adjustments made to the Kalman filtering algorithm and state space formulation [51,143,300].

We address the process by which velocity measurements are obtained from GPS equipped vehicles similarly. Let $\widetilde{\boldsymbol{H}}^n$ be a linear operator which maps the velocity of vehicles which send measurements to the GPS velocity value measured by the vehicle at time n, and let y^n denote the GPS measurements which are received at time n. The network observation model is given by

$$y^n = \widetilde{\boldsymbol{H}}^n \tilde{z}^n + \tilde{\chi}^n \tag{71}$$

where $\tilde{\chi}^n$ is the measurement error of the GPS device in each vehicle.

Several comments can be made about (71). First, note that $\widetilde{\boldsymbol{H}}^n$ is in fact a linear operator, since it simply indicates the subset of vehicles from which measurements are obtained. Second, in order for $\widetilde{\boldsymbol{H}}^{\boldsymbol{n}}$ to be defined, vehicles must report a unique ID along with the velocity measurement, so that each measurement may be correctly mapped to the correct vehicle. If the identifiers are withheld from the measurements, for example for privacy reasons, then $\widetilde{\boldsymbol{H}}^n$ is unknown and a data association problem must be solved to determine $\widetilde{\boldsymbol{H}}^n$. The use of GPS position information may help solve the data association problem but does not replace a known $\widetilde{\boldsymbol{H}}^n$. Finally, note that the GPS error $\tilde{\chi}^n$ may be correlated in time and across vehicles [244].

Because we are working with an aggregate state v^n, we derive an equivalent form of (71) as follows:

$$\begin{aligned} y^n &= \left[\widetilde{\boldsymbol{H}}^n \tilde{z}^n + \tilde{\chi}^n - \boldsymbol{H}^n P(\tilde{z}^n)\right] + \boldsymbol{H}^n P(\tilde{z}^n) \\ &= \boldsymbol{H}^n P(\tilde{z}^n) + \chi^n \\ &= \boldsymbol{H} v^n + \chi^n \end{aligned} \tag{72}$$

The linear observation matrix $\boldsymbol{H}^n \in \{0,1\}^{p^n \times \kappa}$ encodes the p^n discrete cells on the highway for which the velocity is observed during discrete time step n and $\kappa = \sum_{e \in \mathcal{E}} (i_{\max,e} + 1)$ is the corresponding (total) number of cells in the network. The term χ^n now includes both the GPS error and the sampling error introduced when the sample vehicle's velocity is different from the average velocity of all vehicles in the discrete road segment from which the measurement is obtained. Note also that determination of the error statistics χ^n requires knowledge of the true state $\tilde{z}^n$, although approximate statistics could be computed through the use of microscopic simulation models and enhanced error modeling techniques [206,207]. In this work, we approximate the statistics with a white, zero-mean observation noise $\chi^n \sim (0, \boldsymbol{R}^n)$, although again these assumptions can also be relaxed [51].

Interestingly, the term $\boldsymbol{H}^n$ itself may contain error, even in the macroscopic setting. This is because the GPS position of the vehicle is used to determine the location of the measurement, and therefore it determines the corresponding state vector associated to the measurement (which is stored in $\boldsymbol{H}^n$). The amount of error in $\boldsymbol{H}^n$ is determined by the amount of error in the GPS position, the size of the discretized road segments, and the proximity of the measurement to be discretized road segment boundaries. For example, if a measurement is received exactly at the boundary between to discrete road segments, then it is not clear to which element of the state vector the measurement should be mapped. This difficulty is circumvented in our application through the use of a spatial sampling technique known as a virtual trip line [190], which is a virtual marker in the form of a line segment encoded by two latitudes longitude coordinates, which triggers a measurement from a mobile phone when it is crossed. By careful placement of the virtual trip lines, and by employing virtual trip line specific filters, the amount of error in $\boldsymbol{H}^n$

can be made negligible, at the cost of perhaps fewer measurements. We describe virtual trip lines in more detail in Section 1.

We briefly describe an alternate formulation for estimating velocities, using the discretized LWR PDE directly, to compare against (70) and (72). Using a similar approach to the velocity derivation, the density ρ^n evolves according to

$$\rho^{n+1} = \mathcal{M}_\rho\left(\rho^n, \bar{\theta}_\rho^n\right) + \eta_\rho^n \tag{73}$$

$$y^n = \boldsymbol{H}^n V(\rho^n) + \chi_\rho^n \tag{74}$$

where $\mathcal{M}_\rho(\cdot,\cdot)$ is the discretized LWR PDE, $\bar{\theta}_\rho^n$ are the model parameters and boundary data, and η_ρ^n is the error in the conservation law associated with the approximation of velocity as a function of density $v = V(\rho)$ only, the nonuniqueness of entropy solutions to the LWR PDE, and the nonuniqueness of the Riemann problems at junctions. The term y^n is again the vector of GPS velocity measurements, $\boldsymbol{H}^n$ is the same linear operator which mapsthe measurements to the corresponding elements in the state vector, and χ_ρ^n is the error associated with the GPS error, the sampling error, and errors in the velocity function approximation $v = V(\rho)$.

When comparing the velocity formulation (70) and (72) to the density formulation (73) and (74), several observations can be made.

- Both the evolution equation for velocity (70) and the evolution equation for density (73) are in general nonlinear and nondifferentiable, due to the min operator in the Godunov discretization schemes (32) and (49), and in particular the existence of standing shockwaves as solutions to the LWR PDE (see Section 2 for the proof that the model is not differentiable the around this state). This comes in addition to the potential nondifferentiability of the flux function itself, for example like in the Newell–Daganzo flux function. Moreover, the generalized Riemann solver at junctions takes the form of an optimization problem (often a linear program), which also is not differentiable in general.
- Both models make the same fundamental assumption that the velocity can be represented as a function of density only. The velocity evolution equation places a further restriction on the velocity function, requiring that the velocity function the invertible. Note, however, when the velocity function is not invertible, the observation equation (74) makes estimating the density from velocity measurements more ill posed.
- By construction, both models use the same entropy solution and the same Riemann solver at junctions.
- While the observation model for the velocity state (72) is linear, the observation model for the density state (74) is linear only when the velocity function is linear (i.e. Greenshields). For nonlinear velocity functions, the observation equation would have to be linearized to fit a standard Kalman filtering framework.
- The observation model for the density state (74) relies on the velocity function $V(\cdot)$, as does the density evolution equation (73). Thus, in this formulation the model error η_ρ^n in the observation error χ_ρ^n are correlated. This is not the case in the velocity formulation, where the errors η^n and χ^n are independent.

In the remainder of this chapter, we elect to use the velocity evolution equation (70) and observation equation (72) due to the linearity of (72). As a possible extension of this work, it would be interesting to compare how the different formulations perform in practice.

14.1.2. Extended Kalman filtering for nonlinear systems

If equation (50) was differentiable in v^n, so would be the operator $\mathcal{M}(\cdot,\cdot)$ in (70), in which case the estimate for the state v^n could be obtained using the following traditional extended Kalman filtering equations:

- Forecast step (Time-update):

$$v_f^n = \mathcal{M}(v_a^{n-1}, \bar{\theta}^{n-1})$$

$$\boldsymbol{P}_f^n = \mathcal{M}_L^{n-1} \boldsymbol{P}_a^{n-1} (\mathcal{M}_L^{n-1})^T + \boldsymbol{Q}^{n-1} \tag{75}$$

where v_f^n (resp. v_a^n) is the forecast (analyzed) state estimate at time n, and $\mathcal{M}_L$ is the Jacobian matrix of mapping $\mathcal{M}$ (also known as the *tangent linear model*) defined as

$$\mathcal{M}_L^n = \frac{\partial \mathcal{M}(v_a^n, \bar{\theta}^n)}{\partial v_a^n} \tag{76}$$

- Analysis step (Measurement-update):

$$v_a^n = v_f^n + \boldsymbol{G}^n \left(y^n - \boldsymbol{H}^n v_f^n \right) \tag{77}$$

$$\boldsymbol{P}_a^n = \boldsymbol{P}_f^n - \boldsymbol{G}^n \boldsymbol{H}^n \boldsymbol{P}_f^n \tag{78}$$

$$\boldsymbol{G}^n = \boldsymbol{P}_f^n (\boldsymbol{H}^n)^T \left(\boldsymbol{H}^n \boldsymbol{P}_f^n (\boldsymbol{H}^n)^T + \boldsymbol{R}^n \right)^{-1} \tag{79}$$

where $\boldsymbol{P}_f^n$ (resp. $\boldsymbol{P}_a^n$) is the error covariance of the forecast (analyzed) state at time n.

The initial conditions for the recursion are given by $v_a^0 = v^0$ and $\boldsymbol{P}_a^0 = \boldsymbol{P}^0$.

14.1.3. Ensemble Kalman filter

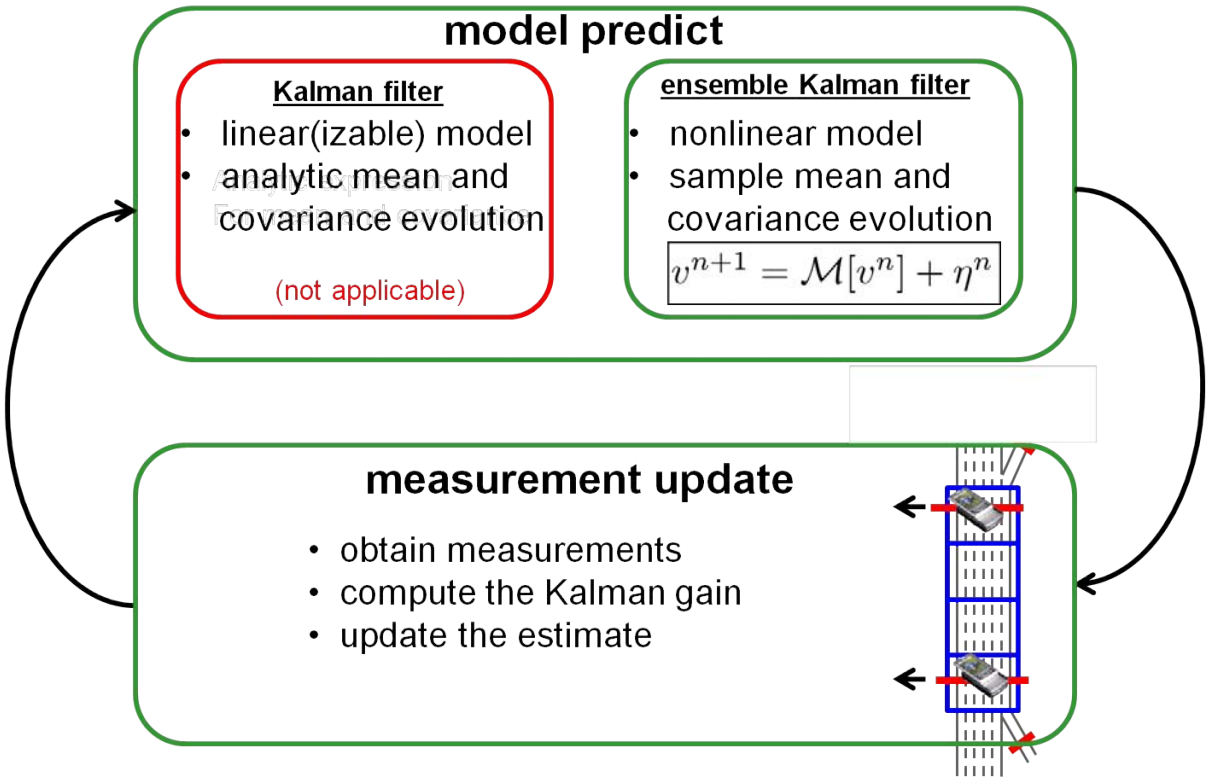


Figure 1.1. Illustration of the difference between extended Kalman filtering and ensemble Kalman filtering, and the iterative process predict–update.

The ensemble Kalman filter was introduced by Evensen in [143] as an alternative to EKF to overcome specific difficulties with nonlinear state evolution models, including non-differentiability of the model and closure problems. Closure problems refer to the fact that in EKF, it is assumed that discarding the higher order moments from the evolution of the error covariance in (75) yields a good approximation. In cases in which this linearization approximation is invalid, it can cause an unbounded error variance growth [143]. To tackle this issue EnKF uses Monte Carlo (or ensemble integrations). By propagating the ensemble of model states forward in time, it is possible to calculate the mean and the covariances of the error needed at the analysis (measurement-update) step [83] and avoid the closure problem. Furthermore, a strength of EnKF is that it uses the standard update equations of EKF, except that the gain is computed from the error covariances provided by the ensemble of model states. Figure 1.1 illustrates the difference.

EnKF also comes with a relatively low computational cost. Namely, usually a rather limited number of ensemble members is needed to achieve a reasonable statistical convergence [83].

In traditional Kalman filtering, the error covariance matrices are defined in terms of the true state as $\boldsymbol{P}_f = E[(v_f - v_t)(v_f - v_t)^T]$ and $\boldsymbol{P}_a = E[(v_a - v_t)(v_a - v_t)^T]$ where $E[\cdot]$ denotes the average over the ensemble, v is the model state vector at particular time, and the subscripts f, a, and t represent the forecast, analyzed, and true state, respectively. Because the true state is

not known, ensemble covariances for EnKF have to be considered. These covariance matrices are evaluated around the ensemble mean $\bar{v}$, yielding $\boldsymbol{P}_f \approx \boldsymbol{P}_{\text{ens},f} = E[(v_f - \bar{v}_f)(v_f - \bar{v}_f))^T]$ and $\boldsymbol{P}_a \approx \boldsymbol{P}_{\text{ens},a} = E[(v_a - \bar{v}_a)(v_a - \bar{v}_a)^T]$ where the subscript ens refers to the ensemble approximation. In [83], it is shown that if the ensemble mean is used as the best estimate, the ensemble covariance can consistently be interpreted as the error covariance of the best estimate. For complete details of derivation of the EnKF algorithm, the reader is referred to [143].

The ensemble Kalman filter algorithm can be summarized as follows [83,143]:

1. *Initialization*: Draw K ensemble realizations $v_a^0(k)$ (with $k \in \{1, \cdots, K\}$) from a process with a mean speed $\bar{v}_a^0$ and covariance $\boldsymbol{P}_a^0$.
2. *Forecast*: Update each of the K ensemble members according to the CTM-v forward simulation algorithm in Section 2. Then update the ensemble mean and covariance according to:

$$v_f^n(k) = \mathcal{M}(v_a^{n-1}(k), \bar{\theta}^{n-1}) + \eta^n(k) \tag{80}$$

$$\bar{v}_f^n = \frac{1}{K}\sum_{k=1}^{K} v_f^n(k) \tag{81}$$

$$\boldsymbol{P}_{\text{ens},f}^n = \frac{1}{K-1}\sum_{k=1}^{K}\left(v_f^n(k) - \bar{v}_f^n\right)\left(v_f^n(k) - \bar{v}_f^n\right)^T \tag{82}$$

3. *Analysis: Obtain measurements, compute the Kalman gain, and update the network forecast:*

$$\boldsymbol{G}_{\text{ens}}^n = \boldsymbol{P}_{\text{ens},f}^n(\boldsymbol{H}^n)^T\left(\boldsymbol{H}^n\boldsymbol{P}_{\text{ens},f}^n(\boldsymbol{H}^n)^T + \boldsymbol{R}^n\right)^{-1} \tag{83}$$

$$v_a^n(k) = v_f^n(k) + \boldsymbol{G}_{\text{ens}}^n\left(y^n(k) - \boldsymbol{H}^n v_f^n(k) + \chi^n(k)\right) \tag{84}$$

4. Return to 2.

In (84), an important step is that at measurement times, the measurement vector y^n is represented by an ensemble indexed by k. This ensemble has the actual measurement as the mean and the variance of the ensemble is used to represent the measurement errors. This is done by adding perturbations $\chi^n(k)$ to the measurements drawn from a distribution with zero mean and covariance equal to the measurement error covariance matrix $\boldsymbol{R}^n$. This ensures that the updated ensemble has the correct analyzed covariance [83].

14.1.4. Large scale real–time implementation

The ensemble Kalman filter algorithm presented in the previous section is in a framework in which all of the unknown state variables on each edge in the network are updated simultaneously. This introduces the following problems. First, because the state covariance is represented through a limited number of ensemble members, non-physical correlations may arise. This means that the correlation matrix may incorrectly show correlation between distant

parts of the highway network which do not correlate in practice. Secondly, the framework described previously requires the forecast error covariance in (82) to be computed for the entire highway network, for use in computing the Kalman gain in (83). When operating on large scale networks such as the San Francisco Bay Area, CA, the loading the covariance matrix into memory can easily require more than 2 GB of space, creating computational limitations for implementation.

To circumvent the above mentioned problems for practical implementations, we employ a *covariance localization method*. This approach limits the correlation between the velocity states on all edges in the network. For a given edge e, only nearby links (upstream and downstream in the network) can exhibit correlation, thereby removing correlation across distant parts of the network. These techniques have also been implemented for oceanography data assimilation problems (see e.g. [245]).

For the large scale traffic network estimation problem, localization also provides a computationally efficient way to update the state variables at the measurement update time in (83)–(84). Namely, due to the localization, the computation of the covariance matrix in (82) is transformed into a computation of numerous small localized covariance matrices for each edge in the network. These small scale covariance matrices are computed for each edge given its neighboring edges on which the correlation is assumed to be physically meaningful. Finally, this allows for the distributed solving of the update equations.

For the localization, we introduce a localization operator $\mathcal{L}_e$ for each edge e, which is constructed at the initialization stage. This operator indicates which velocity states on the other edges of the network are allowed to have correlation with the velocity state on the eth edge. The implementation of the EnKF algorithm described previously can be modified for localization by replacing the measurement update equations (82)-(84) with the following sub-algorithm:

For each edge $e \in \mathcal{E}$:

1. Using the localization operator $\mathcal{L}_e$, compute the localized forecast error covariance:

$$\boldsymbol{P}^n_{\text{ens},f,e} = \frac{1}{K-1}\sum_{k=1}^{K} \mathcal{L}_e\left(v_f^n(k) - \bar{v}_f^n\right) \times$$

$$\left(\mathcal{L}_e(v_f^n(k) - \bar{v}_f^n)\right)^T \tag{85}$$

2. Analysis: Obtain measurements $\mathrm{y}^{\mathrm{n}}_{\mathrm{meas,e}}$ from edges that are indicated in $\mathcal{L}_{\mathrm{e}}$, compute the Kalman gain, and update the the local forecast:

$$\boldsymbol{G}^n_{\text{ens},e} = \boldsymbol{P}^n_{\text{ens},f,e}(\boldsymbol{H}^n_e)^T \times$$

$$\left(\boldsymbol{H}^n_e \boldsymbol{P}^n_{\text{ens},f,e}(\boldsymbol{H}^n_e)^T + \boldsymbol{R}^n_e\right)^{-1} \tag{86}$$

$$v^n_{a,e}(k) = \mathcal{L}_e\left(v^n_f(k)\right) +$$

$$\boldsymbol{G}^n_{\text{ens},e}\left(y^n_{\text{meas},e} - \boldsymbol{H}^n_e v^n_f(k) + \chi^n_e(k)\right) \tag{87}$$

3. Return to 1.

It is worth noting that in practice, the operator $\mathcal{L}_e$ does not need to be constructed as a matrix in the computer memory and subsequently be used to do the relatively demanding matrix multiplications. In other words, the eth edge has references to the forecasts and measurements of its neighboring edges needed to construct the localized forecast error covariance matrix.

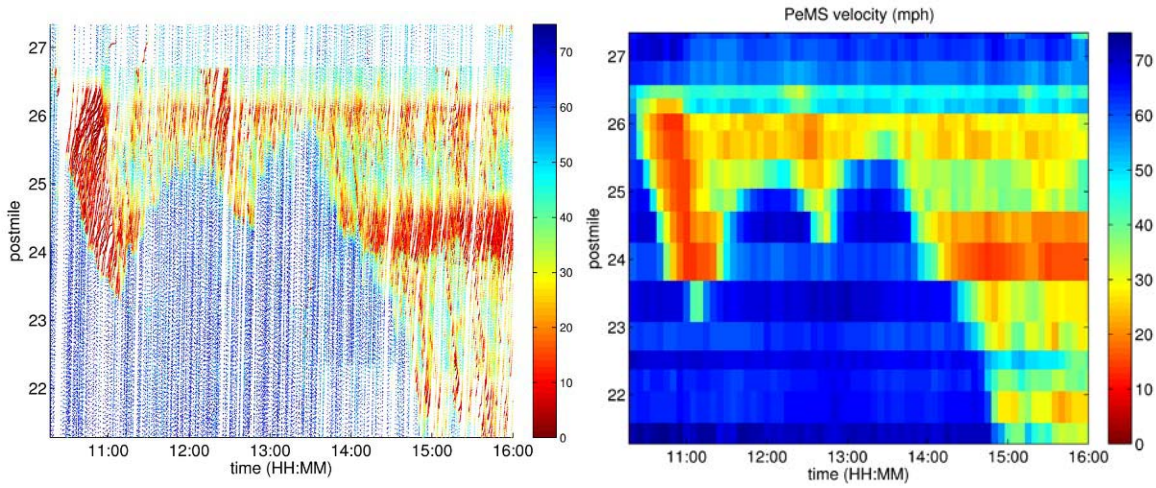
A second performance optimization is achieved by avoiding construction of the covariance matrices directly. When the number of ensemble members is small with respect to the total state space, the covariance matrix $\boldsymbol{P}^n_{\text{ens},f}$ is low rank, and therefore significant computational savings can be achieved by working with a Cholesky decomposition of the covariance matrix. Algorithmic optimizations to the ensemble Kalman filter are explained in detail in the tutorial article [142], complete with pseudocode. For implementation in the *Mobile Millennium* system at UC Berkeley [14], we follow the implementation optimizations of [142].

14.2. Experimental Results

14.2.1. Experimental setup

A complete description of the experiment and comparison of the virtual trip line (VTL) data and PeMS data can be found in the *Mobile Century* Final Report. The data collected during the experiment is downloadable from the project website [14].

For the purposes of *Mobile Millennium*, we remind the reader that this section is monitored with 17 inductive loop detectors, which are processed by the PeMS system to produce speed estimates every five minutes [22]. To construct a velocity contour (Figure 1), the roadway is discretized into 17 links centered around the detectors.



(a) (b)

Figure 2.1. I-880N experiment data. (a) Vehicle GPS logs stored locally on the phone. (b) PeMS velocity contour plot. Color denotes speed in mph. x-axis: time of day. y-axis: postmile.

During the *Mobile Century* experiment at approximately 10:30 am, a multiple car accident created significant unanticipated congestion for northbound traffic south of CA-92 (see Figure 1). The California Highway Patrol reported an incident located at postmile 26.64 at 10:27 am, lasting 34 minutes [22], although GPS readings in Figure 2.1(a) show slowdowns in the area as early as 10:10 am. An earlier version of the EnKF CTM-v algorithm, running in real-time during the experiment, detected the accident's resulting bottleneck and corresponding shockwave [340], and broadcast the results to the web.

14.2.2. Numerical implementation

The network implemented for the results presented here is a 6.8 mile stretch of I-880N from the Decoto Rd. entrance ramp at postmile 20.9, to the Winton Ave. exit ramp at postmile 27.7. The network model consists of 13 edges and 14 junctions (six exit ramps, seven entrance ramps, and one lane drop), shown in Figure 2.2.

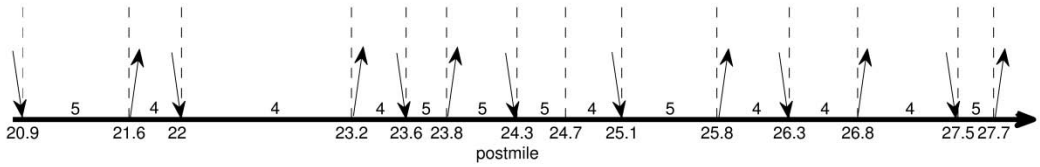


Figure 2.2. Road geometry of I-880N between Decoto Rd. (postmile 20.9) to the south and Winton Ave. (postmile 27.7) to the north. Arrows represent ramp entrance and exit locations, numbers represent the number of lanes on each of the 13 links.

The following link parameters are selected for this experiment: $\rho_{\max} = 200$ vehicles per lane per mile, $v_{max} =$ 70 mph, and w_f= 13 mph. Each link is discretized into equal maximal length cells such that $\Delta x \leq 0.11$ miles and a time step $\Delta t = 5$ seconds is used to ensure numerical stability. The mainline boundary conditions are assumed to be free flowing at 67 mph with standard deviation of 2 mph, and the ramps are set at 30 mph with a standard deviation of 2 mph. The boundary conditions are implemented in the weak sense, and thus are not always imposed on the computational domain. The state noise covariance matrix $\boldsymbol{Q}^n$ is assumed to be diagonal with standard deviation 2 mph, and the measurement error covariance $\boldsymbol{R}^n$ is assumed to be diagonal with standard deviation 4 mph. Parameter estimation and characterization of the error covariance structures is the subject of ongoing work. An initial ensemble with 100 members with mean 67 mph is drawn from $\boldsymbol{P}_a^0$, which is assumed diagonal with standard deviation 4 mph. In one scenario, measurements are collected from ten evenly spaced VTLs, while a second scenario considers measurements collected from 40 evenly spaced VTLs. The estimation algorithm is implemented in Matlab and run on a dual core Intel i5 M540 2.53GHz machine with 4 GB RAM. The estimation algorithm on this experiment site runs just over 14 times faster than real time. For example, a six–hour simulation takes just under 25 minutes.

14.2.3. Comparison with inductive loop detectors

We present a comparison of the velocity estimate from the EnKF CTM-v algorithm using measurements from 10 and 40 VTLs (Figure 3(a)–3(b)) with the velocity estimate obtained from the PeMS system [22]. In order to compare the velocity contours, the EnKF CTM-v estimates are projected onto the coarse discretization induced by the location of the PeMS inductive loop detectors and their corresponding update frequency, then averaged. Because the inductive loops used in the PeMS system are also subject to errors, the resulting velocity contour should not be taken as the ground truth velocity contour.

The results of the EnKF CTM-v with 10 VTLs (Figure 3(c)) and 40 VTLs (Figure 3) show good agreement with the PeMS velocity estimate (Figure 1(b)). Both VTL and PeMS estimates capture important features of the congestion pattern, including the extent of the queue resulting from the accident, which propagates upstream to postmile 23.25 just after 11:00, before it begins to clear (see Figs. 2.3 and 1(b)). The effects of bottlenecks created by capacity decreases at postmiles 25.8 and 24.7 are also well described, and differ by less than 10 mph throughout most of the experiment when 40 VTLs are used (Figure 2.4(b)).

Features of the velocity model are also evident in Figure 3(c)-3(d). In freeflow, information propagates downstream along characteristics, while in congestion information propagates upstream. Also, the discontinuities in the solution joining free flowing upstream sections with congested downstream sections are resolved with high granularity (see in particular the discontinuity caused by the morning accident, Figure 3(c)-3(d)). On the other hand, the PeMS estimates in the same region transition from freeflow speeds in excess of 65 mph to congested speeds around 20 mph over a period of 15 min.

One area where the model appears to underestimate the congestion appears between postmiles 24.7 and 25.1, in Figure 3(c). Both the upstream and downstream sections are five lanes, while the intermediate section has only four lanes. The lane drop at postmile 24.7 acts as a bottleneck, and vehicle speeds increase after entering the four lane link. While speeds increase in both the raw GPS logs (Figure 1(a)) and the PeMS estimates (Figure 1(b)), the resulting velocity estimated from 10 VTLs is approximately 15 mph faster than the PeMS estimate (Figure 2.4(a)). The difference decreases with additional VTLs (Figure 2.4(b)).

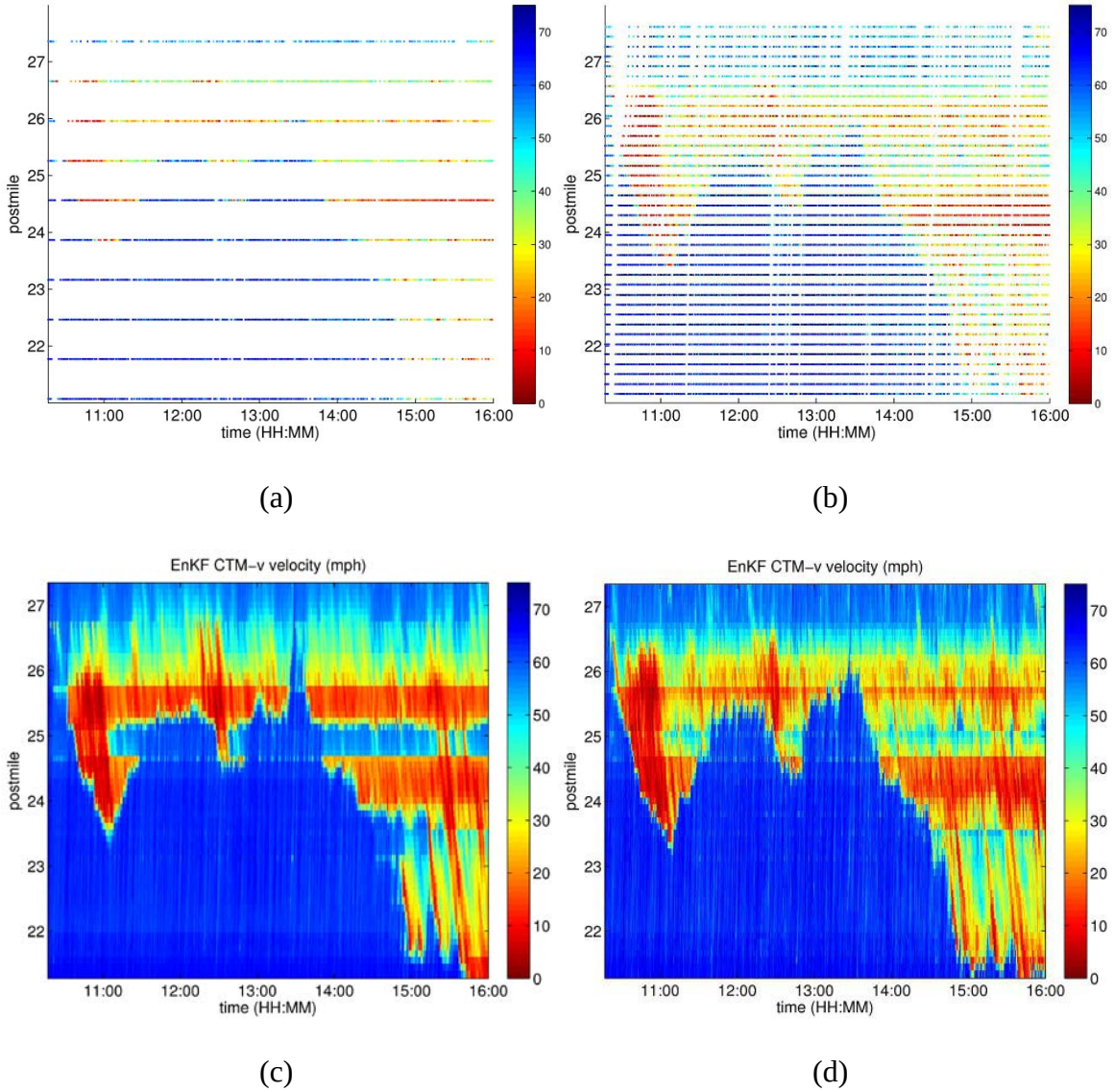
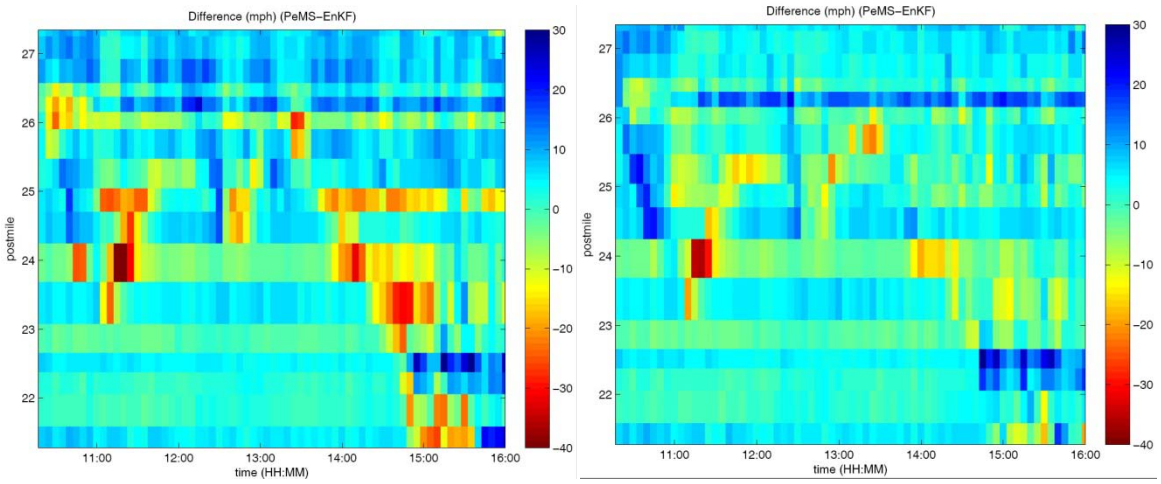


Figure 2.3. VTL measurements with (a) 10 VTLs and (b) 40 VTLs, and EnKF CTM-v velocity contour plots with (c) 10 VTLs and (d) with 40 VTLs. Color denotes speed in mph. x-axis: time of day. y-axis: postmile.The congestion resulting from the morning accident also highlights some of the differences between the EnKF CTM-v estimates created with 10 VTLs and 40 VTLs. Because the model does not predict accidents, measurements are needed to drive the ensemble states into congestion. Because the congestion is recorded on VTLs earlier and more frequently than with the coarser VTL spacing, the ensembles converge to the slower state more quickly. Additionally, because the congested state is slower, the difference in fluxes surrounding the discontinuity is increased, which in turn causes the shockwave speed to increase. Particularly around postmile 25, the decrease in velocity from the shockwave causes the difference between PeMS and EnKF CTM-v velocity measurements to increase with additional VTLs (Figure 2.4(a)–2.4(b)).

At postmile 26.3, the EnKF CTM-v and PeMS estimates differ by almost 20 mph throughout the day (Figure 2.4(a)–2.4(b)). However, there is good agreement on the downstream cell centered at postmile 26.0 which is congested, and the upstream cell centered at postmile 26.5, which is freeflow, so disagreement comes from the transition between the two states. Another area of disagreement occurs in the afternoon rush hour between postmiles 20.9 and 23.6. The EnKF

CTM-v estimates show several distinct shockwaves followed by faster traffic. These features are missed in the average speeds reported by PeMS in the region, which leads to high disagreement in this area.



(a) (b)

Figure 2.4. PeMS and EnKF CTM-v comparison. Color denotes speed difference between PeMS and EnKF CTM-v with (a) 10 VTLs and (b) 40 VTLs, in mph. Color denotes speed in mph. *x*-axis: time of day. *y*-axis: postmile.

Part IV

Mobile Millennium Arterial Models

15. Literature review and background material

The major contribution of the arterial research is the creation of algorithms which fuse together previously disparate disciplines into a single unified framework. Specifically, the goal of the work is to leverage results in traffic flow theory, machine learning theory, sensor networks, and estimation techniques to provide traffic estimation algorithms capable of processing heterogeneous sources of data.

Given the hybrid nature of this work, it is necessary to give some background on several subjects before diving into the contributions of this work. Section 1 introduces the fundamental concepts from machine learning that are used throughout much of the arterial research. It is also necessary to provide an overview of traffic flow theory, which will be presented in chapter 17 as part of the extensions of traffic fundamentals achieved as part of the Mobile Millennium project. The last section of this chapter is a review of the other techniques that have previously been proposed for estimating arterial traffic conditions, presented in section 2. Previous work on arterial traffic estimation has varied greatly in the methodology and the data used. Elements from these previous efforts have provided inspiration for some of the ideas of this work.

15.1. Machine Learning/Probabilistic Graphical Models

This work builds upon algorithms and techniques from machine learning and artificial intelligence. Two introductory books on these subjects are Russell/Norvig [289] and Hastie et al. [178]. The components of machine learning relevant to the problem of estimating arterial traffic are generally categorized as *Probabilistic Graphical Models* [329]. Graphical models provide a logical framework for analyzing complex dependenciesbetween random variables in a system. They can be used in a variety of settings and one can find many examples in [329].

Starting with an overview of graphical models in section 1, the remainder of this chapter then focuses on the key ideas of statistical filtering (section 2) and the expectation maximization algorithm (section 3) used in the estimation algorithms presented in chapters 18 and 19.

15.1.1. Probabilistic Graphical Models

Graphical models can be defined in a very general setting. Consider a graph $G = (V, E)$, where V is a set of nodes and $E \subseteq V \times V$, which can be either directed or undirected. For each vertex $v \in V$ has an associated random variable X_v which can take values in some space $\mathcal{X}_v$ (continuous or discrete). In [329], the authors give an exhaustive introduction to the fundamental properties of a graphical model defined in this general setting. The main concept to note for our purposes is the notion of conditional independence. In an undirected graph setting, variables $X_v, X_u,\ v, u \in V$ are independent given variables $X_A,\ A \subset V$ if there are no paths between v and u that do not intersect some variable $\bar{v} \in A$. A similar definition exists for directed graphs replacing path with directed path. Conditional independence enables the joint distribution of all variables $X_v,\ v \in V$ to be written more compactly. For undirected graphs, it is written

$$p(x_1, x_2, \dots, x_m) = \frac{1}{Z} C \in \mathcal{C} \prod \psi_C\,(x_C), \tag{88}$$

where $\mathcal{C}$ is the set of all (maximal) cliques of the graph, ψ_C is a *compatibility function* defined on each (maximal) clique, x_u is a specific assignment of the random variable X_u for $u \in V$, and Z is a normalization constant. A clique is a complete subgraph, meaning that for the nodes in the clique, every possible edge between them exists. For directed graphs, it is written

$$p(x_1, x_2, \dots, x_m) = v \in V \prod p_v\,(x_v | x_{\pi(v)}), \tag{89}$$

where $\pi(v) \subseteq V$ represents the parents of $v \in V$. In both settings, a sparse graph means that the number and complexity of the probability distributions that need to be defined is small. The only requirement is to specify a function for each clique in the undirected setting. In the directed setting, each vertex needs a probability distribution, but this distribution depends only on the vertex's parents. Computing the full probability distribution $p(x_1, x_2, \dots, x_m)$ for any $(x_1, x_2, \dots, x_m) \in (\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_m)$ can be done efficiently using equations (88) and (89) for the undirected and directed cases, respectively. These concepts are described in full detail in [329].

Given the temporal nature of estimating traffic, a directed graph is the most appropriate for our purposes. This type of graphical model is also known as a *Dynamic Bayesian Network* (DBN). The review of filtering techniques and the expectation maximization algorithm will assume the use of a directed graph in sections 2 and 3.

15.1.2. Statistical Filtering

In the Bayesian approach to dynamic state estimation, one attempts to construct the probability density function of the state at time interval t based on all available measurements up to and including time interval t. This probability density functionis known as *posterior* probability distribution function. The process of estimating the posterior probability distribution function of the state of the network at time interval t is called *filtering*. This filtering process can be used to compute the E step of the EM algorithm presented in section 3, where E stands for expectation and M stands for maximization. Such a filter consists of essentially two stages: *prediction* and

update. The prediction uses the transition probabilities to predict the state probability distribution from one measurement to the next. The update operation uses the latest available measurements to modify the state probability distribution using Bayes theorem.

On small graphs, it is possible to do exact inference (compute the optimal Bayesian estimate) by using the *Junction Tree* algorithm [289]. For graphs that have the structure of a *Hidden Markov Model* (HMM), the probability of a state at time t given observations up to and including time t is computed using alpha recursion (also know as forward algorithm). For more details on inference in HMMs, refer to [279,92]. This algorithm is often presented in the case of unique observation of the hidden state at each time interval. It can be generalized to this model where the number of observations is variable and unknown a priori. HMMs are an important tool built upon in this work for both modeling and parameter estimation (chapters 18 and 19).

When one cannot use the inference techniques for HMMs (because the structure is more complicated) and when the junction tree algorithm is not applicable, an approximation method is needed to perform filtering. One commonly used method is particle filtering (also known as bootstrap filtering or condensation algorithm). For more information on particle filtering, see, for example [289,54]. It is a technique for implementing a recursive Baysesian filter algorithm by Monte Carlo Simulations. The idea is to represent the required posterior density by a set of random samples with associated weights (importance weights) and to compute estimates based on these samples and weights. As the number of samples becomes very large, this Monte Carlo approximation approaches the exact optimal Bayesian estimate. In chapter 19, a specific particle filter for traffic estimation is introduced. There are many issues and potential problems with using particle filters and those will be addressed in that chapter.

Mathematically, the evolution of the state of the system (e.g. the current congestion levels on the road network) is denoted $\{x_0, x_1, \dots, x_n\}$, where x_t is the state of the system at time interval t. It is assumed that x_t is discrete and never observed directly, but rather a series of noisy observations $\{y_0, y_1, \dots, y_n\}$ are observed where $y_t|x_t$ is distributed according to a known function $p_{y|x}(y|x_t)$. A particle filter can be used in the situation where the process to be estimated is modeled as a Markov process with a known transition probability function $p_{x_t,x_{t-1}}(x|x_{t-1})$ (i.e. conditioned on the current state, the next state of the process is independent of the history of the process). Section 3 describes how to use particle filtering when the function is not known with certainty (in the context of the EM algorithm).

The goal of the particle filtering algorithm is to estimate the probability distribution of x_T given the set of observations $\{y_0, y_1, \dots, y_T\}$. The algorithm approximates this distribution using a set of weighted samples of the state (the samples are also called particles). The weights are computed using the observation distribution function, which determines how likely a sample state is. Determining the optimal number of samples to use as representative of the state is dependent upon the sizeof the state space and the dynamics of the process. Experimentation is generally needed to determine a good value for the number of samples to produce.

The particle filtering algorithm is described in algorithm 11, which assumes that the transition and observation probability functions and a total number of samples (denoted N) are given. The end of the algorithm includes a *resampling* stage to avoid the problem of degeneracy (when one

sample has a weight of 1 and all others have weight 0). This algorithm is also known as *Sequential Importance Resampling* (SIR) [169]. The algorithm is presented here for reference and will be referred to later in chapter 19.

Algorithm 1 One iteration of a basic particle filtering algorithm [169].

1. The algorithm for time period t. The samples are kept from the previous time period $x_{t-1}^{(i)}, i = \{1, \dots, N\}$.
2. Generate N random samples of the state $x_t^{(i)}, i = \{1, \dots, N\}$.
3. Generate intermediate weights using $\widehat{w}_t^{(i)} = p(y_t | x_t^{(i)})$.
4. Renormalize the weights: $w_t^{(i)} = \frac{\widehat{w}_t^{(i)}}{\sum_{j=1}^{N} \widehat{w}_t^{(j)}}$.
5. Resample: choose N random samples $\{\hat{x}_t^{(i)}\}_{i=1}^{N}$ from $\{x_t^{(i)}\}_{i=1}^{N}$ with replacement in proportion to the weights $\{w_t^{(i)}\}_{i=1}^{N}$.
6. Replace the sample set with these new samples, i.e., $\{x_t^{(i)}\}_{i=1}^{N} \leftarrow \{\hat{x}_t^{(i)}\}_{i=1}^{N}$, and set the weights to be equal: $w_t^{(i)} = \frac{1}{N}, i = 1, \dots, N$.

15.1.3. Expectation Maximization

In order to use any statistical filtering technique described in section 2, it is necessary to know the value of the parameters of the probability distribution functions used in these techniques. In practical applications, these parameters are frequently unknown and need to be learned from the data. The goal of the *Expectation Maximization* (EM) algorithm is to learn the parameters of a graphical model using incomplete data. An example of incomplete data in the context of traffic is for the traffic *state* to be unobserved (i.e. how congested the road is) while some indirect, noisy measurements are observed (i.e. the travel time of a few vehicles). A graphical model is frequently used in this situation to capture the dynamics of the hidden state as it evolves over time with observation nodes used to represent the noisy measurements that will be collected. The goal of the EM algorithm is to use a set of noisy measurements observed over time along with the model of the process dynamics to estimate the parameters of the transition and observation distribution functions.

The EM algorithm addresses the following paradox: given the (unknown) parameters of the distribution functions, a statistical filtering technique can be used to estimate the probability distribution of the current state of the process; and given the (unknown) the current state of the process, the maximum likelihood estimator of the distribution parameters can be computed. At the outset, neither the true state nor the distribution parameters are known. The EM algorithm deals with this by iterating between these two steps, first fixing the values for distribution parameters and determining the probability distribution for the state (the E-step) and then fixing the state probabilities and determining the distribution parameters (M-step). The EM algorithm is described in detail in [289].

The most important part of the EM algorithm as it pertains to this work is that the E-step can be computed via a particle filter (or any other statistical filter). This fact will be used when applying the EM algorithm to the traffic models presented in chapter 19.

15.2. Previous Arterial Traffic Estimation Techniques

Existing research on arterial traffic estimation varies greatly in the models presented and the data sources assumed available for those models to work. This section categorizes all of these models as either *flow models* (built on traffic flow theory) or *statistical models* (using data-driven knowledge of traffic). To the best of our knowledge, there exists no research that uses a hybrid approach of flow models and statistical models, which is the basis of this work.

Every approach to estimating arterial traffic conditions requires answering two fundamental questions:

- What data is available or assumed available to the model?
- What structure should the model have for processing the data?

Each of these questions is actually very broad and requires answering a series of sub-questions in order to fully characterize the estimation technique. With respect to data, the following points must be addressed:

- **Sensing Infrastructure** Does the model assume that fixed-infrastructure sensors have been placed at specific locations on the roadway? Does the model only use data from probe vehicles (i.e. GPS)? Can it handle both types of data?
- **Frequency** Does the model expect new data at a regular repeating interval (i.e. every 30 seconds)? Can the model handle data streams (i.e. process each new observation when it is received)? For probe data, how often does the probe need to report its position? How much data is transmitted by the probe vehicle?
- **Coverage** In the case of fixed sensors, where do they need to placed to satisfy the demands of the model? For GPS probe data, how much of the road network do the probes need to cover (and how often do they need to cover it)?
- **Latency** How timely must the data be for the model to be accurate? How is the model affected when data arrives 1 minute late, 5 minutes late, half an hour late, etc.?

Coverage and frequency are directly related. One way of characterizing this relationship is to consider the *spatio-temporal* coverage of the data, which means how often and with what spatial resolution is data received. Chapter 3 provides specific details about each of the currently available sensing paradigms and how each type of sensing performs with respect to the list above.

With respect to model structure, the following points must be addressed:

- **Data preprocessing** Does the data need to be changed from its raw form? (i.e. aggregating data within a time window, taking averages, medians, maximums, minimums, etc.)
- **Estimation frequency** How frequently will the estimate be updated? (i.e. every minute, every 5 minutes, every hour, etc.)
- **Estimation quantities** Which state types will the model estimate? Velocity,

density, flow, travel time?

- **Spatial resolution of estimates** Will the model produce an estimate for every link of the network? Will links be aggregated and estimates be produced at the aggregate level? Will the model produce sub-link level estimates?
- **Estimation types** Will the model estimate historic traffic conditions (i.e. a typical Monday at 9am), estimate real-time traffic conditions (i.e. what is happening *right now*), or forecast future traffic conditions (i.e. what will traffic be like in half an hour)?

Previous research efforts have studied how to directly measure delays and travel times through the arterial network through vehicle re-identification (for example [261,286,262,217]). With the direct measurement approach, it is necessary to place sensors everywhere in the network where traffic information is needed. See chapter 3 for an overview of the different sensing technologies available on arterial roads as well as their strengths, weaknesses, and costs. Given that no direct measurement sensors are available for the entire arterial road network (or even close to that coverage), the focus here is on reviewing previous research that infers conditions from incomplete measurements.

In the remainder of this section, previous arterial traffic estimation techniques will be reviewed in the context of how they address each of the points above (for both data and modeling).

15.2.1. Flow Models

The basic idea for all estimation techniques based on flow models is to use the traffic theory principles introduced in chapter 17 as the basis for estimating traffic quantities of interest (flow, density, velocity, travel time). The goal of such techniques is to determine the parameters of the model (fundamental diagram parameters, signal settings, etc.) as well as how to incorporate real-time data into the model. These types of estimation models attempt to relate various traffic variables and assume that measurements of at least one variable are available to estimate the others. For example, one could use flow data to estimate travel times, or one could use travel time measurements to estimate density.

Skabardonis and Geroliminis have written a series of papers on estimating travel times on arterial roads [161,303,304]. Their model shares many similarities with several other previous researchers, such as [157,301,350,344,302,232,345], but the focus here remains on the model of Skabardonis and Geroliminis, which is representative of the issues of these types of modeling approaches. Their model is built upon the traffic flow fundamentals described in chapter 17. The goal of the model is to estimate travel times along a single arterial link, as opposed to estimating travel times through any route in an arterial network. They assume that every traffic signal along the street of interest has the same total cycle length and that the signals are coordinated by a fixed offset from each other. They make the following assumptions about data availability:

- **Sensing Type** Loop detectors are installed along the route and the signal system is capable of providing data about the green and red light times back to the traffic estimation system.
- **Frequency** The loop detectors provide data once per cycle.
- **Coverage** A loop detector is placed upstream of each signalized intersection.

- **Latency** The model needs all current data to estimate the travel time, so the latency of the model is equal to the longest communication delay of all the loop detectors on the route (plus some small model processing time).

In terms of the model, the authors make the following structural decisions:

- **Data preprocessing** No preprocessing is needed. The raw flow and occupancy measurements are used directly and are assumed correct (i.e. no inaccurate measurements enter the system).
- **Estimation frequency** The model estimates the travel time of each arriving vehicle, so the estimation frequency is very high.
- **Estimation quantities** The model estimates travel times on the entire stretch of road.
- **Spatial resolution of estimates** The model only estimates a single travel time value for the road of interest.
- **Estimation types** The model only provides estimates of the travel times that have just occurred. The model does not provide historical traffic conditions and does not perform prediction.

The Skabardonis and Geroliminis model provides a very sound basis for estimating travel times on arterial roads. They build upon the fundamentals of traffic flow theory to be able to incorporate data from one of the most common sensor types available (loop detectors). The primary drawback of the work is the fact that loop detectors and traffic signals are very rarely connected to a communication network that would allow for processing of the data in real-time as well as the fact that this dedicated sensing infrastructure does not have global coverage nor does it have the prospect of good coverage in the near future. The other drawback is that the model focuses on estimating one specific travel time along a road and does not provide a means for calculating arbitrary travel times through the arterial network unless every signal in the network has the same signal cycle length, which is not true in general.

15.2.2. Statistical Models

Many different statistical approaches to arterial traffic estimation have been proposed. There is much variety in terms of the goals of each techniques and the assumptions made to achieve those goals. In general, the techniques reviewed here can be categorized as attempting to apply a standard statistical technique to a particular traffic data type in order to estimate or predict other traffic variables.

Regression is a common statistical tool used frequently in many applications and can be considered one of the simplest forms of machine learning. Given the spatio-temporal nature of traffic conditions, a STARMA model [272] is a logical choice to use for arterial traffic estimation. STARMA is a specific type of regression model that can be used to predict the space-time evolution of a variable (such as link travel time). This type of model requires both space and time to be discrete quantities, so when applying this technique to arterial traffic, one must determine (theoretically or experimentally) the appropriate level of discretization. A common spatial discretization for the arterial network is to look at aggregate link quantities such

as travel time (directly related to average link velocity), average link flow, or average link density. Time discretization is a trickier subject as there is no logical discrete value for how often conditions can change. Researchers often end up choosing values that seem logical, but without a lot of evidence for why the value was chosen. Several papers have presented various STARMA-based approaches to arterial traffic estimation [243,209]. These papers have the following data and modeling characteristics:

Data

- **Sensing Type** Fixed-infrastructure loop detectors.
- **Frequency** The loop detectors provide data at a fixed interval.
- **Coverage** Estimates will be made only at locations where sensors exist, so the coverage is directly equivalent to where the sensors are placed.
- **Latency** The models need a complete set of data for a given time interval in order to compute the next set of predictions, so the latency of the system is equal to the last reporting sensor.

Modeling

- **Data preprocessing** A single aggregate quantity is needed for each time interval. That means using either average flow or average density in the time interval.
- **Estimation frequency** The model produces estimates at the same time interval that data is collected at.
- **Estimation quantities** The model predicts the same quantities that are being collected as input into the model.
- **Spatial resolution of estimates** One estimate is given per sensor location, so estimates are only available where the sensors are located.
- **Estimation types** The model can do short-term prediction and potentially long-term prediction, although accuracy often decreases as a function of the prediction horizon. It is assumed that the data coming in real-time is a perfect description ofthe current state of the road, so no real-time estimation is needed. These models do not provide an estimate of historical traffic conditions.

Previous work has also studied the use of *neural network* [333] and *pattern-matching* [163] algorithms with GPS probe data as input [130]. In this work ([130]), the traffic estimation system makes the critical assumption that the average velocity driven by a few probe vehicles over a link is equal (or nearly equal) to the actual average link velocity for all vehicles. A primary argument of this work is that taking an average of a small number of probe vehicles is insufficient for estimating arterial traffic conditions. After making this critical assumption, the authors then proceed to show how one can use both neural network and pattern-matching algorithms to predict future link velocities. The neural network approach uses two feedforward neural network models per link with one hidden layer. The pattern-matching approach categorizes link velocities

into a small set of discrete categories and then performs prediction by examining the currently available data and seeing which historical pattern the current data is most similar to. The summary of the data and model characteristics is as follows:

Data

- **Sensing Type** GPS probe data from private vehicles.
- **Frequency** The GPS devices send data approximately every 12 minutes when driving on specified roadways.
- **Coverage** A set of roadways is specified and data is collected only on this set. The article assumes that a very large percent of drivers are using a GPS device that will send data. The study was conducted in Italy, where approximately 2% of drivers are using this system. In the United States, no single device (and associated system) has a penetration rate even remotely close to this value.
- **Latency** Given that vehicles report data only every 12 minutes, the travel time values can be 10-15 minutes behind. The model accounts for this by doing prediction, but with a decrease in accuracy due to lack of real-time information.

Modeling

- **Data preprocessing** GPS data is processed into average link velocities per vehicle and then a single average link velocity is computed using an exponential weighting scheme for the individual vehicle values. If data is missing because no vehicles have driven there recently, past values are used. In the case of pattern matching, the velocities were put into discrete categories.
- **Estimation frequency** Estimates are produced every 3 minutes.
- **Estimation quantities** Average link velocities for every link in the studied network.
- **Spatial resolution of estimates** Limited to the pre-specified network where data is collected.
- **Estimation types** The model is capable of both prediction and of providing a historical estimate of traffic conditions. Real-time estimation is assumed perfect by the way they process the GPS probe data.

Belief propagation is one of the standard machine learning tools from the world of probabilistic graphical models (see section 1). The basic idea of the technique is to take observations on a subset of the nodes of a graph and propagate the information contained in those observations to infer the state of all nodes of the graph. In [154], the authors base their arterial traffic model on the Ising model from statistical physics. While the ideas in this type of model originated in the field of statistical physics, the concepts are part of the generalized theory of probabilistic graphical models. In the Ising model, the set of states for each node of the graph are binary. In [154], the authors assume the fundamental states of arterial traffic are undersaturated and

congested. Their model then uses a standard belief propagation algorithm (based on the Ising model) to take "observations" (defined as a measurement between 0 and 1 indicating the level of congestion of a link) on a subset of the links of the road network to then infer the probability of every link of the network being congested.

The key limitation of this model is that it requires pre-processing all of the observations into a single value between 0 and 1 for each link where measurements were recorded. It is not clear that the pre-processing technique described in this article can account for the variability in the measurement coming from a single vehicle. Another limitation is that the algorithm does not always converge and it is unclear under what conditions this situation can arise. Overall, this model shows great potential for being able to estimate traffic conditions on many parts of the network when data is only ever available over a small subset of the network in real-time. The idea of using belief propagation is similar to some of the algorithms described in this work (chapter 19). In summary, the data and model characteristics of this model are as follows:

Data

- **Sensing Type** GPS probe data from private vehicles.
- **Frequency** The GPS devices determine the travel time for the probe vehicle on each link of the network and transmit the link travel time upon completing the traversal of the link. Thus, the frequency of the observations depends on the number of probe vehicles and how often they traverse a link of the network.
- **Coverage** The algorithm is independent of the exact coverage, but the authors state that having data on 10% of the road network is roughly the level that the algorithm needs to perform well. This is the key major benefit of the algorithm, as it does not require data on all links of the network at all times.
- **Latency** The link travel time information sent by the vehicles is only sent after the traversal of the link, so it is necessarily latent by the amount of the travel time. In general, this effect is small and it is reasonable to neglect it, particularly for arterial links that tend to be relatively short (generally no more than several hundred meters).

Modeling

- **Data preprocessing** The link travel times are converted to a single probability on each link where measurements were received. The authors do this by looking at the cumulative distribution function of all travel times ever received on a link and seeing where the received travel time falls. This is a key limitation of the model as it is well-documented that travel times in the undersaturated regime can appear long just

because of the presence of a traffic signal. This issue will be examined in more detail in chapter 16.

- **Estimation frequency** The model can be run as often as desired. The authors do not specifically state how often the traffic estimates are updated.
- **Estimation quantities** Probability of each link of the network being congested. The authors also have a method for determining average travel times from this value (through the historic cumulative distribution function).
- **Spatial resolution of estimates** One estimate per link of the network.
- **Estimation types** The method of collecting data allows the authors to give a historic probability distribution of travel times for each link of the network (without the need of a specific historic model). The model is designed for real-time estimation and it is not clear from the article how one could do prediction with this model.

Similar to the belief propagation algorithm just presented, *Bayesian Networks* can be used to model arterial traffic, as in [267]. In this article, the authors assume that traffic data is discretized into a few categories and that these categories fully represent the traffic conditions. Given the discretization of the data, the message passing (similar to belief propagation) algorithm used is considered standard in the machine learning community. The algorithm works by maximizing the posterior probability of the current traffic state given the data, and the model is able to work with observations on only a subset of the links of the network by propagating the information in the observations to the other links of the network. The main limitation of this work is that the choices for discretization of traffic data are unrealistic and lead to the model just performing classification without interpretation in the context of arterial traffic. The data and model characteristics are as follows:

Data

- **Sensing Type** Dual-loop detector data.
- **Frequency** Data is received every 5 minutes.
- **Coverage** In the study described in the article, there were 5 dual-loop detectors each placed on a single link and there were 6 links total along the route studies.
- **Latency** The model needs the data from each of the loop detectors before computing the next estimate, so the total latency is equal to the last reporting sensor.

Modeling

- **Data preprocessing** The loop detector readings are put into discrete categories by simple interval thresholds.
- **Estimation frequency** One estimate per 5 minutes.
- **Estimation quantities** Average link velocities.
- **Spatial resolution of estimates** One estimate per link (for the 6 links in the study).

- **Estimation types** No historical or prediction model is given. The focus is on a real-time model capable of assessing the current state of traffic.

Another article that assumes high-frequency data is [317]. This article does not present a traffic estimation model, but focuses on simply extracting link travel times from probe vehicles. The primary contribution of the article is that the probe vehicles may have location information based on WiFi (in addition to other vehicles using GPS). The actual estimation of travel times are then just computed by extracting the travel times deduced from the tracking of vehicles through the network. No attempt is made to account for the variability of link travel times, nor to estimate travel times on links where no data has been received. However, the contribution of the article to map-matching and travel time extraction from noisy position measurements is an important practical issue that the authors were able to overcome.

To our knowledge, only one article (outside of the research done by the *Mobile Millennium* team) proposes a model using sparse probe data that may cover many links in between measurements. In [179], the authors propose a *travel time decomposition* approach for determining link travel times when the observations cover several links. The authors do not address the issue of map-matching or path inference, which are two critical pre-processing steps needed for the algorithm to perform well. However, the method proposed in the article for decomposing travel times is a novel one worth mentioning. The authors define likelihood functions for determining how likely it is for a vehicle to get stopped at traffic signal along the path and therefore are able to associate the delay along the path to traffic signals in addition to attempting to identify delay due to congestion. In searching through the literature, this is the first known attempt to use machine learning techniques to determine most likely travel times through the network from sparse probe data. The conclusions of the article ultimately indicate that the likelihood functions proposed are insufficient when the sampling frequency less than once per 90 seconds. For sampling frequencies above ones per 30 seconds, no benefit is gained from the approach as travel times can almost be directly inferred. The maximum benefit seems to occur around the sampling frequency of once every 60 seconds, where travel times cannot be directly inferred. This article served as inspiration for looking at the issue of travel time decomposition, although the approach presented in this work is soundly based on traffic theory principles instead of intuition as in [179].

16. Travel time delay patterns through signalized intersections

Travel times through arterial networks are dependent upon a large number of factors. The primary sources of delay for drivers are the presence of traffic signals and stop signs, as well as the queues that form as a result of intersecting traffic flows. After a brief introduction to the notation and assumptions used (section 1), this chapter presents an idealized model of travel times through a signalized intersection and derives the *delay pattern* that characterizes this theoretical model (section 2).

16.1. Notation and Assumptions

Traffic engineers employ a variety of notations that are considered common in the transportation community. Due to the use of both typical traffic notation and the introduction of new notation for the probabilistic approach, a summary of all notation used is presented in section 1. Also relevant to present in this section are all of the assumptions that are made throughout the development of the models in later chapters (section 2).

16.1.1. Notation

The list below summarizes the notation used in characterizing travel times on arterial links. The parameters are specific for each network link i. The variable t is always used to denote time. Sometimes it is used to refer to a time period (in a discrete time domain) or a time instant (in a continuous time domain). The context will make it clear which use is being employed at any given point.

- **Traffic model parameters**

The traffic model parameters represent the characteristics of the network. They are specific to a link i of the network. For notational simplicity, the subscript i is omitted when the derivations are valid for any link of the network.

ρ_{max}^{i}	Maximum density of link i.
q_{max}^{i}	Capacity (maximum flow) on link i.
ρ_{c}^{i}	Critical density of link i.
w^{i}	$\rho_{c}^{i} v_{f}^{i} / (\rho_{max}^{i} - \rho_{c}^{i})$, Backward shockwave speed of link i.
v_{f}^{i}	Free flow speed of link i.
p_{f}^{i}	Free flow pace (inverse of free flow speed) of link i. Note that $p_{f}^{i} = 1/v_{f}^{i}$.
L^{i}	Length of link i (not a model parameter, but an attribute of the road that is used frequently).

- **Traffic signal parameters**

The traffic signal parameters characterize the properties of the traffic signal at the end of a link i.

C^{i}	Duration of a light cycle on link i.
R^{i}	Duration of the red time on link i.

- **Traffic state variables**

The traffic state variables describe the conditions of traffic that characterize the traffic dynamics on the network. The variables are specific to a link i and a time interval t and represent the dynamic evolution of the traffic state in the different time intervals $t \in \{0 \dots T\}$. The reference to the link or to the time interval may be omitted when the derivations are not link or time specific.

$\rho_{a}^{i,t}$	Arrival density on link i during time interval t.
$v_{a}^{i,t}$	$\rho_{a}^{i,t} v_{f}^{i} / (\rho_{max}^{i} - \rho_{a}^{i,t})$, arrival shockwave speed on link i during time interval t (speed of growth of the queue due to additional vehicles arrival).
$l_{max}^{i,t}$	$R^{i} w^{i} v_{a}^{i,t} / (w^{i} - v_{a}^{i,t})$, length of the triangular queue on link i during time interval t.
$\tau^{i,t}$	$l_{max}^{i,t}(1/w^{i} + 1/v_{f}^{i})$, duration of the clearing time on link i during time interval t, which is the amount of time for the queue to clear in the undersaturated regime (defined for the undersaturated regime only).
ne $l_{r}^{i,t}$	Length of the remaining queue when the light turns red (defined for the congested regime only).

This set of variables is sufficient to characterize the model and the time evolution of the state of

traffic. In particular, through all of the relational formulas, note that only $l_r^{i,t}$ and one of $\rho_a^{i,t}$, $v_a^{i,t}$, $l_{max}^{i,t}$, or $\tau^{i,t}$ are needed to specify the traffic state given that the traffic parameters are fixed. The location x on a link corresponds to the distance from the location to the downstream intersection. From these variables, the other traffic variables can be computed, including velocity v, flow q, and density ρ of vehicles at any x and time t.

The following functions are used for compactness. In general, each function often has a different form depending on whether the undersaturated or the congested regime is being considered. The subscript s is used to denote the regime-specific form of the functions, which can be undersaturated, u, or congested, c.

$d^{s,i}(t)$	The travel time through link i for a vehicle entering the link at time instant t (in the continuous time domain) for regime s.
$\delta^{s,i}(x)$	The delay function for a given location x along link i in regime s.
$g^{s,i}(y_{x_1,x_2})$	The density (in the probability sense) of the travel time y_{x_1,x_2} between locations x_1 and x_2 along link i for regime s.

16.1.2. Traffic Flow Modeling Assumptions

The following assumptions are made on the dynamics of traffic flow:

1. *Triangular fundamental diagram*: this is a standard assumption in transportation engineering.
2. *Stationarity of traffic*: during each estimation interval, the parameters of the light cycles (red and cycle time) and the arrival density ρ_a are constant. Moreover, it is assumed that there is not a consistent increase or decrease in the length of the queue, nor instability. With these assumptions, the traffic dynamics are periodic with period C (length of the light cycle). The work is mainly focused on deriving travel time distributions for cases in which measurements are sparse.Constant quantities (for a limited period of time) do not limit the derivations of the model since the interest here is in trends rather than fluctuations.
3. *First In First Out* (FIFO) model: overtaking on the road network is neglected. When traffic is congested, it is generally difficult or impossible to pass other vehicles. In undersaturated conditions, vehicles can choose their own free flow speed, but it is assumed that the free flow speeds are similar enough that the “no overtaking” assumption is a good approximation.
4. *Model for differences in driving behavior*: the free flow speed is not the same for all vehicles: it is modeled as a random variable with a mean $\bar{v}_f$ and variance σ^2—*e.g.*, a Gaussian or Gamma distribution.

16.2. Travel Time Patterns Through Signalized Intersections

In this section, analytical patterns of intersection delays are developed for the undersaturated and congested regimes. The first regime occurs when queues can be cleared completely during the green phase of a cycle, while the second regime occurs when queues cannot be cleared within one cycle and the remaining queues will have to wait for extra cycles (i.e. more delay) to be cleared. In specific situations (e.g. heavy congestion), queues may spillover to upstream intersections and cause further delays. This third regime is omitted in this analysis and is subject to ongoing research as it requires a network model to determine the effect of neighboring links on the distribution of the link of interest.

Using the assumptions from section 2 and following the standard traffic theory model of vehicles through signalized intersections (presented in chapter 17), the travel time function $d(t)$ is derived. This function represents the amount of time needed for a vehicle entering the link at time t to pass through the intersection. For the remainder of this section, a single link is considered (so the index i is dropped from the notation) and it is assumed that the link parameters are fixed during the time period of the analysis (so the time interval index t as presented in the notation section 2 is dropped). In this section, t refers to a specific time instant in the continuous time domain.

The minimum travel time on the link is equal to the time it takes to traverse the link at the free flow speed, v_f, given the assumption that all vehicles travel at the free flow speed when not stopped. The minimum travel time is called the *free flow travel time* and is equal to $\frac{L}{v_f}$. If the light is red or there is an existing queue when the vehicle approaches the downstream intersection, the vehicle will join the end of the queue and thus be delayed. Otherwise, if a vehicle reaches the intersection when the light is green and there is no queue, the vehicle will pass through the intersection with no delay (thus experiencing the free flow travel time). More importantly, by analyzing the geometry of the triangles in the space-time diagram (figure 1.2), one can easily observe that if a vehicle enters the link at a time that would make it get to the intersection just after the start of the red time (assume no interruption had existed), delay for this vehicle will be the maximum for the specific cycle. After that, delays will be reduced linearly until no delay is reached. The slope, b, of the travel time function can be calculated analytically as (see [64] for the derivation, which can also be seen graphically in figure 1.2):

$$b = -\frac{v_f(w - v_a)}{w(v_f + v_a)} = -1 + \frac{\rho_a}{\rho_c}. \tag{90}$$

Here w is the wave speed, v_f is the free flow speed, v_a is the wave speed when a vehicle joins the queue, ρ_{max} is the jam density, and ρ_a is the arrival density, which is assumed to be constant within a cycle. The three parameters v_f, w, ρ_{max} are specific to actual arterial locations, which also determine the fundamental diagram of the link. Using equation (90), the travel time function for the undersaturated regime is computed as follows:

$$d^u(t) = \frac{L}{v_f} + \max\left\{0, R + b(t + \frac{L}{v_f})\right\}, \text{ for } -\frac{L}{v_f} \leq t < C - \frac{L}{v_f}, \tag{91}$$

where $t = 0$ is defined to be the time at which the light turns red. The function is periodic with a period equal to the signal cycle time, so $d^u(t) = d^u(t + kC), k \in \mathbb{Z}$.

Note that the above analysis and equation (91) only works for the undersaturated regime when the minimum delay reaches 0 in each cycle. In the congested regime, the remaining queue, l_r, must wait for additional cycles to be cleared. In this situation, delay will still be reduced linearly from the maximum value after the start of the red time (when the vehicle arrives at the intersection). However, the delay will never reach zero. Instead, it will have a sudden increase from a nonzero delay to another maximum delay, indicating the vehicle will have to wait for an extra cycle to be cleared. The slope of the curves can all be computed analytically by looking at the geometry of the queue forming and discharging triangles (see figure 1.2). Using the assumption of stationarity from section 2, the slope of the travel time function is still the same as the undersaturated regime (computed in equation (90)). The difference in the congested regime is simply to calculate the delay due to the presence of the remaining queue. First, the maximum number of stops a vehicle will make before exiting the link is calculated as

$$n = \left\lceil \frac{l_r}{l_{max}} \right\rceil. \tag{92}$$

The minimum travel time is then

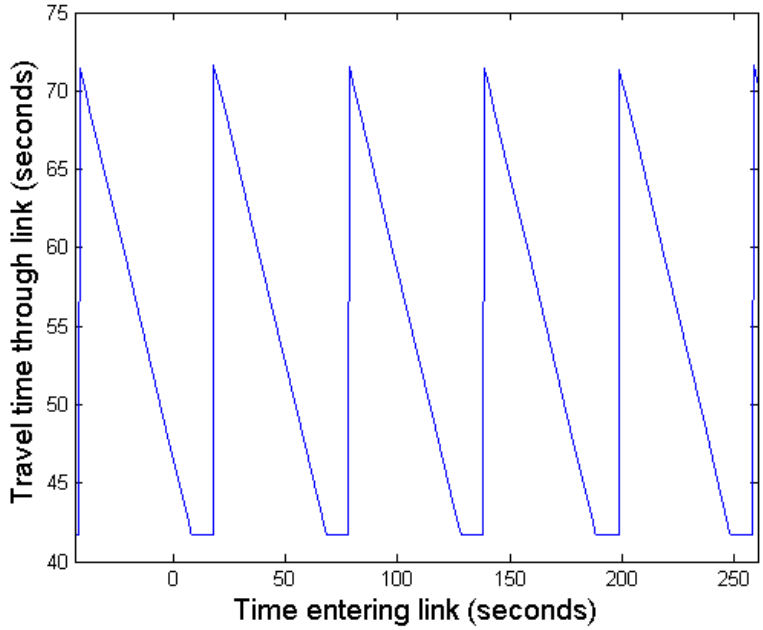
$$tt_{min} = \frac{L}{v_f} + \left((n-1) + \frac{l_r - (n-1)l_{max}}{l_{max}}\right) R. \tag{93}$$

This leads to the expression for the travel time function for the congested regime:

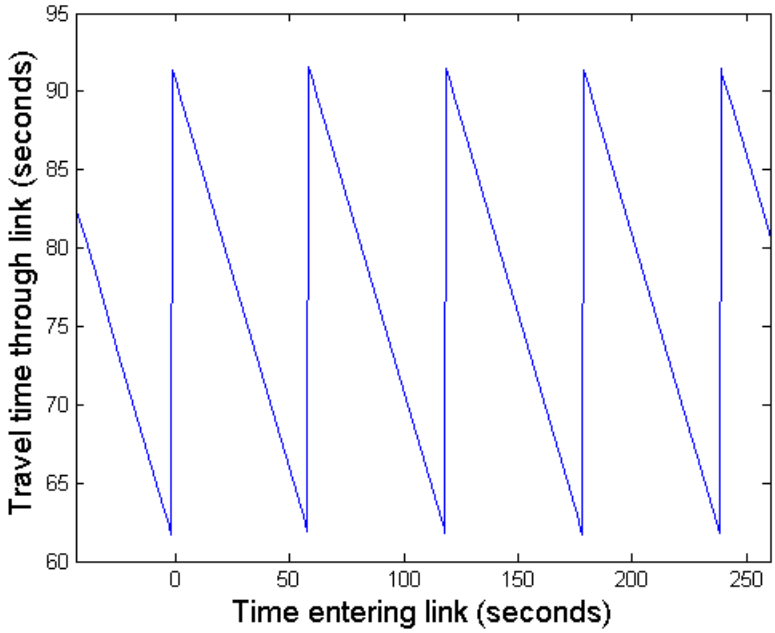
$$d^c(t) = tt_{min} + \max\{0, R + b(t + tt_{min})\}, \text{ for } -tt_{min} \leq t < C - tt_{min}, \tag{94}$$

where $t = 0$ is again the time when the light turns red (for some cycle). Just as in the undersaturated case, this function is periodic with a period equal to the cycle time.

Figure 2.1 depicts the travel time function in the undersaturated and congested regimes for a given set of link parameters. Note that both the undersaturated and congested functions are piecewise linear. The two functions are so similar that it is straightforward to calculate a single travel time function by setting $tt_{min} = \frac{L}{v_f}$ when $l_r = 0$ (the undersaturated regime) and then equation (94) represents both the undersaturated and congested regimes. A complete description of this model with experimental results (using microsimulation data) can be found in [64].



(a)Undersaturated Regime ($d^u(t)$).



(b) Congested Regime ($\boldsymbol{d^c(t)}$)

Figure 2.1. Theoretical link travel time function for the undersaturated (a) and congested regimes (b). The link length was 500 meters and the signal parameters were a cycle time of 60 seconds and a red time of 30 seconds. The free flow travel time was 42 seconds and the length of the remaining queue for the congested case was 100 meters.

17. An extension of traffic fundamentals: A hydrodynamic theory based statistical model of arterial traffic

This chapter presents the most fundamental aspects of the arterial research performed as part of the Mobile Millennium project. The purpose of this part of the work is to introduce these fundamentals and provide the context for which all of the estimation algorithms presented in later chapters is based. All of those algorithms build on top of the results from this chapter.

Starting from the idealized model of travel times in chapter 16, this chapter brings the model into a probabilistic context. The goal is to derive parametric travel time probability distributions for all of the possible states of a link.

Developing these travel time probability distributions is critical to the development of the estimation algorithms to follow in chapter 18. The basic idea is to consider any individual probe measurement as a random sample from the travel time distribution through the links that the vehicle traveled. By understanding the shape of the distribution from which the travel time was generated, it is possible to infer the most likely traffic conditions through which the vehicle traveled. This can then be used to propagate information about the state of one link to its neighboring links, all of which is the subject of chapters 18 and 19.

In arterial networks, the dynamics of traffic flows are driven by the presence of traffic signals. A comprehensive model of the dynamics of arterial traffic flow is necessary to capture the specifics of arterial traffic and provide accurate traffic estimation. From hydrodynamic theory of traffic flow, we model the dynamics of arterial traffic under specific assumptions which are standard in transportation engineering. We use this flow model to develop a statistical model of arterial traffic. First, we derive an analytical expression for the spatial distribution of vehicles. This encompasses the fact that the average density of vehicles is higher close to the traffic signals because of the delays experienced by the vehicles. Second, we derive the probability distribution of *total* and *measured* delay (to be defined specifically in the document). The delays experienced by vehicles traveling on a link of the network depend on the time (from the beginning of the cycle) when they enter the link. We model the probability of delay for a path between two

arbitrary points on the link. The probability distribution of *measured* delay takes into account the sampling scheme to derive the probability of the observed delay from probe vehicles sampled uniformly in time. Finally, we use the probability distribution of delays and a model of driving behavior to derive the probability distribution of travel times between any two arbitrary points on a link. The analytical derivations are parameterized by traffic variables (cycle time, red time, model of free flow speed, queue length and queue length at saturation). The models estimates queue length (and thus congestion levels), signal parameters and variability of driving behavior. We show that the probability distributions of travel times on an arterial links are quasi-concave. The probability distributions of travel times between any arbitrary location on the link are finite mixture distributions where each component represents a class of vehicles depending on the characteristics of its delay. We prove that each component of the mixture distribution is log-concave, which enables the use of specific optimization algorithm. The distributions derived in this report are used as fundamental building block for arterial traffic estimation using sparse travel time measurements from probe vehicles used in subsequent work.

17.1. Modeling arterial traffic

17.1.1. Traffic model

In traffic flow theory, it is common to model vehicular flow as a continuum and represent it with macroscopic variables of *flow* $q(x,t)$ (veh/s), *density* $\rho(x,t)$ (veh/m) and *velocity* $v(x,t)$ (m/s). The definition of flow gives the following relation between these three variables [226,285]:

$$q(x,t) = \rho(x,t)\, v(x,t). \tag{95}$$

We will use this property throughout our derivations of traffic models.

For low values of density, experimental data shows that the velocity of traffic is relatively insensitive to the density; and all vehicles travel close to the so called free flow velocity of the corresponding road segment v_f. As density increases, there is a critical density ρ_c at which the flow of vehicles reaches the capacity q_{max} of the road. As the density of vehicles increases beyond ρ_c, the velocity decreases monotonically until it reaches zero at the maximal density ρ_{max}. The maximal density can be thought of as the maximum number of vehicles that can physically fit per unit length, and at this density, vehicles are unable to move without additional space between vehicles. Experimental data indicates a decreasing linear relationship between flow and density, as the density increases beyond ρ_c. The slope of this line is referred to as the congested wave speed, noted w. Thisleads to the common assumption of a triangular *fundamental diagram* (FD) to model traffic flow dynamics [126].

The triangular FD (illustrated in Figure 1.1) is thus fully characterized by three parameters: v_f, the free flow speed (m/s); ρ_{max}, the jam (or maximum) density (veh/m); and q_{max}, the capacity (veh/m).

We note that ρ_c represents the boundary density value between (i) free flowing conditions for

which cars have the same velocity and do not interact and (ii) saturated conditions for which the density of vehicles forces them to slow down and the flow to decrease. When a queue dissipates, vehicles are released from the queue with the maximum flow—capacity q_{max}—which corresponds to the critical density $\rho_c = q_{max}/v_f$.

For a given road segment of interest, the arrival rate at time t, i.e. the flow of vehicles entering the link at t, is denoted by $q_a(t)$. Conservation of flow relates it to the arrival density $\rho_a(t) = q_a(t)/v_f$.

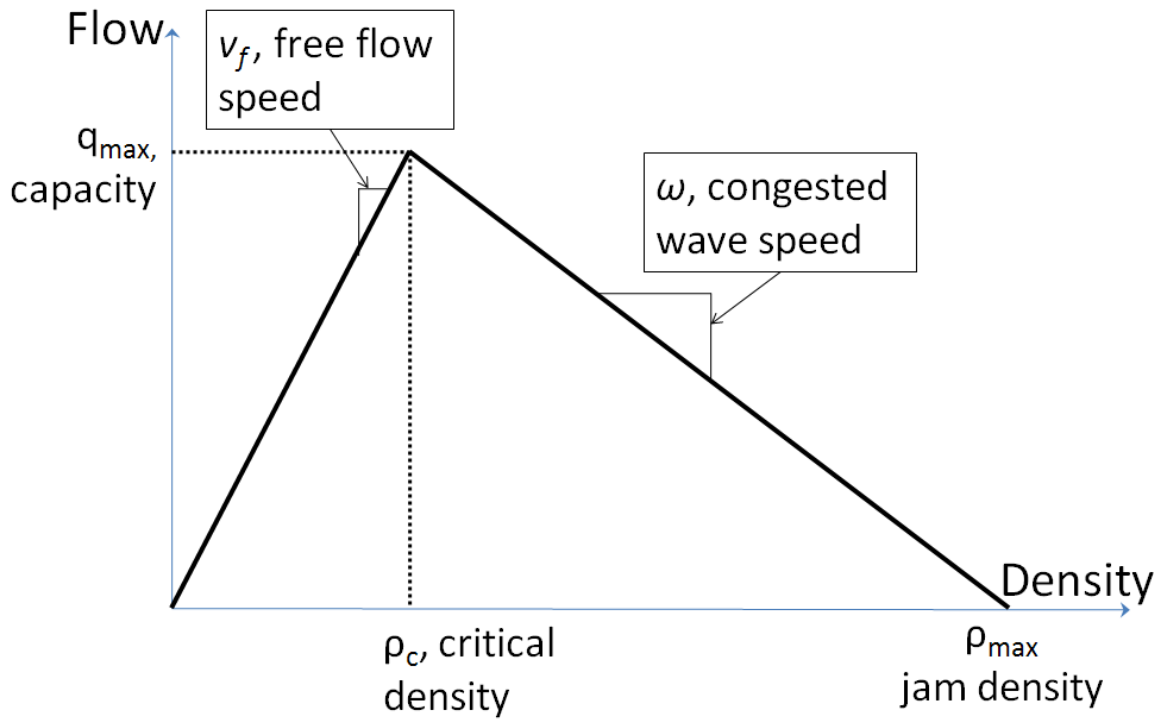


Figure 1.1. The fundamental diagram: empirically constructed relation between flow and density of vehicles.

17.1.2. Traffic flow modeling assumptions

We make the following assumptions on the dynamics of traffic flow and discuss their range of validity:

1. *Triangular fundamental diagram*: this is a standard assumption in transportation engineering.
2. *Stationarity of traffic*: during each estimation interval, the parameters of the light cycles (red and cycle time) and the arrival density ρ_a are constant. Moreover, we assume that there is not a consistent increase or decrease in the length of the queue, nor instability. With these assumptions, the traffic dynamics are periodic with period C (length of the light cycle). The work is mainly focused on deriving travel time distributions for cases in which measurements are sparse. Constant quantities (for a limited period of time) do not

limit the derivations of the model since we are here interested in trends rather than fluctuations.

3. *First In First Out* (FIFO) model: overtaking on the road network is neglected. When traffic is congested, it is generally difficult or impossible to pass other vehicles. In undersaturated conditions, vehicles can choose their own free flow speed, but we assume that the free flow speeds are similar enough that the "no overtaking" assumption is a good approximation.
4. *Model for differences in driving behavior*: the free flow pace (inverse of the free flow speed) is not the same for all vehicles: it is modeled as a random variable with vector of parameter θ_p—*e.g.* the free flow pace has a Gaussian or Gamma distribution with parameter vector $\theta_p = (\bar{p}_f, \sigma_p)^T$ where $\bar{p}_f$ and ma_p are respectively the mean and the standard deviation of the random variable.

17.1.3. Arterial traffic dynamics

In arterial networks, traffic is driven by the formation and the dissipation of queues at intersections. The dynamics of queues are characterized by shocks, which are formed at the interface of traffic flows with different densities.

We define two discrete traffic regimes: *undersaturated* and *congested*, which represent different dynamics of the arterial link depending on the presence (respectively the absence) of a remaining queue when the light switches from green to red. Figure 1.2 illustrates these two regimes under the assumptions made in Section 2. The speed of formation and dissolution of the queue are respectively called v_a and w. Their expression is derived from the Rankine-Hugoniot [139] jump conditions and given by

$$v_a = \frac{\rho_a v_f}{\rho_{max} - \rho_a} \quad \text{and} \quad w = \frac{\rho_c v_f}{\rho_{max} - \rho_c}. \tag{96}$$

<u>Undersaturated regime</u>. In this regime, the queue fully dissipates within the green time. This queue is called the *triangular queue* (from its triangular shape on the space-time diagram of trajectories). It is defined as the spatio-temporal region where vehicles are stopped on the link. Its length is called the maximum queue length, denoted $l_{\max}$, which can also be computed from traffic theory:

$$l_{max} = R\frac{w v_a}{w - v_a} = R\frac{v_f}{\rho_{max}}\frac{\rho_c \rho_a}{\rho_c - \rho_a}. \tag{97}$$

The duration between the time when the light turns green and the time when the queue fully dissipates is the *clearing time* denoted τ. We have

$$\tau = l_{max}\left(\frac{1}{w}+\frac{1}{v_f}\right). \tag{98}$$

Replacing the l_{max} and w by their expressions derived in equations(97) and (96), we have

$$\tau = R\frac{\rho_a}{\rho_c-\rho_a}. \tag{99}$$

Congested regime. In this regime, there exists a part of the queue downstream of the triangular queue called *remaining queue* with length l_r corresponding to vehicles which have to stop multiple times before going through the intersection.

All notations introduced up to here are illustrated for both regimes in Figure 1.2.

Stationarity of the two regimes. Assumption 2 made earlier implies the periodicity of these queue evolutions (see Figure 1.2). As indicated by the slopes of the trajectories in the figure, when vehicles enter the link, they travel at the free flow speed v_f. The distance between two vehicles is the inverse of the arrival density $1/\rho_a$. The time during which vehicles are stopped in the queue is represented by the horizontal line in the queue. The length of this line represents the delay experienced at the corresponding location. The distance between vehicles stopped in the queue is the inverse of the maximum density $1/\rho_{max}$. When the queue dissipates, vehicles are released with a speed v_f and a density ρ_c. The trajectory is represented by a line with slope v_f, the distance between two vehicles is $1/\rho_c$.

We next use these two discrete regimes to derive the pdf for the location of vehicles on a link and for the travel time along a link. The estimation of the distribution of vehicle location uses the individual measurement locations reported by the vehicles. The measurements being sent uniformly in time, vehicles are more likely to report their location where their speed is lower, where they experience delay. Because of the presence of traffic lights, vehicle are more likely to report their location on the downstream part of the link than on the upstream part. Section 2 develops a model for estimating vehicle distribution location. A probabilistic model based on the assumptions formulated in this section provides the pdf of delays (Section 3) and travel times (Section 4) between any two arbitrarylocations on the network.

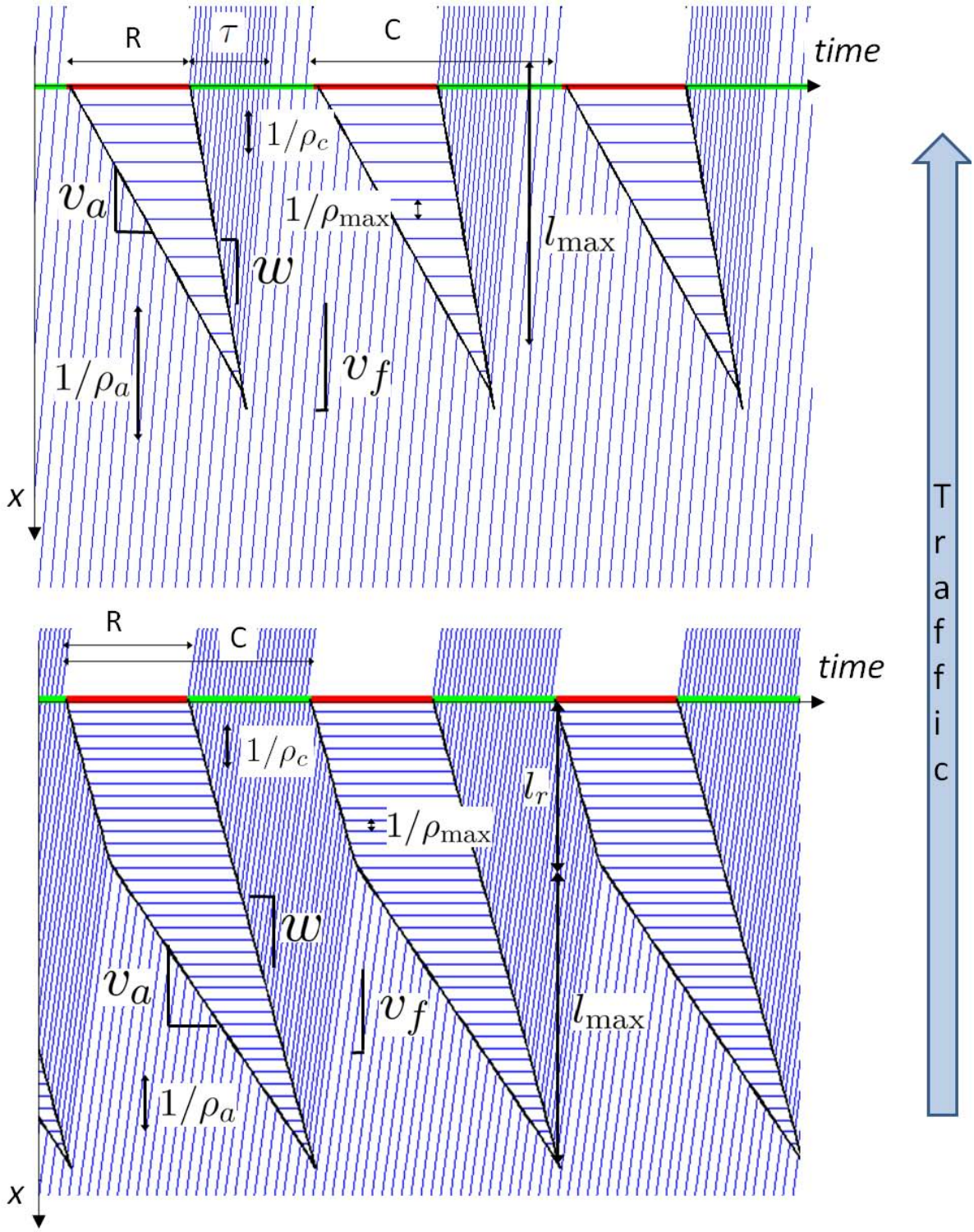


Figure 1.2. Space time diagram of vehicle trajectories with uniform arrivals under an undersaturated traffic regime (top) and a congested traffic regime (bottom).

17.1.4. Notation

The list below summarizes the notation introduced earlier and to be used in the rest of the article. The parameters are specific for each network link j. The index j is omitted for notational simplicity.

- Model parameters:

  - Free flow pace, p_f (seconds/meter), inverse of the free flow velocity v_f. The free flow pace is a random variable. Its *probability distribution function* (p.d.f.) is denoted $\varphi^p(p)$; it models the different driving behavior by assuming a distribution of the free flow pace among the different drivers,
  - Cycle time, C (seconds),
  - Red time, R (seconds),
  - Length of the link, L (meters).
  - Traffic state variables: $\diamond$
  - Clearing time τ,
  - Triangular queue length,
  - Remaining queue length, l_r.

This set of variables is sufficient to characterize the model and the time evolution of the state of traffic. The location x on a link corresponds to the distance from the location to the downstream intersection. From these variables, we can compute the other traffic variables, including velocity, flow, and density of vehicles at any x and time t and queue length. The remaining queue length l_r is specific to the congested regime ($l_r = 0$ in the undersaturated regime). Similarly, the existence of a clearing time is specific to the undersaturated regime (the clearing time is null and thus $\tau = C - R$ in the congested regime).

The undersaturated and congested regimes are labeled u and c respectively. In the following, we derive probability distribution of vehicle locations $f^s(x)$, $s \in \{u, c\}$ based on statistical analysis of queuing theory. We use the variable x to indicate the distance *to* the intersection, so the location of the intersection is at $x = 0$ and the start of the link is at $x = L$. The function f^s encodes the probability of a vehicle to be at location x, which depends on x because of the spatial heterogeneity of the density, due to the formation of queues at intersections, as can be seen in Figure 1.2. We also derive probability distributions for the delay δ_{x_1,x_2} and travel time y_{x_1,x_2} between two locations x_1 and x_2 on a link of the network, noted respectively $h(\delta_{x_1,x_2})$ and $g(y_{x_1,x_2})$. Using the stationarity assumption, we define temporal averages of the traffic variables. These averages are then taken over a light cycle C. For example, we define Z as the average number of vehicles present on a link, with index u (resp. c) for the undersaturated (resp. the congested) regime.

Finally, given that the term *density* has a very specific meaning in traffic theory, we use the term *probability distribution* to refer to a *probability density function*.

17.2. Modeling the spatial distribution of vehicles on an arterial link

In typical traffic monitoring systems relying on probe data, probe vehicles send periodic location measurements, which provide two sources of indirect information about the arterial traffic link parameters. (i) As the location measurements are taken uniformly over time, more densely populated areas *of the link* will have more location measurements. (ii) The time spent between two consecutive location measurements provides information on the speed at which the vehicle drove through the corresponding arterial link(s).

We use the traffic flow model presented in Section 1 to derive the probability distribution of vehicle locations (averaged over time), which corresponds to the probability distribution of measurement locations. The derivation relies on the computation of the average vehicle density over a cycle.

17.2.1. General case

Using the stationarity assumption, the density at location x is time periodic with period C. We define the average density $d(x)$ at location x as the temporal average of the density $\rho(x,t)$ at location x and time t.

$$d(x) = \frac{1}{C}\int_0^C \rho\,(x,t)\,dt$$

In practice, flow is never perfectly periodic of period C (even in stationary conditions), but we will assume that the above averaging over a duration C is a good proxy of a longer average.

According to the model assumptions, the density at location x and time t takes one of the three following values, numbered 1 to 3 for convenience: (1) $\rho_1 = \rho_{max}$, when vehicles are stopped, (2) $\rho_2 = \rho_c$ when vehicles are dissipating from a queue, (3) $\rho_3 = \rho_a$ when vehicles arrive at the link and have not stopped in the queue.

The average density at location x is thus

$$d(x) = \sum_{i=1}^{3} \alpha_i\,(x)\rho_i$$

where $\alpha_i(x)$ represents the fraction of time that the density is equal to ρ_i at location x.

The probability distribution $f(x)$ of vehicle location at location x is proportional to the

average density $d(x)$ at location x, with the proportionality constant given by $Z = \int_0^L d\,(x)\,dx$ so that

$$f(x) = \frac{d(x)}{\int_0^L d\,(x)\,dx}.$$

In the undersaturated and the congested regime, the computation of the $\alpha_i(\cdot)$, $i = 1 \ldots 3$ enables the derivation of the probability distribution of vehicle locations.

17.2.2. Undersaturated regime

Upstream of the maximum queue length, the density remains constant at ρ_a throughout the whole light cycle. Downstream of the maximum queue length, the value of the density varies over time during the light cycle and takes one of the three density values ρ_1, $\rho_2$andρ_3.

Using the assumption that the FD is triangular and that the arrival density is constant, the average density increases linearly from ρ_a to the value it takes at the intersection, where $x = 0$. At the intersection, the density is ρ_{max} during the red time R. The density is ρ_c when the queue dissipates, *i.e.* during the clearing time $\tau = l_{max}(\frac{1}{w} + \frac{1}{v_f})$. Replacing w and l_{max} by their expressions, the time during which the queue dissipates is $R\frac{\rho_a}{\rho_c - \rho_a}$. The rest of the cycle has a duration $C - R\frac{\rho_c}{\rho_c - \rho_a}$ and it has density ρ_a. The average density at the intersection is the sum of the arrival, maximum and critical densities, weighted by the fraction of the cycle during which each of the density is experienced. The average density at the intersection is:

$$\begin{aligned} d(0) &= \frac{1}{C}\left(\underbrace{R\rho_{max}}_{\substack{\text{Red time } R \\ \text{at density } \rho_{max}}} + \underbrace{R\frac{\rho_a}{\rho_c - \rho_a}\rho_c}_{\substack{\text{Clearing time } \tau \\ \text{at density } \rho_c}} + \underbrace{\left(C - \left(R + R\frac{\rho_a}{\rho_c - \rho_a}\right)\right)\rho_a}_{\substack{\text{Extra green-time } C-(R+\tau) \\ \text{at density } \rho_a}} \right) \\ &= \frac{R}{C}\rho_{max} + \rho_a \end{aligned}$$

Given that the density grows linearly between the end of the queue and the intersection, the density at location x is given by

$$
\begin{aligned}
d(x) &= \rho_a && \text{if} x \geq l_{\max} \\
d(x) &= \rho_a + \frac{R}{C}\rho_{\max} \frac{l_{max} - x}{l_{max}} && \text{if} x \leq l_{\max}
\end{aligned} ,
$$

which can be summarized as

$$
d(x) = \rho_a + \frac{R}{C}\rho_{\max} \frac{max\,(l_{max} - x, 0)}{l_{max}}.
$$

We introduce the normalization constant Z_u, which is defined by $Z_u = \int_0^L d\,(x)\,dx$ and represents the temporal average of the number of vehicles on the link. Its explicit value is given by $Z_u = L\rho_a + \frac{l_{max}}{2}\frac{R}{C}\rho_{\max}$. The normalized density of vehicles as a function of the position on the link, defined by $f^u(x) = d(x)/Z_u$ is thus equal to

$$
\begin{aligned}
f^u(x) &= \frac{\rho_a}{Z_u} && \text{if} x \geq l_{\max} \\
f^u(x) &= \frac{\rho_a}{Z_u} + \frac{R}{C}\rho_{\max} \frac{l_{max} - x}{l_{max}\ Z_u} && \text{if} x \leq l_{\max}
\end{aligned} .
$$

When vehicles report their location arbitrarily in time, this function represents the probability of receiving a measurement at location x.

17.2.3. Congested regime

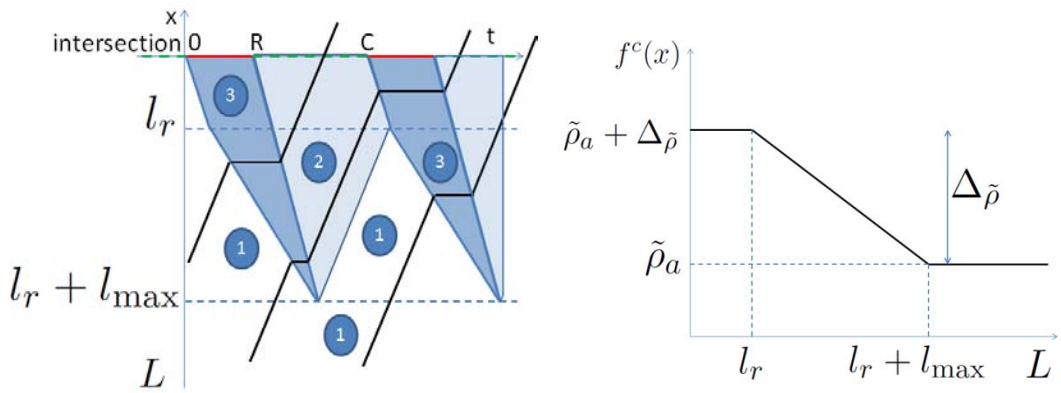


Figure 2.1. Left: The estimation of vehicle spatial distribution on a link is derived from the queue dynamics of the traffic flow model. The space-time plane represents the space-time domain in which density of vehicles is constant. Domain 1 represents the arrival density $\boldsymbol{\rho_a}$, domain 2 represents the critical density $\boldsymbol{\rho_c}$ and domain 3 represents the maximum density $\boldsymbol{\rho_{\max}}$. Right: Using the stationarity assumption, we compute the average density at location $\boldsymbol{x}$ and normalize to derive the probability distribution of vehicle locations on the link.

In the congested regime, the average density is constant upstream of the maximum queue length—equal to ρ_a—and increases linearly until the remaining queue. In the remaining queue, it is constant and equal to $\frac{R}{C}\rho_{\max} + (1-\frac{R}{C})\rho_c$. The different spatio-temporal domains of the density regions are illustrated Figure 2.1 (left). The probability distribution of vehicle locations is:

$$\begin{aligned} f^c(x) &= \frac{\rho_a}{Z_c} && \text{if } x \geq l_{\max} + l_r \\ f^c(x) &= \frac{\rho_a}{Z_c} + \left(\frac{R}{C}\rho_{\max} + \left(1-\frac{R}{C}\right)\rho_c - \rho_a\right)\frac{-x + l_{max} + l_r}{l_{max}\ Z_c} && \text{if } x \in [l_r, l_{\max} \quad (100) \\ f^c(x) &= \frac{R}{C}\frac{\rho_{max}}{Z_c} + \left(1-\frac{R}{C}\right)\frac{\rho_c}{Z_c} && \text{if } x \leq l_r \end{aligned}$$

where Z_c is the normalizing constant that ensures that the integral of the function on $[0, L]$ equals 1. We have

$$Z_c = L\rho_a + \left(\frac{l_{max}}{2} + l_r\right)\left(\frac{R}{C}\rho_{\max} + \left(1-\frac{R}{C}\right)\rho_c - \rho_a\right).$$

Notice that the undersaturated regime is a special case of the congested regime, in which the

remaining queue length l_r is equal to zero. In the remainder of this report, we consider the congested regime as the general case for the spatial distribution of vehicle location. This distribution is fully determined by three independent parameters. We choose the following parameterization: the remaining queue length l_r, the triangular queue length $l_{\max}$ and the normalized arrival density $\tilde{\rho}_a = \rho_a / Z_c$ to specify the distribution f^c. Using this parameterization, the probability distribution of vehicle location is illustrated in Figure 2.1 (right) and reads:

$$\begin{aligned} f^c(x) &= \tilde{\rho}_a && \text{if} x \geq l_{\max} + l_r \\ f^c(x) &= \tilde{\rho}_a + \frac{(l_r + l_{max}) - x}{l_{max}} \Delta_{\tilde{\rho}} && \text{if} x \in [l_r, l_{\max} + l_r], \\ f^c(x) &= \tilde{\rho}_a + \Delta_{\tilde{\rho}} && \text{if} x \leq l_r \end{aligned}$$

$$\text{with} \Delta_{\tilde{\rho}} = \frac{1 - \tilde{\rho}_a L}{l_{max}/2 + l_r}. \tag{101}$$

The expression of $\Delta_{\tilde{\rho}}$ above, can be obtained easily by noticing that $\int_0^L f^c(x)\, dx = 1$ or by direct computation from Equation (100), by replacing Z_c by its expression (Equation (101)) and ρ_a by $Z_c \tilde{\rho}_a$.

<u>Remark:</u> The undersaturated regime is a special case of the congested regime in which $l_r = 0$.

17.3. Modeling the probability distribution of delay among the vehicles entering the link in a cycle

The travel time experienced by vehicles traveling on arterial networks is conditioned on two factors. First, the traffic conditions, given by the parameters of the network, dictate the state of traffic experienced by all the vehicles entering the link. Second, the time (after the beginning of a cycle) at which each vehicle arrives at the link determines how much delay will be experienced in the queue due to the presence of a traffic signal and the presence of other vehicles. Under similar traffic conditions, drivers experience different travel times depending on their arrival time. Using the assumption that the arrival density (and thus the arrival rate) is constant, arrival times are uniformly distributed on the duration of the light cycle. This allows for the derivation of the pdf of delay, which depends on the characteristics of the traffic light and the traffic conditions as defined in Section 4.

In this work, we assume that we receive travel time measurements from vehicles traveling on the network. The vehicles are sampled uniformly in time and they send tuples of the form

(x_1, t_1, x_2, t_2) where x_1 is the location of the vehicle at t_1 and x_2 is the position of the vehicle at t_2. This is representative of taxi fleets or truck delivery fleets which typically send data every minute in urban networks. We consider all the tuples sent by the vehicles independently. For example, we assumethat the sampling strategy is such that we cannot reconstruct the trajectories of vehicles from the tuples (*e.g.* at each sampling time, the vehicles send tuples with a defined probability).

17.3.1. *Total delay* and *measured delay* between locations x_1 and x_2

We consider a vehicle traveling from location x_1 to location x_2 and sending its location x_1 at time t_1 and its location x_2 at time t_2. We call *measured delay from* x_1 to x_2, *experienced in the time interval* $[t_1, t_2]$, in short "measured delay from x_1 to x_2", the difference between the travel time of the vehicle $(t_2 - t_1)$ and the travel time that the vehicle would experience between x_1 and x_2 without the presence of other vehicles nor signals. For a vehicle with free flow pace p_f, we call free flow travel time between x_1 and x_2, the quantity $y_{f;x_1,x_2} = p_f(x_1 - x_2)$, representing the travel time between x_1 and x_2 if the vehicle is not slowed down or stopped on its trajectory. The delay experienced between x_1 and x_2 is the difference between the travel time y_{x_1,x_2} of the vehicle between x_1 and x_2—not necessarily at free flow speed—and the free flow travel time $y_{f;x_1,x_2}$. In this model, vehicles are either stopped or driving at the free flow speed. The measured delay from x_1 to x_2, experienced in the time interval $[t_1, t_2]$ is the cumulative stopping time between t_1 and t_2.

We call *total delay* from x_1 to x_2 the cumulative stopping time of the vehicle on its trajectory from x_1 (from the first time it joined the queue, if the vehicle was in the queue at x_1) to x_2 (until the time it left the queue, if it was in the queue at x_2). In particular, if the vehicle stops at x_1 or at x_2 the total delay from x_1 to x_2 covers the full delay experienced during the stop, without taking into account the sampling scheme. Note that for vehicles sampled at x_1 and x_2 that do not stop at x_1 nor at x_2 the total delay is equal to the measured delay. For vehicles stopping in x_1 or in x_2, the measured delay is less than or equal to the total delay experienced by the vehicle (Figure 1 (right)).

To gain more insight in the difference between *measured* and *total* delay, we can study a simple case. Let a vehicle be sampled every 30 seconds. Assume that the vehicle stops at the traffic signal ($x = 0$) and that the duration of the red time is 40 seconds. The vehicle sends its locations x_1 at t_1 and x_2 at $t_2 = t_1 + 30$. We do not receive additional information on the trajectory prior to t_1 or past t_2. The measured delay is at most 30 seconds (sampling rate); the total delay is 40 seconds. As a general remark, a vehicle reporting its delay during a stop reports a delay that is less than or equal to the total delay experienced on the trajectory, it represents the delay experienced between the two sampling times.

Using the modeling assumptions defined in Section 1, we derive the pdf of the *measured* and the *total* delay between any two locations x_1 and x_2. Given two sampling locations x_1 and x_2, the probability distribution of the total (resp. measured) delay δ_{x_1,x_2} is denoted $h^t(\delta_{x_1,x_2})$ (resp. $h^m(\delta_{x_1,x_2})$). We use the stationarity and constant arrival assumptions to derive the speed of formation and dissolution of the queue, respectively denoted v_a and w (96). Under the

stationarity assumption, the traffic variables are periodic with period C. For each arrival time, we compute the delay corresponding to the trajectory of the vehicle (Figure 1.2). The arrivals being uniform, we can compute the probability distribution of delays.

17.3.2. Probability distribution of the total and measured delay between x_1 and x_2 in the undersaturated regime

Pdf of the *total* delay between x_1 and x_2:

In the undersaturated regime, we call $\eta^u_{x_1,x_2}$, the fraction of the vehicles entering the link during a cycle that experience a delay between x_1 and x_2. The remainder of the vehicles entering the link in a cycle travels from x_1 to x_2 without experiencing any delay. The proportion $\eta^u_{x_1,x_2}$ of vehicles delayed between x_1 and x_2 in a cycle, is computed as the ratio of vehicles joining the queue between x_1 and x_2 over the total number of vehicles entering the link in one cycle (Figure 1, left). The number of vehicles joining the queue between x_1 and x_2 is the number of vehicles stopped between x_1 and x_2: $(\min(l_{\max}, x_1) - \min(l_{\max}, x_2))\rho_{\max}$. The number of vehicles entering the link is $v_f C \rho_a$. The proportion of vehicles delayed between x_1 and x_2 is thus:

$$\eta^u_{x_1,x_2} = (\min(x_1, l_{\max}) - \min(x_2, l_{\max}))\frac{\rho_{max}}{v_f C \rho_a}.$$

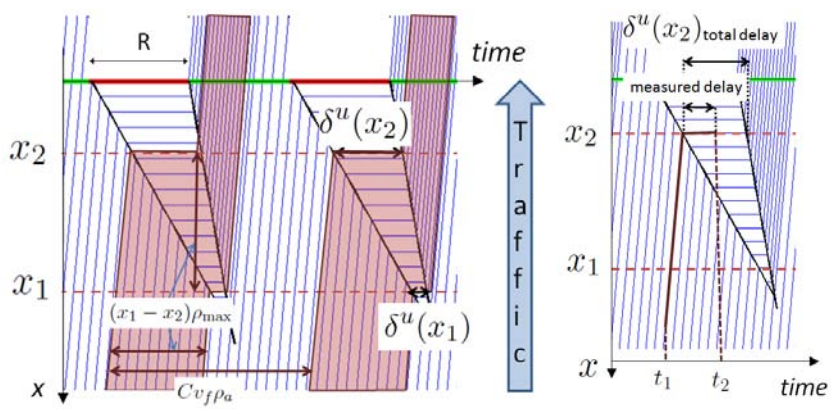


Figure 3.1. (Left) The proportion of delayed vehicles $\eta^u_{x_1,x_2}$ is the ratio between the number of vehicles joining the queue between x_1 and x_2 over the total number of vehicles entering the link in one cycle. The trajectories highlighted in purple represent the trajectories of vehicles delayed between x_1 and x_2. (Right) The vehicles reporting their location during a stop at x_2 experience a delay $\delta \in [0, \delta^u(x_2)]$ in the time interval $[t_1, t_2]$. This delay is less than or equal to the total delay ($\delta^u(x_2)$) experienced on the trajectory.

The total stopping time experienced when stopping at x is denoted by $\delta^u(x)$ for the undersaturated regime. Because the arrival of vehicles is homogenous, the delay $\delta^u(x)$ increases linearly with x. At the intersection ($x = 0$), the delay is maximal and equals the

duration of the red light R. At the end of the queue ($x = l_{\max}$) and upstream of the queue ($x \geq l_{\max}$), the delay is null. Thus the expression of $\delta^u(x)$:

$$\delta^u(x) = R\left(1 - \frac{min\,(x, l_{max})}{l_{max}}\right).$$

Given that the arrival of vehicles is uniform in time, the distribution of the location where the vehicles reach the queue between x_1 and x_2 is uniform in space. For vehicles reaching the queue between x_1 and x_2, the probability to experience a delay between locations x_1 and x_2 is uniform. The uniform distribution has support $[\delta^u(x_1), \delta^u(x_2)]$, corresponding to the minimum and maximum delay between x_1 and x_2.

The *total* delay experienced between x_1 and x_2 is a random variable with a mixture distribution with two components. The first component represents the vehicles that do not experience any stopping time between x_1 and x_2 (mass distribution in 0), the second component represents the vehicles reaching the queue between x_1 and x_2 (uniform distribution on $[\delta^u(x_1), \delta^u(x_2)]$). We note $\mathbf{1}_A$ the indicator function of set A,

$$\mathbf{1}_A(x) = \begin{cases} 1 \text{if} x \in A \\ 0 \text{if} x \notin A \end{cases}$$

We note $\mathrm{Dir}_{\{a\}}(\cdot)$ the Dirac distribution centered in a, used to represent the mass probability. The pdf of total delay between x_1 and x_2 (Figure 3.3, left) reads:

$$h^t(\delta_{x_1,x_2}) = (1 - \eta^u_{x_1,x_2})\mathrm{Dir}_{\{0\}}(\delta_{x_1,x_2}) + \frac{\eta^u_{x_1,x_2}}{\delta^u(x_2) - \delta^u(x_1)}\mathbf{1}_{[\delta^u(x_1),\delta^u(x_2)]}(\delta_{x_1,x_2})$$

The cumulative distribution function of total delay $H^t(\cdot)$ reads:

$$H^t(\delta_{x_1,x_2}) = \begin{cases} 0 & \text{if} \quad \delta_{x_1,x_2} < 0 \\ (1 - \eta^u_{x_1,x_2}) & \text{if} \quad \delta_{x_1,x_2} \in [0, \delta^u(x_1)] \\ (1 - \eta^u_{x_1,x_2}) + \eta^u_{x_1,x_2}\dfrac{\delta_{x_1,x_2} - \delta^u(x_1)}{\delta^u(x_2) - \delta^u(x_1)} & \text{if} \quad \delta_{x_1,x_2} \in [\delta^u(x_1), \delta^u(x_2)] \\ 1 & \text{if} \quad \delta_{x_1,x_2} > \delta^u(x_2) \end{cases}$$

Pdf of the *measured* delay between x_1 and x_2:

because of the sampling scheme, the measured delay differs from the total delay experienced by the vehicles.

In the following, i refers to the upstream or the downstream measurement locations ($i \in \{1,2\}$). When sending their location x_i, some vehicles are stopped at this location. These vehicles may not report the full delay associated with location x_i (Figure 1, right). In particular, a vehicle stopped at x_1 when sending its location at time t_1 will only report the delay experienced after t_1. Similarly, a vehicle stopped at x_2 when sending its location at time t_2 will only report the delay experienced before t_2.

- For a measurement received at x_i, the probability that it comes from a vehicle stopped in the queue is $\frac{\delta^u(x_i)}{c}$, which is the ratio of the time spent by the stopped vehicle at x_i over the duration of the cycle.
- For a vehicle stopped at x_2, observed at t_2 coming from a previous observation point x_1 (at t_1), the probability of a stopping time experienced by the vehicle until t_2 is uniform between 0 and $\delta^u(x_2)$, since the vehicle is sampled arbitrarily during its stopping phase.

We classify the vehicles traveling from x_1 to x_2 (where x_1 and x_2 are measurement points) depending on the locations of their stop with respect to x_1 and x_2. This classification of the trajectories is also illustrated in Figure 3.2:

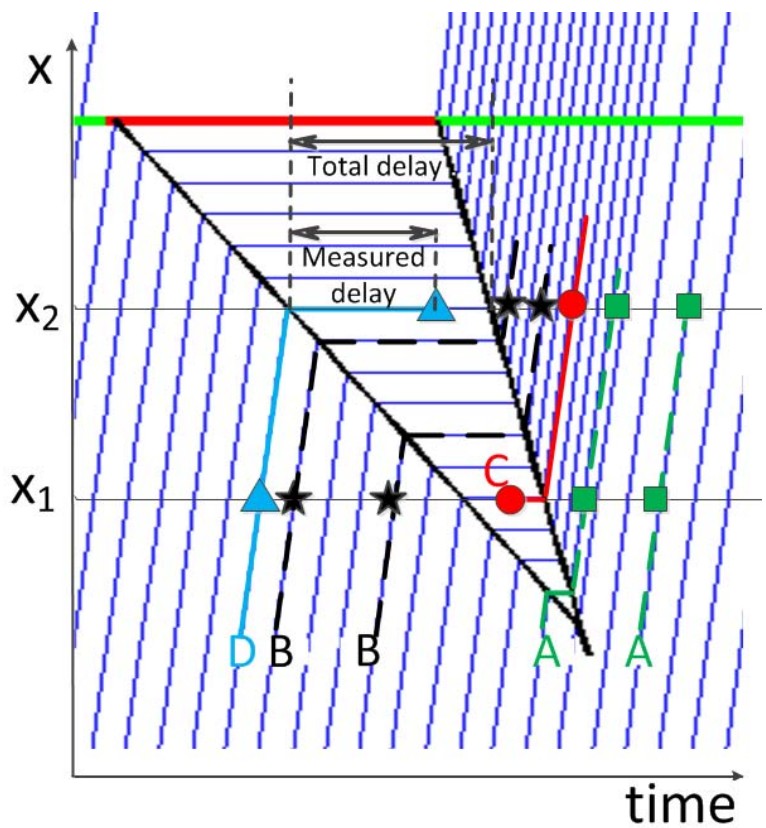


Figure 3.2. Classification of the trajectories depending on the stopping location. We consider vehicles traveling between the two measurement points x_1 and x_2, sampled uniformly in time.

- Vehicles do not experience any delay between the measurement points x_1 and x_2.
- Vehicles reach the queue at x with $x \in (x_2, x_1)$. These vehicles are not stopped when they send their location at x_1 and x_2.
- Vehicles reach the queue at x_1, where they report their location at time t_1. At t_1, the vehicle was already stopped (or was just stopping) and the measurement only accounts for the delay occurring after t_1, which is less than or equal to$\delta^u(x_1)$. Because of the uniform sampling in time, the reported delay has a uniform distribution on $[0, \delta^u(x_1)]$
- Vehicles reach the queue at x_2, where they report their location at time t_2. At t_2, the vehicle is still stopped and the measurement only represents the delay occurring up to t_2, which is less than or equal to $\delta^u(x_2)$. Because of the uniform sampling in time, the reported delay has a uniform distribution on $[0, \delta^u(x_2)]$

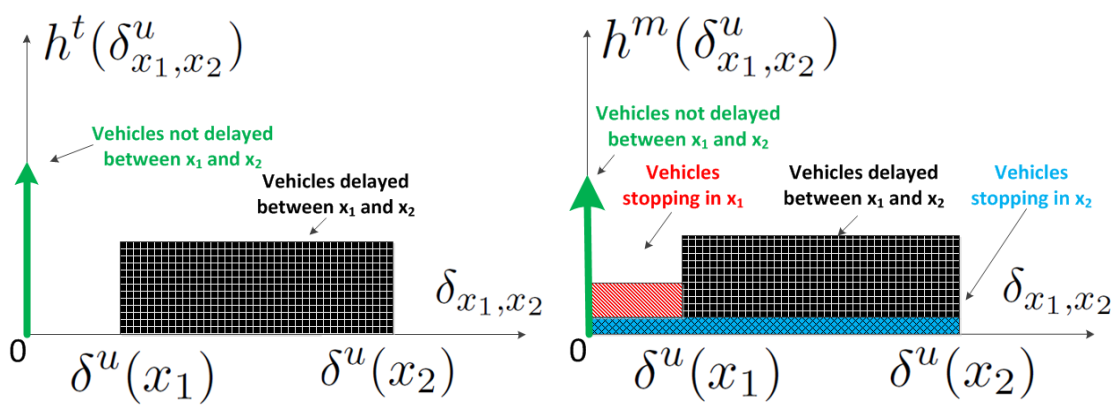


Figure 3.3. (Left) Probability distribution of the total delay between x_1 and x_2 in the undersaturated regime. (Right) Probability distribution of the measured delay between x_1 and x_2 in the undersaturated regime. Vehicles are assumed to be sampled uniformly in time.

We denote by s_{x_i} the event "*vehicle stops at location* x_i". Denoting by $\mathcal{P}(A)$ the probability of event A, we have $\mathcal{P}(s_{x_i}) = \frac{\delta^u(x_i)}{C}$. The notation $\bar{s}_{x_i}$ represents the event "*vehicle does not stop at location* x_i". The notation $(\bar{s}_{x_1}, \bar{s}_{x_2})$ represents the event "*vehicles do not stop at location* x_1 *nor* x_2". We assume that the events $\bar{s}_{x_1}$ and $\bar{s}_{x_2}$ are independent. The probability of event $(\bar{s}_{x_1}, \bar{s}_{x_2})$ reads:

$$\begin{aligned} \mathcal{P}(\bar{s}_{x_1}, \bar{s}_{x_2}) &= \mathcal{P}(\bar{s}_{x_1})\mathcal{P}(\bar{s}_{x_2}) && \text{Independence assumption} \\ &= (1 - \mathcal{P}(s_{x_1}))(1 - \mathcal{P}(s_{x_2})) && \text{Complementary events} \end{aligned}$$

The event $(\bar{s}_{x_1}, \bar{s}_{x_2})$ corresponds to trajectories of type A (vehicles do not stop between x_1 and x_2) and trajectories of type B (vehicles stop strictly between x_1 and x_2 but neither in x_1 nor in x_2). Among the vehicles stopping at none of the measurement points, a fraction $\eta^u_{x_1,x_2}$ is

delayed between x_1 and x_2 (trajectories of type B) and a fraction $1-\eta^u_{x_1,x_2}$ does not experience delay between x_1 and x_2 (trajectories of type A). Given that we receive a delay measurement between locations x_1 and x_2, the probability that it was sent by a vehicle with a trajectory of type A is $\mathcal{P}(\bar{s}_{x_1},\bar{s}_{x_2})(1-\eta^u_{x_1,x_2})$. Similarly, the probability that it was sent by a vehicle with a trajectory of type B is $\mathcal{P}(\bar{s}_{x_1},\bar{s}_{x_2})\eta^u_{x_1,x_2}$.

Given that a measurement is received at location x_i, the probability that this measurement is sent by a vehicle that joined the queue at x_i is proportional to the delay experienced at location x_i. Given successive measurements at locations x_1 and x_2, the probability that a vehicle reports its location x_i ($i \in \{1,2\}$) while being stopped at this location is denoted ζ_{x_i}. From this definition and given that we receive a delay measurement between locations x_1 and x_2, the probability that it was sent by a vehicle with a trajectory of type C is ζ_{x_1}. The probability that it was sent by a vehicle with a trajectory of type D is ζ_{x_2}. Note that vehicles cannot stop both at x_1 and x_2 (they stop only once in the queue); thus $\mathcal{P}(s_{x_1},s_{x_2})=0$. Given that a vehicle was sampled at x_1 and x_2, we have:

$$\underbrace{\zeta_{x_1}}_{\substack{\text{Prob.thattheveh}\\ \text{stopped at } x_1 \text{ only}}} + \underbrace{\zeta_{x_2}}_{\substack{\text{Prob.that the veh}\\ \text{stopped at } x_2 \text{ only}}} + \underbrace{\mathcal{P}(\bar{s}_{x_1},\bar{s}_{x_2})}_{\substack{\text{Prob.that the veh}\\ \text{stopped neither at } x_1 \text{ nor at } x_2}} + \underbrace{\mathcal{P}(\bar{s}_{x_1},\bar{s}_{x_2})}_{\substack{\text{Prob.that the veh}\\ \text{stopped both at } x_1 \text{ and at } x_2 (=0)}} = 1$$

The probability of stopping either at x $_1$ or at x_2 is $1-\mathcal{P}(\bar{s}_{x_1},\bar{s}_{x_2})$ (complementary of stopping neither at x_1 nor at x_2). Among these vehicles, the proportion that stops in x_1 is proportional to the delay experienced in x_1. We have:

$$\begin{cases} \zeta_{x_i} & \propto & \delta^u(x_i),\ i \in \{1,2\} \\ \zeta_{x_1}+\zeta_{x_2} & = & 1-\mathcal{P}(\bar{s}_{x_1},\bar{s}_{x_2}) \end{cases} \Rightarrow \zeta_{x_i} = \left(1-\mathcal{P}(\bar{s}_{x_1},\bar{s}_{x_2})\right)\frac{\delta^u(x_i)}{\delta^u(x_1)+\delta^u(x_2)} \quad i \in \{1,2\}$$

The probability distribution of measured delay is a finite mixture distribution, in which each component is a mass probability or a uniform distribution. The theoretical probability distribution function is illustrated Figure 3.3, right. It is the sum of the following terms that also refer to Figure 3.2:

A. a mass probability in 0 with weight $(1-\eta^u_{x_1,x_2})\mathcal{P}(\bar{s}_{x_1},\bar{s}_{x_2})$, representing the vehicles that do not reach the queue between x_1 and x_2,
B. a uniform distribution on $(\delta^u(x_1),\delta^u(x_2))$ with weight $\eta^u_{x_1,x_2}\mathcal{P}(\bar{s}_{x_1},\bar{s}_{x_2})$, representing the vehicles that reach the queue strictly between x_1 and x_2,
C. a uniform distribution on $[0,\delta^u(x_1)]$ with weight ζ_{x_1}, representing the vehicles that stop in x_1,
D. a uniform distribution on $[0,\delta^u(x_2)]$ with weight ζ_{x_2}, representing the vehicles that stop in x_2.

The pdf of the measured delay is related to the pdf of the total delay as:

$$h^m(\delta_{x_1,x_2}) = \mathcal{P}(\bar{s}_{x_1}, \bar{s}_{x_2}) h^t(\delta_{x_1,x_2}) + \frac{\zeta_{x_1}}{\delta^u(x_1)} \mathbf{1}_{[0,\delta^u(x_1)]}(\delta_{x_1,x_2}) + \frac{\zeta_{x_2}}{\delta^u(x_2)} \mathbf{1}_{[0,\delta^u(x_2)]}(\delta_{x_1,x_2})$$

17.3.3. Probability distribution of the measured delay between x_1 and x_2 in the congested regime

In the congested regime, the delay distribution can be computed using a similar methodology as for the undersaturated regime, by deriving the delay experienced between x_1 and x_2 for each arrival time. We call n_s the maximum number of stops experienced by the vehicles in the remaining queue between the locations x_1 and x_2. The delay experienced at location x when reaching the triangular queue at x is readily derived from the expression of the delay in the undersaturated regime. The delayexperienced when reaching the remaining queue is the duration of the red time R. The expression of the delay at location x is then

$$\delta^c(x) = \begin{cases} R & \text{if} x \leq l_r \\ R \dfrac{l_r + l_{max} - x}{l_{max}} & \text{if} x \in [l_r, l_r + l_{\max}] \\ 0 & \text{if} x \geq l_r + l_{\max} \end{cases}$$

The details of the derivation are given in Appendix A and illustrated in Figures A.1–A.4. Note that to satisfy the stationarity assumption, the distance traveled by vehicles in the queue in the duration of a light cycle is $l_{\max}$.

We summarize the derivations, classified depending on the location of the positions x_1 and x_2 with respect to the remaining and triangular queue lengths:

1. x_1 *Upstream* – x_2 *Remaining* (Figure A.1): The location x_1 is upstream of the queue and the location x_2 is in the triangular queue. We define the critical location x_c by $x_c = x_2 + n_s l_{\max}$. Vehicles reaching the triangular queue upstream of x_c stop n_s times in the remaining queue. On the road segment $[x_1, x_2]$, vehicles reaching the triangular queue downstream of x_c stop $n_s - 1$ times in the remaining queue. The vehicles experience a delay uniformly distributed on $[\delta_{\min}, \delta_{\max}]$ with $\delta_{\min} = (n_s - 1)R + \delta^c(x_c)$ and $\delta_{\max} = n_s R + \delta^c(x_c) = \delta_{\min} + R$. The probability distribution of total delay reads:

$$h^t(\delta_{x_1,x_2}) = \frac{1}{\delta_{max} - \delta_{min}} \mathbf{1}_{[\delta_{\min}, \delta_{\max}]}(\delta_{x_1,x_2}), \quad \begin{array}{l} \delta_{\min} = \delta^c(x_c) + (n_s - 1)R \\ \delta_{\max} = \delta^c(x_c) + n_s R \end{array}$$

2. x_1 *Triangular* – x_2 *Triangular* (Figure A.2): Both locations x_1 and x_2 are upstream

of the remaining queue (in the triangular queue or upstream of the queue). Given that the path is upstream of the remaining queue, this case is similar to the undersaturated regime, where derivations are updated to account for the fact that the triangular queue starts at $x = l_r$. We adapt the notation from Section 2 and denote by $\eta^c_{x_1,x_2}$ the fraction of the vehicles entering the link in a cycle that experience delay between locations x_1 and x_2.

$$\eta^c_{x_1,x_2} = \frac{min\,(\,x_1 - l_r, l_{max}\quad) - min\,(\,x_2 - l_r, l_{max}\quad)}{l_{max}}$$

This delay is uniformly distributed on $[\delta^c(x_1), \delta^c(x_2)]$. The reminder do not stop between x_1 and x_2. The probability distribution of total delay reads:

$$h^t(\delta_{x_1,x_2}) = (1 - \eta^c_{x_1,x_2})\mathrm{Dir}_{\{0\}}(\delta_{x_1,x_2}) + \frac{\eta^c_{x_1,x_2}}{\delta^c(x_2) - \delta^c(x_1)}\mathbf{1}_{[\delta^c(x_1),\delta^c(x_2)]}(\delta_{x_1,x_2})$$

3. x_1 *Remaining* – x_2 *Remaining* (Figure A.3): Both locations x_1 and x_2 are in the remaining queue. We define the critical location x_c by $x_c = x_2 + (n_s - 1)l_{\max}$. The vehicles reaching the queue between x_1 and x_c stop n_s times in the remaining queue between x_1 and x_2, their stopping time is $n_s R$. The reminder of the vehicles stop $n_s - 1$ times in the remaining queue and their stopping time is $(n_s - 1)R$. The probability distribution of total delay reads:

$$h^t(\delta_{x_1,x_2}) = \frac{x_1 - x_c}{l_{max}}\mathrm{Dir}_{\{n_s R\}}(\delta_{x_1,x_2}) + \left(1 - \frac{x_1 - x_c}{l_{max}}\right)\mathrm{Dir}_{\{(n_s-1)R\}}(\delta_{x_1,x_2})$$

4. x_1 *Triangular* – x_2 *Remaining* (Figure A.4): The upstream location x_1 is in the triangular queue and the downstream location x_2 is in the remaining queue. We define the critical location x_c by $x_c = x_2 + n_s l_{max}$.

   - If $x_1 \geq x_c$, a fraction $(x_1 - x_c)/l_{\max}$ of the vehicles entering the link in a cycle join the triangular queue between x_1 and x_c. They stop once in the triangular queue and n_s times in the remaining queue. Among these vehicles, the stopping time is uniformly distributed on $[\delta^c(x_1) + n_s R,\, \delta^c(x_c) + n_s R]$. A fraction $(x_c - l_r)/l_{\max}$ of the vehicles entering the link in a cycle join the triangular queue between x_c and $l_{\max}$. Among these vehicles, the stopping time is uniformly distributed on $[\delta^c(x_c) + (n_s - 1)R,\, n_s R]$. The remainder of the vehicles reach the remaining queue between l_r and $x_1 - l_{\max}$ and their stopping time is $n_s R$. The probability distribution of total delay reads:
   -

$$h^t(\delta_{x_1,x_2}) = \frac{x_1 - x_c}{l_{max}} \frac{\mathbf{1}_{[\delta^c(x_1)+n_sR,\, \delta^c(x_c)+n_sR]}(\delta_{x_1,x_2})}{\delta^c(x_c) - \delta^c(x_1)} \quad \text{Vehicles stopping between } x_1 \text{ and } x_c$$

$$+ \frac{x_c - l_r}{l_{max}} \frac{\mathbf{1}_{[\delta^c(x_c)+(n_s-1)R,\, n_sR]}(\delta_{x_1,x_2})}{R - \delta^c(x_c)} \quad \text{Vehicles stopping between } x_c \text{ and } l_r$$

$$+ \left(1 - \frac{x_1 - l_r}{l_{max}}\right) \mathrm{Dir}_{\{n_sR\}}(\delta_{x_1,x_2}) \quad \text{Vehicles stopping between } l_r \text{ and } x_1 - l_{\max}$$

- If $x_1 \leq x_c$, a fraction $(x_1 - l_r)/l_{\max}$ of the vehicles entering the link in a cycle join the triangular queue between x_1 and l_r. They stop once in the triangular queue and $n_s - 1$ times in the remaining queue. Among these vehicles, the stopping time is uniformly distributed on $[\delta^c(x_1) + (n_s - 1)R,\, n_sR]$. A fraction $1 - (x_c - l_r)/l_{\max}$ of the vehicles entering the link in a cycle join the remaining queue between l_r and $x_c - l_{\max}$. The stopping time of these vehicles is n_sR. The remainder of the vehicles experiences a stopping time of $(n_s - 1)R$. The probability distribution of total delay reads:

$$h^t(\delta_{x_1,x_2}) = \frac{x_1 - l_r}{l_{max}} \frac{\mathbf{1}_{[\delta^c(x_1)+(n_s-1)R,\, n_sR]}(\delta_{x_1,x_2})}{R - \delta^c(x_1)} \quad \text{Vehicles stopping between } x_1 \text{ and } l_r$$

$$+ \left(1 - \frac{x_c - l_r}{l_{max}}\right) \mathrm{Dir}_{\{n_sR\}}(\delta_{x_1,x_2}) \quad \text{Vehicles stopping between } l_r \text{ and } x_c - l_{\max}$$

$$+ \frac{x_c - x_1}{l_{max}} \mathrm{Dir}_{\{(n_s-1)R\}}(\delta_{x_1,x_2}) \quad \text{Vehicles stopping between } x_c - l_{\max} \text{ and } x_1 - l_{\max}$$

These cases represent the pdf of total delay. From the results derived in Section 2, we derive the pdf of measured delay. From the previous derivations, we have:

$$\begin{aligned} \mathcal{P}(\bar{s}_{x_1}, \bar{s}_{x_1}) &= (1 - \mathcal{P}(\bar{s}_{x_1}))(1 - \mathcal{P}(\bar{s}_{x_2})) \\ \zeta_{x_i} &= \left(1 - \mathcal{P}(\bar{s}_{x_1}, \bar{s}_{x_2})\right) \frac{\delta^u(x_i)}{\delta^u(x_1) + \delta^u(x_2)} \quad i \in \{1,2\} \end{aligned}$$

It is the sum of the following terms:

i. the delay distribution given that the vehicles stop neither in x_1 nor in x_2, with weight $\mathcal{P}(\bar{s}_{x_1}, \bar{s}_{x_2})$,
ii. the delay probability distribution given a stop in x_1, with weight ζ_{x_1},
iii. the delay probability distribution given a stop in x_2, with weight ζ_{x_2}.

We summarize the different components of the delay distribution, described as a mixture distribution for all the different cases in Table 12.

17.4. Probability distributions of travel times

On a path between x_1 and x_2, the travel time y_{x_1,x_2} is a random variable. It is the sum of two random variables: the delay δ_{x_1,x_2} experienced between x_1 and x_2 and the free flow travel time of the vehicles $y_{f;x_1,x_2}$.The free flow travel time is proportional to the distance of the path and the free flow pace p_f such that $y_{f;x_1,x_2} = p_f(x_1 - x_2)$. We have $y_{x_1,x_2} = \delta_{x_1,x_2} + y_{f;x_1,x_2}$.

In the following, we assume that the delay and the free flow pace are independent random variables, thus so are the delay and the free flow travel time.

We model the differences in traffic behavior by assuming a prior distribution on the free flow pace p_f. The free flow pace is modeled as a random variable with distribution φ^p and support $\mathcal{D}_{\varphi^p}$. For convenience, we define for a pdf φ with support $\mathcal{D}_\varphi$, its prolongation by zero of out of $\mathcal{D}_\varphi$. With a slight abuse of notation, we call this new function φ.

Using a linear change of variables, we derive the probability distribution $\varphi^y_{x_1,x_2}$ of free flow travel time $y_{f;x_1,x_2}$ between x_1 and x_2:

$$p_f \sim \varphi^p(p_f) \Rightarrow \varphi^y_{x_1,x_2}(y_{f;x_1,x_2}) = \varphi^p\left(\frac{y_{f;x_1,x_2}}{x_1 - x_2}\right)\frac{1}{x_1 - x_2}$$

To derive the pdf of travel times we use the following fact:

Fact 1. *Sum of independent random variables* If X and Y are two independent random variables with respective pdf f_X and f_Y, then the pdf f_Z of the random variable $Z = X + Y$ is given by $f_Z(z) = f_X * f_Y(z)$

This classical result in probability is derived by computing the conditional pdf of Z given X and then integrating over the values of X according to the total probability law.

For each regime s, the probability distribution of travel times reads:

$$g^s(y_{x_1,x_2}) = \left(h^s * \varphi^y_{x_1,x_2}\right)(y_{x_1,x_2})$$

We notice that the delay distributions are mixtures of mass probabilities and uniform distributions. We derive the general expression of the travel time distributions when vehicles experience a delay with mass probability in Δ and when vehicles experience a delay with uniform distribution on $[\delta_{\min}, \delta_{\max}]$.

Case	Trajectories	Weight	Dist.	Support
<u>Case1</u> $x_1 \geq l_r + l_{\max}$, $x_2 \leq l_r$, $x_c = x_2 + n_s l_{\max}$	Does not stop at x_2	$\mathcal{P}(\bar{s}_{x_1}, \bar{s}_{x_2})$	Unif.	$[(n_s - 1)R + \delta^c(x_c),$ $n_s R + \delta^c(x_c)]$
	Stop at x_2	$\zeta_{x_2} = 1 - \mathcal{P}(\bar{s}_{x_1}, \bar{s}_{x_2})$	Unif.	$[(n_s - 1)R + \delta^c(x_c),$ $n_s R + \delta^c(x_c)]$
<u>Case2</u> $x_1 \geq l_r$, $x_2 \geq l_r$	No stop between x_1 and x_2	$\mathcal{P}(\bar{s}_{x_1}, \bar{s}_{x_2}) \times (1 - \eta^c_{x_1,x_2})$	Mass	$\{0\}$
	Reach the (triangular) queue between x_1 and x_2	$\mathcal{P}(\bar{s}_{x_1}, \bar{s}_{x_2}) \times \eta^c_{x_1,x_2}$	Unif.	$[\delta^c(x_2), \delta^c(x_1)]$
	Stop at x_1	ζ_{x_1}	Unif.	$[0,\ \delta^c(x_1)]$
	Stop at x_2	ζ_{x_2}	Unif.	$[0,\ \delta^c(x_2)]$
<u>Case3</u> $x_1 \leq l_r$, $x_2 \leq l_r$, $x_c = x_2 + (n_s - 1)l_{\max}$	Reach the (remaining) queue between x_1 and x_c	$\mathcal{P}(\bar{s}_{x_1}, \bar{s}_{x_2}) \times \frac{x_1 - x_c}{l_{max}}$	Mass	$\{n_s R\}$
	Reach the (remaining) queue between x_c and $x_1 - l_{\max}$	$\mathcal{P}(\bar{s}_{x_1}, \bar{s}_{x_2}) \times \frac{x_c - x_1 + l_{max}}{l_{max}}$	Mass	$\{(n_s - 1)R\}$
	Stop at x_1	ζ_{x_1}	Unif.	$[(n_s - 1)R,\ n_s R]$
	Stop at x_2	ζ_{x_2}	Unif.	$[(n_s - 1)R,\ n_s R]$
<u>Case4a</u> $x_1 \in [l_r, l_r + l_{\max}]$, $x_2 \leq l_r$, $x_c = x_2 + n_s l_{\max}$, $x_c \leq x_1$	Reach the (triangular) queue between x_1 and x_c	$\mathcal{P}(\bar{s}_{x_1}, \bar{s}_{x_2}) \times \frac{x_1 - x_c}{l_{max}}$	Unif.	$[n_s R + \delta^c(x_1),$ $n_s R + \delta^c(x_c)]$
	Reach the (triangular) queue between x_c and l_r	$\mathcal{P}(\bar{s}_{x_1}, \bar{s}_{x_2}) \times \frac{x_c - l_r}{l_{max}}$	Unif.	$[(n_s - 1)R + \delta^c(x_c),$ $n_s R]$
	Reach the (remaining) queue between l_r and $x_1 - l_{\max}$	$\mathcal{P}(\bar{s}_{x_1}, \bar{s}_{x_2}) \times \frac{l_r - x_1 + l_{max}}{l_{max}}$	Mass	$\{n_s R\}$
	Stop at x_1	ζ_{x_1}	Unif.	$[n_s R,\ n_s R + \delta^c(x_1)]$
	Stop at x_2	ζ_{x_2}	Unif.	$[(n_s - 1)R + \delta^c(x_c),$ $n_s R + \delta^c(x_c)]$
<u>Case4b</u> $x_1 \in [l_r, l_r + l_{\max}]$, $x_2 \leq l_r$, $x_c = x_2 + n_s l_{\max}$, $x_c \geq x_1$	Reach the (triangular) queue between x_1 and l_r	$\mathcal{P}(\bar{s}_{x_1}, \bar{s}_{x_2}) \times \frac{x_1 - l_r}{l_{max}}$	Unif.	$[(n_s - 1)R + \delta^c(x_1),$ $n_s R]$
	Reach the (remaining) queue between l_r and $x_c - l_{\max}$	$\mathcal{P}(\bar{s}_{x_1}, \bar{s}_{x_2}) \times \frac{l_r - x_c + l_{max}}{l_{max}}$	Mass	$\{n_s R\}$
	Reach the (remaining) queue between $x_c - l_{\max}$ and $x_1 - l_{\max}$	$\mathcal{P}(\bar{s}_{x_1}, \bar{s}_{x_2}) \times \frac{x_c - x_1}{l_{max}}$	Mass	$\{(n_s - 1)R\}$
	Stop at x_1	ζ_{x_1}	Unif.	$[(n_s - 1)R,$ $(n_s - 1)R + \delta^c(x_1)]$
	Stop at x_2	ζ_{x_2}	Unif.	$[(n_s - 1)R,\ n_s R]$

Table 12. The pdf of measured delay is a mixture distribution. The different components and their associated weight depend on the location of stops of the vehicles with respect to the queue length and sampling locations.

17.4.1. Travel time distributions

Travel time distribution when the delay has a mass probability in Δ

The stopping time is Δ. This corresponds to trajectories with n_s stops ($n_s \geq 0$) in the remaining queue. This includes the non stopping vehicle in the undersaturated regime, when the remaining queue has length zero. The travel time distribution is derived as

$$g(y_{x_1,x_2}) = \mathrm{Dir}_{\{\Delta\}} * \varphi^y_{x_1,x_2}(y_{x_1,x_2})$$

$$= \varphi^y_{x_1,x_2}(y_{x_1,x_2} - \Delta). \tag{102}$$

Travel time distribution when the delay is uniformly distributed on $[\delta_{\min}, \delta_{\max}]$.

Vehicles experience a uniform delay between a minimum and maximum delay respectively denoted $\delta_{\min}$ and $\delta_{\max}$. The probability of observing a travel time y_{x_1,x_2} is given by

$$g(y_{x_1,x_2}) = \frac{1}{\delta_{max} - \delta_{min}} \int_{-\infty}^{+\infty} \mathbf{1}_{[\delta_{\min}, \delta_{\max}]}(y_{x_1,x_2} - z)\ \varphi^y_{x_1,x_2}(z)\, dz. \tag{103}$$

The integrand is not null if and only if $y_{x_1,x_2} - z \in [\delta_{\min}, \delta_{\max}]$, *i.e.* if and only if $z \in [y_{x_1,x_2} - \delta_{\max}, y_{x_1,x_2} - \delta_{\min}]$. Since $\varphi^y_{x_1,x_2}(z)$ is equal to zero for $z \in \mathbb{R} tminus \mathcal{D}_\varphi$, the integrand is not null if and only if $z \in [y_{x_1,x_2} - \delta_{\max}, y_{x_1,x_2} - \delta_{\min}] \cap \mathcal{D}_\varphi$.

As an illustration, we derive the probability distribution of travel times on an entire link in the undersaturated regime, for a pace distribution with support on $\mathbb{R}^+$ (Figure 4.1 (left)). The length of the link is denoted L. A fraction $1 - \eta^u_{L,0}$ of the vehicles entering the link in a cycle has a delay with mass probability in 0 (vehicles do not stop on the link). The probability distribution of travel times of these vehicles is computed via Equation (102) with $\Delta = 0$. The reminder of the vehicles (fraction $\eta^u_{L,0}$) experiences a delay that is uniformly distributed on $[0, R]$. The probability distribution of travel times of these vehicles is computed via Equation (103) with $\delta_{\min} = 0$ and $\delta_{\max} = R$. The probability distribution of travel times on an undersaturated arterial link reads:

$$g^u(y_{L,0}) = \begin{cases} 0 & \text{if} \quad y_{L,0} \leq 0 \\ (1-\eta_{L,0}^u)\varphi_{L,0}^y(y_{L,0}) + \frac{\eta_{L,0}^u}{R}\int_0^{y_{L,0}} \varphi_{L,0}^y(z)\,dz & \text{if} \quad y_{L,0} \in [0,R] \\ (1-\eta_{L,0}^u)\varphi_{L,0}^y(y_{L,0}) + \frac{\eta_{L,0}^u}{R}\int_{y_{L,0}-R}^{y_{L,0}} \varphi_{L,0}^y(z)\,dz & \text{if} \quad y_{L,0} \geq R \end{cases}, \tag{104}$$

In the more general case of a travel time distribution on an undersaturated partial link between locations x_1 and x_2, we write the delay distribution as a mixture of mass probabilities and uniform distributions. We use the linearity of the convolution to treat each component of the mixture separately and sum them with their respective weights to derive the probability distribution of travel times.

The derivations are similar in the congested regime. For the different cases described in Section 3, the delay is a mixture of mass probabilities and uniform distributions. For example, the probability distribution of link travel times (Case 1) is illustrated in Figure 4.1 (right)). When the delay is uniformly distributed on $[\delta_{\min}, \delta_{\max}]$, the probability distribution of travel times is computed via Equation (103) and reads

$$g^c(y_{L,0}) = \begin{cases} 0 & \text{if} \quad y_{L,0} \leq \delta_{\min} \\ \frac{1}{\delta_{max} - \delta_{min}}\int_0^{y_{L,0}-\delta_{\min}} \varphi_{L,0}^y(z)dz & \text{if} \quad y_{L,0} \in [\delta_{\min}, \delta_{\max}] \\ \frac{1}{\delta_{max} - \delta_{min}}\int_{y_{L,0}-\delta_{\max}}^{y_{L,0}-\delta_{\min}} \varphi_{L,0}^y(z)\,dz & \text{if} \quad y_{L,0} \geq \delta_{\max} \end{cases} \tag{105}$$

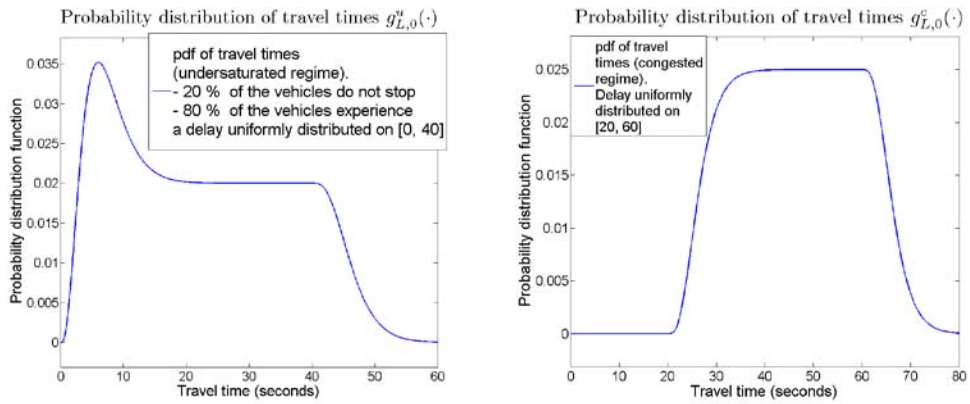


Figure 4.1. Probability distributions of link travel times. Left: Undersaturated regime. The figure represents pdf for a traffic light of duration 40 seconds when 80% of the vehicles stop at the light ($\eta^u_{L,0} = .8$). Right: Congested regime. The figure represents the pdf for a traffic light of duration 40 seconds when all the vehicles stop in the triangular queue and 50% of the vehicles stop once in the remaining queue. Both figures are produced for a link of length 100 meters. The free flow pace is a random variable with Gamma distribution. The mean free flow pace is 1/15 s/m and the standard deviation is 1/30 s/m. We recall that the probability distribution γ of a Gamma random variable $x \in \mathbb{R}^+$ with shape $\alpha$and inverse scale parameter β is given by $\gamma(x) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x}$, where Γ is the Gamma function defined on $\mathbb{R}^+$ and with integral expression $\Gamma(x) = \int_0^{+\infty} t^{x-1}\, e^{-t}\, dt$.

17.4.2. Quasi-concavity properties of the probability distributions of link travel times

The pdf of travel times depend on a set of parameters that must be estimated to fully determine the statistical distribution of the travel times. A parameter with true value θ_0 is estimated via an estimator $\hat{\theta}$. We require this estimator to have some optimality properties—extremum point based on an objective function, *e.g.* least square estimator, maximum likelihood estimator. In particular, the maximum likelihood estimator is widely used in statistics for its convergence properties. Its computation requires the maximization of the likelihood (or log-likelihood) function, which represents the probability of observing a set of data points, given the value of a parameter.

In this work, the function to maximize is the probability distribution of travel times. Properties on the concavity of this function are important for designing efficient maximization algorithms with guaranties of global optimality. In this section, we present the proof of the quasi-concavity of the link travel time distributions in both the undersaturated and the congested regimes. We also prove the log-concavity of the different components of the distribution of travel times, considered as mixture distributions.

Definition 2. *Quasi-concavity (Boyd)* [78] A function $f: \mathbb{R}^n \to \mathbb{R}$ is called quasi-concave if its domain is convex and if $\forall \alpha \in \mathbb{R}$, the superlevel set Sf_α ($Sf_\alpha = \{x \in \mathcal{D}_f | f(x) \geq \alpha\}$) is convex.

From this definition, one can derive equivalent characterization of quasi-concavity, when f has

first (and second) order derivatives. The reader should refer to [78] for further references on quasi-concavity. In particular, we use the characterization of continuous quasi-concave functions on $\mathbb{R}$ (Lemma and the second order characterization:

Lemma 3. *Characterization of continuous quasi-concave functions on* $\mathbb{R}$ [78] A continuous function $f: \mathcal{D}_f \to \mathbb{R}$ is quasi-concave if and only if at least one of the following conditions holds:

- f is nondecreasing
- f is nonincreasing
- there is a point $x_c \in \mathcal{D}_f$ such that for $x \leq x_c$ (and $x \in \mathcal{D}_f$), f is nonincreasing, and for $x \geq x_c$ (and $x \in \mathcal{D}_f$), f is nondecreasing

Lemma 4. *Second order characterization of quasi-concave functions* [78] $f \in \mathcal{C}^2$ is quasi-concave if and only if $\forall (x, y) \in {\mathcal{D}_f}^2,\ y^T \nabla f(x) = 0 \Rightarrow y^T \nabla^2 f(x) y \leq 0$. If f is unidimensional, f is quasi-concave if and only if $f'(x) = 0 \Rightarrow f''(x) \leq 0$.

Note that for probability distributions, we are interested in the properties of the log of the probability function.

Definition 5. *Log-concavity (Boyd)* [78] A function $f: \mathbb{R}^n \to \mathbb{R}_*^+$ is log-concave if and only if $\ln(f)$ is concave. The second order characterization is as follows:

$f \in \mathcal{C}^2$ is log-concave is and only if $\forall x\, f(x) f''(x) - (f'(x))^2 \leq 0$.

Fact 6. For f a twice differentiable function taking values in $\mathbb{R}_*^+$, f is quasi-concave $\Leftrightarrow$ $\ln(f)$ is quasi-concave.

Proof. We have $\nabla \ln f(x) = \frac{\nabla f(x)}{f(x)}$ and $\nabla^2 \ln f(x) = \frac{f(x)\nabla^2 f(x) - \nabla f(x) \nabla f(x)^T}{f(x)^2}$. •

<u>We assume that f is quasi-concave, we want to show that $ln(f)$ is quasi-concave:</u>

We assume $\forall (x, y),\ y^T \nabla f(x) = 0 \Rightarrow y^T \nabla^2 f(x) y \leq 0$.

Let x and y be such that $y^T \nabla \ln(f(x)) = 0$, *i.e.* $y^T \nabla f(x) = 0$. From the quasi-concavity of f, we have $y^T \nabla^2 f(x) y \leq 0$

$$y^T \nabla^2 \ln(f(x)) y = \frac{f(x)\, y^T \nabla^2 f(x) y - y^T \nabla f(x) \nabla f(x)^T y}{f(x)^2}$$

So $y^T \nabla \ln(f(x)) = 0 \Rightarrow y^T \nabla^2 \ln(f(x)) y \leq 0$ and $\ln(f)$ is quasi-concave.

<u>We assume that $\ln(f)$ is quasi-concave, we want to show that f is quasi-concave:</u>

We assume $\forall (x,y),\ y^T \nabla \ln(f(x)) = 0 \Rightarrow y^T \nabla^2 \ln(f(x)) y \leq 0$. Using the expression of $\nabla \ln(f(x))$ and $\nabla^2 \ln(f(x))$, this condition can be rewritten as follows:

$$\forall (x,y),\ y^T \nabla f(x) = 0 \Rightarrow y^T \nabla^2 f(x) y \leq 0$$

And this proves that f is quasi-concave. □

In the following, we assume that the pdf φ^p of the free flow pace is strictly log-concave, and thus so is the pdf $\varphi^y_{x_1,x_2}$ of the free flow travel time between location x_1 and x_2. Note that most common probability distributions (*e.g.* Gaussian or Gamma with shape greater than 1) are log-concave.

Fact 7. In one dimension, a strictly log-concave probability distribution function φ defined on $\mathcal{D}_\varphi \subset \mathbb{R}$ has a unique critical point $y_c \in \overline{\mathcal{D}_\varphi}$. On its domain, φ is strictly increasing for $y \leq y_c$ and strictly decreasing for $y \geq y_c$

This result comes from the fact that log-concavity implies quasi-concavity (see [78]). Note that we allow y_c to be at the bounds of the domain $\mathcal{D}_f$. If $\exists a$ such that $\forall x \in (a, +\infty), \varphi(x) > 0$, then hi is either strictly decreasing or has a unique critical point (reasoning by contradiction and using the integrability of φ). Similarly, if $\exists b$ such that $\forall x \in (-\infty, b), \varphi(x) > 0$, then φ is either strictly increasing or has a unique critical point.

Proof of the quasi-concavity of the travel time probability distribution in the undersaturated regime

The goal of this section is to prove that the undersaturated travel time probability distribution function is a quasi-concave function. Let Δ denote the maximum delay experienced (i.e. the red time R) and η the fraction of delayed vehicles (previously denoted $\eta^u_{L,0}$). The length of the link L is a scale parameter that does not change the concavity properties of the function. For notational simplicity, we denote φ the pdf of travel times and omit the locations $x_1 = L$ and $x_2 = 0$ in this section. Recall the travel time distribution on an undersaturated link:

$$g^u(y) = (1-\eta)\varphi(y) + \frac{\eta}{\Delta} \int_{y-\Delta}^{y} \varphi(z)\, dz$$

with the convention $\varphi(z) = 0$ for $z \leq 0$.

The function g^u is continuously differentiable on $\mathbb{R}^+$ and $\forall\, y \in \mathbb{R}^+$ we have:

$$(g^u)'(y) = (1-\eta)\varphi'(y) + \frac{\eta}{\Delta}(\varphi(y) - \varphi(y-\Delta)). \tag{106}$$

The function $(g^u)'$ is continuously differentiable on $\mathbb{R}^+$ and $\forall\, y \in \mathbb{R}^+$ we have:

$$(g^u)''(y) = (1-\eta)\varphi''(y) + \frac{\eta}{\Delta}(\varphi'(y) - \varphi'(y-\Delta)) \tag{107}$$

Using the expression of $(g^u)'(y)$, we have

$$(g^u)'(y) = 0 \Leftrightarrow (1-\eta)\varphi'(y) = \frac{\eta}{\Delta}(\varphi(y-\Delta) - \varphi(y)). \tag{108}$$

Our goal is to prove that $(g^u)'(y) = 0 \Rightarrow (g^u)''(y) \leq 0$, so let y be such that $(g^u)'(y) = 0$.

<u>Case 1: $\varphi'(y) > 0$</u>

Using Fact 7, we know that φ is strictly increasing on $(-\infty, y]$. Thus $\varphi(y-\Delta) < \varphi(y)$. Plugging back into (106), we prove that $\varphi'(y) > 0 \Rightarrow (g^u)'(y) > 0$ which contradicts the hypothesis $(g^u)'(y) = 0$.

<u>Case 2: $\varphi'(y) \leq 0$</u>

From (107) and the log concavity of φ we have

$$\begin{aligned}
(g^u)''(y) &\leq (1-\eta)\frac{(\varphi'(y))^2}{\varphi(y)} + \frac{\eta}{\Delta}(\varphi'(y) - \varphi'(y-\Delta)) \\
\text{Using (108) we replace } & (1-\eta)\varphi'(y) \text{by} \frac{\eta}{\Delta}(\varphi(y-\Delta) - \varphi(y)) \\
&= \frac{\eta}{\Delta}\left(\frac{(\varphi'(y))}{\varphi(y)}(\varphi(y-\Delta) - \varphi(y)) + \varphi'(y) - \varphi'(y-\Delta)\right) \\
&= \frac{\eta}{\Delta}\left(\frac{(\varphi'(y))}{\varphi(y)}\varphi(y-\Delta) - \varphi'(y-\Delta)\right)
\end{aligned}$$

Moreover, equation (108) and the condition $\varphi'(y) \leq 0$ imply that $\varphi(y-\Delta) \leq \varphi(y)$. Reasoning by contradiction, we assume that $\varphi'(y-\Delta) \leq 0$. From Fact 7, we know that $\varphi'(y-\Delta) \leq 0$ implies $\varphi(y-\Delta) > \varphi(y)$, which contradicts the assumption of Case 2. Thus necessarily, we have $\varphi'(y-\Delta) \geq 0$ and plugging into (107), $(g^u)''(y) \leq 0$.

We conclude that $(g^u)'(y) = 0 \Rightarrow (g^u)''(y) \leq 0$. From the definition of quasi-concavity (Definition 2), we conclude that $g^u(y)$ is quasi-concave.

Proof of the quasi-concavity of the travel time probability distribution in the congested regime

The goal of this section is to prove that the congested travel time probability distribution function is a quasi-concave function[14]. Let $\delta_{\min}$ (resp. $\delta_{\max}$) denote the minimum (resp. maximum) delay experienced. Recall the travel time probability distribution on a congested link:

$$g_c(y) = \frac{1}{\delta_{max} - \delta_{min}} \int_{y-\delta_{\max}}^{y-\delta_{\min}} \varphi(y)\, dy$$

The function g_c is continuously differentiable on $\mathbb{R}^+$ and $\forall\, y \in \mathbb{R}^+$ we have:

$$g'_c(y) = \frac{1}{\delta_{max} - \delta_{min}} (\varphi(y - \delta_{\min}) - \varphi(y - \delta_{\max})).$$

We prove that there exists an interval I such that $y \notin I \Rightarrow g'_c(y) \neq 0$. From the characterization of quasi-concave function given in Lemma 3, we conclude that g_c is quasi-concave.

Referring to Fact 7, we note y_c the critical point of the pace distribution φ. We have:

- For $y \in [0, y_c + \delta_{\min}]$, we have $y - \delta_{\max} < y - \delta_{\min} \leq y_c$. Thus $\varphi(y - \delta_{\max}) < \varphi(y - \delta_{\min})$ and $g'_c(y) > 0$.
- For $y \in [y_c + \delta_{\max}, +\infty]$, we have $y_c \leq y - \delta_{\max} < y - \delta_{\min}$. Thus $\varphi(y - \delta_{\max}) > \varphi(y - \delta_{\min})$ and $g'_c(y) < 0$.
- For $y \in [y_c + \delta_{\min}, y_c + \delta_{\max}]$, we have $y - \delta_{\max} \leq y_c \leq y - \delta_{\min}$. For all $y \in [y_c + \delta_{\min}, y_c + \delta_{\max}]$, $y - \delta_{\max} \leq y_c$ and thus the function $y \mapsto \varphi(y - ẹlta_{\max})$ is strictly increasing on $[y_c + \delta_{\min}, y_c + \delta_{\max}]$. Similarly, for all $y \in [y_c + \delta_{\min}, y_c + \delta_{\max}]$, $y - \delta_{\max} \geq y_c$ and the function $y \mapsto \varphi(y - \delta_{\min})$ is strictly decreasing on $[y_c + \delta_{\min}, y_c + \delta_{\max}]$. The function g'^c is strictly decreasing on $[y_c + \delta_{\min}, y_c + \delta_{\max}]$. Moreover $g'^c(y_c + \delta_{\min}) > 0$ and $g'^c(y_c + \delta_{\max}) < 0$. Using the monotonicity of g'^c on $[y_c + \delta_{\min}, y_c + \delta_{\max}]$ and the theorem of intermediate values, we show that g'_c is equal to zero in a unique point on $[y_c + \delta_{\min}, y_c + \delta_{\max}]$.

[14]We will show in Section 3 that the pdf is log-concave, but this involves results on log-concave functions that are not trivial to prove.

The function g_c has a unique critical point (unique point where g'^c equals zero). From the characterization of quasi-concavity given in Lemma 3, we conclude that $g_c(y)$ is quasi-concave. Note that we have also proven *strict* quasi-concavity, the critical point of g_c is unique.

17.4.3. Log-concavity properties of the different components of the mixture model

For any locations x_1 and x_2 and any regime (undersaturated or congested) the probability distribution of travel times is a mixture distribution. Each component of the mixture, denoted $\psi_i(y_{x_1,x_2})$, is the probability distribution of travel times associated with a delay with either a mass or a uniform probability. In this section, we prove that each of the component is log-concave. For each of the component ψ_i, one of the following statement is true:

$\exists \Delta \geq 0$ such that $\psi_i(y_{x_1,x_2}) = \mathbf{1}_{\{\Delta\}} * \varphi^y_{x_1,x_2}(y_{x_1,x_2})$,

$\exists \delta_{\min} \geq 0$, $\delta_{\max} > \delta_{\min}$ such that $\psi_i(y_{x_1,x_2})$

$$= \frac{1}{\delta_{max} - \delta_{min}} \mathbf{1}_{[\delta_{\min}, \delta_{\max}]} * \varphi^y_{x_1,x_2}(y_{x_1,x_2})$$

To conclude on the log-concavity of each component, we use the following facts:

Fact 8. *Integration of log-concave functions [275,276]* If $f: \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is a log-concave function, then $g(x) = \int_{\mathbb{R}^m} f(x,y)\, dy$ is a log-concave function of x on $\mathbb{R}^n$.

In particular, log-concavity is closed under convolution.

Fact 9. *Log-concavity is closed under convolution [78]* If f and g are log-concave on $\mathbb{R}^n$, then so is the convolution h of f and g, $h(x) = \int_{\mathbb{R}^n} f(x-y) g(y)\, dy$. Indeed the function $(x,y) \mapsto f(x-y)g(y)$ is log-concave as the product of two log-concave functions (log-concavity is closed under the multiplication since concavity is closed under summation). The result follows from Fact 8.

We have written each component of the mixture distribution as a convolution between (i) a Dirac distribution and the pdf of free flow travel time or (ii) a Uniform distribution (constant function on a convex interval) and the pdf of free flow travel times. Under the assumption that the probability distribution of pace is log-normal, so is each component of the mixture.

Remark that when the delay distribution has a mass probability, this result can also be derived by noticing that the probability distribution of travel times is a translation of the probability distribution of free flow travel times.

For any locations x_1 and x_2 on a link and any congestion regime (undersaturated or congested), we have derived an expression for the probability distribution of travel times. These probability distributions are mixture distributions. Each of the component is the convolution between the probability distribution of free flow travel time and either a mass probability or a uniform probability distribution. We have proven that the link travel times are quasi-concave for both the undersaturated and the congested regime. Moreover, for any locations x_1 and x_2 on a link and any congestion regime, the probability distribution of travel times is a mixture of log-concave probability distributions.

17.5. Conclusion

This report presents the application of traffic flow theory to the construction of a statistical model of arterial traffic conditions based on standard assumptions in transportation engineering. In particular, we assume that time can be discretized into periods of stationary conditions and we then study the traffic dynamics for the duration of one period.

The model validates the intuition that the average density of vehicles is higher close to the end of the links because of the presence of traffic signals. We provide analytical derivations for the average density and spatial distribution of vehicles on a link, parameterized by traffic parameters. When probe vehicles send their location periodically in time, this model is used to learn traffic conditions via the estimation of queue length.

Under similar traffic conditions, the delay experienced by vehicles depends on the time at which they enter the link. Assuming uniform arrivals, we compute the probability distribution of delays among the vehicles entering the link in a cycle to model the differences in delay experienced by the vehicles. We show that the delay distribution is a finite mixture distribution, where each mixture corresponds to an interval of arrival times. Each mixture distribution corresponds to a delay with either a mass ora uniform distribution.

Under free-flow conditions, vehicles may have different driving behavior. We model this fact by considering the free flow pace as a random variable with a specific distribution. We use the model of driving behavior and the probability distribution of delays to derive the probability distribution of travel times between any two locations on an arterial link. We show that the probability density functions of link travel times are quasi-concave and that the probability distributions of travel times between any two arbitrary location on the link are mixtures of log-concave distributions. The probability distributions of travel times between arbitrary locations are parameterized by the traffic signal parameters (red time and cycle time), the driving behavior, the queue length and the queue length at saturation. When probe vehicles send pairs of successive locations, we can compute their travel times to go from one location to the next. As we receive data from different probes, we can estimate the parameters that maximize the probability of receiving the observations. This modeling approach is used and developed further in subsequent work to produce accurate arterial traffic estimates and short-term forecast [187]. It can also be used with historical data to estimate the parameters of the network (parameters of the traffic signals, queue length at saturation). The concavity properties of the travel time distributions are used to guarantee global optimality of a sub-problem of the estimation algorithm.

18. Traffic Models for Arterial Estimation and Prediction

This chapter presents models for estimating arterial traffic conditions using GPS probe data as the only source of data. Two different data collection methodologies are considered as input into three different estimation models. The model presented in section 1 uses a VTL-based system as the only source of input data (see section 8 for a description of the VTL). This model is based on regression techniques considered standard in the statistics community and presents two variants, one with a discrete representation of the state of traffic and one with a continuous representation. The models presented in sections 2 and 3 use sparsely-sampled GPS probe data from fleets (see section 9). The sparsely-sampled GPS data format is more general than travel times from VTLs (since the two VTLs can just be considered to be the start and end points of each of the sparsely sampled vehicle trajectories), so data from VTLs can also be easily incorporated into the models presented in sections 2 and 3. The first of these two models focuses on how to allocate travel time data to the different links traveled between GPS observations. The last model is a graphical model (common in the machine learning community, see section 1), which is a much more generic and flexible framework than the other models.

18.1. Regression Models

This section introduces the use of regression models for estimating *Level of Service* (LoS) indicators which are the aggregate travel times and congestion states for an arterial road network. After stating the general assumptions of the regression models (section 1), the general problem of sensing on a graph (section 2) is presented followed by the formal definitions of LoS indicators (section 3). The problem description of estimating the LoS indicators based on STARMA and logistic regression is presented in section 4. The algorithms to solve these models are presented later in chapter 19.

18.1.1. Assumptions

There are a few key general assumptions that enable the regression models to be computationally tractable. First, it is assumed that there is a VTL-based infrastructure collecting travel time data

between all adjacent pairs of VTLs on the road network. The VTL pairs are the fundamental building block for these models. All data and model estimates are assumed to be in the form of travel times for each VTL pair. Additionally, the models estimate the average travel time for a discrete time interval and do not estimate the probability distribution of travel times. The second general assumption is that there exist a discrete number of congestion states for each VTL pair and that this number of states is the same for all VTL pairs. Other technical assumptions are made in the context of deriving the models and will be presented when needed.

18.1.2. Graph Model of the Road Network

Consider an arterial network with a total of N pairs of VTLs deployed. Each pair has a unique identification number $i \in \{1, \dots, N\}$. The set of all VTL pairs is denoted by $\mathcal{V} = \{1, \dots, N\}$. Each VTL pair has a segment of road in between with a possibility of one or more road features such as an intersection (with or without traffic lights), pedestrian walkways, stop/slow signs etc. The characteristics of these road features can be static (such as presence of a stop sign) or dynamic (such as phase of a signalized intersection) with respect to time. The travel time experienced by a vehicle traveling through a VTL pair depends on the characteristics of the road features as well as the demand-capacity restrictions imposed by the dynamics of traffic flow.

The upstream (resp. downstream) VTL for the pair i is the VTL at which the traffic enters (resp. leaves) the corresponding stretch of road. For pair i, let the upstream and downstream VTLs be denoted by i_u and i_d, respectively. The VTL sensor network can be represented as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of all VTL pair as defined earlier and $\mathcal{E}$ is the set of all edges. Two VTL pairs i and j form an edge directed from pair i to pair j, denoted e_{ij}, if i_d and j_u correspond to same VTL. Then i (resp. j) is called the upstream (resp. downstream) node of edge e_{ij}.

Define the set of first order neighbors for VTL pair j as

$$\mathcal{N}^1(j) = \{j\} \cup \{i \in \mathcal{V}: e_{ij} \in \mathcal{E}\} \cup \{\kappa \in \mathcal{V}: e_{j\kappa} \in \mathcal{E}\}$$

which is simply the set of all the upstream and downstream VTL pairs for the pair j (in which pair j itself is included).

The above definition can be extended to define n^{th} $(n \geq 1)$ order neighbors as:

$$\begin{cases} \mathcal{N}^0(j) & = & \{j\} \\ \mathcal{N}^n(j) & = & \mathcal{N}^{n-1}(j) \cup \left(\bigcup_{l \in \mathcal{N}^{n-1}(j)} \{i \in \mathcal{V}: e_{il} \in \mathcal{E}\} \cup \{\kappa \in \mathcal{V}: e_{l\kappa} \in \mathcal{E}\} \right) \end{cases} \tag{109}$$

18.1.3. Traffic Level of Service Indicators

It is assumed that for any VTL pair $i \in \mathcal{V}$, the travel time data is available at times $0 \leq t_1 \leq t_2 \leq \cdots$. As an alternative representation to travel time data, the pace, can also be used (travel time divided by the length of road for the VTL pair). The data obtained at time t_1 for VTL pair i is denoted $X_{t_1,i}$ (i.e. the travel time or pace of a vehicle traversing VTL pair i starting at time t_1).

Since the data obtained is event-based, it cannot be directly used for training statistical models that needs regular sampling rates (i.e. one quantity per discrete time step). To address this, the travel time data is aggregated in t second windows to obtain a time series of observations at times $k = 0, t, 2t, \ldots$. Here t is the aggregation interval. Henceforth, k is used to denote the time interval $[(k-1)t, kt)$. The set of available observations during the time period k for any VTL pair i is denoted as $A_{k,i}$, that is,

$$A_{k,i} = \{X_{t_m,i} \mid (k-1)t \leq t_m < kt\}.$$

Define the spatial aggregation function for VTL pair i, $h_i(\cdot): A_{k,i} \mapsto [0, \infty)$, as the function that aggregates the set of observations $A_{k,i}$ in to an *aggregate representative quantity*, denoted $Z_{k,i}$. In the remainder of this section, $Z_{k,i}$ is an aggregated travel time (seconds). Thus, the aggregate travel time for VTL i during interval k is

$$Z_{k,i} = h_i(\{X_{t_m,i} \mid (k-1)t \leq t_m < kt\}).$$

The *mode* of a VTL pair is defined as the categorical variable indicative of the extent of delay experienced in navigating through the VTL pair. For example, a binary mode classification can be *uncongested or congested*. Thus, the mode of a VTL pair can also interpreted as a *congestion state*. Let the mode of VTL pair i during time interval k be denoted as $Q_{k,i}$. In order to convert the total number of observations available for VTL pair i during time interval k into a congestion state, a *congestion indicator function* is defined as $g_i(\cdot): A_{k,i} \mapsto \{1, \ldots, M\}$ where M is the number of congestion states consider. M is a meta-parameter of the model that is chosen based on a preliminary analysis of the data for the site under consideration. Several values of M can be chosen to see which fits best. With these definitions, $Q_{k,i}$ is determined as

$$Q_{k,i} = g_i(\{X_{t_m,i} | (k-1)t \leq t_m < kt\}).$$

From a statistical modeling perspective, both the aggregate speed or travel time, $Z_{k,i}$, and the congestion state, $Q_{k,i}$, for $i \in \mathcal{V}$ and $k \in \{0,1, \ldots\}$ can be considered as random processes

generated by space-time varying traffic flow phenomena on the arterial network. Both $Q_{k,i}$ and $Z_{k,i}$ can be regarded as LoS indicators.

18.1.4. Estimating Level of Service Indicators

If data was available from all the vehicles for all the VTL pairs over the entire time horizon of interest, the entire probability distribution of $Z_{k,i}$ and $Q_{k,i}$ could be computed directly. However, the challenge of arterial traffic state estimation and forecast is that the typical penetration rates are very low. The focus here is to develop reliable estimation and forecasting methods for such situations.

Typically only a small percentage of the total number of vehicles provide data to the system. Thus, the choice of the aggregation function $h_i(\cdot)$ (resp. the congestion indicator function $g_i(\cdot)$) becomes critical to obtain reliable estimates of $Z_{k,i}$ (resp. $Q_{k,i}$). For a given choice of $h_i(\cdot)$, the best estimate of the aggregate travel time or speed for VTL pair i during interval k is given by the conditional expectation of $Z_{k,i}$ given the aggregate travel times up-to (and excluding) the current time interval:

$$\hat{Z}_{k,i} = \mathbb{E}[h_i(A_{k,i})|h_j(A_{j,v}), j < k, v \in \mathcal{V}] = \mathbb{E}_{h_i}[Z_{k,i}|Z_{j,v}, j < k, v \in \mathcal{V}],$$

where $\mathbb{E}_{h_i}[\cdot]$ is used to indicate the dependence of the expectation on the aggregation function h_i. It is assumed that $Z_{k,i}$ is conditionally independent of all other data conditioned on the data from the past r time intervals for VTL pairs in the set $\mathcal{N}^s(i)$. Under this assumption, $\hat{Z}_{k,i}$ can be rewritten as

$$\hat{Z}_{k,i} \approx \mathbb{E}_{h_i}[Z_{k,i}|Z_{j,v}, k - r \leq j < k, v \in \mathcal{N}^s(i)] \tag{110}$$

Thus, $\hat{Z}_{k,i}$ only depends on data with r *temporal dependencies* in the past and s *spatial dependencies* from the neighbors. Similarly, for given choices of the aggregation function $h_i(\cdot)$ and the congestion indicator function $g_i(\cdot)$, the conditional expectation of $Q_{k,i}$ given all the aggregate travel times up-to (and excluding) the current time interval is

$$\hat{Q}_{k,i} = \mathbb{E}[g_i(A_{k,i})|h_j(A_{j,v}), j < k, v \in \mathcal{V}] \approx \mathbb{E}_{h_i,g_i}[Q_{k,i}|Z_{j,v}, k - r \leq j < k, v \in \mathcal{N}^s(i)], \tag{111}$$

In the statistics terminology, the quantities $Z_{k,i}$ and $Q_{k,i}$ in (110) and (111) are known as the response variables; the conditioned variables $Z_{j,v}$ and $Q_{j,v}$ are called the dependent variables

or covariates.

This section compares two estimators. The first estimator is based on expressing (110) as a *linear regression problem*. For a temporal and spatial dependence of orders r and s respectively, a linear dependence of response $Z_{k,i}$ on the covariates $Z_{j,v}$ is assumed:

$$\hat{Z}_{k,i} = \beta_i^0 + \sum_{v \in \mathcal{N}^s(i)} \left(\sum_{j=k-r}^{k-1} \beta_i^{j,v} \, Z_{j,v} \right). \tag{112}$$

In order to make the notation concise, let $\boldsymbol{Z}_{k,i}^{r,s}$ be the $r \times \mathcal{N}^s(i)$ vector of covariates or dependent variables obtained by stacking the aggregate travel times $Z_{j,v}$ for $k - r \leq j < k$ and $v \in \mathcal{N}^s(i)$, β_i be the corresponding $r \times \mathcal{N}^s(i) + 1$ vector of parameters to be estimated. Then the equation (112) can be rewritten as

$$\hat{Z}_{k,i} = \beta_i^\top \, \boldsymbol{Z}_{k,i}^{r,s}.$$

As described later in section 2, instead of a simple regression model (112), a STARMA model is used, which is an extended version of the simple regression model.

The second estimator is based on expressing (111) as a logistic regression problem which assumes a linear dependence of the *logit* or the log-odds ratio of conditional expectation $\hat{Q}_{k,i}$ on the response variables. That is, for a temporal and spatial dependence of orders r and s respectively,

$$\log \left(\frac{\hat{Q}_{k,i}}{1 - \hat{Q}_{k,i}} \right) = \beta_i^\top \, \boldsymbol{Z}_{k,i}^{r,s}.$$

This equation can be expressed as

$$\hat{Q}_{k,i} = f_{\beta_i}(\boldsymbol{Z}_{k,i}^{r,s}) := \frac{1}{1 + exp\left(-\beta_i^\top \, \boldsymbol{Z}_{k,i}^{r,s}\right)}, \tag{113}$$

where the subscript β_i in $f_{\beta_i}(\cdot)$ encodes the dependence on the β_i.

The implementation of a logistic regression estimator is detailed in section 1 and the STARMA-based estimator in section 2. However, two important points need to be mentioned here. First, the above formulation can be modified to include the case of multiple steps forecast. For example, an m −step forecast at time k for VTL pair i can be written as

$$\hat{Z}_{k+m,i} = \mathbb{E}_{h_i}[Z_{k+m,i} | Z_{j,v}, j \leq k, v \in \mathcal{V}], \tag{114}$$

where data up to time k is used to predict traffic at time $k + m$.

Second, note that for some VTL pairs and time intervals, there may not be any available data, that is, $A_{j,v} = \emptyset$ for some $j \in \{k-r, \dots, k\}$ and $v \in \mathcal{N}^s(i)$. In this case, one has to employ a technique of *estimation with missing data*. The forecast problem is addressed for the STARMA model but the issue of missing data is something that is handled by the more robust graphical model approach presented in section 3.

18.2. Historic modeling using Bayesian inference for real-time estimation

In this section, an *independent link travel time* model of arterial traffic is proposed. This model learns the historic traffic patterns of the network and uses these historic patterns (section 1) as prior information to be used in a Bayesian real-time estimation model (section 2). The assumptions of the model are as follows:

- The travel time distribution for each link of the network is independent from all other links of the network. The set of links of the network is denoted $\mathcal{L}$.
- Any given moment in time belongs to exactly one *historic time period*, during which traffic conditions are assumed constant. The set of historic time periods are denoted $\mathcal{T}$. All of the travel time observations from a specific link l are independent and identically distributed with a given time period, $t \in \mathcal{T}$.
- Sparse probe measurements are the only data available to the model (see section 9).

18.2.1. Historic Model of Traffic

The historic model of arterial traffic estimates the average travel time as well as the standard deviation of travel time within a given historic time period (i.e. Mondays 4pm-5pm, 5pm-6pm, etc.) for each link of the network. The model parameters for link l and time period t are denoted $Q_{l,t}$. The set of all link parameters for a given time period is denoted $\boldsymbol{Q}_t$. The link travel time probability density function for link l during time period t is denoted $g_{Q_{l,t}}(y)$ for a given travel time y.

The observations available to the estimation model are assumed to be in the form of *path observations*. The pth path observation, y_p, is defined as a set of consecutive links traveled, L_p, through the network along with the fraction of the first and last link traversed as well as the total travel time associated with the entire path. The fraction of link l traversed for the pth observation on link l is denoted $w_{p,l}$. The set of path observations for time period t are denoted $mathbf P_t$. For the pth path observation, the path travel time distribution is denoted $G_{P_{L_{p,t}}}(y_p)$, which is the convolution of the link travel time distributions that make up the path, where $P_{L_{p,t}}$ denotes the parameters of the L_p links along the pth path observation.

The goal is to determine the values of $Q_{l,t}$ for each link and time period that are most consistent with the probe data received. This is achieved by maximizing the likelihood of the data given the parameters, which is written

$$\arg\max_{Q_t} \sum_{p \in \boldsymbol{P}_t} \ln(\quad), \tag{115}$$

where N_t is the number of observations available for time period t. This optimization problem is challenging due to the high number of variables (number of links times number of parameters per link travel time distribution) and it is not directly decomposable. In section 1, an intuitive decomposition scheme is presented for finding near-optimal solutions to this optimization problem. Given the assumption that each time period is independent, equation (115) can be solved for each value of $t \in \mathcal{T}$ separately.

18.2.2. Real-Time Estimation

The parameters learned by the historic model (section 1) are used as priors to estimate current traffic conditions via a *Bayesian update* (see [287] for more on Bayesian statistics). The process of doing a Bayesian update maximizes the likelihood of the current data given the prior distribution, with the assumption that the travel times are normally distributed.

In the general form, given a prior $p(Q_{l,t})$ for some set of link parameters (from the historic model) and a set of measured travel times $y_{l,t}$ for that same link, the posterior function is written

$$p(Q_{l,t}|y_{l,t}) = \frac{p(y_{l,t}|Q_{l,t})p(Q_{l,t})}{\int p\,(y_{l,t}|Q_{l,t})p(Q_{l,t})dQ_{l,t}}. \tag{116}$$

The goal is then to choose $Q_{l,t}$ that maximizes $p(Q_{l,t}|y_{l,t})$, which is dependent upon the specific distributions chosen for the model. The details of how to compute the Bayesian update are left for chapter 19. The specifics regarding the real-time estimation step are found in section 2.

18.3. Probabilistic Graphical Model

The modeling approaches presented in the first two sections of this chapter made a number of strong assumptions. The goal of the model presented in this section removes some of these assumptions and presents a modeling framework that is flexible and extensible. The key features that this model possesses are:

- Each link has a discrete traffic state that cannot be directly observed.
- Traffic states of nearby links are correlated and evolve over time in a Markov manner (i.e. the future is independent of the past given the present).
- Expectation maximization is an appropriate tool for learning the transition and observation model parameters.

While some assumptions are still made for computational tractability, these assumptions can gradually be removed as more advances are made on this topic.

18.3.1. Assumptions

The graphical model presented in this section makes the following assumptions:

1. *Discrete congestion states*: for each day d and each time interval t, the traffic conditions on link l are represented by a *discrete* value, $s_{d,t}^{l}$, which indicates the level of congestion. There are S discrete levels of congestion.
2. *Conditional independence of link travel times*: conditioned on the state $s_{d,t}^{l}$ of a link l, the travel time distribution of that link is independent from all other traffic variables.
3. *Conditional independence of state transitions*: conditioned on the states of the spatial neighbors of link l of order n (denoted $\boldsymbol{N}_n^l$) at time t, the state of link l at time $t+1$ is independent from all other current link states, all past link states and all past travel time observations.

$\boldsymbol{N}_n^l$ denotes the spatial neighbors of link l of order n, where first order neighbors ($\boldsymbol{N}_1^l$) are the links sharing an intersection with link l (including link l). The higher order spatial neighbors are defined by the following recursive formula:

$$\boldsymbol{N}_{n+1}^l = \bigcup_{j \in \boldsymbol{N}_n^l} \boldsymbol{N}_1^j \tag{117}$$

Assumption 2 implies that link travel times are not correlated across links, which is an assumption made for computational tractability. Assumption 3 implies that each link is correlated with some (small) subset of neighboring links, but independent of the rest of the network. Neither of these assumptions must hold all of the time in a real traffic network, but some approximation is necessary for computational tractability of the model. The full effect of these assumptions has not been studied to date, but is necessary for full validation of this model.

18.3.2. Graphical Model

Arterial traffic conditions vary over space and time. Given the assumptions in section 1, the spatio-temporal conditional dependencies of arterial traffic are modeled using a probabilistic graphical model known as a *Coupled Hidden Markov Model* (CHMM) [79]. A *Hidden Markov Model* (HMM) is a statistical model in which the system being modeled is assumed to be a Markov process with unobserved states. CHMMs model systems of multiple interacting processes. In the present case, the multiple processes evolving over time are the discrete *states* (assumption 1) of each link in the arterial network. The discrete state of each link l and time period t is denoted $Z_{l,t}$. Since the state of each link for all times is not directly observed, these processes are considered *hidden*. The travel time distribution on each link is conditioned on its

hidden state (assumption 1) from which come sparse observations from probe vehicles traveling through the arterial network. The observations for link l and time period t are denoted by the set $\boldsymbol{y}_{l,t}$. Assumption 1 gives the *coupled* structure to the HMM by specifying local dependencies between adjacent links of the road network. Figure 3.1 illustrates our model representation of link states and probevehicle observations. Each circular node in the graph represents the state of a link in the road network. The state is a discrete quantity defined based on the application (e.g. the possible states could be undersaturated/congested or the number of vehicles in the queue). The forward arrows indicate the local spatial dependency of links from one time period to the next. Each square node in the graph represents observations on the link to which it is attached (e.g. travel time from probe vehicles, flow data from loop detectors if available, etc.).

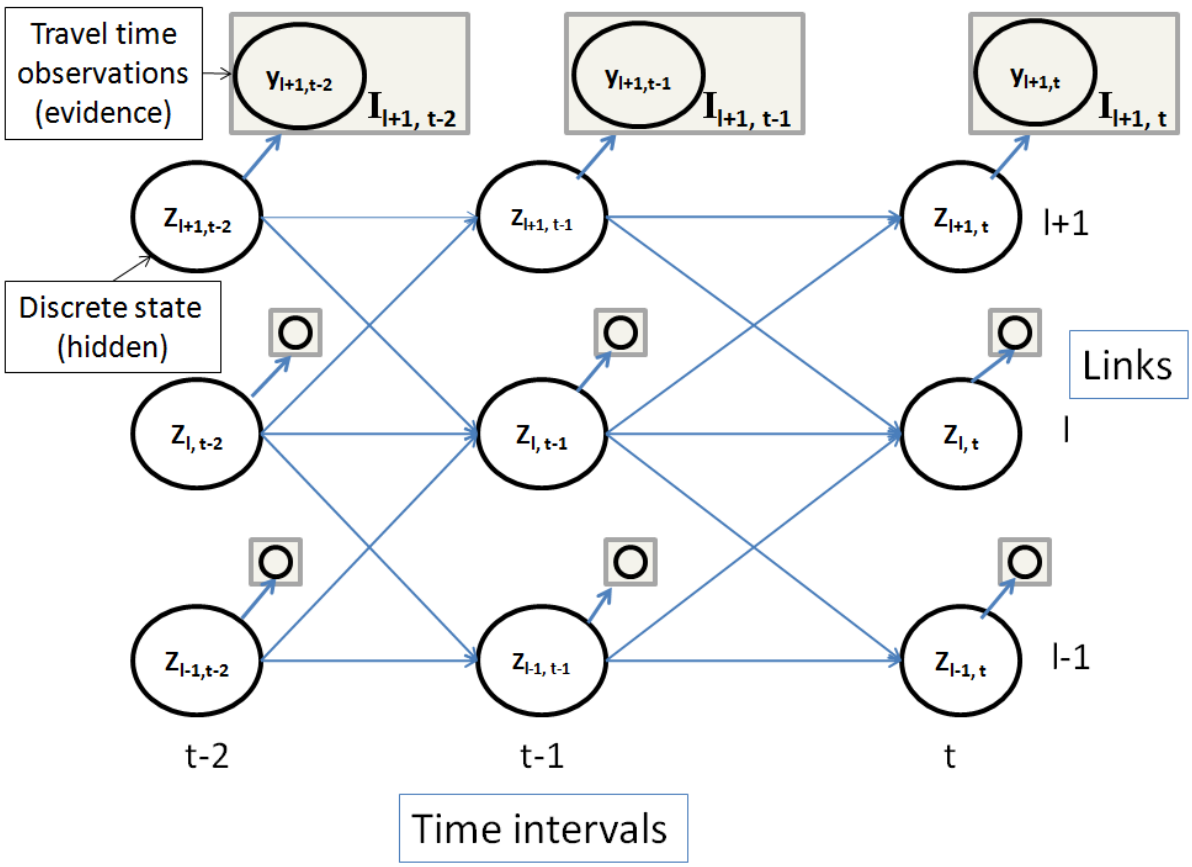


Figure 3.1. Spatio-temporal model of arterial traffic evolution represented as a coupled hidden Markov model. The circular nodes represent the (hidden) discrete state of traffic for each link at each time interval, denoted $\boldsymbol{Z}_{l,t}$. The square nodes represent travel time observations from the distribution defined by the traffic state, denoted $\boldsymbol{y}_{l,t}$.

To completely specify the CHMM-based model, it is necessary to estimate (i) the initial state probabilities for each link, denoted $\pi_{l,s}$, (ii) the discrete transition probability distribution functions (assumption 3), denoted $A_{l,t}$, and (iii) the distribution of travel time on a link given the state of that link (assumption 2), denoted $g_{l,s,t}$. If data sources other than link travel times are available, then additional observation nodescan be attached to the hidden state nodes. They

would be incorporated into the model by defining a probability distribution function for the data source given the hidden states.

For each link l and each time interval t, the probability of link l to be in state s at time $t+1$ given the state of its neighbors at time t is given by the *discrete transition probability distribution* function of link l. It is fully characterized by a matrix of size $S^{N_n^l} \times S$, denoted $A_{l,t}$. The element of line r and column s, $A_{l,t}(r,s)$, represents the probability of link l to be in state s at time $t+1$ given that the neighbors of l are in state$r$ at time t.

A simplifying assumption for computational tractability is to assume that for each link l, the state transition matrix $A_{l,t}$ and the conditional travel time distribution function $g_{l,s,t}$ do not depend on time. They are denoted respectively by A_l and $g_{l,s}$ in the remainder of this chapter. To relax this assumption, one can assume that these functions are piecewise constant in time and estimate them for each period of time during which the stationarity assumption is satisfied. It is also assumed that, given the state of a link, the travel time distribution on that link is independent from all the other random variables. In general, travel time distributions across links are not independent (due to light synchronization, platoons, and other factors). Future work will specifically address the challenge of using correlated distributions, which have the potential to capture more complex dynamics in the arterial road network.

19. Learning and Inference Algorithms for Arterial Traffic Estimation

The models presented in chapter 18 were all designed to leverage machine learning tools for solving them. This chapter takes each of the models and provides the detailed algorithms needed for using each one to estimate real-time arterial traffic conditions. These algorithms are the core of the arterial research done and the primary building blocks for future research on this topic.

19.1. Solution Methods for Regression Models

r	Temporal aggregation level, which is an integer representing how many previous time intervals to include in producing an estimate
s	Spatial aggregation level, which is an integer that indicates what order-level neighboring VTL pairs to include for the previous time intervals in producing an estimate
$\mathcal{N}^s(i)$	The set of s order neighbors for VTL pair i
$Q_{k,i}$	Congestion state of VTL pair i for time interval k
$\boldsymbol{Z}_{k,i}^{r,s}$	Aggregate representative quantity for VTL pair i during time interval k for temporal and spatial aggregation levels r and s
Z_k	Vector of aggregate travel times for all VTL pairs for time interval k (for STARMA model)
$w_{k,i}^{(n)}$	Spatial weights of order n for $Z_{k,i}$ for VTL pair i and time interval k (for STARMA model)
β_i	Regression parameters for VTL pair i
$f_{\beta_i}(\boldsymbol{Z}_{k,i}^{r,s})$	Estimation function for $Q_{k,i}$ for VTL pair i and time interval k (for logistic regression model)
$\varphi_i^{(n)}(Z_j)$	Spatially-weighted travel time function (for STARMA model)

Table 13. Notation used in the regression algorithms.

This section demonstrates how to solve the regression models presented in section 1. The solution methods for the logistic regression (section 1) and STARMA (section 2) models are given followed by two sets of experimental results (section 3), one using simulation data and one using GPS data from an experiment in Manhattan, New York. This work is based on a previously published article [184]. A summary of the notation used for the regression algorithms is provided

in table 13.

19.1.1. Logistic Regression

Consider the estimator based on the logistic model (113) to estimate the congestion state $Q_{k,i}$ for a VTL pair i and time interval k. A reminder that the aggregate representative quantity for the incoming data for time interval$k$ on link i using temporal and spatial aggregation levels r and s is denoted $\boldsymbol{Z}_{k,i}^{r,s}$ (see section 3 for details). Suppose that $Q_{k,i}$ is binary-valued, that is $Q_{k,i} \in \{0,1\}$ and $M = 2$. $Q_{k,i} = 1$ (resp. $Q_{k,i} = 0$) corresponds to the VTL pair i during interval k being in the *congested mode* (resp. *undersaturated mode*). The estimator $\hat{Q}_{k,i}$ gives the conditional probability of the $Q_{k,i}$ given the dependent variables:

$$\hat{Q}_{k,i} = \mathbb{E}_{h_i,g_i}\left[Q_{k,i}\middle|\boldsymbol{Z}_{k,i}^{r,s}\right] = 1 \cdot \mathbb{P}_{h_i,g_i}\left[Q_{k,i} = 1\middle|\boldsymbol{Z}_{k,i}^{r,s}\right] + 0 \cdot \mathbb{P}_{h_i,g_i}\left[Q_{k,i} = 0\middle|\boldsymbol{Z}_{k,i}^{r,s}\right]$$

Now using the definition of β_i from equation (113), the conditional probability of $Q_{k,i}$ given the aggregate travel time for r temporal and s spatial dependencies is written

$$\mathbb{P}_{h_i,g_i}[Q_{k,i}|\boldsymbol{Z}_{k,i}^{r,s};\beta_i] = [f_{\beta_i}(\boldsymbol{Z}_{k,i}^{r,s})]^{Q_{k,i}}[1 - f_{\beta_i}(\boldsymbol{Z}_{k,i}^{r,s})]^{1-Q_{k,i}}$$

It is assumed that for a VTL pair i, the response process $\{Q_{k,i}\}$ and the covariate process $\{\boldsymbol{Z}_{k,i}^{r,s}\}$ is available for a number of time intervals $k = 0, \dots, K$. Introducing the conditional independence assumption that the response variable $Q_{k,i}$ is independent of all other data given $\boldsymbol{Z}_{k,i}^{r,s}$. Then the joint conditional probability of $\{Q_{k,i}\}$ given $\{\boldsymbol{Z}_{k,i}^{r,s}\}$ (also known as the conditional likelihood) can be expressed as

$$\mathbb{P}_{h_i,g_i}[\{Q_{k,i}\}_{k=0}^{K}|\{\boldsymbol{Z}_{k,i}^{r,s}\}_{k=0}^{K};\beta_i] = \prod_{k=0}^{K}[f_{\beta_i}(\boldsymbol{Z}_{k,i}^{r,s})]^{Q_{k,i}}[1 - f_{\beta_i}(\boldsymbol{Z}_{k,i}^{r,s})]^{1-Q_{k,i}}$$

For a given training data $\{Q_{k,i}\}_{k=0}^{K}$ and $\{\boldsymbol{Z}_{k,i}^{r,s}\}_{k=0}^{K}$, the *best* estimate of parameter β_i is obtained by maximizing the logarithm of the conditional likelihood which is stated explicitly as follows:

$$\mathcal{L}(\beta_i;\{Q_{k,i}\}_{k=0}^{K},\{\boldsymbol{Z}_{k,i}^{r,s}\}_{k=0}^{K}) = \sum_{k=0}^{K}\left(Q_{k,i} \cdot \beta_i^{\top}\boldsymbol{Z}_{k,i}^{r,s} - \log\left[1 + \exp\left(\beta_i^{\top}\boldsymbol{Z}_{k,i}^{r,s}\right)\right]\right)$$

The optimal estimate so obtained and denoted β_i^*, is called the *maximum likelihood estimate*

(MLE). A number of standard iterative methods, all similar to the Newton-Raphson method, can be used to obtain the MLE β_i^*. Examples of such method include Fisher scoring method and the iterative reweighted least squares. The details of the algorithm can be found in [257].

Once the parameters are learned, *validation* can be done on a similar data set as the one used to obtain β_i^*. Validation is done to assess the ability of the learned model to correctly estimate the traffic status (congestion state in this case) on previously unseen data.

19.1.2. STARMA

The STARMA model is a more efficient estimator than the simple linear regression model (112). The number of parameters to be estimated for (112), given by $r \times |\mathcal{N}^s(i)| + 1$, can increase significantly as the spatial dependency s increases. In order to explain the model, the *spatio-temporal autoregressive* (STAR) model is presented first and subsequently generalized to a full STARMA model.

Following (109), the set of n order neighbors ($0 \leq n \leq s$) for a VTL pair i can be expressed as follows

$$\mathcal{N}^s(i) = \left(\bigcup_{n=0}^{s} \mathcal{N}^n(i)\right) \backslash \mathcal{N}^{n-1}(i),$$

where $\mathcal{N}^0(i) \backslash \mathcal{N}^{-1}(i) = \{i\}$ by convention. Now, for the linear regression model (112), for any temporal order j, ($k - r \leq j < k$) and spatial order n, ($0 \leq n \leq s$), assume that

$$\text{for all } v \in \mathcal{N}^n(i) \backslash \mathcal{N}^{n-1}(i), \quad \beta_i^{j,v} \equiv \beta_i^{j,n}, \tag{118}$$

and the definition of *n-th order, spatially-weighted travel time* as

$$\varphi_i^{(n)}(Z_k) = \frac{\sum_{l \in \mathcal{N}^n(i) \backslash \mathcal{N}^{n-1}(i)} w_{k,i}^{(n)} \boldsymbol{Z}_{k,i}}{\sum_{l \in \mathcal{N}^n(i) \backslash \mathcal{N}^{n-1}(i)} w_{k,i}^{(n)}}, \tag{119}$$

where $Z_j = (Z_{j,1}, \dots, Z_{j,N})$ is the vector of aggregate travel times for all the N VTL pairs during time interval j and $w_{i,l}^{(n)}$ are the pre-defined *spatial weights of order* n for $Z_{j,l}$.

The goal of the STAR model is to predict a future $\boldsymbol{Z}_{k,i}^{r,s}$ from currently available data. Under the assumption (118) and the definition (119), the STAR model of *autoregressive* (AR) temporal order r and spatial order s is

$$\boldsymbol{Z}_{k,i} = \sum_{j=k-r}^{k-1} \sum_{n=0}^{s} \beta_i^{j,n} \varphi_i^{(n)}(Z_j) + \epsilon_{k,i} \tag{120}$$

where $\epsilon_{k,i}$ is the normally distributed error term with variance σ^2 with the properties that $\mathbb{E}[\epsilon_{k,i}] = 0$ for all k and $i \in \mathcal{V}$; and for all $i, j \in \mathcal{V}$

$$\mathbb{E}[\epsilon_{k,i}\epsilon_{k+s,j}] = \begin{cases} \sigma^2 & \text{if } s = 0 \\ 0 & \text{otherwise.} \end{cases}$$

The number of parameters to be estimated for the STAR model (120), including σ^2, is $r(s+1)+1$ which is (typically) much smaller than $r \times \mathcal{N}^s(i) + 1$ for (112). The STAR model can now be generalized to STARMAmodel of autoregressive temporal order r and spatial order s, and *moving average* (MA) temporal order p and spatial order q as[15]

$$Z_{k,i} = \sum_{j=k-r}^{k-1} \sum_{n=0}^{s} \beta_i^{j,n} \varphi_i^{(n)}(Z_j) - \sum_{j=k-p}^{k-1} \sum_{n=0}^{q} \alpha_i^{j,n} \varphi_i^{(n)}(\epsilon_j) + \epsilon_{k,i}, \tag{121}$$

where $\epsilon_j = (\epsilon_{j,1}, \dots, \epsilon_{j,N})^\top$.

Here $\alpha_i^{j,n}$ are the moving average parameters. The total number of parameters (including σ^2) to be estimated for the STARMA model (121), denoted as STARMA(r, s, p, q) are $r(s+1) + p(q+1) + 1$.

Following [272], assume that the STARMA parameters are the same for all VTL pairs, that is, $\alpha_1^{j,n} = \cdots = \alpha_N^{j,n} \equiv \alpha_{j,n}$ and $\beta_1^{j,n} = \cdots = \beta_N^{j,n} \equiv \beta_{j,n}$. Then model (112) can be vectorized for all VTL pairs $i \in \mathcal{V}$ as

$$Z_k = \sum_{j=k-r}^{k-1} \sum_{n=0}^{s} \beta^{j,n} \Phi^{(n)}(Z_j) - \sum_{j=k-p}^{k-1} \sum_{n=0}^{q} \alpha^{j,n} \Phi^{(n)}(\epsilon_j) + \epsilon_k. \tag{122}$$

where $\Phi^{(n)}(\cdot) = (\varphi_1^{(n)}(\cdot), \dots, \varphi_N^{(n)}(\cdot))^\top$ and $\epsilon_k = (\epsilon_{k,1}, \dots, \epsilon_{k,N})^\top$.

For given training data $\{Z_k\}$, $(k = 0, \dots, K-1)$, the best estimate of the parameters $A := [\alpha^{j,n}]_{p\times(q+1)}$, $B := [\beta^{j,n}]_{r\times(s+1)}$ and σ^2 is given by maximizing the conditional likelihood expressed as

$$\mathbb{P}(\{Z_k\}_{k=0}^{K-1}; A, B, \sigma^2) = (2\pi)^{-\frac{KN}{2}} |\sigma^2 \boldsymbol{I}_{KN\times KN}|^{-\frac{1}{2}} \exp\left(-\frac{S(A,B)}{2\sigma^2}\right) \tag{123}$$

where $I_{KN\times KN}$ is the identity matrix, $S(A,B) := (\epsilon_0, \dots, \epsilon_{K-1})^\top (\epsilon_0, \dots, \epsilon_{K-1})$ and according to (122), ϵ_k is written

[15]More generally, the AR spatial order s (resp. the MA spatial order q) can vary with the temporal order r (resp. p).

$$\epsilon_k = Z_k - \sum_{j=k-r}^{k-1} \sum_{n=0}^{s} \beta^{j,n}\, \Phi^{(n)}(Z_j) + \sum_{j=k-p}^{k-1} \sum_{n=0}^{q} \alpha^{j,n}\, \Phi^{(n)}(\epsilon_j).$$

The *maximum likelihood estimate* parameters, denoted A^*, B^*, are obtained by maximizing the logarithm of the conditional likelihood (123), and the corresponding σ^* is estimated by

$$\sigma^* = \sqrt{\frac{S(A^*, B^*)}{KN}}.$$

Additional technical details for the STARMA model can be found in [272].

19.1.3. Results

The results from logistic regression-based classification and STARMA-based continuous linear regression are presented here. Each algorithm is implemented and tested on simulation and field experiment data. A framework for quantifying accuracy is introduced. Results are then presented for one-step forecast, followed by multi-step forecast for the STARMA model. Additionally, a study of the effect of the *penetration rate* on the forecast accuracy is presented.

Simulation and Field Experiment Data

There are two data sets used to validate the regression models. The first set was generated from Paramics micro-simulation software. The road network modeled consists of 1,961 nodes, 4,426 links, 210 zones and is based on the SR41 corridor in Fresno, CA.The analysis presented here is for a sub-network that includes 9 arterial roads, 20 signals and 15 stop signs. Paramics simulates every car in the network. From this simulation, the position of every vehicle at one-second time intervals is extracted. This provides detailed information about speed and travel time through the network. The sub-network studied here includes 380 different links, each one of which is characterized with a specific length, a number of lanes, a direction, a speed limit and signal information. 99 VTLs were placed on different links, which corresponds to 156 different pairs of VTLs, in order to capture travel times along links and through intersections.

The second data set was obtained as part of the official *Mobile Millennium* launch demonstration in New York City at the *ITS World Congress*. Twenty drivers, each carrying a GPS equipped cell phone, drove for 3 hours (9:00am to 12:00pm) around a 2.4 mile loop of Manhattan (see figure 1.1). This number of drivers constituted approximately 2% of the total vehicle flow through the road of interest. The experiment was repeated 3 times in order to use two of the experiments as training data for the models and the other to validate the model results. The operational capabilities of the system were demonstrated at the *ITS World Congress* [14] on November 18, 2008, when live arterial traffic was displayed for conference attendees.

(a) Paramics Map (b) New York Map (c) Test Vehicle

Figure 1.1. Experiment Design. (a) Map of the Paramics network in Fresno, CA. (b) Experiment route for New York City field test used to collect data (arrows represent the direction of traffic probe vehicles). (c) Test vehicle used for the New York test.

Validation Framework

In order to compute the accuracy of the model, one needs to define the "ground truth" state of traffic. In this set of experiments, travel times are aggregated into a single value per time interval (5 minutes for Paramics, 15 minutes for the New York test). This single value per time interval is considered the true state for the interval. Determining ground truth for the logistic regression method requires classifying each time interval as congested or uncongested. The STARMA method uses the average travel time during each interval as the ground truth value. Both of these methods correspond to choosing appropriate $h_i(\cdot)$ and $g_i(\cdot)$ functions as described in section 3.

The aggregation function $h_i(\cdot)$ should capture the pattern of change in pace over different intervals to provide an aggregate quantity that is sufficiently representative of the congestion state, thus providing better accuracy in training the model and obtaining the logistic regression parameters. Based on extensive testing and simulation, it is observed that aggregating the travel times based on the entire data available in an interval fails to capture the congestion state due to the high variance of travel times when a link is congested. The probes most effected by congestion should thus have more weight in the aggregation process. A simple yet fairly effective data-driven aggregation method is as follows: given the set of observations for VTL pair i and interval k, $A_{k,i}$ is sorted such that $t_{m_1} < t_{m_2} X_{t_{m_1},i} > X_{t_{m_2},i}$, then take

$$Z_{k,i} = h_i(\{X_{t_m,i} \mid (k-1)t \le t_m < kt\}) := \frac{1}{w.M_{k,i}} \sum_{m=1}^{\lfloor w.M_{k,i} \rfloor} X_{t_m,i},$$

where $M_{k,i}$ is the number of observations in $A_{k,i}$ and $0 < w \le 1$ is the fraction of

observations used for aggregation. The symbol $\lfloor a \rfloor$ denotes the floor value of a. In words, the aggregate pace is the mean of the $100 \times w\%$ observations with highest pace or equivalently the worst observations. The simulation results for different values of w are shown in figure 1.2. From an application-driven point of view, the w that maximizes estimation accuracy is selected, in the present case $w = 0.3$. At this value, the travel time envelope of the time series of observations is best captured.

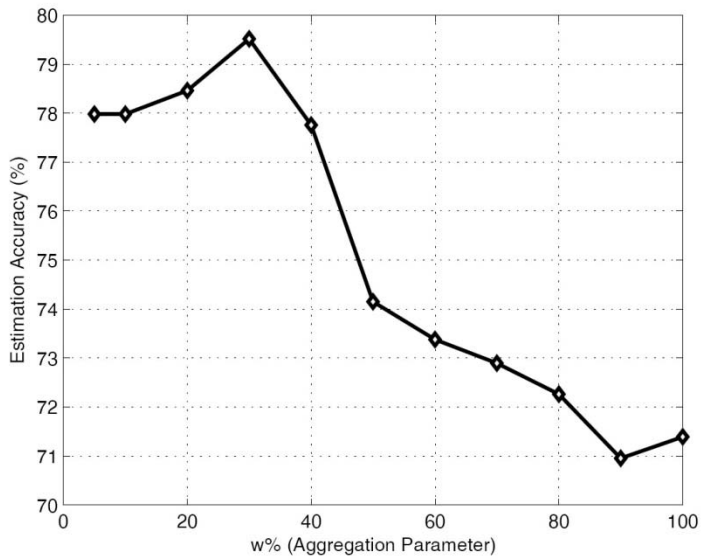


Figure 1.2. Average estimation accuracy vs. aggregation parameter w.

The training phase of logistic regression requires as input a congestion threshold along with the aggregate travel times $Z_{k,i}$. Since the congestion threshold should be chosen to be consistent with the choice of aggregate travel times to provide meaningful classification, the congestion threshold, T_i, is defined as the mean of the $100 \times w\%$ observations in D_i with highest travel time where D_i is the set of available observations in all intervals and w is essentially be the same value chosen for aggregation ($w = 0.3$ in this section). This corresponds to choosing

$$Q_{k,i} = g_i(\{X_{t_m,i} | (k-1)t \leq t_m < kt\}) = I(h_i(\{X_{t_m,i} \mid (k-1)t \leq t_m < kt\}) > T_i),$$

where $I(\cdot)$ is the indicator function. The STARMA model does not use a g_i function because it forecasts a continuous quantity.

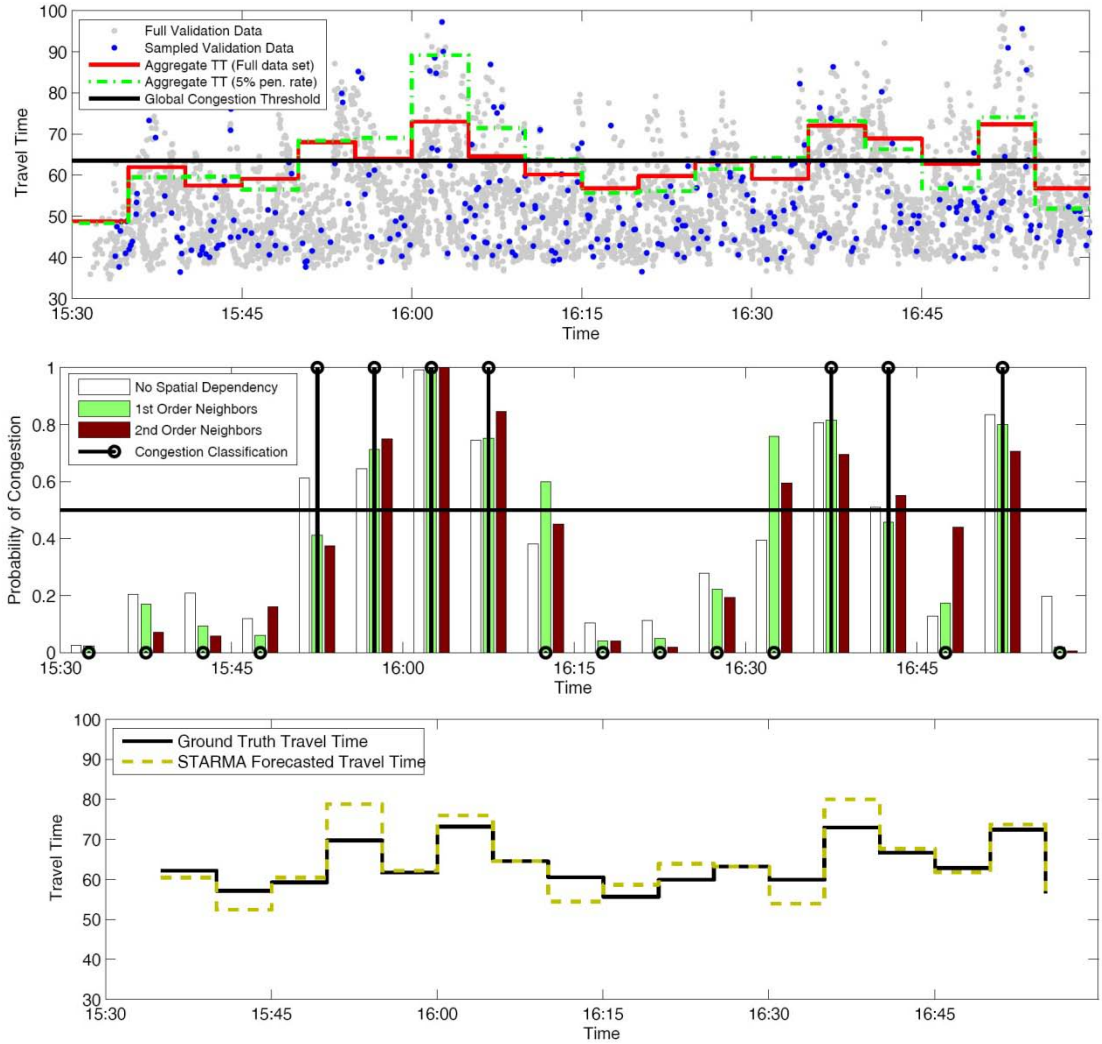
The logistic regression algorithm produces a probability of congestion for each VTL pair studied. If this probability is greater than .5, then the forecasted state is congested. The accuracy of the logistic regression forecasts is defined as the percentage of correctly forecasted states over all intervals and VTL pairs studied. For the STARMA model, the accuracy is defined as the mean percentage error between the forecasted travel time value and the actual travel time value as defined by the h_i function described earlier.

Short-Term Forecast

Both regression methods are designed to do one-step (short-term) forecasts. For each data set (as described in section 1), the performance of each model was evaluated by dividing the data set into a training set and a validation set. For the Paramics simulation data, the training set consisted of three simulation runs and the validation set consisted of a separate, fourth simulation run. For the New York experiment data, two days of data were used for training and the other day for validation. Through a-priori experimentation, the temporal dependency for the logistic regression model was set to $r = 1$ for the logistic regression, $r = 2$ for the STARMA model. The spatial dependency is varied for comparison in the result figures described in the following paragraph.

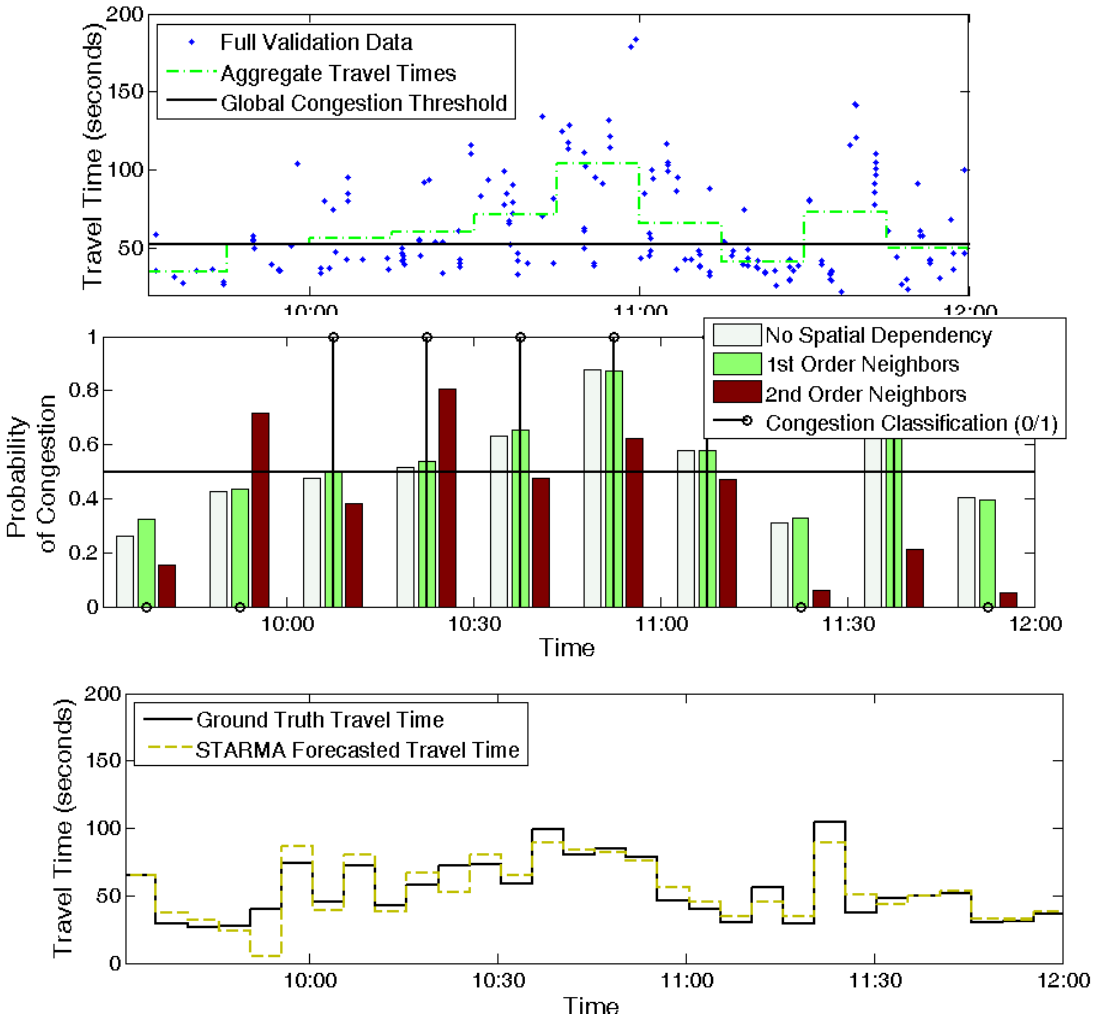
The Paramics simulations give information about every vehicle. For testing the methods, only a subset of the data is used for training and inference, corresponding to the penetration rate. This was incorporated into the following analysis by requiring each regression method to produce estimates for the validation data set using only a small percentage of the available travel times. Figure 1.3 displays the one-step forecast results of the logistic regression and STARMA methods on the Paramics validation set respectively, using a penetration rate of 5%. Similarly, figure 1.4 displays the one-step forecast results on the New York validation set.

[] []



[]

Figure 1.3. **One-step forecast validation results on a given VTL pair of the Paramics simulation network (penetration rate: 5%).** (a) Travel time data of the VTL pair and its aggregate value on 5-minute intervals. Both the data and the aggregate value are shown for the whole data set and for a 5% penetration rate. (b) Onestep forecast of the congestion state produced by the logistic regression algorithm. The bars represent the probability of congestion estimated by the models for different levels of spatial dependency. The real state of congestion is represented with circles. (c) One-step forecast of travel time produced by the STARMA algorithm.



[]

Figure 1.4. **One-step forecast validation results for logistic regression on one VTL pair of the New York network.** (a) Travel time data of the VTL pair and its aggregate value on 15 minutes time intervals. (b) One-step forecast of the congestion state produced by the logistic regression algorithm. The bars represent the probability of congestion estimated by the models for different levels of spatial dependency. The ground truth state of congestion is represented with circles. (c) One-step forecast of travel time produced by the STARMA algorithm.

Penetration Rate Study

The value of 5% for the penetration rate used in the previous subsection was chosen based on the prospects for future adoption of GPS equipped cell phones running traffic information software (such as that provided by *Mobile Millennium*). Therefore, a study of the effect of the penetration rate on results is of interest to quantify the influence of technology adoption on estimation and forecast accuracy. Figure 1.5 shows the one-step forecast accuracy for the logistic regressionand STARMA methods as a function of the penetration rate. From these figures, one can infer that 2% penetration rate can give reasonably good results, while 5% and higher give very accurate results. Also note that using spatial neighbors of order 1 (direct neighbors) generally provides

better results. One can interpret this as indicating that second order neighbors lead to an overfit model while no neighbors lead to an underfit model.

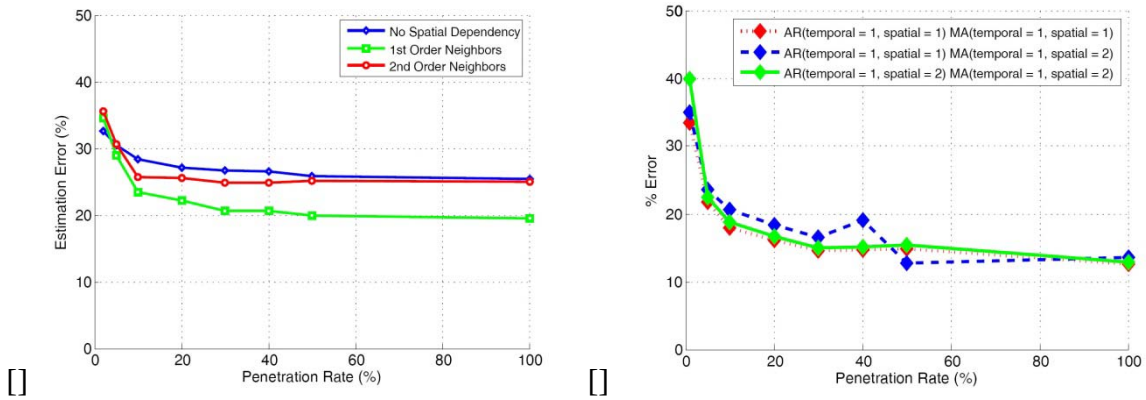


Figure 1.5. **Average one-step forecast error vs. penetration rate for all VTL pairs in the Paramics dataset**. (a) Logistic Regresion Forecast Classification Error. (b) STARMA Travel Time Forecast Error.

Multi-Step Forecast

The STARMA model is capable of producing forecasts of any number of steps by using the output of the model as input for the next time interval. It is not straightforward to do the same for the logistic regression model since it has an output that is fundamentally different from the input it requires. Therefore, the discrete output of the logistic regression model must be transformed back to a continuous value in order to do forecast in the same way. This avenue was not considered in this work and is left as further research.

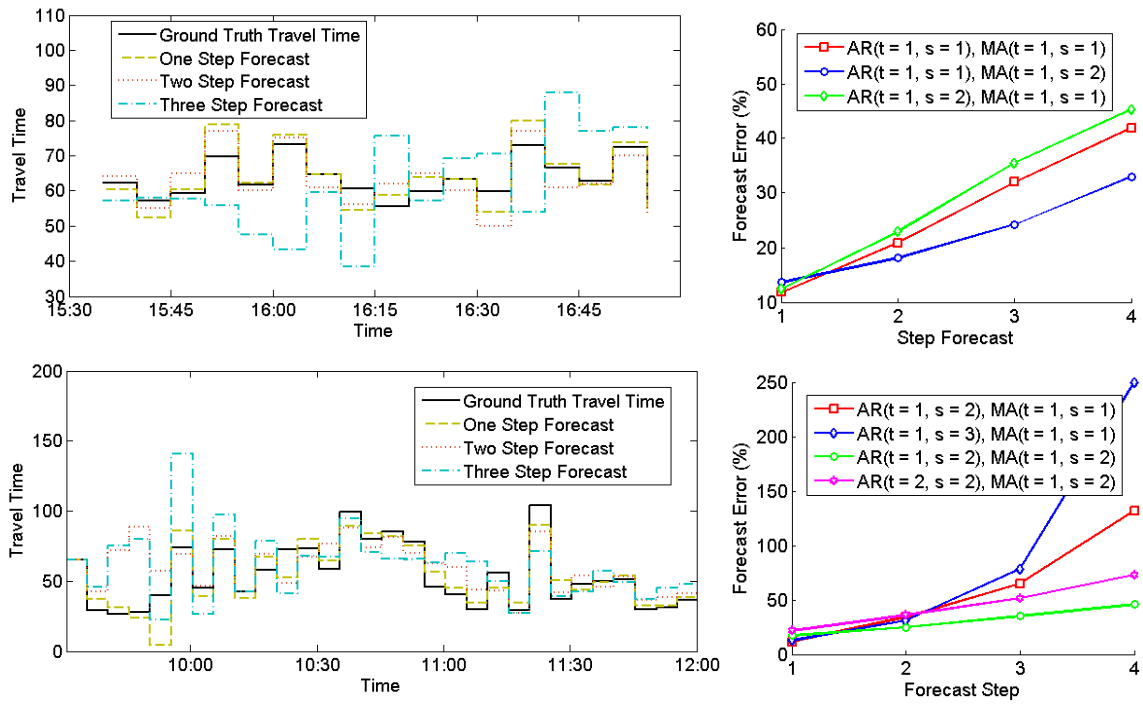
In this section, the results of multi-step forecast for the STARMA model are presented. Figure 1.6 shows the forecast results for the New York data set. The best results for the first step forecast are obtained for an autoregressive temporal order of 1, a spatial order of 2, a moving average temporal order of 1 and a spatial of 1. The two plots for which the moving average temporal and spatial orders are both equal to 1 show the best result for the first step forecast, but the error becomes quickly significant when the forecast step increases. On the other hand, the two other plots for which the moving average orders are one temporally and two spatially show a worse result for the first step forecast but considerably better results for more than one step. The choice of the parameters is therefore a very important step and should take into consideration the performance of the forecasting for more than one step ahead. Analysis of a larger data set is necessary to come to a statistically significant conclusion about the best way to chose the spatio-temporal parameters for the STARMA model.

19.2. Solution Methods for Bayesian Model

Solving the estimation model presented in section 2 is a two-step process. First, the historic parameters of the network are learned (section 1) and then these parameters are usedto perform a Bayesian update for real-time estimation (section 2). Learning the historic parameters is a

process that is intended to be run infrequently, although the more often the historic parameters are re-learned to take into account changes in traffic patterns, the more accurate the real-time estimations will be. The real-time estimation step is structured to take advantage of the more intensive computations required in learning the historic parameters, which means that the real-time estimation step can be solved efficiently online.

In this section, the case where the link travel time distributions are assumed to be Gaussian distributions is considered. The parameters, $Q_{l,t}$ (see section 2), are denoted $\mu_{l,t}$ and $\sigma_{l,t}$ for the mean andstandard deviation, respectively. The methodology extends to cases beyond the Gaussian distribution, but leads to more difficult optimization problems. The Gaussian case is presented here to show an example of the algorithm from start to finish in complete detail. A summary of the notation used for the algorithms from this section is provided in table 14.



[]

Figure 1.6. Forecast accuracy. (a) and (c): Forecast error for a VTL pair in the Paramics and New York networks, respectively. (b) and (d): Average forecast error as a function of the number of forecast steps into the future, Paramics and New York networks,respectively. One step is 5 minutes. In (b) and (d), t represents the temporal dependency and s represents the spatial dependency.

$\mathcal{L}$	A set of links of the road network
$\mathcal{T}$	A set of historic time periods
t	A historic time period (such as Mondays from 2:00pm-2:15pm)
t_n	A current time interval (such as the current 15 minute period
$\mu_{l,t}$	The historic mean travel time for link l during time period t
$\sigma_{l,t}$	The historic travel time standard deviation for link l during time period t
$\hat{\mu}_{l,t_n}$	The current mean travel time for link l during current time period t_n
$\hat{\sigma}_{l,t_n}$	The current travel time standard deviation for link l during current time period t_n
$\boldsymbol{P}_t$	The set of probe path observations for time period t
y_p	The travel time for probe path observation $p \in \boldsymbol{P}_t$
$w_{p,l}$	The proportion of link l driven for path observation $p \in \boldsymbol{P}_t$
b_l	The minimum travel time for link l
$\boldsymbol{X}_{l,t}$	The set of travel times allocated to link l for time period t
$\boldsymbol{N}(y,\mu,\sigma)$	The natural logarithm of the Gaussian distribution for a given travel time y, mean μ and standard deviation σ

Table 14. Notation used in the historic and Bayesian real-time algorithms.

19.2.1. Learning Historic Model Parameters

It is computationally challenging to solve the optimization problem in equation (115) because it is simultaneously solving for the mean and variance of every link in the network. It is possible to solve this problem directly if using a commercial grade non-linear optimization engine with a lot of computational power. It is assumed that such resources may not be available and an alternative solution strategy is proposed. The basic idea is to decouple the optimization into two separate subproblems, each of which is easier to solve on its own, and then to iterate between these subproblems until converging to an optimal solution. These two subproblems are *travel time allocation* and *parameter optimization*.

The insight into the decoupled solution approach is to realize that if it were known exactly how much each probe vehicle drove on each link of its path (instead of just the total travel time), it would be easy to estimate the mean and standard deviation for each link in the network (by simply taking the mean and standard deviation of all the link travel time observations). However, it is not known exactly how long each probe vehicle spent on each link without high frequency probe data. For sparsely-sampled data, the most likely amount of travel time spent on each link is determined instead. The problem in doing this is that computing the most likely link travel times is dependent upon the link travel time parameters (μ and σ) that need to beestimated. This would appear to a chicken-and-egg sort of problem, but there is a sound mathematical justification for iterating between these two steps. The link parameters are used to determine the most likely travel times and then the most likely travel times are used to update the parameters.

Travel Time Allocation

The travel time allocation problem assumes that estimates of the link parameters are available for

all links of the network (which means that $\mu_{l,t}$ and $\sigma_{l,t}$ are fixed for this part of the algorithm). In addition to these link parameters, it is necessary to specify *lower bounds* on the travel time allocated for each link of the network, denoted b_l for any link $l \in \mathcal{L}$. These bounds should be computed conservatively using the maximum speed that is realistically possible for the link, which is likely to be quite a bit higher than the speed limit (on a 25 mph arterial, one might choose $40 - 50$ mph to compute the minimum link travel time). For each observation $p \in \boldsymbol{P}_t$, the goal is to solve the following optimization problem

$$\begin{aligned} \underset{x}{\arg\max} \quad & \sum_{l \in L_p} N(x_{l,p} w_{p,l} \mu_{l,t}, \sqrt{w_{p,l} \sigma_{l,t}^2}) \\ \text{s.t.} \quad & \sum_{l \in L_p} x_{p,l} = y_p \\ & x_{p,l} \geq w_{p,l} b_l, \forall l \in L_p \end{aligned} \tag{124}$$

where the goal is to obtain the probe vehicle's link travel time ($x_{p,l}$) for each link on that vehicle's path ($l \in L_p$) subject to the constraint that the sum of the link travel times is equal to the observed travel time. There are also lower bounds ($w_{p,l} b_l$) on the link travel time variables to ensure a sensible solution is returned. Algorithm 2 gives the details for how to solve optimization problem 124. The optimal solution is primarily contained in lines 11-15, which computes the total expected path variance (V) and the difference between expected and actual travel times (Z). With these two quantities, each link is allocated the expected link travel time adjusted by some proportion of Z, where thisproportion is computed using the link variance divided by the total path variance. This procedure can lead to some links being allocated a travel time below the minimum for that link. The set $\boldsymbol{J}$ is introduced to track the links with initial allocated travel times below the lower bound and the main procedure is repeated by setting the travel times for these links to the lower bound and optimizing with respect to the remaining links. Note that for some path observations, it may not be possible to produce a travel time allocation that satisfies the constraints. This means that the vehicle traveled faster than the bounds dictated was possible. In this case, this observation is considered an outlier and is discarded from the set of observations. The final data structure that is kept from this part of the algorithm is denoted $\boldsymbol{X}_{l,t}$ which contains all of the individual travel times allocated to link $l \in \mathcal{L}$ for time period $t \in \mathcal{T}$. These travel times are scaled by the proportion of the link traveled, which explains the division by $w_{p,l}$ on line 19 of algorithm 2.

Algorithm 2: Travel Time Allocation.

Require $t \in \mathcal{T}$ is fixed to some particular time period.

  for $l \in \mathcal{L}$ **do**

    $\boldsymbol{X}_{l,t} = \emptyset$ {Initialize allocated travel time sets to be empty.}

  end for

  for $p \in \boldsymbol{P}_t$ **do** {For all probe path observations.}

    if $\sum_{l \in L_p} w_{p,l}\, b_l > y_p$ **then**

      Travel time allocation infeasible for this path. This means that the observation represented travel that is considered faster than realistically possible, so the observation is considered an outlier. Remove p from $\boldsymbol{P}_t$.

    else

      $\boldsymbol{J} = \emptyset$ {$\boldsymbol{J}$ contains all links for which the travel time allocation is fixed to be equal to the lower bound.}

      repeat

        $x_{p,l} = w_{p,l} b_l, \forall l \in \boldsymbol{J}$ {For all links that had an infeasible allocation in the previous pass through this loop, set the allocation to the lower bound.}

        $V = \sum_{l \in L_p \setminus \boldsymbol{J}} w_{p,l}\, \sigma_{l,t}^2$ {Calculate the path variance for the links not fixed to the lower bound.}

        $Z = y_p - \sum_{l \in \boldsymbol{J}} w_{p,l}\, b_l - \sum_{l \in L_p \setminus \boldsymbol{J}} w_{p,l}\, \mu_{l,t}$ {Calculate the difference between expected and actual travel time for the links not fixed to the lower bound.}

        for $l \in L_p$ {Allocate excess travel time in proportion of link variance to path variance.}

          $x_{p,l} = w_{p,l}\mu_{l,t} + \frac{w_{p,l}\sigma_{l,t}^2}{V} Z$

        end for

        $\boldsymbol{J} = \boldsymbol{J} \cup \{l \in L_p : x_{p,l} < w_{p,l} b_l\}$ {Find all links violating the lower bound.}

      until $x_{p,l} >= w_{p,l} b_l, \forall l \in L_p$

      for $l \in L_p$ **do**

        $\boldsymbol{X}_{l,t} = \boldsymbol{X}_{l,t} \cup \left(\frac{x_{p,l}}{w_{p,l}}\right)$ {Add the allocated travel time to $\boldsymbol{X}_{l,t}$.}

end for

end if

end for

return $\boldsymbol{X}_{l,t}, \forall l \in \mathcal{L}$

Parameter Optimization

Given $\boldsymbol{X}_{l,t}$ (a vector of allocated travel times for link l during time period t) from algorithm 2, it is straightforward to optimize the model parameters for each link ($\mu_{l,t}$ and $\sigma_{l,t}$). As was stated in the previous section, time period t is fixed for these computations and the algorithms are repeated for each time period independently. Algorithm 3 provides the detailed procedure for computing new values for these parameters. For this algorithm, $\boldsymbol{X}_{l,t}(i)$ denotes the ith element of $\boldsymbol{X}_{l,t}$ from algorithm 2.

Algorithm 3: Parameter Optimization.

Require $t \in \mathcal{T}$ is fixed to some particular time period.

Require Input: $\boldsymbol{X}_{l,t}, \forall l \in \mathcal{L}$ as returned from the travel time allocation.

for $l \in \mathcal{L}$ **do**

if $|\boldsymbol{X}_{l,t}| < \tilde{n}$ **then**

Link has too little data. Keep original $\mu_{l,t}$ and $\sigma_{l,t}$, or use values from other time periods if appropriate. The estimate for this link should be given a very low confidence value. $\tilde{n}$ represents the minimum number of observations to have a reliable estimate of the mean and variance. This should be set to at least 10.

else

$\mu_{l,t} = \frac{1}{|\boldsymbol{X}_{l,t}|}\sum_{i=1}^{|\boldsymbol{X}_{l,t}|} \boldsymbol{X}_{l,t}(i)$ {Compute the sample mean.}

$\sigma_{l,t} = \sqrt{\frac{1}{|\boldsymbol{X}_{l,t}|}\sum_{i=1}^{|\boldsymbol{X}_{l,t}|}\left(\boldsymbol{X}_{l,t}(i) - \mu_{l,t}\right)^2}$ {Compute the sample standard deviation.}

end if

end for

11. **return** $\mu_{l,t}, \sigma_{l,t}, \forall l \in \mathcal{L}$

Full Historic Arterial Traffic Algorithm

Putting the travel time allocation and parameter optimization pieces together, the entire historic traffic model is presented in algorithm 3. A global parameter is required for this algorithm, which is the maximum number of iterations $M^{\max}$ (to go back and forth between travel time allocation and parameter optimization).

Algorithm 4: Historic Arterial Traffic.

1. **Require** $t \in \mathcal{T}$ is fixed to some particular time period.
2. Initialize the model parameters $\mu_{l,t}$ and $\sigma_{l,t}$ to typical values based on the physical characteristics of the links using guidelines from transportation handbooks such as the Highway Capacity Manual [30].
3. **for** $i = 1$ **to** $M^{\max}$ **do**
4. Use $\mu_{l,t}, \sigma_{l,t}$ for all $l \in \mathcal{L}$ to obtain $\boldsymbol{X}_{l,t}$ using algorithm 2.
5. Use $\boldsymbol{X}_{l,t}$ to compute new values of $\mu_{l,t}, \sigma_{l,t}$ using algorithm 3.
6. **end for**
7. **return** $\mu_{l,t}, \sigma_{l,t}, \boldsymbol{X}_{l,t}, \forall l \in \mathcal{L}$

When first running the historic model, it is important to run it iteratively as described in algorithm 4. After this initial run, future observations can be incorporated into the historic model in two ways. The first and most robust way is to run the entire algorithm 4 as before using all previous and newly acquired data. A second way which is less computationally intensive is to do an incremental update of the model parameters, $\mu_{l,t}$ and $\sigma_{l,t}$. Algorithm 5 describes how to do this incremental version. It would be appropriate to do this incremental version to update the historic model weekly. It is ideal to re-run the full version (algorithm 4) every few months.

Algorithm 5: Incremental update of historic traffic parameters.

1. **Require** $t \in \mathcal{T}$ is fixed to some particular time period.

2. **Require** $\mu_{l,t}, \sigma_{l,t}, \boldsymbol{X}_{l,t}, \forall l \in \mathcal{L}$ as given by the last run of the historic algorithm.

3. Run the travel time allocation (algorithm 2) on the new observations $\widetilde{\boldsymbol{P}}_t$ using $\mu_{l,t}, \sigma_{l,t}$ and denote $\widetilde{\boldsymbol{X}}_{l,t}$ the returned values from the algorithm.
4. $\boldsymbol{X}_{l,t} = \boldsymbol{X}_{l,t} \cup \widetilde{\boldsymbol{X}}_{l,t}$

5. Run the parameter optimization (algorithm 3) using $\boldsymbol{X}_{l,t}$ to obtain new values for $\mu_{l,t}, \sigma_{l,t}$.

18.2.2. Bayesian Real-time Traffic Estimation

The real-time model uses the output of the travel time allocation to perform a Bayesian update of the link parameters. Algorithm 6 provides the details for the complete algorithm to go from path-inferred probe data to travel time distributions for each link. This requires specifying the *current time window* (t_1, t_2). Denote t_2 as the current time and t_1 as some amount of time back in the past that must be specified. Both quantities are defined such that the algorithm is run once every $t_2 - t_1$ amount of time. The frequency at which the model should be run is dependent upon the amount of data coming in real-time. The frequency should be at least as high as the historic model, so if the historic model produces estimates for each 15 minute period of the day, then the real-time model should be run at least once per 15 minutes. If the data volume is large, the model can be run up to every 5 minutes. Running the model more frequently will likely not increase the performance and may lead to estimates that fluctuate too much.

The algorithm works by considering the historic link travel time distributions as *priors* on the real-time distribution. The variance is not updated as part of the real-time algorithm because it is considered to be constant within a historic time period (primarily due to a lack of data to be confident in a real-time estimate of the variance). In algorithm 6, lines 5-6 define the prior parameters (denoted α and β, which are from the *conjugate normal* prior distribution [287] used here) based on the historic travel time distribution. This requires specifying ν, which expresses the precision of the estimate of the historic mean $\mu_{l,t}$. It is chosen based on experimentation to determine how much real-time traffic conditions can deviate from the historic value. The ν parameter allows one to give more or less weight to real-time data. Line 7 computes the average of the current data. Line 8 computes the current estimate of the meanfor the link travel time distribution using the formula for performing a Bayesian update [287]. The normal prior is used because it is a computationally efficient (conjugate) prior when the observations (link travel times) are normally distributed with known mean (see [287] for details on the expression of the prior).

Algorithm 6: One time step of the real-time algorithm.

1. **Require** t_2 is the current time and t_1 is the time of the last estimate.
2. Select $\mathcal{P}_{t_1,t_2}$, the set of current probe path observations.
3. Run the travel time allocation algorithm over the set $\mathcal{P}_{t_1,t_2}$ and obtain the link travel time sets $\boldsymbol{X}_{l,t_2}$, $\forall l \in \mathcal{L}$.
4. Define $n_{l,t_2} := |\boldsymbol{X}_{l,t_2}|$, $\forall l \in \mathcal{L}$.
5. **for** $l \in \mathcal{L}$ **do**
6. $\alpha = \mu_{l,h(t_2)}$
7. $\beta = \nu\sigma_{l,h(t_2)}$
8. $\bar{x} = \frac{1}{n}\sum_{x_i \in \boldsymbol{X}_{l,t_2}} x_i$ {The average of the current observations. x_i is the ith element of $\boldsymbol{X}_{l,t_2}$.}
9. $\hat{\mu}_{l,t_2} = \frac{\frac{\alpha}{\beta}+\frac{\bar{x}}{\sigma_{l,h(t_2)}}}{\frac{1}{\beta}+\frac{n_{l,t_2}}{\sigma_{l,h(t_2)}}}$ {The updated mean travel time corresponds to a weighted average between the historic mean and the real-time mean.}
10. **end for**

18.3. Solution Methods for Probabilistic Graphical Model

In this section, the solution methods for solving the graphical model presented in section 3 are introduced. These solution methods employ both particle filtering and the Expectation Maximization algorithm, both of which are introduced in more detail in section 1.

The way the graphical model is constructed gives rise to the following paradox. Given the parameters of the model, it is possible to estimate the most likely state of the links given observations and their evolution over time. Similarly, given the state of the links of the network over a period of time, it is possible to estimate the parameters of the model (state transition matrix, and conditional travel time probability distributions). This well known type of problem is solved using an EM algorithm which iterates between finding the probability of each state for each link of the network and each time interval given some values of the model parameters (E step). Then, the probabilities of each state for each link and each time interval are used to update the value of the parameters by maximizing the log likelihood (M step).

A high-level description of the parameter estimation is presented in algorithm 128 (this work is based on a previously published article [183]). A summary of the notation used for the graphical

model algorithms is provided in table 20. In general, bold notation refers to the vectorized version of a particular variable (e.g. $\boldsymbol{a} = \{a_i | \forall i\}$).

N	Number of links of the road network
S	Number of congestion states per link
D	Number of days in the training data set
T_d	Number of time intervals for day $d \in D$
$g_{l,s}(\cdot)$	The travel time probability density function for link l when in congestion state s
$P_{l,s}$	The parameters of the probability density function $g_{l,s}(\cdot)$
$z_{d,t,l}^{s}$	The probability of link l being in congestion state s for time period t on day d
$q_{d,t,l}^{s,r}$	The probability of link l being in congestion state s for time period t on day d given that its neighboring links are in state r
A_l	The state transition probability matrix for link l with respect to its neighbors
$\pi_{l,s}$	The initial probability that link l begins the day in state s
$I_{d,t,l}$	The set of probe observations (after travel time allocation) for day d, time t and link l
$\alpha_{x_1^l, x_2^l}$	The proportion of travel time of link l when driving the partial distance from start location x_1^l to end location x_2^l.

Table 20. Notation used in the graphical model algorithms.

18.3.1. State Estimation

The goal of this section is to describe how to perform state estimation given the parameters of the model. This also turns out to be the E-step of the EM algorithm as will be described later.

The challenge of the graphical model approach is that the link travel times are not directly observed since the probe observations received can span several links of the network between two consecutive measurements. This difficulty is addressed by computing the most likely link travel times that make up the path of the probe vehicle (*travel time allocation*). It is possible to have a graphical model representation that does not have this decomposition approach, but it leads to a difficult non-linear parameter optimization (M-step) problem, for which the number of variables increase quadratically in the number of links. This optimization problem would require an approximation technique to solve, which is why a more intuitive decomposition scheme called *travel time allocation* is proposed. For data from fixed location sensors, this is not an issue as one simply needs to define an appropriate parameterized observation distribution for each link. Since all of the available data at the present time is sparse GPS probe data, only this case is analyzed here. Once travel time allocation has been performed, it is straight forward to apply a particle filter for performing real-time estimation.

Travel Time Allocation

An observation consists of a travel time over a path consisting of multiple (partial) links. In order to use the graphical model presented in section 3, the total travel time must be decomposed into a travel time for each (partial) link on the path. This can be achieved by maximizing the log-likelihood of the link travel times for each observation given the model parameters. This optimization problem for a single observation is

$$\underset{\mathrm{y}}{\arg\max}\left\{\sum_{l\in P}\ln\left(\sum_{s=1}^{S} z_l^s\, g_{l,s}(y_l)\right) : \sum_{l\in P}\alpha_{x_1^l,x_2^l}\, y_l = \tilde{y}\right\}, \tag{125}$$

where P is the set of links on the path, $\boldsymbol{y} := \{y_l\}_{l\in P}$ is a vector of the travel times assigned to each link on the path, x_1^l and x_2^l are the start and end location on link l, $\tilde{y}$ is the observed travel time betweenthe GPS measurements, z_l^s is the probability of link l to be in state s, and $g_{l,s}$ is the travel time distribution for link l when in congestion state s. The values of x_1^l and x_2^l will be equal to the start and end of the link for all intermediate links and will only have non-trivial values for the first and last link of the path (where the actual GPS observations are). The values of z_l^s are obtained from the E-step of the EM algorithm, except in the first iteration where they have been initialized with reasonable values (see algorithm 128). The optimization problem in equation (125) has a number of variables equal to the number of links of the path between consecutive GPS measurements, which is always a relatively small number. This makes the optimization problem easy to solve using numerical methods.

To compute α_{x_1,x_2}, the proportion of the full link travel time to use, the method proposed in [188] is used (see the section on density estimation).

Particle Filtering

On small networks, it is possible to do exact inference in the CHMM by converting the model to an HMM (with a number of links-dimensional state vector) [289]. However, the transition matrix is a $S^N \times S^N$ matrix (N isthe number of links in the network), which is intractable for traffic network with more than a few dozen links. Instead, an approximation based on particle filtering [169] is used. Each particle represents an instantiation of the time evolution of the network. Each particle has a weight proportional to the probability of having this instantiation of the state evolution of the network given the available data. It is necessary to simulate a high number of particles that evolve through the graphical model. These particles are used to estimate the probabilities of the state of each link and each time interval and the probabilities of transition between the state of the neighbors of link l at time $t-1$ and the state of link l at time t. See section 2 for more details on particle filtering.

18.3.2. Parameter Estimation: the EM Algorithm

As described earlier, the EM algorithm iterates between finding the most likely state of the network given the model parameters and then uses those state estimates to find the new most likely model parameters. Section 3 provides additional details about the EM algorithm. The details of how to apply the EM algorithm for the graphical model are presented here.

E step

As stated earlier, the E-step of the EM algorithm is identical to the state estimation section just discussed (section 1).

M step

For each link and each state, it is assumed that the travel time distribution $g_{l,s}$ is parameterized by a set of parameters $P_{l,s}$ and the set of all parameters is denoted $\boldsymbol{P} = (P_{l,s})_{l,s}$. To update these parameters, the expected complete log-likelihood is maximized given the probability ($z_{d,t,l}^{s}$) that each link l is in state s at time t and day d and the probability ($q_{d,t,l}^{s,r}$) of link l to be in state s given that the neighbors of link l are in state r at time $t-1$ and day d. The updates of the transition matrices A_l and the initial state probabilities π_l for each link of the network corresponds to optimizing on the set of parameters $\boldsymbol{A} = (A_l)_l$ and $\boldsymbol{\pi} = (\pi_{l,s})_{l,s}$. The expected complete log likelihood is

$$\begin{aligned}
&\Lambda(Y|\,\boldsymbol{z},\boldsymbol{q},\boldsymbol{P},\boldsymbol{A},\boldsymbol{\pi}) = \\
&\sum_{l=1}^{N}\sum_{s=1}^{S}\sum_{d=1}^{D}\sum_{t=1}^{T_d} z_{d,t,l}^{s}\left(\sum_{i=1}^{I_{d,t,l}} \ln\left(g_{l,s}(y_i)\right)\right) + \\
&\sum_{l=1}^{N}\sum_{d=1}^{D}\sum_{t=2}^{T_d}\sum_{s=1}^{S}\sum_{r=1}^{S^{N_n^l}} q_{d,t,l}^{s,r} \ln\left(A_l(r,s)\right) + \\
&\sum_{l=1}^{N}\sum_{d=1}^{D}\sum_{s=1}^{S} z_{d,0,l}^{s} \ln\left(\pi_{l,s}\right),
\end{aligned} \tag{126}$$

where $I_{d,t,l}$ is the set of travel time observations for day d, time interval t, and link l as provided by the travel time allocation method presented in section 1.

The typical optimization problem (as described in section 3) is modified to take into account the varying number of observations for each link and each time interval. The optimization problem is stated as

$$\max_{\mathrm{P,A}} \quad \Lambda(Y|\,\boldsymbol{z},\boldsymbol{q},\boldsymbol{P},\boldsymbol{A},\boldsymbol{\pi}): \begin{cases} \sum_{s=1}^{S} A_l\,(r,s) = 1, \forall\, l,r \\ A_l(r,s) \in [0,\,1], \forall l,r,s \\ \sum_{s=1}^{S} \pi_{l,s} = 1, \forall\, l \\ \pi_{l,s} \in [0,\,1], \forall l,s \end{cases} \tag{127}$$

The updates of the transition probabilities A_l and of the initial state probabilities π_l are straightforward.

The update of the travel time distributions depends on the type of distribution used in the model. Due to the travel time allocation, the optimization problem on all the parameters $\boldsymbol{P}$ of the network decouples in $S \times N$ smaller optimizationproblems, one for each state and link of the network. For state s and link l, the optimization problem is

$$\max_{\mathrm{p_{l,s}}} \quad \sum_{d=1}^{D} \sum_{t=1}^{T_d} z_{d,t,l}^{s} \left(\sum_{i=1}^{I_{d,t,l}} \ln\,(\,g_{l,s}(y_i)) \right), \tag{128}$$

where $p_{l,s}$ represents the parameters of the travel time distribution $g_{l,s}$. Decoupling the optimization problem makes it highly scalable as each of the optimization subproblems can be performed in parallel. If the travel time allocation method is not used, then the resulting optimization problem is coupled across the whole network resulting in a large non-linear optimization problem that does not scale well.

Algorithm 7: Estimation of the historical distribution of travel time and state transition probability matrices.

Initialize the parameters $P_{l,s}$ of the distributions, the state transition probability matrices A_l, the initial state probabilities $\pi_{l,s}$, and the state probabilities $z_{d,t,l}^{s}$

EM-algorithm with travel time allocation:
while {The algorithm has not converged} **do**
  <u>Travel time allocation</u> (section 1)
  $y_l \leftarrow$ Allocated travel times given the parameters $P_{l,s}$ and the state probabilities $z_{d,t,l}^{s}$

  <u>E Step</u> (section 1): compute the expected state probabilities $z_{d,t,l}^{s}$ and transition probabilities $q_{d,t,l}^{r,s}$ given $(y_l)_l$, $(P_{l,s})_{l,s}$ and $(A_l)_l$
  $z_{d,t,l}^{s} \leftarrow E(z_{d,t,l}^{s}|y_l,\, P_{l,s},\, A_l)$
  $q_{d,t,l}^{r,s} \leftarrow E(q_{d,t,l}^{r,s}|y_l,\, P_{l,s},\, A_l)$

  <u>M Step</u> (section 2): maximize the expected complete log-likelihood, given the state probabilities $z_{d,t,l}^{s}$ and the transition probabilities $q_{d,t,l}^{r,s}$.
  $(P_{l,s}, A_l, \pi_l) \leftarrow \boldsymbol{argmax}_{\boldsymbol{P},\boldsymbol{A},\boldsymbol{\pi}} \Lambda(Y|\,\boldsymbol{z},\boldsymbol{q},\boldsymbol{P},\boldsymbol{A},\boldsymbol{\pi})$
end while

Results

The model was tested using probe data from a fleet of about 500 taxis in San Francisco as provided to us by the Cabspotting project [4], which has been integrated into *Mobile Millennium* system through a data feed. Each taxi provides a measurement of its location approximately once every minute (generally between 40 and 100 seconds), which falls into the category of sparse GPS probe data (as described in section 9). In addition to its location, the taxi also reports whether or not it is carrying a customer. This information allows for filtering out the points when a taxi is loading or unloading a passenger. This data is sent to the *Mobile Millennium* traffic system, where it is processed and visualized in real-time.

In the case study, data from November 25, 2009 through February 27, 2010 was used, focusing on weekdays from 3pm-8pm in the subnetwork of San Francisco depicted in figure 3.1. This subnetwork contains 322 links (where a link is defined as the road between two signals) and has an average of 600 observations per half hour time interval for the whole network. The time interval used was 30 minutes (half an hour) for the graphical model. These results are for the case where the observation probability distribution functions g (section 2) are independent Gaussians. In general, the choice of a Gaussian distribution restricts the flexibility of the model to capture unique traffic characteristics, but itis also far more tractable to solve in practice. While this section has described how to solve this model using the distribution introduced in chapter 17, no experimental results have been generated to date.

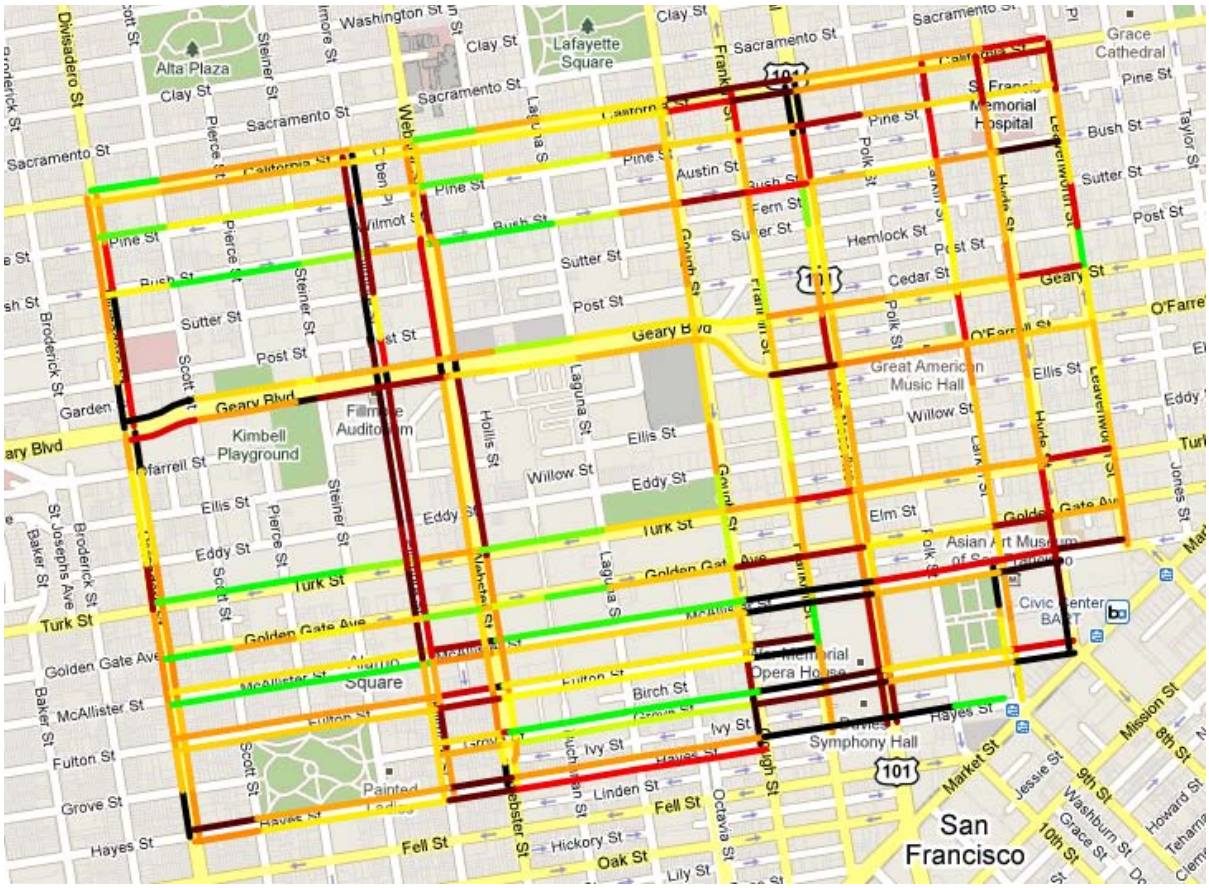


Figure 3.1. Real-time traffic estimation for a subnetwork of San Francisco. The color scale represents the estimated travel time divided by the speed limit travel time. Green is for values close to 1 (travel time is about the same as driving at the speed limit) and black indicates values around 5 (travel time is 5 times slower than driving at the speed limit).

The approach requires a training period (section 2) before it can be used to make predictions in real-time. Data from November 25, 2009 through February 19, 2010 was used for the training period. Only Tuesdays, Wednesdays and Thursdays were used to train the model, which totalled 18 training days (after removing holidays and days with system malfunctions that prevented data collection). The model was then tested by running it over all Tuesdays, Wednesdays and Thursdays between February 20, 2010 and February 27, 2010, which totalled 3 days.

The traffic density parameters (see the travel time allocation algorithm of section 1) for each hour of the day from 3pm to 8pm, where each hour period is assumed to have its own characteristics in terms of the average density on a link. The EM algorithm (section 2) was run over all the training data, with the assumption that the transition matrix $\boldsymbol{A}$ and the Gaussian distributions for each link are stationary over the study period. Once the parameters were learned through the EM algorithm, a particle filter was used to compute the most likely state of each link given real-time data on a test day. Figure 3.1 shows a map of the subnetwork of San Francisco with each link colored according to its level of congestion, defined as the mean travel time divided by a reference free flow travel time. The free flow travel time is computed as the travel

time experienced when traveling at the speedlimit and accounting for an expected delay (due to traffic signals) under light traffic conditions.

To quantify the validity of the estimates, the actual travel time of an observed path is compared to the estimate obtained by summing over the mean travel time for all links of the path. Table 21 shows the root mean squared error (RMSE) and mean percentage error (MPE) of our travel time predictions as compared to a baseline approach. The baseline approach computes the average speed for each observation and assigns it to each link along its path. Then all of the speeds recorded on each link are averaged to give a historical average speed for each link. The real-time version of this approach also computes the average speed on each link within the real-time estimation interval and then takes a weighted average between the historical and the real-time speed to give a speed estimate for each link of the network, which can be used to estimate travel times.

These results were computed on the data obtained between February 20 and February 27, 2010. The data was split into two sets, one for computing the real-time traffic estimates and one for computing the error metrics. This was done to ensure an unbiased comparison of the proposed graphical model and the baseline model. Approximately 70% of the data was used for computing the real-time traffic estimates with the other 30% used for computing the error metrics.

Model	RMSE (sec)	MPE
Graphical model	46	30.1%
Baseline model	63	44.4%

Table 21. Experimental results comparison between the proposed graphical model and the baseline model.

Part V

Mathematical and Algorithmic Contributions

20. Enhancing Privacy and Accuracy in Probe Vehicle Based Traffic Monitoring

This chapter extends and refines the spatial sampling scheme previously described in the Mobile Century final report [269]. That description employed the use of virtual trip lines (VTLs), which are geographic markers that indicate where vehicles should provide speed updates. These markers are placed to avoid specific privacy sensitive locations. They also allow aggregating and cloaking several location updates based on trip line identifiers, without knowing the actual geographic locations of these trip lines. Thus, they facilitate the design of a distributed architecture, in which no single entity has a complete knowledge of probe identities and fine-grained location information.

The Mobile Century final report [269] included only a single architecture utilizing VTLs to achieve probabilistic privacy. A second architecture is presented here that achieves guaranteed privacy using k-anonymous temporal cloaking. The data set from the Mobile Century experiment is revisited to evaluate performance trade-offs between these two architectures.

As an extended version of the descriptions in the Mobile Century final report, this chapter offers substantial improvements by:

- Arguing that spatial sampling (through virtual trip lines) rather than temporal sampling leads to increased privacy because it allows omitting location samples from more sensitive areas.
- Describing a privacy-aware placement approach that creates the virtual trip line database.
- Demonstrating that the virtual trip line concept can be implemented on a GPS-enabled cellular phone platform.
- Evaluating accuracy and privacy by revisiting the extensive datasets from the Mobile Century experiment.
- Developing a light-weight trip line crossing detection algorithm against inaccurate GPS readings and intermittent wireless connectivity.

The remainder of this chapter is organized as follows. Section 1 describes the challenges in probe

vehicle based traffic monitoring. Section 2 introduces the virtual trip line concept and discusses its potential uses in the domain of traffic monitoring. Section 3 describes the use of VTLs in two different traffic monitoring architectures and discusses the privacy features of each system. We implement and evaluate proposed architectures in section 4 and 5. Then we discuss limitations and outlooks in section 6, and propose some conclusions.

20.1. Traffic Monitoring Challenges in Probe Vehicle Systems

In this section we describe two challenges faced by probe monitoring systems, and our design goals to overcome these challenges through the implementation of traffic monitoring with virtual trip lines.

20.1.1. Privacy Risks

Traffic monitoring using GPS-equipped vehicles raises significant privacy concerns, because the external traffic monitoring entity acquires fine-grained movement traces of the probe vehicle drivers. These location traces might reveal sensitive places that drivers have visited, from which, for example, medical conditions, political affiliations, traffic violations, or potential involvement in traffic accidents could be inferred.

Threat Model and Assumptions. This work assumes that adversaries can compromise any single infrastructure component to extract information and can eavesdrop on network communications. We assume that different infrastructure parties do not collude. We believe that this model is useful in light of the many data breaches that occur due to dishonest insiders, hacked servers, stolen computers, or lost storage media (see [8] for an extensive list, including a dishonest insider case that released 4500 records from California's FasTrak automated road toll collection system). These cases usually involve compromised log files or databases in a single system component and motivate our approach of ensuring that no single infrastructure component can accumulate sensitive information.

We assume that a handset (i.e., a client application) itself is trustworthy but its owner can be malicious. Thus an owner cannot reverse-engineer the client code, so that he or she cannot intentionally manipulate a GPS reading, speed, timestamp of measurements, or cryptographic keys. However, as we will consider in Section 20.6.1, an owner can use the client application for malicious purposes within the legitimate use of a handset. We call this situation *compromised phones*. For example, a company competing for the same service (e.g., traffic monitoring services) can hire multiple users and ask them to intentionally drive slowly in non-congested roads.

We label sensitive information any information from which the precise location of an individual at a given time can be inferred. Traffic monitoring does not need to rely on individuals or personal information, only on the aggregated statistics from a large number of probe vehicles. Thus, an obvious privacy measure is to anonymize the location data by removing identifiers such as network addresses. This approach is insufficient, however, because drivers can often be re-identified by correlating anonymous location traces with identified data from other sources. For example, home locations can be identified from anonymous GPS traces [191,214] that may

be correlated with address databases to infer the likely driver. Similarly, records on work locations or automatic toll booth records could help identify drivers. Even if anonymous point location samples from several drivers are mixed, it is possible to reconstruct individual traces because successive location updates from the same vehicle inherently share a high spatio-temporal correlation. If overall probe vehicle density is low, location updates close in time and space likely originate from the same vehicle. This approach is formalized in target tracking models [283].

As an example of tracking anonymous updates, consider the following problem: given a time series of anonymous location and speed samples mixed from multiple users, extract a subset of samples generated by the same vehicle. To this end, an adversary can predict the next location update ($\hat{x}_{t+\Delta t}$) based on the prior reported speed $\hat{x}_{t+\Delta t} = v_t \cdot \Delta t + x_t$ of the actual reported updates, where x_t and $x_{t+\Delta t}$ are locations at time t and $t + \Delta t$, respectively, and v_t is the reported speed at t. The adversary then associates the prior location update with the next update closest to the prediction, or more formally with the most likely update, where likelihood can be described through a conditional probability $P(x_{t+1}|x_t)$ that primarily depends on spatial and temporal proximity to the prediction. The probability can be modeled through a probability density function of distance (or time) differences between the predicted update and an actual update (under the assumption that the distance difference is independent of the given location sample).

Privacy Metrics. As observed in [192], the degree of privacy risk depends on how long an adversary successfully tracks a vehicle. Longer tracking increases the likelihood that an adversary can identify a vehicle and observe it visiting sensitive places. We thus adopt the *time-to-confusion* [192] metric and its variant *distance-to-confusion*, which measures the time or distance over which tracking may be possible. Distance-to-confusion is defined as the travel distance until tracking uncertainty rises above a defined threshold. Tracking uncertainty is calculated separately for each location update in a trace as the entropy $H = -\sum p_i \log p_i$, where the p_i are the normalized probabilities derived from the likelihood values described in [176]. These likelihood values are calculated for every location update generated within a temporal and spatial window after the location update under consideration.

These tracking risks and the observations regarding increased risks at certain locations further motivate the virtual trip line solution described next. Compared to a periodic update approach, in which clients provide location and speed updates at regular time intervals, virtual trip lines can be placed in a way to avoid updates from sensitive areas.

Goal. We aim to achieve privacy protection by design so that the compromise of a single entity, even by an insider at the service provider, does not allow individual users to be tracked or reidentified.

20.1.2. Lack of Guaranteed Accuracy of Sensor Data

The quality of traffic monitoring is contingent on the accuracy of the sensor data. In turn, the accuracy of this data is affected by technical limitations of sensor and the potential for maliciously injected bogus data. Thus, a key strategy to provide high quality traffic monitoring is

to ensure accurate speed and location measurements in the presence of GPS error and to prevent malicious injection attacks.

To address the issue of GPS position errors, some level of client–side or server–side data filtering is required. If a light–weight algorithm running on the client can manage this job efficiently, it not only reduces user privacy concerns by avoiding data transmission, but also reduces the server–side computational burden, thereby achieving better scalability. To prevent bogus measurements from entering the data stream, some security countermeasures can be introduced to validate data authenticity. However, device authentication conflicts with user anonymity desired for privacy, and authentication alone cannot prevent fraudulent updates. Recent studies have presented a trusted platform module (TPM) [162,291] for preventing fraudulent updates.

Goal. The client software must cope with the resource constraints of current cellphone platforms where the use of computationally expensive algorithms such as map-matching and Kalman filtering is limited. We mainly focus on designing a light-weight component that detects trip line crossings accurately while suppressing false positives in the presence of noisy GPS readings and intermittent wireless connectivity (which affects A-GPS performance). Additionally the system should not allow adversaries to insert spoofed data, which would compromise the data quality and thus traffic information. This is especially challenging because it conflicts with the desire for anonymity.

20.2. Virtual Trip Lines

To address these challenges our proposed traffic monitoring system builds on the concept of virtual trip lines and the notion of separating the communication and traffic monitoring responsibilities (as introduced in [191]). A *virtual trip line* (VTL) is a line segment in geographic space that, when crossed, triggers a client's location update to the traffic monitoring server. More specifically, it is defined by

$$[vtlid, x_1, y_1, x_2, y_2, d]$$

where $vtlid$ is the virtual trip line ID, x_1, y_1, x_2, and y_2 are the (x, y) coordinates of two line endpoints, and d is a default direction vector (e.g., N-S or E-W). The default direction vector encodes the valid direction in which the virtual trip line can be crossed. This directionally specific attribute can be used to reject location updates from vehicles crossing VTLs in the opposite direction, which can occur due to GPS errors and dense road networks. Also in case that a single VTL covers both directions on highways if it is long enough (to cover both northbound and southbound, or westbound and eastbound), the clients detect the direction from two successive coordinates and simply code the direction into 0 or 1 based on default direction vector.

When a vehicle traverses the trip line, its measurement update includes the time, trip line ID, speed, and the direction of crossing. The trip lines are pre-generated, downloaded, and stored in clients. To check any crossings, we set the sampling period of a single-chip GPS/A-GPS module in each smartphone and retrieve the position readings. Since our setup did not provide speed

information, we calculate the mean speed using two successive location readings (in our implementation, every 3 seconds). The client software registers the task for checking the traversal of trip lines as an event handler for GPS module location updates, which is automatically invoked whenever a new position reading becomes available. As an example of required storage and bandwidth consumption, consider the San Francisco Bay Area, the total road network of which contains about 20,000 road segments, according to the Digital Line Graph 1:24K scale maps of the San Francisco Bay Area Regional Database managed by USGS. Assuming that the system on average places one trip line per segment this results in 166KB of storage.

Virtual trip lines control disclosure of location updates by sampling in space rather than sampling in time, since clients generate updates at predefined geographic locations (compared to sending updates at periodic time intervals). The rationale for this approach is that at specific locations, traffic information is more valuable and certain locations are more privacy-sensitive than others. Through careful placement of trip lines, the system can thus better manage data quality and privacy than through a uniform sampling interval. In addition, the ability to store trip lines on the clients can reduce the dependency on trustworthy infrastructure for coordination.

20.2.1. Strengths

The VTL concept can be extended to provide several additional benefits. First, as will be discussed throughout the article, it allows system designers to choose several different options for privacy protection. The levels of privacy protection range from forcing the location sampling in sensitive areas to achieving guaranteed privacy via k-anonymous cloaking. Second, for a given number of location updates from drivers, the VTL paradigm allows system designers to predefine measurement locations for high-value updates. For example, location updates from low priority residential streets can be avoided. Third, the use of VTLs removes the need for map matching the measurement update to road segments, since each VTL is already associated with a road segment. Fourth, system designers can embed traffic alerts or warnings on VTLs by piggybacking on the system's acknowledgement packet which responds to a user's location update. For example, VTLs may be defined with location descriptors associated with school zones, construction zones, or icy roads. Fifth, we can define a timer attribute for each VTL which specifies the allowable latency for each measurement. Thus, increasing the timer on a VTL allows users to delay the measurement report time, which aids in the prevention of adversarial tracking. Sixth, we can dynamically turn on/off VTLs depending on the time of day and congestion levels. Also around construction sites or detours, one can dynamically place more VTLs.

20.2.2. Virtual Trip Line Measurements

Noisy GPS readings can be filtered either on the client side or the server side. Server side processing can allow for a computationally expensive algorithm to filter out noisy GPS readings, for example using map-matching algorithms. However, it requires clients to send detailed traces to a server, which incurs increased network bandwidth consumption and privacy concerns. Instead we address filtering on the client, with the specific goals of subsampling GPS readings to reduce the frequency of trip line measurement computations (i.e., checking whether the line

between two GPS readings intersects with any trip lines), and removing the need of any client side or server side map-matching algorithm, which is a computationally expensive algorithm for resource constrained devices.

We have observed that GPS position error can create false VTL crossings and inaccurate VTL velocity measurements in the following cases:

- *GPS position error.* When a vehicle stops near a trip line, error in the GPS position can create successive position measurements with a zigzag pattern over the VTL, which can lead to multiple false trip line crossings. These crossings can be eliminated by requiring a minimum distance between successive GPS readings.
- *Intermittent GPS.* When the time interval between two GPS positions becomes large (e.g., due to lost GPS signal), the inferred trajectory connecting these two location measurements no longer describes the actual movement of a vehicle. To eliminate false trip line crossings by this type of unrealistic trajectory, an upper bound of time gap between successive GPS samples is required.
- *Infeasible speed.* In areas prone to high GPS position error (e.g., urban areas with high-rise buildings), the speed computed from a finite difference approximation of the successive positions (required by the GPS receiver in our implementation) becomes infeasible. We refer to these errors as *speed glitches* in the remainder of the article.

Algorithm 22 below describes in detail our implementation of a light-weight client filtering algorithm to treat the common situations above. The algorithm proceeds as follows. First, if the GPS sample l is the first update, it is simplysaved to *CurrLocationFiltered* (line 4-7). Without a previous update, we cannot compute the speed or heading of the current update or confirm it as valid. Assuming a previous update exists, the validity of the next update can be determined based on the computed speed and the temporal/spatial gap from previously filtered GPS reading called *PrevLocationFiltered*. We consider the current location invalid if it is updated long after the previous update (line 10-14), if it has not traveled a minimum distance (line 16-18, e.g., stopped at the traffic signal), or if has a speed glitch (line 19-30).

Additionally, we maintain two reference points, *LastGoodRefPoint* and *LastBadRefPoint*. If a series of locations have speed glitches against *LastGoodRefPoint*, but do not have speed glitches against *LastBadRefPoint*, we consider *LastBadRefPoint* and the most recent location in the series as valid (line 22-26). Next, the location update after the validity check is injected to a smoothing filter (called *SmoothingFilter* in algorithm 22), which is implemented by an exponentially-weighted moving average lowpass filter (line 24, 31). The smoothing filter produces a smoothed version of speed profile by cutting off abrupt speed changes. The final step is used to reduce the computational overhead created by the frequent checking of virtual trip line crossings on the output of algorithm 22. Instead of returning a location update at the maximal rate allowed by the GPS receiver, we return a location update only after every T seconds, which is encoded by the function *checkReportingInterval* (line 25, 33). A larger T makes computation of trip line crossings more efficient, but if it becomes too large, valid trip line crossings can be missed and false trip line crossings can be computed.

The output returned from the algorithm 22 is then used by the software routine that computes

virtual trip line crossings from consecutive filtered GPS positions (line 34-48). We check if any line defined by two end positions of each trip line intersects with a trajectory (built by two consecutive filtered GPS positions) in a two dimensional space. If crossed, the algorithm returns a tripline measurement including trip line ID, speed, heading, and timestamp information. All trip lines in downloaded tiles are tested, but limited to valid trip lines. Validity of trip lines can be subject to a combination of trip line's expiration time and user's privacy guidelines.

Algorithm 22. Tripline Crossing Detection Algorithm

```
θ = thresholdToSwitchBadToGood
T = subsampling interval
for all GPS sample l do
    if PrevLocationFiltered is null then
        CurrLocationFiltered = LastGoodRefPoint = l;
        LastLocationUpdateTimestamp = l.t; goto TripLineChecking;
    end if
    TimeGap = l.t - LastLocationUpdateTimestamp;
    LastLocationUpdateTimestamp = l.t;
    if TimeGap is too large then
        LastGoodRefPoint = l; LastBadRefPoint = null; n = 0;
        CurrLocationFiltered = l; PrevLocationFiltered = null;
        goto TripLineChecking;
    end if
    Calculate speed against LastGoodRefPoint;
    if a vehicle has not moved far enough then
        LastBadRefPoint = null; n = 0; CurrLocationFiltered = null;
        goto TripLineChecking;
    else if speed glitch is true then
        Re-calculate speed against LastBadRefPoint;
        if speed glitch is false then
            if ++n is greater than θ then
                n = 0; LastBadRefPoint = null; LastGoodRefPoint = l;
                filteredLoc = SmoothingFilter(LastBadRefPoint, l);
                CurrLocationFiltered =
            end if
            goto TripLineChecking;
        end if
        LastBadRefPoint = l; goto TripLineChecking;
    end if
    n = 0; filteredLoc = SmoothingFilter(LastGoodRefPoint, l);
    LastBadRefPoint = null; LastGoodRefPoint = l;
    CurrLocationFiltered = checkReportingInterval(filteredLoc, T);
    // TripLineChecking
    if both CurrLocationFiltered and PrevLocationFiltered not null then
        traj = SetTrajectory(PrevLocationFiltered, CurrLocationFiltered);
        for all tripline j in each tile(i) do
            if tile(i).status is valid then
                triplineCrossed = CheckCrossing(tripline j, traj);
                if triplineCrossed is true then
                    Compute speed and heading with traj for
                end if
            end if
        end for
```

```
        end if
        if CurrLocationFiltered is not null then
                PrevLocationFiltered = CurrLocationFiltered;
        end if
end for
```

Discussion. The most challenging situation potentially experienced by the above algorithm occurs when the sampling frequency is too slow given the road geometry, and the route driven. The following example of a missed virtual trip line at an intersection illustrates this challenge. During a right turn maneuver at an intersection, if the position is sampled infrequently, then the two consecutive location updates may occur on two different roads, with one update occurring in the middle of the road before the right turn, and one update occurring in the middle of the road after the right turn. Then the straight line segment connecting these two points does not follow the road geometry, and any virtual trip lines placed near the intersection on either road segment will be missed. Moreover, if the sampling interval becomes large enough, the line segment between two consecutive location updates may intersect with virtual trip lines on the road segments not driven by the reporting vehicle, generating false measurements. This problem arises because of our removal of the computationally intensive map matching algorithm.

20.3. Architecture Designs

We present two different architectures, one focused more on traffic estimation accuracy with probabilistic privacy preservation and an improved version that achieves guaranteed privacy using a k-anonymous temporal cloaking. The main purpose of temporal cloaking is to prevent an adversary from compromising anonymity, even in a very low user participation scenario. First, section 20.3.1 describes the common parts for both architectures, then particular changes for each architecture follow in section 20.3.2 and 20.3.3.

20.3.1. Achieving Authenticated but Anonymous Data Collection

In order to achieve the anonymization of measurement uploads from clients while authenticating the sender of the measurements, we split the actions of authentication and data processing into two different entities, which we call the ID proxy server and the traffic monitoring server. By separately encrypting the identification information and the sensing measurements (i.e., trip line ID, speed, and direction) with different keys, we prevent each entity from observing both the identification and the sensing measurements.

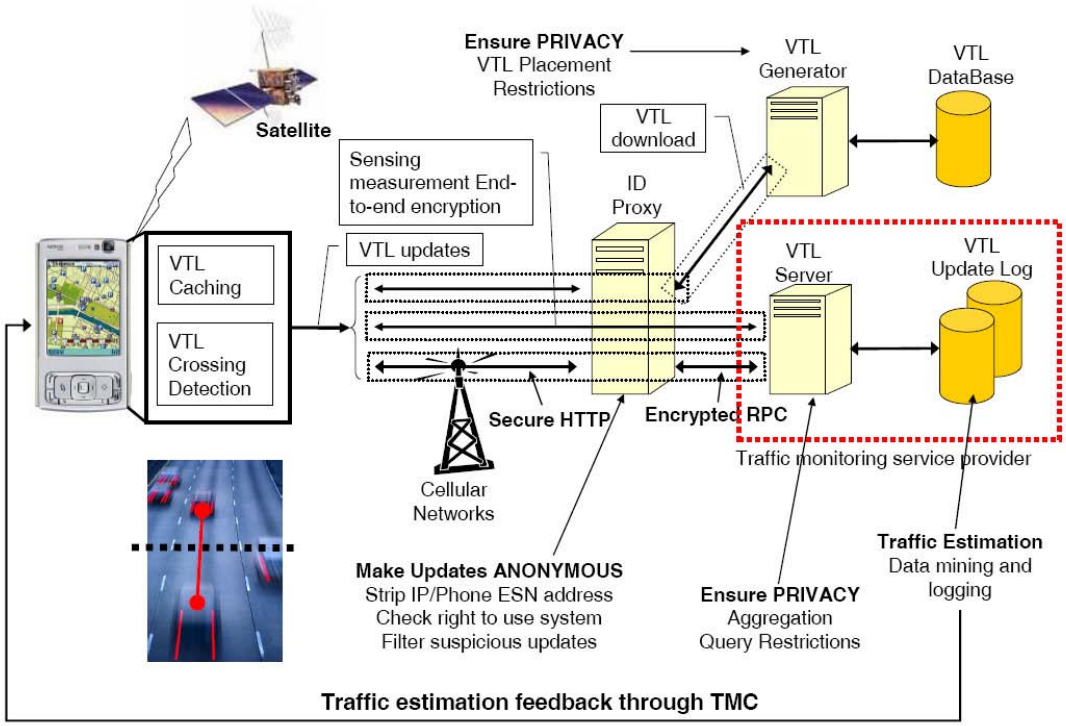


Figure 3.1. Virtual Trip Line: Privacy-Preserving Traffic monitoring System Architecture.

Figure 3.1 shows the resulting system architecture. It includes four key entities: probe vehicles with the cell phone handsets, an ID proxy server, a traffic monitoring service provider, and a VTL generator. Each probe vehicle carries a GPS-enabled mobile handset that executes the client application. This application is responsible for the following functions: downloading and caching trip lines from the VTL server, detecting trip line traversal, and sending measurements to the service provider. To determine trip line traversals, probe vehicles check if the line between the current GPS position and the previous GPS position intersects with any of the trip lines in its cache. Upon traversal, handsets create a VTL measurement including trip line ID, speed readings, timestamps, and the direction of traversal and encrypt it with the VTL server's public key. Handsets then transmit this measurement to the ID proxy server over an encrypted and authenticated communication link set up for each handset separately. The handset and the ID proxy server share an authentication key in advance.

The ID proxy server's responsibility is to first authenticate each client to prevent unauthorized measurements and then forward anonymized measurements to the VTL server. Since the VTL measurement is encrypted with the VTL server's key, the ID proxy server cannot access the VTL measurement content. It has knowledge of which phone transmitted a VTL measurement, but no knowledge of the phone's position. The ID proxy server strips off the identifying information and forwards the anonymous VTL measurement to the VTL server over another secure communication link.

The VTL server aggregates measurements from a large number of probe vehicles and uses them for estimating traffic conditions. The VTL generator determines the position of trip lines, stores

them in a database, and distributes trip lines to probe vehicles when any download request from probe vehicles is received. Similar to the ID proxy server, each handset and the VTL generator share an authentication key in advance. The VTL generator first authenticates each download requester to prevent unauthorized requests and can encrypt trip lines with a key agreed upon between the requester and the VTL generator.[16] Both the download request message and the response message are integrity protected by a message authentication code.

Discussion. The above architecture improves location privacy of probe vehicle drivers through several mechanisms. First, the VTL server must follow specific restrictions on trip line placements that we will describe in section 20.4.1. This means that a handset will only generate measurements in areas that are deemed less sensitive and not send any information in other areas. By splitting identity-related and location-related processing, a breach at any single entity would not reveal the precise position of an identified individual. A breach at the ID proxy would only reveal which phones are generating measurements (or are moving) but not their precise positions. Similarly, a breach at the VTL server would provide precise position samples but not the individual's identities. Separating the VTL server from the VTL generator prevents active attacks that modify trip line placement to obtain more sensitive data. This is, however, only a probabilistic guarantee because tracking and eventual identification of outlier trips may still be possible. For example, tracking would be straightforward for a single probe vehicle driving along on empty roadway at night. The outlier problem in sparse traffic situations can be alleviated by changing trip lines based on traffic density heuristics. Trip lines could be locally deactivated by the client based on time of day or the clients speed. They could also be deactivated by the VTL generator based on traffic observations from other sources such as loop detectors. At the cost of increased complexity, the system can also offer k-anonymity guarantees regardless of traffic density. We will describe this approach next.

20.3.2. Guaranteeing K-Anonymity at Low Density Using Temporal Cloaking

We now demonstrate how virtual trip lines can help computing k-anonymous VTL measurements via temporal cloaking without using a single trusted server. Motivated by a well-known concept called a secret splitting scheme, we distribute secret information through multiple parties so that no central entity has complete knowledge of all three types of information: location, timestamp, and identity information. In doing so, we focus on minimizing any possible degradation of traffic information quality introduced by the information splitting scheme.

[16]While VTL positions are not highly sensitive, encryption reduces the possibility of timing analysis (see section 20.6.1).

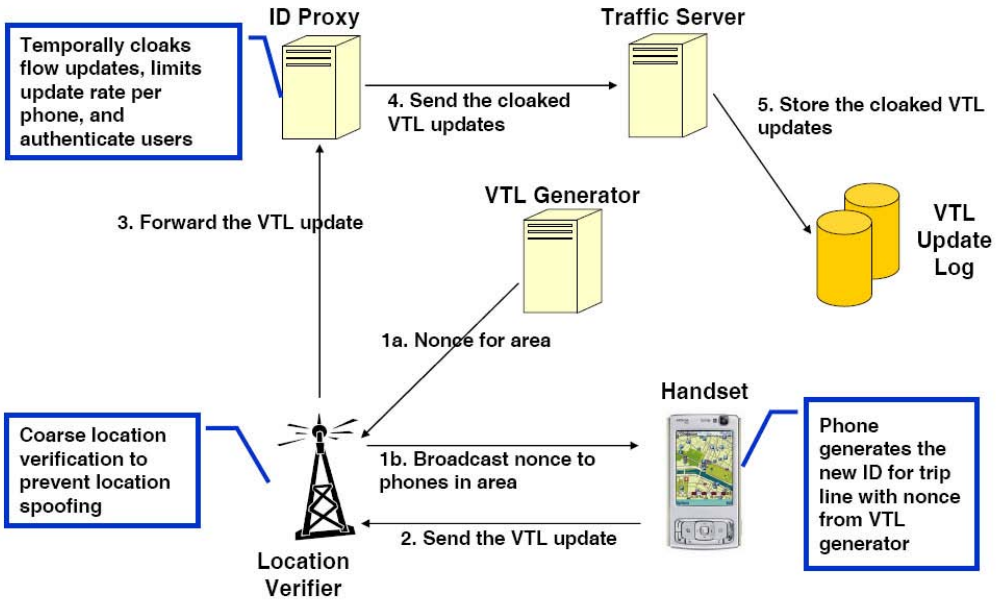


Figure 3.2.

Distributed Architecture for VTL-based Temporal Cloaking.

We propose a distributed VTL-based temporal cloaking scheme that reduces timestamp accuracy to guarantee a degree of k-anony−mity in the dataset accumulated at the VTL server. This provides a stronger privacy guarantee than probabilistic privacy, sinceit prevents the tracking or reidentification of an individual phone even when user participation is very low. The key challenge in applying temporal cloaking is to conceal the locations of the probe vehicles from the cloaking entity. To calculate the time interval for probe vehicles at the same location, the cloaking entity typically needs access to the detailed records of each data subject [175,313], which itself can raise privacy concerns.

Using virtual trip lines, however, it is possible to execute the cloaking function without access to precise location information. The cloaking entity can aggregate measurements by trip line ID, without knowing the mapping of trip line IDs to locations. It renders each measurement k-anonymous by replacing the measurement timestamp with a time window during which at least k measurements were generated from the same VTL (i.e., $k-1$ other phones passed the VTL). In effect, k VTL measurements are aggregated into a new measurement $(vtlid, \frac{s_1 \ldots s_k}{k}, \max(t_1 \ldots t_k))$, where s_i denotes the speed reading of each VTL measurement i. Since now k-phones generate the same measurement, it becomes harder to track one individual phone. The cloaking function can be executed at the ID proxy server, if handsets add a VTL ID to the measurement that can be accessed by the ID proxy server.

Entity	Role	ID	Location	Time
Handset	Sense	Yes	Accurate	Accurate
Verifier	Broadcast VTL ID updates	Yes	N/A	Accurate
ID Proxy	Anonymize and cloak	Yes	N/A	Accurate
Traffic Server	Compute traffic	No	Accurate	Cloaked

Table 23. Splitting of roles and sensitive information across entities.

Beyond the cloaking function at the ID proxy server, two further changes are needed in the architecture to prevent an adversary from obtaining the mapping of VTL IDs to actual VTL locations. The system uses two techniques to reduce privacy leakage in the event of phone database compromises. First, the road network is divided into tiles, and phones can only obtain the trip line ID to location mapping for the area in which the phone is located. This assumes that the approximate position of a phone can be verified (for example, through the cellular network). Second, the VTL server periodically randomizes the VTL ID for each trip line and updates phone databases with the new VTL IDs for their respective location.

This leads to the extended distributed architecture depicted in Figure 3.2, in which again no central entity has knowledge of all three types of information: location, timestamp, and identity information. As before, VTL measurements from phones to the ID proxy server are encrypted, so that network eavesdroppers do not learn position information. It first checks the authenticity of the message and limits the upload rate per phone to prevent spoofing of measurements. It then strips off the identification information and forwards the anonymous measurement to the traffic server. With knowledge of the mapping of VTL IDs to locations, the traffic server can calculate road segment travel times. In this architecture, the ID proxy server cloaks anonymous measurements with the same VTL ID before forwarding to the traffic server. It also requires a location verification entity, which can coarsely verify phone location claims (e.g., in range of a cellular base station) and distribute the VTL ID updates to only the phones that are actually present within a specified tile. Table 23 summarizes the roles of each entity and how information is split across them.

The temporal cloaking approach can be vulnerable to spoofing attacks unless it is equipped with proper protection mechanisms. For instance, malicious clients can send a large number of measurements to shorten the cloaking time window. To prevent this denial of service attack, the ID proxy server limits the upload rate per phone.

To reduce network bandwidth consumption of the periodic VTL updates, clients can independently update the VTL IDs based on a single nonce per geographic area (tile). The VTL generator generates the nonces using a cryptographically secure pseudo random number generator and distributes each nonce and its expiration time to the clients currently in the tile area. Both clients and server can then compute $VTLID_{new} = h(nonce, VTLID_{old})$, where h is a secure hash function such as SHA. Then clients update the ID and the expiration time of each VTL in the current tile. In case that clients do not know the old ID (for example, as they have missed some updates or are new to a tile), the VTL generator still allows clients to download the set of whole VTLs with their new IDs in the tile. Each VTL has an expiration time beyond which its ID becomes invalid. If the connection is accidentally lost during downloading VTLs or the

nonce, clients retry n times more until a successful downloading. The incomplete downloading can be easily checked by the header that includes the total number of VTLs in the corresponding tile (in our implementation). The expiration time of each VTL is used to synchronize the traffic server and clients. Clients decide whether or not to apply the ID update (using the nonce currently downloaded from the VTL generator), depending on whether the current ID of VTLs expires or not. Thus the synchronization based on the expiration time prevents clients from reapplying the ID update to VTLs that are already updated, so that it helps the procedure for calculating $VTLID_{new}$ idempotent.

Temporal cloaking fits well with the travel time estimation method used in the VTL system because the mean speed calculation does not depend on accurate timestamp information. To estimate the travel time, the traffic server calculates the mean speed for a trip line only based on the speed information in the VTL measurements. Typically, the travel time would be periodically recomputed. The use of temporal cloaking adaptively changes this mean speed calculation interval so that at least k phones have crossed the trip line. If k is chosen large, it reduces the update frequency. The rationale for temporal cloaking is that real-time traffic incident information such as congestion, potholes, and accidents requires more accurate location accuracy than timestamp accuracy. Since temporal information can be relaxed to provide enhanced user privacy as long as the monitoring events change relatively slowly, temporal cloaking can be generally applicable to other kinds of incident reports.

20.3.3. Balancing Privacy and Accuracy Requirements

The temporal cloaking architecture has several drawbacks in terms of real-time traffic estimation. First, since the ID proxy server needs to wait until it receives k VTL measurements, the system may fail to reflect brief events and incur unavoidable delay. This impact increases when a larger k is chosen. Second, in order to offer k-anonymity guarantees regardless of user participation rates, the system complexity is increased. Third, when the k measurements are averaged over a large period of time, the resulting measurement cannot be directly integrated into traffic estimation algorithms relying on the dynamics of traffic flow, which are commonly used in the transportation engineering community. To overcome these limitations, we propose an alternative architecture which focuses on real time traffic monitoring accuracy by relaxing the privacy requirements down to probabilistic privacy guarantee. The main idea is to remove k-anonymous temporal cloaking to allow the traffic server to receive k individual anonymous VTL measurements. Thus, at the cost of sacrificing a privacy guarantee, we alleviate system complexity, and enable the use of flow based traffic estimation algorithms.

20.4. Experimental Evaluation

The data collected during the *Mobile Century* field experiment were used to reevaluate our k-anonymous temporal cloaking and its privacy-relaxed version for accuracy improvement. A detailed description regarding the system implementation can be found in the *Mobile Century* final report [269].

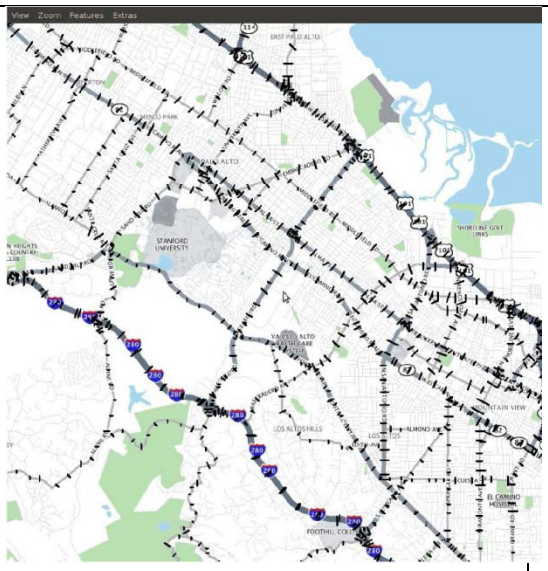
A total of 77 cell phones running the experimental client software were able to properly record

the probe vehicles' positions and velocities, which generated 2200 vehicle trajectories across the experiment site during the eight hour experiment. These trajectories make up between 0% and 5% of the total traffic flow depending on the time of day [181]. Using the data obtained from these vehicles, we were able to assess the impact of virtual trip line spacing and the number of participating vehicles on the accuracy of computing travel times.

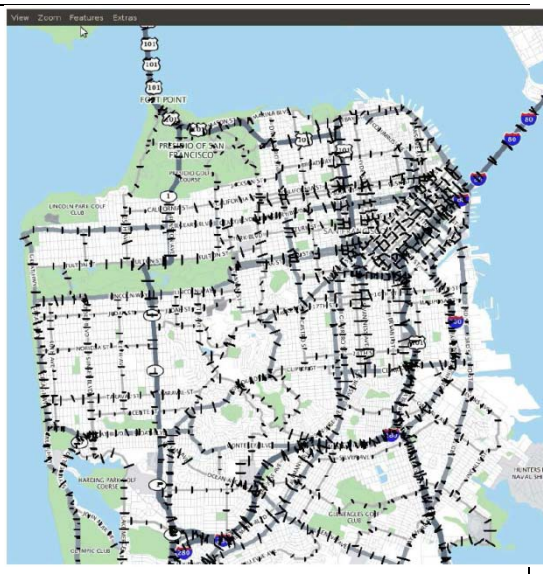
20.4.1. Trip Line Placement

We use the combination of the following techniques to determine the positions of VTLs.

Exclusion Area via Road Category. Privacy can be significantly improved by restricting trip line placement to high traffic roadways, such as highways and arterials, which are also typically less sensitive areas. We extend the concept of exclusion area in our earlier work by restricting placement to these roadways. To determine our placement, we use the road category information provided by the Navteq street database, which classifies each road segment from 1 (highest capacity roads) to 5 (the lowest capacity roads). We only place VTLs on road categories 1 to 3, which avoids trip line placement in residential areas. Figures 4.1(a) and 4.1(b) show examples of virtual trip lines placements in Palo Alto and San Francisco respectively. This approach prevents an adversary from identifying the precise origin and destination of the tracked user in many situations, but it cannot deliver guaranteed privacy protection when sensitive locations are on high capacity road segments.



(a) Palo Alto



(b) San Francisco

Figure 4.1. Example of Virtual Trip Lines Placements.

Equidistant Spacing with Data Obfuscation. This approach takes as input a network graph of road segments in the category of our interest as explained above. For each road segment, defined by stretches of roadway between intersections or merges/diverges, the algorithm places equidistant trip lines orthogonally to the road. A large spacing makes it harder to track

anonymous users as we demonstrated in our earlier study [190]. In the study, we focus the minimum spacing constraint on straight highway scenarios, in which more regular traffic flows increase the tracking risks. Minimum spacing for longer road segments is determined based on a tracking uncertainty threshold. Recall that to prevent linking compromises, an adversary should not be able to determine with high confidence that two anonymous VTL measurements were generated by the same handset. Tracking uncertainty defines the level of confusion that an adversary encounters when associating two successive anonymous VTL measurements to eachother. We define tracking uncertainty as the entropy $H = -\sum p_i \log p_i$, where p_i denotes the probability (from the adversary's perspective) that anonymous VTL measurement i at the next trip line was generated by the same phone as a given anonymous VTL measurement at a previous trip line. The probability p_i is calculated based on an empirically derived pdf model that takes into account the time difference between the predicted arrival time at the next trip line and the actual timestamp of VTL measurement i. We fit an empirical pdf of time deviation with an exponential function, $\widehat{p_i} = \frac{1}{\alpha} e^{-\frac{t_i}{\beta}}$, where we obtain the values of α and β by using unconstrained nonlinear minimization. Higher penetration rates lead to more VTL measurements around the projected arrival time, which decreases certainty. As spacing increases, the likelihood that speeds and the order of vehicles remain unchanged decreases, leading to more uncertainty.

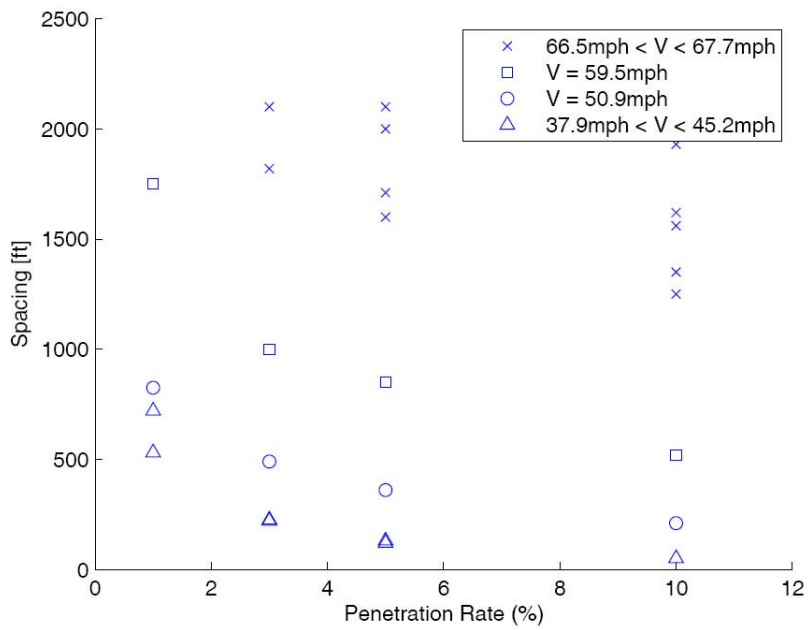


Figure 4.2. Minimum Spacing Constraints for Straight Highway Section.

We empirically validate these observations through simulations using the PARAMICS vehicle traffic simulator [21]. Figure 4.2 depicts the minimum spacing required to achieve a minimum mean tracking uncertainty of 0.2 for different penetration rates and different levels of congestion (or mean speed of traffic). We choose a reasonably low uncertainty threshold, which ensures to an adversary a longer tracking that could have privacy events such as two different places (e.g., origin and destination). Two recent studies [214,191] observe about 15 minutes as a median trip time. The uncertainty value of 0.2 corresponds to an obvious tracking case in which

the most likely hypothesis has a likelihood of 0.97. The penetration rates used were 1%, 3%, 5% and 10%. To evaluate different levels of congestion, we used traces from seven 15 min time periods distributed over one day. We also used three different highway sections (between the junction of CA92 and the junction of Tennyson Rd., between the junction of Tennyson Rd. and the junction of Industrial Rd., and between the junction of Industrial Rd. and the junction of Alvarado-Niles Rd.) to reduce location-dependent effects. The simulations show that the needed minimum spacing decreases with slower average speed and higher penetration rate. The clear dependency of the tracking uncertainty on the penetration rate and the average speed allows creating a model that provides the required minimum spacing for a given penetration rate and the average speed of the target road segment.

20.5. Results

This section first evaluates the performance of the tripline crossing detection algorithm presented in section 20.2. Then, we analyze the travel time estimation accuracy and privacy preservation of our spatial sampling approaches using virtual trip lines. Spatial sampling approaches to be evaluated here include k-anonymous temporal cloaking and its privacy-relaxed version where we strip off the requirement of guaranteed privacy via temporal cloaking. The former is the proposed scheme but the latter is still meaningful in that it is a baseline technique to be compared with the proposed scheme and a less complex system with an acceptable privacy protection in the real world.

20.5.1. VTL Measurement Accuracy

# Sample Types	Number of Samples
Intermittent GPS samples	5
Zigzag GPS samples	745
Speed glitches	13
Good GPS samples	894

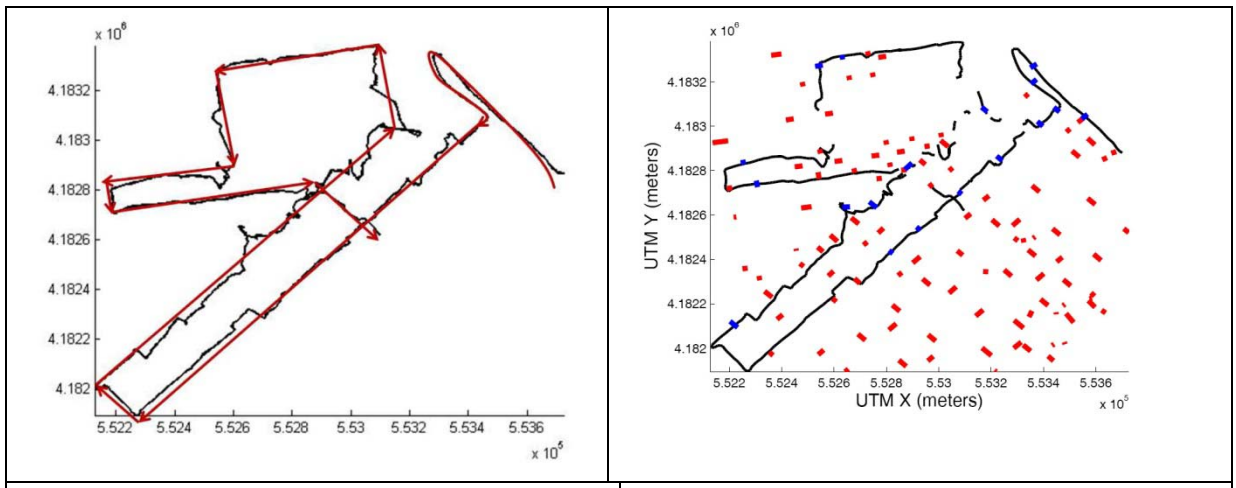
Table 24. Removals of GPS sample errors help reduce the frequency of tripline crossing checks.

	Detection	False Alarm
San Francisco	47%	11%
San Jose	98%	3.6%

Table 25. The worst GPS accuracy in SF approximately degrades the performance of tripline crossing detection algorithm by half.

Tripline Crossing Detection. We observed that GPS position error creates false VTL crossings and drops the detection performance in experimental GPS traces collected in San Francisco downtown, as shown in figure 5.1(a). Collected traces cover Market st., Mission st., Pine st., Bush st., and Washington st., where the worst GPS positioning accuracy is expected, due to highrise buildings and cloudy weather. Figure 5.1(b) illustrates the filtered GPS trace, a

smoothed version of original GPS trace after intermittent GPS samples, zigzag GPS samples, and speed glitches are removed by algorithm 22. Table 24 summarizes the number of removed samples corresponding to each case. The number of "Good GPS samples" are reported 894 samples in our algorithm, but its definition is based on whether the speed of two successive filtered locations lies within a valid range. Thus if a map-matching algorithm is additionally applied to the collected trace, the number of "Good GPS samples" should be larger. To observe the dependency of the presented algorithm on GPS positioning accuracy and wireless connectivity, we collected traces in San Jose downtown and measured the detection probability and false alarm probability of VTL crossings for both cities. In San Jose (where better GPS accuracy is expected than San Francisco), the presented algorithm detects 98% of VTLs placed on the route; only 3.6% of reported crossings were false. However, the detection probability was significantly degraded to 47% in San Francisco, and the false alarm probability increased up to 11%. Dense road network in San Francisco downtown makes the situation worse even with a few meter GPS error. In this experiment, there was a very slight change in the number of false alarms and detections for the two different situations (with the algorithm and without the algorithm) since removals of GPS samples do not coincide with the VTL locations in the collected traces. However, the benefit of these removals should be observed if more traces are tested. To see how many false crossings these removals would potentially save in unfiltered GPS traces, we count the number of crossings that unfiltered and filtered GPS traces create on wrong road segments and compare with the counts. Unfiltered GPS traces create 17 crossings on wrong road segments while filtered GPS traces have 9 crossings. If we consider these wrong road segment crossings into false alarm probability assuming that VTLs are placed dense enough to coincide with these wrong crossings, unfiltered GPS traces have almost two times false alarm probability compared to filtered GPS traces. We find that the presented algorithm has two major benefits; it removes potential false alarms by removing GPS position errors and removing samples (almost 50% of GPS samples removed in this experiment) helps reduce the frequency of tripline crossing checking, thereby relieving the computation overhead.



(a) GPS Traces: ground truth (red) vs. unfiltered (black).	(b) VTLs and Filtered GPS Traces (black).

Figure 5.1. In the right figure, red colored lines denote VTLs placed in downtown while blue lines denote detected VTL crossings.

20.5.2. Guaranteed Privacy via VTL-based Temporal Cloaking

To evaluate the performance of VTL-based temporal cloaking, we compute its travel time estimation accuracy in offline mode with the collected traces from *Mobile Century*. The procedure for computing travel time consists of three steps. First, we divide the I-880 northbound highway segment (used in the experiment) into multiple sections, putting one VTL in the middle of each section. Second, we compute the speed profile for each section, where the speed profile denotes the change of mean speed overtime. The mean speed is updated when the VTL on the section receives k-anonymous VTL measurements. Lastly, we compute the time taken to traverse each section and compute the sum from the first section to the last one. To compute the travel time for each section, we read the initial speed of the vehicle at the moment of entering the section from the speed profile of the section and let the vehicle follow the speed profile of the section until the vehicle exits the section.

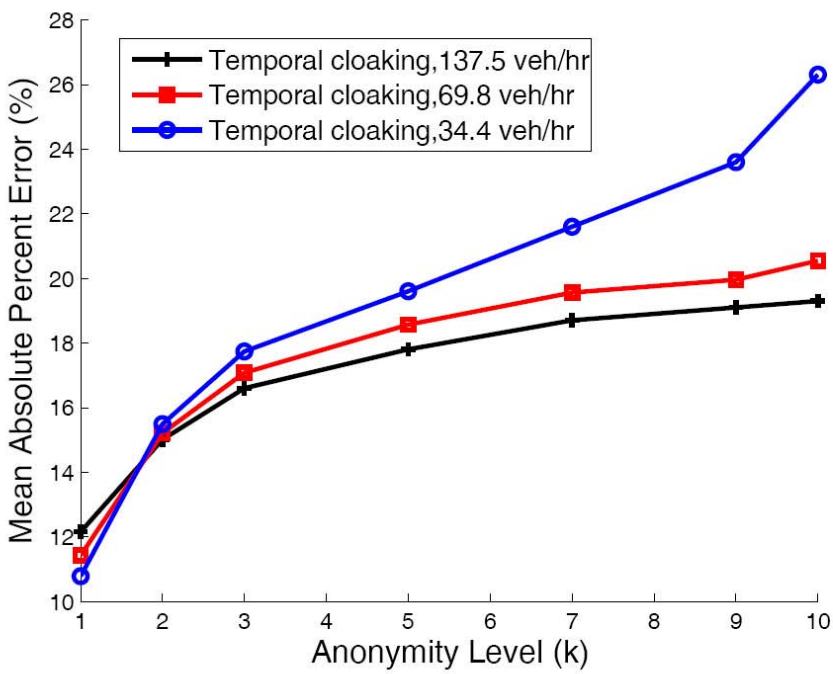


Figure 5.2. Travel Time Accuracy versus Anonymity Level using 10 VTLs/mile.

To see the effect of k on travel time estimation accuracy, we vary k up to 10. In order to see the sensitivity of travel time estimation accuracy on penetration rates, we control the penetration rate by respectively using the full set of probe vehicles (about 137.5 veh/hr), half of them, and one fourth of them. Figure 5.2 shows that temporal cloaking achieves less than 18% travel time error using $k = 5$ and a probe rate of 137.5 veh/hr, which corresponds to about 2% penetrationrate in the morning and about 1% in the afternoon [181]. The cases for $k = 1$ can be considered periodic sampling techniques with the same number of VTL measurements collected as temporal cloaking. Compared to a periodic sampling, the proposed scheme sacrifices about 5% accuracy to achieve the guaranteed privacy (k=5).

20.5.3. Reconstructing VTLID-Location Mapping

Concealing the mapping between each VTL ID and its location is a key enabler of temporal cloaking. However, the mapping can be partially reconstructed by an active attack at the level of the compromised ID proxy server. For example, let us consider a scenario in which a malicious ID proxy server performs an active attack with a small fraction of handsets and the VTL generator refreshes each VTL ID every 10 minutes. Each compromised handset sends VTL measurements associated with random VTL identifiers and timestamps to the ID proxy server. Later, all VTL measurements can be cross-checked against GPS logs (containing GPS position with timestamp) collected separately by a compromised vehicle.

To evaluate the difficulty of reconstructing the VTL ID and location mapping that is randomly changed by a secure hash function and a nonce, we use the VTL database that contains all virtual trip lines placed over the United States. To build the database, we ran the automated algorithm explained in section 1 with an average spacing of 1000ft. The 90 percentile of tiles in SF Bay area have about 500 VTLs as shown in figure 5.3, so that the total length of roads covered by VTLs would be $500{,}000ft \simeq 94miles$. Following the attack described above, an ID proxy server would require about 14 vehicles (assumed to run 40mph) per tile to reveal the mapping of all VTLs. As more frequent VTL ID updates are used to randomize the mapping, the number of compromised vehicles required per tile should increase linearly for reconstruction. In Los Angeles metro and New York, for example, more number of compromised vehicles (equipped with handsets) are required to cover larger number of VTLs due to their more dense road networks.

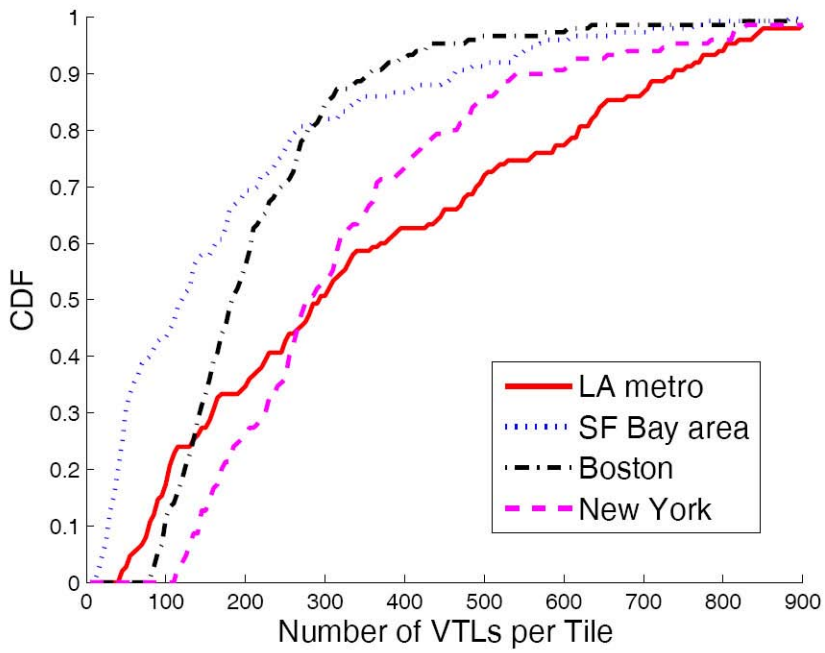


Figure 5.3. The CDFs of number of virtual triplines per tile (8km by 8km) in different major cities in US.

20.5.4. Accuracy-Centric Architecture

In order to compute travel times from VTL data with temporal cloaking relaxed, we use a highway traffic estimation algorithm which estimates the average velocity field along the roadway as described in Chapter 14. For the numerical experiments presented next, a subset of virtual trip lines and a subset of the participating vehicles are selected for northbound I880 and the resulting measurements are fed into the velocity estimation algorithm. The impact of the virtual trip line spacing and the number of participating vehicles on the travel time accuracy are shown in Figure 5.4. Each curve corresponds to a different number of equipped vehicles, ranging from 13.75 veh/hr (10% of the 2200 trajectories) to 137.5 veh/hr (100% of the 2200 trajectories). Similarly, we adjusted the number of trip lines deployed on the experiment site, from nine trip lines to 99 trip lines in increments of 10. Figure 5.4 shows how improvements in accuracy can be achieved either by increasing the number of vehicles sending measurements or by increasing the number of locations where measurements are obtained from a fixed number of vehicles. In the case in which numerous vehicles are participating and the virtual trip line spacing is sparse, the experiment shows that it is possible to reconstruct travel times with less than 10% error while maintaining a high degree of anonymity for the participating users. Furthermore, the travel times can be computed without measurements of the travel times of the equipped vehicles, which would have required disclosure of the full vehicle trajectories.

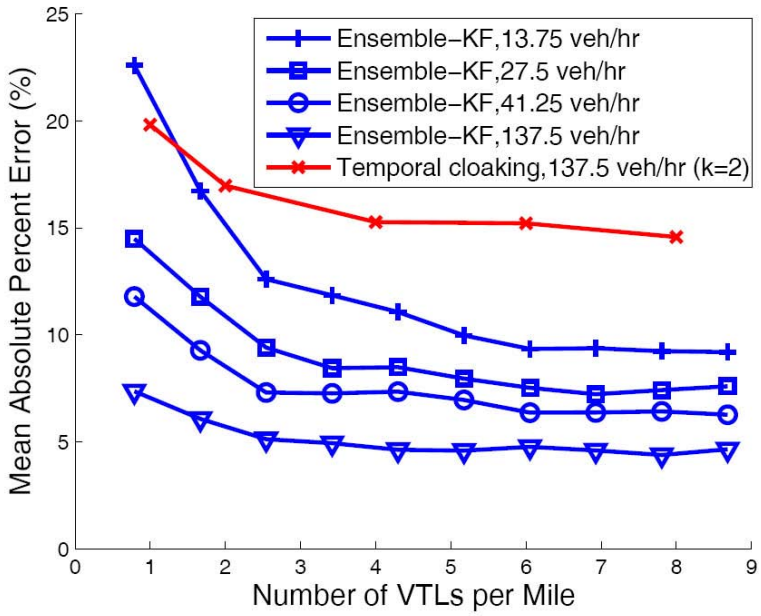


Figure 5.4. Tradeoffs between number of virtual trip lines per mile and the travel time error for different values of the number of equipped vehicles sending measurements per hour.

Compared to temporal cloaking (at best when $k = 2$ and full probe vehicles used), the accuracy centric architecture enhances the travel time estimation error by almost 10% (achieving about 5% travel time estimation error when more than two VTLs are place per mile), which is again even better than periodic sampling techniques. For example, 2.5 VTLs per mile has about 2100ft spacing, which easily meets the minimum spacing constraint of 1750ft that maintains the tracking uncertainty less than 0.2 for roads where 1 to 10% penetration rates of probe vehicles run at the average speed of 0 to 60mph as shown in figure 4.2. The comparison between the accuracy centric architecture and temporal cloaking architecture demonstrates the price the system designer pays for privacy, which is about 10% accuracy reduction. Moreover, even when the penetration rate of probe vehicles for the accuracy centric architecture is 1/10 that of the temporal cloaking architecture, the accuracy centric architecture produces travel time estimates with lower error. So the guaranteed privacy comes at the cost of significantly more vehicles required to achieve the same level of accuracy.

In Figure 5.5, we show a comparison of the mean travel time obtained from our video data compared to the mean travel time computed using our spatial sampling approaches using virtual trip lines. This comparison corresponds to a mean absolute percent error of about 5% for the accuracy centric architecture, which was achieved using 100% of our equipped vehicles and about 8.6 VTLs per mile. The largest error in this simulation occurs in the morning around 10:40 AM, with an error of about six minutes. The high travel times experienced by drivers at this time are caused by a five car accident. The estimation algorithm performs poorly here for two reasons. First, the traffic model used in the estimation algorithm does not predict accidents. Second, because the accident occurred at the beginning of the experiment, some of the equipped vehicles had not yet been deployed resulting in few measurements to correct the model.

Throughout the rest of the day, the estimated travel times are significantly closer to the mean travel times. The Temporal cloaking approach used for the comparison achieves about 15% travel time error, where k is set to two and 100% of our equipped vehicles upload VTL measurements from 8 VTLs per mile.

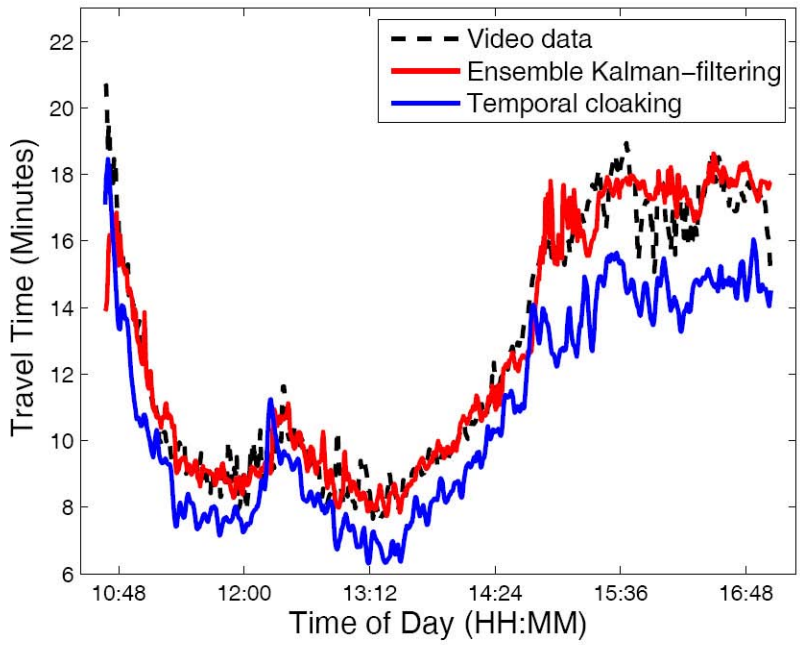


Figure 5.5. Comparison between mean travel time obtained from video data, mean travel time obtained from temporal cloaking, and mean travel time obtained from the accuracy centric architecture.

20.6. Discussion

We now discuss limitations and outlooks of our presented approaches as well as share lessons learned from the field operational deployment.

20.6.1. Security

The proposed architecture significantly improves privacy protection over earlier proposals, by distributing the traffic monitoring functions among multiple entities, none of which have access to location, timestamp, and identity records at the same time.

The system protects privacy against passive attacks under the assumption that only a single infrastructure component is compromised. One passive attack that remains an open problem for further study is timing analysis by the ID Proxy server or by network eavesdroppers between the Location verifier and the ID Proxy server. For the case of an adversary at the ID Proxy server, the adversary can hire multiple handsets (their IDs known to the adversary) and ask them to move around a target area. By comparing the GPS traces driven by those handsets with their trip

line updates, the adversary can learn the exact trip line locations. In addition, the adversary could estimate the time needed to travel between any two trip lines from public travel time information on the road network. Then the adversary could attempt to match a sequence of observed VTL update message inter-arrival times to these trip line locations. This attack can be also conducted by network eavesdroppers passively observing the channel between the Location verifier and the ID Proxy server. One may expect that the natural variability of driving times provides some protection against this approach. Protection could be further strengthened against network eavesdroppers by inserting random message delays on the handset (client application) side. Under the temporal cloaking scheme, however, the ID proxy server also obtains trip line identifier information. If trip line identifier information is used for extended durations, an adversary may match them to actual VTL positions based on the sequence in which probe vehicles have passed them. This threat can be alleviated through frequent VTL ID updates. Quantifying these threats and choosing exact tile size and update frequency parameters to balance privacy and network overhead concerns remain open research problems.

The system also protects the privacy of most users against active attacks that compromise a single infrastructure component and a small fraction of handsets. It does not protect user privacy against injecting malware directly onto users' phones, which obtains GPS readings and transfers them to an external party. This challenge remains outside of the scope of this paper, because this vulnerability is present on all networked and programmable GPS devices even without the use of a traffic monitoring system. Instead, the objective of the presented architecture is to limit the effect of such compromises on other phones. For the temporal cloaking approach, compromised phones result in two concerns. First, an adversary at the ID proxy can learn the current temporary trip line IDs. To limit the effectiveness of this attack, the architecture periodically changes trip lines and verifies the approximate location of each phone so that a tile of trip line updates can only be sent to phones in the same location. Second, a handset could spoof trip line updates at a certain location to limit the effectiveness of temporal cloaking. Our proposed architecture already eliminates updates from unauthorized phones and can easily limits the update rate per phone and verify that the approximate phone position matches the claimed update. This renders extended tracking of individual difficult because it would either require a large number of compromised phones spread around the area in which the individual moves, or set of compromised phones that move together with the individual. The system could also incorporate other sanity checks and blacklist phones that deliver suspicious updates.

The same methods also offer protection against spoofing attacks that seek to reduce the accuracy of traffic monitoring data. The system does not offer full protection against any active attack on traffic monitoring accuracy, however. For example, a compromised ID proxy could drop messages to reduce accuracy. These challenges remain an open problem for further work.

As in any secret-splitting scheme, the proposed architecture cannot offer protection if adversaries within the different entities collude or if an adversary manages to break into multiple entities. Experience from current privacy violations has shown, however, that the vast majority are due to accidental disclosures or a single disgruntled or curious insider [172,229]. If implemented correctly, no individual insider would have access to more than one of the proposed entities, thus our secret-splitting architecture provides adequate protection against this important class of privacy violations.

20.6.2. Involvement of Cellular Networks Operators

While this work was based on cellular handsets, the question of how to improve location privacy within cellular networks themselves was outside of the scope of this work. The Phase II E911 requirements [9] mandate that cellular networks be able to locate subscriber phones within 150-300m 95% of the time, and A-GPS solutions should achieve similar accuracy. In addition, phones are identifiable through IMSI (International Mobile Subscriber Identity, in the GSM system) and operators typically know their owner's names and addresses. While precise phone location information is accessible, to our knowledge, it is not widely collected and stored by operators at this level of accuracy.

This work investigated how traffic monitoring services can be offered without access to sensitive location information. It was primarily motivated by third party organizations that currently do not yet have access to identity and location information and want to implement privacy-preserving traffic monitoring services. Our solution is general enough so that in actual implementations, different levels of involvement of network operators are possible. One case may be four separate organizations, each operating a different component of the system with no involvement of the network operator.[17] Another extreme case would be a cellular network operator creating separate entities within the company to protect itself against dishonest insiders and accidental data breaches of their customers records. Clearly, the first would be preferable from a privacy perspective, but in the end both lead to a significant improvement in privacy over a naive implementation.

20.6.3. Challenges in Arterial Roads Traffic Estimation

In comparison to highway traffic, arterials present additional challenges. The underlying flow physics that governs arterials is more complex (traffic lights often with unknown cycles, intersections, stop signs, parallel queues). While our work [105,104,342] explicitly derives techniques to reconstruct traffic from VTL type data, such a reconstruction becomes harder for arterials. Also, while macroscopic flow models such as the ones used in [105,104,342] exist and can be used for secondary networks [160,303], their parameters are in general unknown or inaccessible and only documented for few cities, making their use impossible without going to the field and measuring them. In addition, even if they were known, the complexity of the underlying flows makes it challenging to perform estimation of the full macroscopic state of the system at low penetration rates. In light of these challenges, statistical approaches for characterizing a subset of the macroscopic state (for example travel times and aggregated speeds) have proved to work well and seem to be one of the only alternatives to traffic flow model based traffic reconstruction [154,267]. Ongoing work has focused on sampling policies for arterial networks [184,189].

[17]The only limitation is that for temporal cloaking one of the identities needs to be able to approximately (at the level of a tile size) verify client location claims. This verification could be provided by a network operator but other forms of verification are also plausible.

20.6.4. The Mobile Millennium Field Operational Test

For one year starting in November 2008, the *Mobile Millennium* pilot project was deployed in Northern California. Residents of the Bay Area were able to download a traffic client on Java enabled mobile phones, which displayed real time traffic conditions while collecting virtual trip line data. Unlike the *Mobile Century* experiment described earlier, *Mobile Millennium* monitored traffic conditions on highways and arterials continuously throughout the year.

This pilot project highlighted a fundamental challenge for launching privacy preserving traffic monitoring systems using GPS data, which is to achieve high participation rates amongst the driving public when launching the system. Even with more than 5000 application downloads, only a few hundred users ran the *Mobile Millennium* application at any given time across a large geographic area. Thus, the resulting data from these devices was sparse both in space and time. At low participation rates, an architecture relying on temporal cloaking is insufficient to monitor real–time traffic conditions accurately due to the latency required for anonymity. At the same time, without temporal cloaking, reidentification of users at low participation rates becomes much easier.

20.7. Conclusions

This chapter described traffic monitoring system implemented on GPS smartphone platform. The system uses the concept of virtual trip lines to determine when phones reveal a location update to the traffic monitoring infrastructure. We demonstrated that the introduced scheme, Virtual Trip Lines, successfully addresses known weaknesses of probe vehicle based traffic monitoring. First, the VTL paradigm achieves strong anonymity through k-anonymous temporal cloaking. Virtual trip lines allow the application of temporal cloaking techniques to ensure k-anonymity properties of the stored dataset, without having access to the actual location records of phones. Second, they improve the accuracy of traffic monitoring. We show that the temporal cloaking leads to less than 5% reduction in the accuracy of travel time estimates for k values less than 7 compared to periodic sampling techniques and a privacy-relaxed version achieves 5% travel time estimation error using only 1-2% penetration rate. Third, VTLs enable a light-weight client algorithm for collecting VTL measurements, and we achieve the VTL crossing detection between 50% to 98% in downtowns while suppressing false alarm less than 11% without map-matching.

21. Kernel regression for travel-time estimation via convex optimization

This chapter explores a kernel regression technique for the estimation of travel times along a signalized arterial. The technique allows the combination of multiple kernels to improve the accuracy of the final estimates. In this example, the resulting estimates are able to capture the evolution of travel times generated by a paramics simulation. Although this technique is not yet ready for real time application, it opens the door to further innovative schemes in this area of open research.

In terms of the mathematics, this algorithm utilizes a convex optimization framework. Sampled travel-times from probe vehicles are assumed to be known and serve as a training set for a machine learning algorithm to provide an optimal estimate of the travel-time for all vehicles. A kernel method is introduced to allow for a non-linear relation between the known entry times and the travel-times that we want to estimate. To improve the quality of the estimate we minimize the estimation error over a convex combination of known kernels. This problem is shown to be a semi-definite program. A rank-one decomposition is used to convert it to a linear program which can be solved efficiently.

21.1. Background

Travel-time estimation on transportation networks is a valuable traffic metric. It is readily understood by practitioners, and can be used as a performance measure [96] for monitoring applications. The travel-time estimation problem is easier to address on highways than on arterial roads. This is related to the fact that properties of highways can be considered to be 'more spatially invariant' than arterial roads. Indeed the latter present complex features such as intersections and signalization forcing to stop resulting in spatially discontinuous properties.

In this chapter, we describe a new method to estimate travel-time on road segments without any elaborated model assumption. This method is shown to belong to a class of convex optimization problems and provides a non-linear estimate of the travel-time. The kernel regression method introduced here allows for estimation improvement through the online extension of the set of kernels used. In particular we will assess the performance of this technique through a rank one

kernel decomposition.

Travel-time estimation on highways has been approached with different tools. Efforts have been made from the modeling side, assuming local knowledge of density or speed, and producing an estimate given by a deterministic or stochastic model [110,205,271]. This problem has also been addressed using data analysis and machine learning techniques with various types of learning methods [259,268,284,228,346].

Arterial travel-time estimation is more complex because the continuum approximation of the road might not apply at intersections, where the dynamics are not easily modeled [64]. Information about the state of traffic on arterials is also limited because of the sparsity of sensors. Some attempts have been made to estimate travel-time on arterials, but in practice it is not always possible to know the traffic lights cycles or to obtain a dedicated fleet of probe vehicles, often needed for estimation [303,308]. However ubiquitous GPS now enable us to realistically assume the knowledge of sampled travel-times.

Herein, we focus on arterial travel-time estimation using machine learning techniques and convex optimization. We use kernel methods [293] to provide a non-linear estimate of travel-time on an arterial road segment. We assume the knowledge of the travel-times of a subset of vehicles and estimate the travel-time of all vehicles. We use convex optimization [78] to improve the performance of the non-linear estimate through kernel regression. The regression is done on a set of kernels chosen according to their usually good performances, or physical criteria. The kernel framework [119] enables us to add features of the initial objects which are used in the regression problem. The kernel regression gives the possibility to select the most relevant features through an optimization problem.

This chapter is organized as follows. In section 21.2, we describe the optimization problem, introducing the regularization parameter and the kernel in a learning setting. In section 21.3, we pose the kernel regression problem and show that it can be written as a convex optimization problem, transform it into a linear program, which can be solved efficiently and we describe the general learning algorithm used. In section 21.4, we present the simulation dataset used for validating the method and the results obtained. In particular, we discuss the theoretical results stating that kernel regression enables to obtain a better estimate on the validation set. Finally, based on these results, we enumerate in section 21.5 ongoing extensions to this work.

21.2. Problem Statement

21.2.1. Travel-time estimation

We investigate travel-time estimation for a given road segment. Assuming a set of entry times and travel-times on the section, we want to apply machine learning techniques to use the knowledge of a subset of the couples entry time, travel-time, in order to produce an estimate of travel-time for every entry time. The dataset used for validation is described in section 1. We assume the knowledge of a dataset of size N which reads $S = \{(x_i, y_i) \in \mathbb{R}^+ \times \mathbb{R}^+ | i = 1 \ldots N\}$ where for each value of the index $i = 1 \cdots N$, x_i is an entry time on the road section and y_i is the realized travel-time (known as the *a-posteriori travel time* in transportation) for entry time

x_i. We would like to learn a function $h: \mathbb{R}^+ \to \mathbb{R}^+$ which given S, would provide an estimate of the travel-time y for any $x \in S$. This is a typical regression problem as described in [119]. The well-known unconstrained least-squares method can be formulated as an optimization problem.

$$\min_\theta \quad \| y - x^T \theta \|_2^2 \tag{129}$$

where $y \in \mathbb{R}^{N\times 1}$ is the vector of realized travel-time or output, $x^T \in \mathbb{R}^{N\times 1}$ is the vector of entry time or input. One must note that here x is a row vector and y is a column vector so θ is a scalar. The well-known solution of this problem can be computed as:

$$\theta_{\text{opt}} = -(x\, x^T)^\dagger\, x\, y \tag{130}$$

where the notation $(x\, x^T)^\dagger$ denotes the pseudo-inverse of $(x\, x^T)$ and the optimal estimate is given by $\hat{y} = -(x\, x^T)^\dagger\, x^T\, x\, y$. This estimate does not have bias, i.e. the mean of the output y equals the mean of the estimate $\hat{y}$.

21.2.2. Regularization

The regression problem defined in (129) is often ill-posed in the sense that the solution does not depend continuously on the data (the case of multiple solutions falls into that denomination). Formulation (129) could also lead to over-fitting in the case of non-linear regression since there is no penalization for high values of the solution θ_{opt}. In order to prevent these two possible flaws, it is a common practice to add to the objective function a quadratic termcalled *Tikhonov regularization* [319] which has the form $\rho^2\, |\theta|^2$ in the scalar case. Then the optimal estimate becomes:

$$\hat{y} = -(x\, x^T + \rho^2)^{-1}\, x^T\, x\, y. \tag{131}$$

For ρ large enough, the problem is well-posed and over-fitting with respect to θ is prevented [137].

21.2.3. Kernel methods

The regression method described in section 130 in the linear case can be extended to the non-linear case through the use of a kernel. One can consider a mapping function $\phi(\cdot)$ and consider the linear regression problem between the mapped inputs $\phi(x_i)$ and the outputs y_i. This is the main principle of kernel methods, which consist in using a feature space, in which the dataset is represented, and to consider linear relations between objects in this feature space. Given a positive semi-definite matrix $K = (g_{ij})$; we define the kernel function by $K_{\text{f}}: X \times X \to \mathbb{R}$ such that $K_{\text{f}}(x_i, x_j) = K_{ij}$. This implicitly defines a feature mapping $\phi(\cdot)$ between the input set X and a Hilbert space H by $\phi(\cdot): X \to H$ such that $< \phi(x_i), \phi(x_j) >_H = K_{\text{f}}(x_i, x_j)$. In the following we will note x_{map} the matrix whose i-th column is $\phi(x_i)$. When $\phi(\cdot)$ has scalar values, then x_{map} is a row vector.

Remark 18. One does not have to define a mapping function $\phi(\cdot)$ to define a kernel matrix, but can simply take a positive semi-definite matrix and use it a a kernel. Of course, it is also possible to define the kernel matrix from the mapping as $K = x_{\text{map}}^T x_{\text{map}}$.

The inner product in H naturally appears to be given by the value of the Gram matrix K, called the kernel. Kernel techniques [119,293] have several benefits:

- They enable us to work with any types of features of the initial data-set, which has a priori no particular structure, in a Hilbert space.
- They guarantee the computational cost by allowing a complexity related to the number of points represented and not the number of features used (this is known as the kernel trick and will appear in the next section).

Thus, kernel methods provide several extensions to usual regression methods, and can be easily written in a machine learning framework.

21.2.4. Learning setting

W assume the knowledge of a training set $S_{\text{tr}} = \{(x_i, y_i) | i = 1 \cdots n_{\text{tr}}\}$ and we look for the best estimate of the elements of the test set, $S_{\text{t}} = \{x_i | i = n_{\text{tr}} + 1 \cdots n_{\text{tr}} + n_{\text{t}}\}$. In order to match the structure of this problem, we will define the kernel matrix in block form as:

$$K = \begin{pmatrix} K_{\text{tr}} & K_{\text{trt}} \\ K_{\text{trt}}^T & K_{\text{t}} \end{pmatrix} \tag{132}$$

where $K(i,j) =< \phi(x_i), \phi(x_j) >_H$ for $i,j = 1 \cdots n_{\text{tr}}, n_{\text{tr}} + 1 \ldots n_{\text{tr}} + n_{\text{t}}$. The Gram matrix K_{tr} is the result of an optimization problem over the training set, and we learn the cross-term K_{trt}, which expresses the inner-product in the feature space between the elements of the test-set and the elements of the training set.

21.3. Analysis

21.3.1. Convex formulation

Expressing the linear least-squares (129) for the mapped input x_{map} and with the regularization term described in section 131 yields:

$$p^* = \min_\theta \quad \| y - x_{\text{map}}^T \theta \|_2^2 + \rho^2 \ \| \theta \|_2^2 \tag{133}$$

where we note p^* the optimal value of this problem. Using the change of variable $z = x_{\text{map}}^T \theta - y$ yields the equivalent formulation:

$$p^* = \min_{\theta,z} \ \| z \|_2^2 + \rho^2 \ \| \theta \|_2^2 \tag{134}$$

$$\text{subject to } \ z = y - x_{\text{map}}^T \theta \tag{135}$$

The lagrangian dual of this problem reads:

$$d^* = \max_\alpha \quad -2\,\alpha^T\, y - \alpha^T \left(I + \frac{x_{map}^T\, x_{map}}{\rho^2} \right) \alpha \tag{136}$$

And in this equation we see the expression of the kernel matrix

$$K = x_{\text{map}}^T\, x_{\text{map}}. \tag{137}$$

If we denote $K_\rho = I + \frac{x_{map}^T\, x_{map}}{\rho^2}$ the regularized kernel, the dual optimal point and dual optimal value of problem (136) can be expressed as:

$$\alpha^* = K_\rho^{-1}\, y \quad and \quad d^* = y^T\, K_\rho^{-1}\, y \tag{138}$$

Remark 19. Since the primal (134)-(135) and dual (136) are convex and strictly feasible, strong duality holds and primal optimal value p^* and dual optimal value d^* are equal. We note that expression (136) shows that the dual optimal value is a maximum over a set of linear functions of $K_{\rho,\phi}$, so the optimal value is a convex function of the regularized kernel matrix K_ρ. Since the choice of the kernel is crucial for the optimal value it is interesting to minimize the optimal value d^* with respect to the kernel.

Remark 20. Optimizing the kernel matrix physically means looking for the best mapping function $\phi(\cdot)$ such that there is a linear relation between the features of the inputs $\phi(x_i)$ and the outputs y_i. If one takes $\phi(\cdot)$ as the identity mapping, then the optimal value of (136) becomes:

$$y^T\, K_\rho^{-1}\, y = y^T \left(I + \frac{x^T\, x}{\rho^2} \right)^{-1} y \tag{139}$$

which may not be optimal for non-linear systems.

Remark 21. One must note that the kernel matrix (137) is a square matrix which has the dimension of $x^T\, x$, and thus its size does not depend on the number of features represented in x_{map}. The dimension of the image space of $\phi(\cdot)$ which is the dimension of the feature space, does not appear in the kernel matrix. This is the kernel trick mentioned in section 21.2.3.

21.3.2. Cross-validation

The optimal value of (133) as expressed in (138) depends on the kernel matrix (137) and on the

regularization parameter ρ. The parameter ρ is tuned through a re-sampling procedure [135], the K-fold cross-validation (here K does not denote the kernel matrix). This technique consists in dividing the dataset into K parts of approximately equal size, and using a subset for training while the remainder is used for testing [323]. For instance if the different parts are $\{P_i | i = 1 \cdots K\}$ then given $n \in \{1 \cdots K\}$ one would use P_n as a training set and $\bigcup_{i=1 \ldots K,\ i \neq n} P_i$ as a test set. This is useful to make extensive use of the dataset while avoiding bias of the training set. Here we use this method on the training set to pick the optimal value of the regularization parameter ρ.

21.3.3. Kernel regression

As stated in remark 19, the optimal value of the regularized regression problem (133) is a convex function of the regularized kernel matrix K_ρ and can be optimized. The kernel optimization problem, which consists in minimizing the value d^* defined in (138) with respect to the regularized kernel K_ρ reads:

$$\min_{K_\rho} \ y^T K_\rho^{-1} y \tag{140}$$

$$\text{subject to} \quad K_\rho \geq 0 \tag{141}$$

where the constraint on the kernel matrix enforces that the regularized kernel K_ρ must be a Gram matrix. This problem is convex according to remark 19. In order to prevent over-fitting with respect to K_ρ, we follow [218] and constrain K_ρ to be a convex combination of given kernels, i.e. we define a set of kernels $\{K_1 \cdots K_k\}$ and consider the problem:

$$\min_\lambda \ y^T K_\rho^{-1} y \tag{142}$$

$$\text{subject to} \quad \lambda_i \geq 0 \qquad \sum_{i=1}^k \lambda_i = 1 \tag{143}$$

$$K_\rho = \sum_{i=1}^k \lambda_i \ K_i \tag{144}$$

In a learning setting, the optimization problem (142) is defined only on the training set but the expression of the kernel matrix as a linear combination of known kernels must be satisfied on the whole set. Using the notation introduced in section 132 we write $K_\rho = \begin{pmatrix} K_{\text{tr}} & K_{\text{trt}} \\ K_{\text{trt}}^T & K_{\text{t}} \end{pmatrix}$ and under this form the problem reads:

$$\min_\lambda \ y^T K_{\text{tr}}^{-1} y \tag{145}$$

$$\text{subject to} \quad \lambda_i \geq 0 \qquad \sum_{i=1}^k \lambda_i = 1 \tag{146}$$

$$K_\rho = \sum_{i=1}^k \lambda_i \ K_i \tag{147}$$

which can be written in a semi-definite program form using an epigraph property and the Schur

complement:

$$\min_{\lambda,t} \; t \tag{148}$$

$$\text{subject to} \quad \lambda_i \geq 0 \qquad \sum_{i=1}^{k} \lambda_i = 1 \tag{149}$$

$$K_\rho = \sum_{i=1}^{k} \lambda_i \; K_i \;\; and \begin{pmatrix} t & y^T \\ y & I + \frac{K_{tr}}{\rho^2} \end{pmatrix} \geq 0 \tag{150}$$

The solution of this optimization problem is the parameter λ^* giving the optimal convex combination of the set of kernels $\{K_1 \cdots K_k\}$ which minimizes d^* defined in (138).

21.3.4. Rank-one kernel optimization

The kernel optimization problem in the form of (148)-(149)-(150) is not easily tractable and cannot be efficiently solved by standard optimization softwares. In this section we use the rank-one decomposition of kernels to find an equivalent formulation in a linear program form. This is done through the introduction of several intermediate problems. We assume that we can write the regularized kernel as a convex combination of dyads: $K_\rho = \sum_{i=1}^{p} \nu_i \; l_i \, l_i^T$ where l_i are row vectors and ν_i are positive scalars such that $\sum_{i=1}^{p} \nu_i = 1$. Since by definition $K_\rho = I + \frac{K}{\rho^2}$, the decomposition of K_ρ into a sum of dyads is possible whenever the kernel K itself can be written as a sum of dyads. In practice the kernel is a positive semi-definite matrix so it can be diagonalized in an orthonormal basis and this property is satisfied. We can write an equivalent formulation of problem (142)-(143)-(144) as:

$$\Psi = \min_\nu \; y^T K_\rho^{-1} \, y \tag{152}$$

$$\text{subject to} \quad \nu_i \geq 0 \qquad \sum_{i=1}^{p} \nu_i = 1 \tag{153}$$

$$K_\rho = \sum_{i=1}^{p} \nu_i \; l_i \, l_i^T \tag{154}$$

where the vectors l_i are the eigenvectors of the matrices K_j from equation (144). Introducing the change of variable $\kappa = K_\rho^{-1}(\nu)$ and doing some computations enables one to rewrite problem (152)-(153)-(154) as:

$$\Psi = \max_\kappa \left(2 \; y^T \kappa - \max_{1 \leq i \leq p} (l_i^T \, \kappa)^2 \right) \tag{155}$$

and the optimal κ is related to the optimal ν by the relation

$$\kappa^* = K_\rho^{-1}(\nu^*) \tag{156}$$

One can note that solving problem (155) with the vector variable κ is the same as solving the problem:

$$\Psi = \min_{\gamma,\beta} \quad \left(2\ y^T \gamma\, \beta - \max_{1\le i\le p} (\, l_i^T \gamma\, \beta)^2\right) \tag{157}$$

for the variables γ and β. This is simply obtained by writing $\kappa = \gamma\, \beta$ in problem (155), with γ scalar and β vector. The optimal point (γ^*, β^*) of problem (157) satisfies:

$$\Psi^{1/2}\, \beta^* = \gamma^*\, \beta^* = \kappa^*. \tag{158}$$

If we minimize over γ in (157) we obtain the following optimization problem:

$$\Psi^{1/2} = \max_{\beta}\ \ y^T\, \beta \tag{159}$$

$$\text{subject to} \quad |l_i^T\, \beta| \le 1 \quad i = 1 \ldots p \tag{160}$$

The lagrangian of this problem can be written as:

$$L(\beta, u) = y^T\, \beta + \sum_{i=1}^{p} \left(|u_i| - u_i\, (l_i^T\, \beta)\right) \tag{161}$$

and taking the lagrangian dual of problem (159)-(160) yields:

$$\Psi^{1/2} = \min_{u}\ \ \| \, u \, \|_1 \tag{162}$$

$$\text{subject to} \quad y = \textstyle\sum_{i=1}^{p} u_i\ l_i \tag{163}$$

using the strict feasibility of the primal and convexity of the primal and the dual. Problem (163) is a linear program. The optimal ν^* can be retrieved from the optimal u^* from the relation:

$$\nu_i^* = \frac{|u_i|^*}{\Psi^{1/2}}. \tag{164}$$

Indeed one can check that with this value of the vector ν equations (158)-(156) yields $\Psi^{1/2}\, K_\rho(\nu^*)\, \beta^* = y$ and on the other hand we can write $\Psi^{1/2}\, K_\rho(\nu^*)\, \beta^* = \sum_{i=1}^{p} |\, u_i^*|\, (l_i^T\, \beta)\, l_i$ which is equal to $\sum_{i=1}^{p} u_i\ l_i$ using the optimality condition in the lagrangian (161). This proves that if u^* is optimal for (162)-(163) then ν^* given by (164) is optimal for (152)-(153)-(154) and vice-versa.

21.3.5. Choice of kernels

Several types of kernels are commonly used in machine learning [293]. Here, we combine several classical kernels with a kernel motivated by the known physical properties of the phenomenon we want to estimate.

Classical kernels

We consider a gaussian kernel K_σ defined by $K_\sigma(i,j) = \exp\left(-\frac{|x_i - x_j|^2}{\sigma^2}\right)$. We also use a linear kernel $K_{\text{lin}}(i,j) = x_i\, x_j$. Since we use a rank-one decomposition of each kernel, the regression problem with the linear kernel K_{lin} is expected to produce a slightly better estimate than the regular linear least-squares, because in the kernel method the weight of the eigenvectors can be different.

Physics of the phenomenon

Since we are interested in estimating the travel-time across a traffic light intersection, we consider the physical properties of this phenomenon as described in [64]. According to the authors, a reasonable model is the following: the travel-time across a traffic light intersection increases suddenly at the beginning of the red light and decreases linearly from there until the next beginning of the red light. This motivates us to use a piecewise linear function $\phi(\cdot)$ as a mapping.In order to do so we assume the traffic cycle length to be constant of value c. The slope of the linear function is left free and will be an optimization parameter. These considerations lead us to define the mapping function for the third kernel as $hi_{\text{phy}}(x) = x \bmod c$ where c is the traffic cycle length. It makes sense since according to the model the phenomenon is periodic of period c, and on a period the relation between the entry time and the travel-time is linear. We assume that $c = 60$ seconds.

21.4. Simulation results

21.4.1. Dataset description

We use a dataset generated by a traffic micro simulator, Paramics [102]. It is based on the car-following theory and driving behavior modeling and has been the subject of extensive research funded by Caltrans. It is assumed to mimic very accurately the macroscopic properties of traffic as well as inconsistent driving patterns observed in real life. Thus it provides a very interesting and challenging dataset to estimate the performances of our algorithm. The dataset consists of 1055 couples (x_i, y_i) where x_i is an entry time and y_i is the travel-time of a vehicle entering the road section at time x_i. This dataset has been generated for a road segment in Berkeley, California. It consists of an arterial link of length 1207 feet and the simulation has been run for half an hour between $3{:}30$ PM and $4{:}00$ PM on a week day.

21.4.2. Analysis method

Given an entry time x_i, we would like to provide a travel-time estimate $\hat{y}_i$. We are interested in the quadratic error between this estimate $\hat{y}_i$ and the effective travel-time y_i. In order to evaluate the performance of the techniques described in section 21.3, we follow the method described below. The error metric used is a L_2 relative norm.

- We consider a training set whose size is one half of the size the whole dataset. This can be

considered to model the fact that we know one half of the travel-times of vehicles flowing on the road section, and we want to estimate the travel-time of the other vehicles.

- In order to define the optimal regularization parameter for the training set, we define a 5-fold on this set. We use cross-validation as defined in section 21.3.2 on this 5-fold. Namely we use one of the five subsets as a trainingset, and the remainder serves as the test-set. We solve the optimization problem described in section 21.3 on each of the five training sets, and for several values of ρ, and we pick the one which minimizes the error metric.
- Having defined a regularization parameter at the previous step, we compute the optimal weight vector from the training set and evaluate the error on the test-set.
- We iterate this method for different training sets being in size one-half of the whole dataset, and we average the errors obtained. The results are given in table 26 for different convex combinations of kernels.

These computations are executed with Matlab, and the optimization problems are solved by `CVX`, which is a disciplined convex programing tool, using the program `SDPT3`.

21.4.3. Results and discussion

Kernel	Error on training set	Error on test set
K_{lin}	1.09	1.10
K_{phy}	1.07	1.09
K_{σ}	0.79	0.76
$K_{\text{lin}} + K_{\text{phy}}$	1.07	1.10
$K_{\text{lin}} + K_{\sigma}$	0.79	0.76
$K_{\text{phy}} + K_{\sigma}$	0.79	0.75

Table 26. Values of the L_2 relative error on the training set and on the test set for different combinations of kernels, for a training set of size 50 % of the size of the whole dataset. We note K_{σ} the gaussian kernel and use $\sigma = 100$, K_{lin} the linear kernel, and K_{phy} the physical kernel.

The results shown in table 26 yield several observations. First the high value of the relative errors must be compared to usual techniques, such as the conventional linear least-squares. For this dataset, the linear least-squares optimal estimate gives a relative error of 1 on the training set, because it produces an estimate without bias which equal zero. The error on the test set is as close to 1 as the distribution of the training set is close to the distribution of the test set. Whenusing only one kernel, the estimate performs at least almost as well as the linear least-squares estimate. The quality of the estimate is not always improved by taking a convex combination of kernels. The performance of the combination of kernels is at least as good at the performance of the best kernels, and is better when the two kernels have different features. In the case of a gaussian kernel and a physical kernel, the result is improved because the physical

kernel uses a feature (the operator modulo) which does not exist in the gaussian kernel. In the case of the linear kernel and the gaussian kernel, there is no added value compared to the gaussian kernel alone. The benefit of a combination of kernels is that if one does not know a-priori which kernel performs better, the optimization algorithm will choose an optimal combination and yields a mixed non-linear estimate which, as illustrated in figure 4.1, is able to locally capture trends of the phenomenon. This property of the estimate does not appear in the comparison of table 26 but is really useful for traffic applications. Here the linear least-squares estimate has a constant value at the mean value of the dataset, whereas the estimate given by a combination of the kernels oscillates. This is the subject on ongoing research which uses the same formalism with other norms such as the H^1 norm instead of a L^2 norm in the objective function of (133).

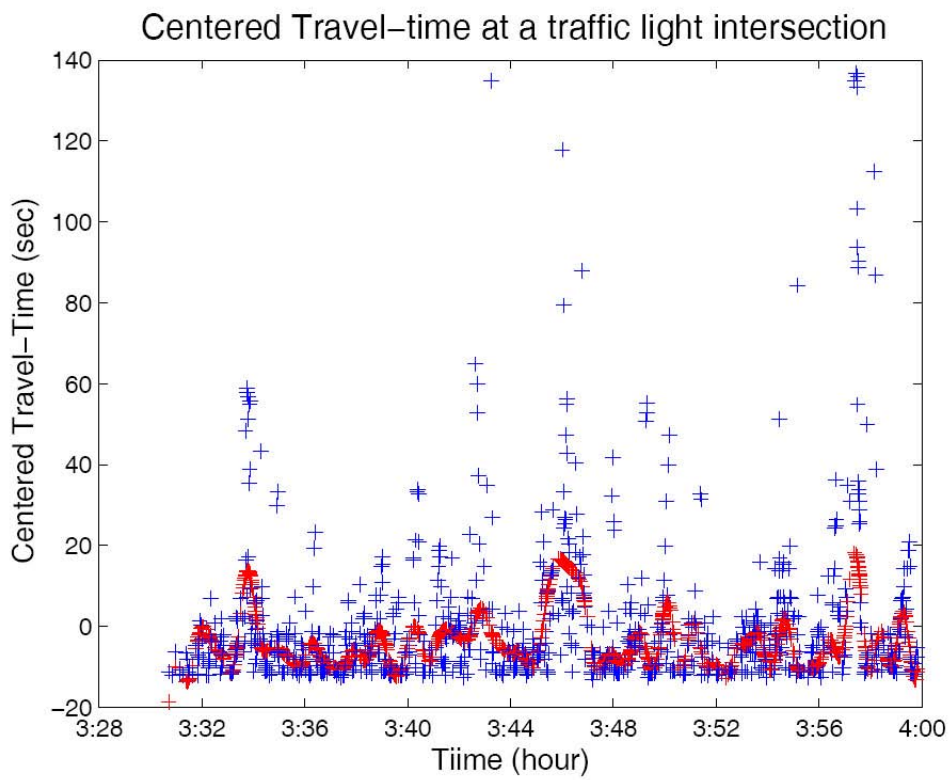


Figure 4.1. Observed travel-times (blue sparse points) and estimated travel-times (red smooth curve) for a linear combination of a gaussian kernel (with $\sigma = 100$) and a linear kernel.

21.5. Conclusions and future work

The results presented in this chapter show that even if the accuracy obtained by the kernel regression technique is not spectacular, an added value is the fact that the estimate is able to follow the trend of the travel-time. The kernel regression technique makes possible the addition of kernels to the set used in order to provide a richer signal providing better accuracy. Thus extensions to this work include the use of different kernels offering other features to improve the

results obtained. In particular, it would be satisfying to reach sufficiently good estimation accuracy with kernels only based on the physical properties of the road and some varying parameters (weather, time of day...). Applying results from the support vector machines theory allowing to bound the error in classifiers would significantly improve the quality of our travel-time estimate by adding robustness to it. The estimated travel-time on a road segment may be as important for practitioners as its range of variations. This is related to the discussion in section 4.1 on the ongoing research focusing on the use of other norms in the regression problem (133), and how to find a tractable and efficient way to solve the problem in these cases.

22. Reliable Routing in Stochastic Networks

The goal of this chapter is to provide the theoretical basis for enabling tractable solutions to the "arriving on time" problem and enabling its use in real-time mobile phone applications. Optimal routing in transportation networks with highly varying traffic conditions is a challenging problem due to the stochastic nature of travel-times on links of the network. The definition of optimality criteria and the design of solution methods must account for the random nature of the travel-time on each link. Most common routing algorithms consider the expected value of link travel-time as a sufficient statistic for the problem and produce least expected travel-time paths without consideration of travel-time variability. However, in numerous practical settings the reliability of the route is also an important decision factor. In this chapter, we consider the following optimality criterion: *maximizing the probability of arriving on time at a destination given a departure time and a time budget*. We present an efficient algorithm for finding an optimal routing policy with a well bounded computational complexity, improving on an existing solution that takes an unbounded number of iterations to converge to the optimal solution. A routing policy is an adaptive algorithm that determines the optimal solution based on en route travel-times and therefore provides better reliability guarantees than an a-priori solution. Novel speed-up techniques to efficiently compute the adaptive optimal strategy and methods to prune the search space of the problem are also investigated. Finally, an extension of this algorithm that allows for both time-varying traffic conditions and spatio-temporal correlations of link travel-time distributions is presented. The dramatic runtime improvements provided by the algorithm are demonstrated for practical scenarios in California.

22.1. Introduction

The design of optimal routing systems for operational scenarios in large scale transportation networks is a challenging problem due to the variability of realized link travel-times. As a result of this variability, the total travel-time on each link (and therefore the entire route) is a random variable with an associated probability distribution. Depending on the user preferences, an optimal route in this setting might need to consider notions of both the travel-time (expected value) and travel-time reliability (variance). This is a multi-criterion optimization problem that is in general hard to solve. The most commonly used formulation of optimality is to consider the route with the *least expected time* (LET) as defined by [233]. The LET problem has been well

researched and many efficient algorithms exist for different variants of the problem; for example [153,241,330]. When the link weights are independent and time-invariant distributions, the LET problem can be reduced to the standard deterministic shortest path problem by setting each link weight to its expected value. The solution to this problem can be computed efficiently using the [132] algorithm. [177] shows that Dijkstra's algorithm does not provide an optimal solution when the link weights are time-varying. If the network satisfies the first-in first-out (FIFO) condition defined by [56], the problem can be solved using dynamic programming using time-dependent weights with the original graph, see [131]. [330] consider the time-invariant LET problem with correlated link travel-times.

However, there are several settings in which the LET solution is not adequate, since it does not take into account the variance of travel-time distributions and gives no reliability guarantees. The optimal path defined by the LET solution can be unreliable and can result in highly variable realizations of travel-time. In numerous cases, travelers have hard deadlines or are willing to sacrifice travel-time to take a more reliable route. In commercial routing, there are delivery guarantees that need to be met and perishables that need to be delivered within a fixed amount of time. [150] presents a very natural definition of a reliable optimal path, as the path that maximizes the probability of realizing a travel-time that is less than a given constant. However, the formulation given by Frank requires enumerating all possible paths and therefore is not tractable for practical problems.

A formulation of the problem using stochastic optimal control (see [73]), which combines static information about the network structure with real-time information about actual travel-times, results in an adaptive solution that is an optimal policy as opposed to an optimal path. An optimal policy generates a node-based decision rule that defines the optimal path from a given node to the destination conditioned on the realized travel-time. It is clear that such an adaptive policy should do better than a static a-priori solution.

[145] consider Frank's formulation of maximizing the probability of realizing a given travel-time, also known as the *stochastic on time arrival* (SOTA) problem, and formulate it as a stochastic dynamic programming problem. The dynamic program is solved using a standard *successive approximation* (SA) algorithm. In an acyclic network, the SA algorithm converges in a number of steps no greater than the maximum number of links in the optimal path. However, in a realistic network an optimal adaptive strategy may contain loops. Moreover, there is no finite bound on the maximum number of links the optimal path can contain, as explained by [145]. Therefore, the number of steps required for the algorithm to converge is unbounded. As an alternative, [258] propose a discrete approximation algorithm for the SOTA problem that converges in a finite number of steps and runs in pseudo-polynomial time.

In this chapter, we present a number of theoretical and numerical results that improve the tractability of the SOTA problem over existing methods. We first solve the unbounded convergence problem, by developing a new algorithm that gives an exact solution to the SOTA problem and has a provable convergence bound. As with [145], this algorithm requires computing a continuous-time convolution product, which is one of the challenges of the method. In general, this convolution cannot be solved analytically when routing in arbitrary networks, and therefore a discrete approximation scheme is required. By exploiting the structure of our

algorithm, we are able to solve the convolution more efficiently than the standard (brute force) discrete time approximation algorithm used in [258] and obtain a faster computation time. We show that the order in which the nodes of the graph are considered greatly impacts the running time of our solution and present an optimal ordering algorithm that minimizes the computation time.

In addition, we present an analysis of the conditions under which our framework can be extended to handle time-varying travel-time distributions and show that these conditions are satisfied in the commonly used travel-time models for road networks. We also consider the problem of correlated travel-time distributions. Finally, we present a network pruning scheme that reduces the search space of the algorithm and thus also improves its computational efficiency. Our goal is to provide the theoretical basis for a tractable implementation of adaptive routing with reliability guarantees in an operational setting. One specific application of interest is to enable adaptive routing on mobile phones using real-time traffic data. Experimental results are provided for San Francisco Bay Area highway and arterial networks using the [14] traffic information system.

The rest of the chapter is organized as follows. In Section 22.2, we define the *stochastic on time arrival* (SOTA) problem and discuss its classical solution method. In Section 22.3, we present a new SOTA algorithm, we prove its convergence properties and discuss how the algorithm can be used with both time-varying and correlated travel-time distributions. In Section 22.4, we present an efficient numerical method to approximate convolution integrals using the Fast Fourier Transform (FFT) and an optimal update algorithm. Experimental results are given in Section 22.5. Finally, we present our conclusions in Section 22.6.

22.2. The Stochastic On-time Arrival (SOTA) Problem

We consider a directed network $G(N, A)$ with $|N| = n$ nodes and $|A| = m$ links. The weight of each link $(i, j) \in A$ is a random variable with probability density function $p_{ij}(\cdot)$ that represents the travel-time on link (i, j). Given a time budget T, an optimal routing strategy is defined to be a policy that maximizes the probability of arriving at a destination node s within time T. A routing policy is an adaptive solution that determines the optimal path at each node (intersection in the road network) based on the travel-time realized to that point. This is in contrast to a-priori solutions that determine the entire path prior to departure. Given a node $i \in N$ and a time budget t, $u_i(t)$ denotes the probability of reaching node s from node i in less than time t when following the optimal policy. At each node i, the traveler should pick the link (i, j) that maximizes the probability of arriving on time at the destination. If j is the next node being visited after node i and ω is the time spent on link (i, j), the traveler starting at node i with a time budget t has a time budget of $t - \omega$ to travel from j to the destination, as described in equation (165)[18].

[18]In this formulation of the problem, the traveler is not allowed to wait at any of the intermediate nodes. In Section 2 we state the conditions under which travel-time distributions from traffic information systems satisfy the first-in-first-out (FIFO) condition, and thus waiting at a node cannot improve the on time arrival probability of the modeled traveler.

Definition 22. The optimal routing policy for the SOTA problem can be formulated as follows:

$$u_i(t) = \max_j \int_0^t p_{ij}(\omega)\, u_j(t-\omega)\, d\omega \qquad \forall i \in N, i \neq s, (i,j) \in A, 0 \leq t \leq T \tag{1}$$

$$u_s(t) = 1 \quad 0 \leq t \leq T$$

where $p_{ij}(\cdot)$ is the travel-time distribution on link (i,j).

The functions $p_{ij}(\cdot)$ are assumed to be known and can for example be obtained using historical data or real-time traffic information.

[145] present the *successive approximations* (SA) algorithm described in Algorithm 27, which solves the system of equations (165) and gives an optimal routing policy.

Algorithm 27. Successive approximations algorithm ([145])

Step 0. Initialization

$k = 0$

$u_i^k(t) = 0, \forall i \in N, i \neq s, 0 \leq t \leq T$

$u_s^k(t) = 1, 0 \leq t \leq T$

{ $u_i^k(t)$ is the approximation of $u_i(t)$ in the kth iteration of the algorithm}

Step 1. Update

$k = k + 1$

$u_s^k(t) = 1, 0 \leq t \leq T$

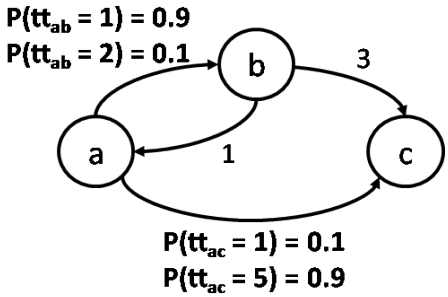
$u_i^k(t) = \max_j \int_0^t p_{ij}(\omega)\, u_j^{k-1}(t-\omega)\, d\omega,\ \forall i \in N, i \neq s, (i,j) \in A, 0 \leq t \leq T$

Step 2. Convergence test

If $\forall (i,t) \in N \times [0,T], \max_{i,t} |u_i^k(t) - u_i^{k-1}(t)| = 0$ stop;

Otherwise go to Step 1.

At each iteration k, $u_i^k(t)$ gives the probability of reaching the destination from node i within a time budget t, using a path with no more than k links, under the optimal policy. The approximation error monotonically decreases with k and the solution eventually reaches an optimal value when k is equal to the number of links in the optimal path. A formal proof of the convergence is given in Section 3 of [145] using the bounded monotone convergence theorem. However, since an optimal routing policy in a stochastic network can have loops (see Example 23), the number of iterations required to attain convergence is not known a-priori.



Path	Travel-time	Probability
$\{(a,b),(b,c)\}$	4	0.9
$\{(a,c)\}$	1	0.1
$\{(a,b),(b,a),(a,c)\}$	4	0.01

Figure 2.1. A simple network with an optimal routing policy that may contain a loop. Links (b,c) and (b,a) have deterministic travel-times of respectively 3 and 1 time units. Link (a,b) has a travel-time of 1 with probability 0.9 and a travel-time of 2 with probability 0.1. Link (a,c) has a travel-time of 5 with probability 0.9 and a travel-time of 1 with probability 0.1. The table presents on time arrival probabilities for all possible realizations of optimal paths.

Example 23. Figure 23 shows a simple network in which an optimal path can contain a loop. Consider finding the optimal route from node a to node c with a budget of 4 time units. There are two choices at the origin, of which it is clear that link (a,b) gives the highest probability of reaching the destination on time, since there is a 0.9 probability of the travel-time on (a,b) being 1 time unit, which leads to a total travel-time of 4. However, assume that the realized travel-time on link (a,b) is actually 2 time units, which can happen with probability 0.1. In this case, taking the path $\{(a,b),(b,c)\}$ results in zero probability of reaching the destination on time. The optimal path is therefore $\{(a,b),(b,a),(a,c)\}$, which is the only path that has a non zero probability of reaching the destination on time. This optimal path contains a loop. An a-priori algorithm such as the least expected time (LET) algorithm will always route on path $\{(a,b),(b,c)\}$ regardless of the realized travel-times and thus have a lower probability of reaching the destination on time.

As stated in [145], as the network gets arbitrarily complex, it is possible to have an infinite-horizon routing process. [74] showed that this is unlikely to happen in realistic networks, but whether a successive approximations solution to this problem converges in a finite number of steps is an open problem to the best of our knowledge.

To solve the problem raised by the unbounded convergence of Algorithm 27, [258] present a discrete time approximation of the SOTA problem. This algorithm has a computational complexity of $O(m(Td)^2)$, where m is the number of links in the graph, d is the number of discretization intervals per unit time and T is the time budget. The drawback of this method is the numerical discretization error in the representation of the probability density function. A smaller discretization interval leads to a more accurate approximation, but increases the computation time quadratically. In this chapter, we present both a continuous time exact solution that does not require successive approximations and a discretization scheme with a computation time to approximation error trade off that is lower than the standard discretization scheme in most practical cases.

22.3. Continuous time exact formulation of the SOTA problem with single iteration convergence algorithm

In this section, we present an algorithm that finds the optimal solution to the continuous time SOTA problem in a single iteration through time space domain of the problem. The complexity of the algorithm does not depend on the number of links in the optimal path.

22.3.1. Solution algorithm for single iteration convergence

The key observation used in this algorithm is that there exists a minimum physically realizable travel-time on each link of the network. Let β be the minimum realizable link travel-time across the entire network. β is strictly positive since speeds of vehicles have a finite uniform bound, and the network contains a finite number of links with strictly positive length. Therefore, given $\epsilon \in (0, \beta)$, $\delta = \beta - \epsilon$ is a strictly positive travel-time such that $p_{ij}(t) = 0 \; \forall \, t \leq \delta, (i,j) \in A$. Given a time budget T discretized in intervals of size δ, let $L = \lceil T/\delta \rceil$. We propose the solution Algorithm 28.

Algorithm 28. Single iteration SOTA algorithm

Step 0. Initialization.
$k = 0$
$u_i^k(t) = 0, \forall i \in N, i \neq s, t \in [0, T)$
$u_s^k(t) = 1, \forall \, t \in [0, T)$
Step 1. Update
For $k = 1,2, \dots, L$
$\tau^k = k\delta$
$u_s^k(t) = 1, \forall \, t \in [0, T)$
$u_i^k(t) = u_i^{k-1}(t)$
$\forall i \in N, i \neq s, t \in [0, \tau^k - \delta]$
$u_i^k(t) = \max_j \int_0^t p_{ij}(\omega) \, u_j^{k-1}(t - \omega) \, d\omega$ {computation of the convolution product}
$\forall i \in N, i \neq s, (i,j) \in A, t \in (\tau^k - \delta, \tau^k]$

In this formulation of the SOTA problem, the functions $u_i^k(\cdot)$ are computed on $[0, T]$ by increments of size δ. This algorithm relies on the fact that for $t \in (\tau^k - \delta, \tau^k]$, $u_i^k(t)$ can be computed exactly using only $u_j^{k-1}(\cdot)$, $(i,j) \in A$, on $(\tau^k - 2\,\delta, \tau^k - \delta]$, where τ^k is the budget up to which $u_i^k(\cdot)$ is computed at the kth iteration of Step 1.

Proposition 24. Algorithm 28 finds the optimal policy for the SOTA problem in a single iteration.

Proof. Proof by induction over the sub-steps $\tau = \delta$ to $L\delta$.

Base case: When $k = 1$ the convolution product is computed on the interval $(0, \delta]$. From the definition of δ, we know that there does not exist a realizable travel-time that is less than or equal to δ. Therefore, $p_{ij}(\omega) = 0$ on the interval $(0, \delta]$ of the convolution product, and $u_i^1(t) = u_i^0(t)$ $\forall i \in N$. This is indeed the correct solution $\forall\, t$ such that $0 \leq t \leq \delta$, since no feasible path to the destination exists in this time interval.

Induction step: Assume that the algorithm generates an optimal policy for $k < L$. We show that the algorithm also provides an optimal policy at step $k + 1$. When $t = (k + 1)\delta$, the convolution product is computed on the interval $(k\delta, (k + 1)\delta]$. Since $p_{ij}(\omega) = 0$, $\forall\, \omega$ such that $\omega \leq \delta$, to find the optimal policy for $u_i^{k+1}(t)$, $\forall\, t$ such that $k\delta < t \leq (k + 1)\delta$, we only need to know the optimal policy for $u_j^k(t)$, $\forall, t$ such that $0 \leq t \leq k\delta$, $(i, j) \in A$. By the induction hypothesis we know that $u_i^k(t)$ gives the optimal policy $\forall\, t$ such that $0 \leq t \leq k\delta$ for all nodes and it is known. This implies that the optimal policy iscomputed for the range $0 \leq t \leq (k + 1)\delta$ at the end of the $(k + 1)^{th}$ step. □

The most computationally intensive step of this algorithm is the computation of the convolution product, which is represented algebraically in the above algorithm. If the link travel-time distributions $p_{ij}(\cdot)$, $(i, j) \in A$, and the cumulative optimal travel-time distributions $u_i(\cdot)$, $i \in N$, belong to a parametric distribution family which is closed under convolution (e.g. Gaussian, Erlang, see [77,55]), the convolution product can be computed analytically. However, since $u_i(\cdot)$ is the point wise maximum of the convolution products of the link travel-time distributions $p_{ij}(\cdot)$ and the cumulative distributions $u_j(\cdot)$, $(i, j) \in A$, it does not have an analytical expression in general.

Numerical approximations of the distributions involved in the convolution product have been proposed in the literature. [146] argue that since $u_i(\cdot)$ is a continuous monotone increasing function, it can be approximated by a low degree polynomial. When the approximating polynomial is of degree $2n$, the convolution integral can be solved exactly with n evaluation points using the Gaussian quadrature method (see [72]).

The applicability of these methods is highly dependent on the shape of the travel-time distributions, which can be very complex, depending on the traffic conditions and the topology of the network. Therefore, we solve the convolution product via a time discretization of the distributions involved, which results in a computational complexity that is independent of the shape of the optimal cumulative travel-time distributions $u_i(\cdot)$. An incremental update scheme that exploits the structure of the SOTA algorithm is used to efficiently compute the discrete convolution product. This solution is shown to be computationally less expensive than the existing convolution methods for the SOTA problem (e.g [258]). A detailed explanation is given in Section 4.

22.3.2. Extended algorithm for time-varying link travel-times

The solution algorithm in the previous section makes the assumption that link travel-time distributions are static. However, in real transportation networks, it is clear that link travel-time distributions are time-varying. In this section, we present an extension to Algorithm 28 that

accounts for time-varying distributions. A common approach when solving shortest path problems on graphs with time-dependent distributions is to consider the corresponding time expanded graph with static weights. See [131] for a discussion on the various flavors of this problem. If the *first-in-first-out* (FIFO) condition holds, the problem can be solved without time-expanding the graph using a trivially modified version of Dijkstra's algorithm and indexing the link weights by time (see [133]). We describe a similar algorithm based on the fact that waiting at a node is never optimal when the FIFO conditions holds. First we show that most commonly used travel-time estimates satisfy the the FIFO condition and then show that waiting at a node is never optimal in such a model.

Definition 25. In a deterministic setting, let α_P^t denote the travel-time on path[19] P when departing at time t. The graph satisfies the FIFO condition if and only if:

$$\alpha_P^{t_1} \leq \alpha_P^{t_2} + (t_2 - t_1) \; \forall \text{ path } P \text{ and } \forall \, t_1, t_2 \text{ such that } \; 0 \leq t_1 \leq t_2 \tag{166}$$

This definition states that on a given path P, the travel-time $\alpha_P^{t_1}$ when leaving at t_1 is lower than the travel-time $\alpha_P^{t_2} + (t_2 - t_1)$ obtained by waiting at the departure node for $t_2 - t_1$ and departing at t_2.

In the time-varying setting, the link travel-time estimates given by a travel-time model can change as a vehicle moves through a link. We assume an elastic vehicle travel-time model, where the vehicle link travel-time is calculated based on all the link travel-time estimates the vehicle might encounter as it moves through a link. This is in contrast to a frozen vehicle travel-time model, where the travel-time is calculated simply based on the link travel-time estimate when the vehicle enters the link[20].

Proposition 26. Under an elastic vehicle travel-time model, a deterministic discrete-time traffic estimate such that link travel-time is single-valued on each time discretization yields a deterministic FIFO path.

Proof. Consider two vehicles traveling along the same path P from node i to node k departing at times t_1 and t_2 respectively for $0 \leq t_1 \leq t_2$. Their respective travel-times are denoted $\alpha_P^{t_1}$ and $\alpha_P^{t_2}$. For the vehicle departing node i at time t_2 to arrive at node k before the vehicle departing at time t_1, it must overtake the vehicle that departed first. Overtaking can only occur when both vehicles are in the same space-time cell. However, since we assume that the model gives a single-valued speed in each space-time cell, both vehicles will travel at the same speed when in the same cell and no overtaking can occur. Therefore, a single-valued speed

[19]The FIFO condition is typically defined on a link. Here, we use a path-based definition to make the subsequent explanations and proofs more intuitive. This leads to equivalent results under the assumption that the network topology is static.

[20] Please see [264] for further discussion on the elastic and frozen travel-time models.

in each space-time cell implies that the vehicle that departed first will always arrive first. □

This guarantees that the shortest path problem on transportation networks, with time-varying link travel-times generated by a traffic information system, can be solved with the same complexity as in the static case by time-indexing the link weights ([131]). In the case of transportation networks with stochastic link travel-times, for the SOTA problem (equation (165)), a similar stochastic FIFO condition is needed to guarantee correctness of the algorithm with time-indexed link travel-times.

Definition 27. Let $u_P^t(\cdot)$ denote the cumulative travel-time distribution on path P when departing at time t. The graph satisfies the stochastic FIFO condition if and only if:

$$u_P^{t_1}(T) \geq u_P^{t_2}(T-(t_2-t_1))\ \forall \text{ path } P \text{ and } \forall\, T, t_1, t_2 \text{ such that } 0 \leq t_1 \leq t_2, t_2 - t_1 \leq T \qquad (167)$$

This definition states that at any given time and on any given path on the network, departing as soon as possible yields a greater probability of arriving on time than delaying the departure. The stochastic FIFO property is obtained if the conditions defined by Proposition 28 are satisfied.

Proposition 28. A stochastic discrete-time traffic estimate such that link travel-time distributions are fixed for each time discretization yields a stochastic FIFO path.

Proof. Proof by induction over the length of the path $(v_n, \dots, v_1)$.
Base case $(n = 2)$: From definition 168, a stochastic FIFO network satisfies the following condition:

$$u_{v_2 v_1}^{t_1}(T) \geq u_{v_2 v_1}^{t_2}(T-(t_2-t_1))\ \forall\ 0 \leq t_1 \leq t_2 \qquad (168)$$

where $u_{ij}^t(T)$ is the probability of arriving at node j in time T when departing from node i at time t. We want to show that a delayed departure cannot improve the probability of on time arrival when traveling from node v_2 to node v_1 on link (v_2, v_1). Without loss of generality, let vehicle w_1 depart from node v_2 at time t_1 and vehicle w_2 depart from node v_2 at time t_2, where w_1 is in cell $c_1 = (v_2, v_1) \times [t_{c_1}, t_{c_2}]$ and w_2 is in cell $c_2 = (v_2, v_1) \times [t_{c_2}, t_{c_3}]$. Also, let $X_w(t_a, t_b)$ be the distribution of the distance traveled by vehicle w in the interval (t_a, t_b). We have the following:

$$X_{w_2}(t_1, t_2) = 0$$

$$X_{w_1}(t_1, t_2) \geq 0$$

Let x be the length of link (v_2, v_1). We want to show that:

$$\begin{aligned} & u_{v_2 v_1}^{t_1}(T) \\ \Leftrightarrow\quad & P(X_{w_1}(t_1, t_1+T) \geq x) \\ \Leftrightarrow\quad & P(X_{w_1}(t_1, t_{c_2}) + X_{w_1}(t_{c_2}, t_2) + X_{w_1}(t_2, t_1+T) \geq x) \end{aligned}$$

This clearly holds because:

$$X_{w_1}(t_2, t_1 + T) = X_{w_2}(t_2, t_1 + T)$$

since both vehicles are in the same space-time cells from time t_2.

Induction step $(n = k)$: We assume that a single-valued travel-time distribution for each space-time cell implies a stochastic FIFO path $(v_k, \dots, v_1)$ of k nodes and show that it holds for $k + 1$ nodes. The explanation in the base case shows that departing from node v_{k+1} earlier gives a higher probability of reaching node v_k within a given time budget. The induction hypothesis implies that arriving at node v_k earlier increases the probability of reaching the destination v_1 within a given time budget. Therefore, leaving the node v_{k+1} earlier increases the probability of reaching the destination within a given time budget. □

Under the stochastic FIFO condition, we now show that an optimal policy for the SOTA problem does not prescribe waiting at a node. Therefore, the SOTA problem on transportation networks with time-varying link travel-time distributions can be solved by time-indexing the link travel-time distributions.

Proposition 29. In a stochastic FIFO network, according to the optimal policy for the SOTA problem, waiting at a non-terminal node is not optimal.

Proof. Proposition 28 shows that waiting at a node cannot improve the probability of arriving within a certain budget using the same path. We now show that waiting at a node cannot improve the probability of arriving within a given budget T when using the optimal path for each departure time. Assume that path P_1 is the optimal path when departing at time t_1 and that path P_2 is the optimal path when departing at time t_2. From proposition 28 we know that $u_{P_2}^{t_1}(T) \geq u_{P_2}^{t_2}(T - (t_2 - t_1))$. Furthermore, since P_1 is the optimal path when leaving at time t_1 with a budget of T, we have $u_{P_1}^{t_1}(T) \geq u_{P_2}^{t_1}(T)$. Therefore, $u_{P_1}^{t_1}(T) \geq u_{P_2}^{t_2}(T - (t_2 - t_1))$ and waiting cannot improve the probability of on time arrival. □

When the assumptions from Proposition 28 are satisfied, according to Proposition 29, waiting at a node is not part of the optimal policy. Therefore, the optimal policy in the time-varying case can be defined as follows:

$$u_i^\tau(t) = \max_j \int_0^t p_{ij}^\tau(\omega)\, u_j^{\tau+\omega}(t - \omega)\, d\omega \qquad (169)$$
$$\forall i \in N, i \neq s, (i,j) \in A, 0 \leq t \leq T, 0 \leq \tau$$

$$u_s^\tau(t) = 1 \quad \forall\, 0 \leq t \leq T, 0 \leq \tau.$$

where $u_i^\tau(t)$ is the maximum probability of arriving at destination s within time budget t when leaving node i at time τ. Waiting is not allowed in the optimal policy as enforced by the fact that the departure time from node$i$ (superscript of $u_i^\tau(\cdot)$) is the same as the time at which

the link (i,j) is traversed (superscript of $p_{ij}^{\tau}(\cdot)$). This policy is optimal according to Proposition 29.

This algorithm uses the same network as the static SOTA problem and simply replaces the travel-time distribution query with a time-indexed version, as defined in Equation 169. i.e. it modifies the convolution step to query the appropriate link travel-time distribution based on the current time offset τ. This algorithm has the same computational complexity as the static algorithm because the structure of the graph remains the same and no additional queries are performed. The required memory is larger in the time-varying case than in the static case because there are multiple travel-time distributions for each link.

22.3.3. Generalized algorithm for correlated link travel-times

In this section, we extend the capabilities of the previous algorithm, by relaxing another assumption that was made when presenting the algorithm in Section 22.2. The formulation presented so far assumes that the link travel-times on the network are uncorrelated. However, in reality the travel-times of neighboring links are correlated. Assuming that link travel-times satisfy the Markov condition, they only depend on their upstream and downstream neighbors. Therefore, the travel-time on each link is a joint distribution over the link and its neighbors. If we do not have any information regarding the travel-times on these other links, the correct approach is to marginalize them out and use the marginal distribution of the link we are considering. However, in the SOTA formulation each decision could be preceded by conditioning on the travel-time of an upstream link. Assuming independence in this case results in an inaccurate expected travel-time and an overestimation of the variance. Therefore, to minimize such errors one must incorporate this observation and use the conditional probability distribution of the link travel-time. We present a simple extension to our formulation that considers the correlation between a link and the upstream neighbors via which the link is reached. The problem formulation is as follows:

$$u_i(t,k,y) = \max_j \int_0^t p_{ij}\,(tt_{ij} = \omega | tt_{ki} = y)\, u_j(t-\omega, i, \omega)\, d\omega \tag{170}$$

$$\forall i \in N, i \neq s, (i,j) \in A, (k,i) \in A,$$

$$0 \leq y \leq (T-t), 0 \leq t \leq T$$

$$u_s(t,k,y) = 1 \quad \forall\, 0 \leq t \leq T, (k,s) \in A, 0 \leq y \leq T-t$$

where $u_i(t,k,y)$ is the cumulative distribution function (CDF) when t is the remaining time budget, k is the node from which the vehicle is arriving and y is the realized travel-time on this upstream link, and $p_{ij}(tt_{ij} = \omega | tt_{ki} = y)$ is the probability that the travel-time on link (i,j) is ω conditioned on the travel-time on link (k,i) being y. The joint probability density function of a link and its upstream neighbors is assumed to be known. In this case, each CDF $u_i(\cdot,\cdot,\cdot)$ and travel-time distribution $p_{ij}(\cdot)$ is now conditioned on the upstream travel-time, but the structure of the problem remains unchanged. We are simply propagating more information at

each step. Therefore, the correctness analysis of our algorithm remains unchanged and the same proof holds. However, the time complexity of the algorithm increases since a new dimension is being added to the problem. Equation (170) needs to be solved for each incoming node k and the travel-time on link (k, i), which is in the range $0 \leq y \leq (T - t)$. This increases the runtime of the original formulation (Equation (165)) by a factor of ΦT, where Φ is the maximum in-degree of the network.

In most practical cases this is likely to make the algorithm intractable for real-time applications, as the complexity is now cubic in T. Therefore, we use a discrete approximation of the conditional distribution function. In the simplest case, the conditioning can be done based on whether the upstream link was in congestion or free flow, which will only increase the complexity by a factor of 2Φ. If upstream travel-time is discretized in to d ranges, the increase in complexity will be a factor of $d\Phi$. The most appropriate value to use for d depends on the quality of the conditional travel-time distributions available and the computing resources that can be utilized.

22.4. Discrete formulation of the SOTA algorithm with a Fast Fourier Transform solution

For practical networks, a numerical approximation of the convolution integral is necessary, with a proper discretization. In the discrete setting, the algorithm can be formulated as shown in Algorithm 29 below.

Table 29. Discrete SOTA algorithm

Step 0. Initialization.

$k = 0$

$u_i^k(x) = 0, \forall i \in N, i \neq s, x \in \mathbb{N}, 0 \leq x \leq \frac{T}{\Delta t}$

$u_s^k(x) = 1, x \in \mathbb{N}, 0 \leq x \leq \frac{T}{\Delta t}$

Step 1. Update

For $k = 1,2, \dots, L$

$\tau^k = k\delta$

$u_s^k(x) = 1, x \in \mathbb{N}, 0 \leq x \leq \frac{T}{\Delta t}$

$u_i^k(x) = u_i^{k-1}(x)$

$\forall i \in N, i \neq s, (i,j) \in A, x \in \mathbb{N}, 0 \leq x \leq \frac{(\tau^k - \delta)}{\Delta t}$

$u_i^k(x) = \max_j \sum_{h=0}^{x} p_{ij}(h)\, u_j^{k-1}(x-h)$

$\forall i \in N, i \neq s, (i,j) \in A, x \in \mathbb{N}, \frac{(\tau^k - \delta)}{\Delta t} + 1 < x \leq \frac{\tau^k}{\Delta t}$ $\{\Delta t$ is selected such that $\delta > \Delta t\}$

where Δt is the length of a discretization interval and T is the time budget. The functions $u_i(\cdot)$ and $p_{ij}(\cdot)$ are vectors of length $L = \lceil \frac{T}{\Delta t} \rceil$. For notational simplicity, we assume that T is a multiple of Δt. In general, the link travel-time distributions are available as either discrete or continuous time distributions. If the link travel-time distribution is discrete and the length of the discretization interval d is not equal to Δt, the probability mass needs to be redistributed to intervals of Δt. If the distribution is continuous, the probability mass function $p_{ij}(\cdot)$ is computed as follows:

$$p_{ij}(h + \Delta t) = \int_{h}^{h+\Delta t} p_{ij}(\omega)\, d\omega, \forall h = 0, \Delta t, \dots, (L-1)\Delta t \tag{171}$$

22.4.1. Complexity analysis

Obtaining the appropriately discretized probability mass functions can be done in time $O(\frac{mT}{\Delta t})$, since there are m links and each link travel-time distribution function is of length $\frac{T}{\Delta t}$. This can also be computed in advance and reused during each call to the algorithm[21]. In step 0, initializing k vectors (one for each node i) of length $\frac{T}{\Delta t}$ takes $O(\frac{kT}{\Delta t})$ time. In step 1, notice that for each link (i, j) the algorithm progressively computes a sum of increasing length from $x = 1$ to $x = \frac{L\delta}{d} = \frac{T}{\Delta t}$. Therefore, the time complexity of the summation for each link is $O((\frac{T}{\Delta t})^2)$. The assignment $u_i^k(x) = u_i^{k-1}(x)$ can be done in constant time by manipulating pointers instead of a memory copy or by simply having one array for all $u_i(\cdot)$ that keeps getting updated at each iteration of the loop. Since there are m links, the total time complexity of step 1 is $O(m(\frac{T}{\Delta t})^2)$. This dominates the complexity of step 0 and therefore is the total time complexity of the entire algorithm. Recall that this is identical to the time complexity of the discrete approximation algorithm proposed by [258].

Algorithm 29 can perform more efficiently by computing the convolution products via the *Fast Fourier Transform* (FFT). The FFT computes the convolution of two vectors of length n in $O(n \log(n))$ time, see [114]. Notice however that the algorithm does not compute the entire convolution at once. The computation is required to be done in blocks of length δ to preserve optimality. Therefore, L convolution products of increasing length $\delta, 2\delta, \dots, L\delta$ have to be computed. One inefficiency of this approach is that successive convolutions recompute the results that have already been obtained. However, using the FFT can still be shown to perform better than a brute force convolution product with quadratic complexity. For each link, the time complexity of the sequence of FFTs is $O(\sum_{k=1}^{L} \frac{\delta k}{\Delta t} \log(\frac{\delta k}{\Delta t}))$, where $L = \lceil \frac{T}{\delta} \rceil$. Since there are m links, the total time complexity is:

[21]In the case of time-varying link travel-times, this needs to be recomputed for each time step at which the travel-times differ.

$$O\left(m\sum_{k=1}^{\lceil\frac{T}{\delta}\rceil}\frac{\delta k}{\Delta t}\ \log\left(\frac{\delta k}{\Delta t}\right)\right) \tag{172}$$

As $T \to \infty$, the complexity of the FFT based approach $O((\frac{T}{\Delta t}^2 \log(\frac{T}{\Delta t}))$ is asymptotically larger than the run-time of the brute force approach $\sum_{k=1}^{\frac{T}{\Delta t}} k = O((\frac{T}{\Delta t})^2)$.

However, the running time of the FFT approach is significantly smaller than the brute force approach in the time range of interest for most practical applications, as shown in proposition 5.

Proposition 30. The travel budget $t = \Delta t \cdot 2^{\frac{1}{4}\left(3+\frac{\delta}{\Delta t}\right)} - \delta$ is a lower bound for the largest budget at which the FFT based approach has a faster run-time than the brute force approach. See Appendix A for proof.

Proof. We compare the runtime of the FFT approach and the brute force approach. In order to compute the optimal solution up to a given time τ, the brute force method needs to be executed for $\frac{\tau}{\Delta t}$ steps and the FFT method needs to be executed for $\frac{\tau}{\delta}$ steps. The runtime of the brute force method is $\sum_{k=1}^{\frac{\tau}{\Delta t}} k$ and the running time of the FFT approach is $\sum_{k=1}^{\frac{\tau}{\delta}} \frac{\delta k}{\Delta t}\ \log(\frac{\delta k}{\Delta t})$. The FFT approach is faster than the brute force convolution for $t = K\,\delta$ such that:

$$\sum_{k=1}^{K}\frac{\delta k}{\Delta t}\ \log\frac{\delta k}{\Delta t} \le \sum_{k=1}^{\frac{K\delta}{\Delta t}} k = \frac{1}{2}\frac{K\delta}{\Delta t}\left(\frac{K\delta}{\Delta t}+1\right).$$

If we use the fact that the function $x \mapsto x \log x$ is increasing on $[1;+\infty)$, we can bound the left hand side of the above inequality as follows:

$$\begin{aligned}\sum_{k=1}^{K}\frac{\delta k}{\Delta t}\ \log\frac{\delta k}{\Delta t} \quad &< \int_{z=0}^{K}\frac{\delta(z+1)}{\Delta t}\ \log\frac{\delta(z+1)}{\Delta t}\ dz \\ &= \frac{1}{2}\frac{\delta}{\Delta t}\,(K+1)^2\left(\log\frac{\delta\,(K+1)}{\Delta t}-\frac{1}{2}\right)-\frac{1}{2}\frac{\delta}{\Delta t}\left(\log\frac{\delta}{\Delta t}-\frac{1}{2}\right)\end{aligned}$$

where the inequality is a right Riemann integral bound. A lower bound t on the time up to which the FFT approach is faster than the brute force convolution thus satisfies:

$$\frac{1}{2}\frac{\delta}{\Delta t}\,(K+1)^2\left(\log\frac{\delta\,(K+1)}{\Delta t}-\frac{1}{2}\right)-\frac{1}{2}\frac{\delta}{\Delta t}\left(\log\frac{\delta}{\Delta t}-\frac{1}{2}\right) \le \frac{1}{2}\frac{K\delta}{\Delta t}\left(\frac{K\delta}{\Delta t}+1\right).$$

If we assume $\delta \geq \Delta t \exp \frac{1}{2}$, we have $-\frac{1}{2}\frac{\delta}{\Delta t}\left(\log \frac{\delta}{\Delta t} - \frac{1}{2}\right) \leq 0$ and thus a sufficient condition to have the inequality above satisfied is to have:

$$\frac{1}{2}\frac{\delta}{\Delta t}(K+1)^2\left(\log \frac{\delta\,(K+1)}{\Delta t} - \frac{1}{2}\right) \leq \frac{1}{2}\frac{K\delta}{\Delta t}\left(\frac{K\delta}{\Delta t} + 1\right)$$

We can equivalently rewrite the above inequality as:

$$(K+1)^2 \log \frac{\delta\,(K+1)}{\Delta t} \leq \frac{1}{2}(K+1)^2 + \frac{\delta}{\Delta t}K^2 + K.$$

We wish to find $\alpha \in \mathbb{R}$ such that $\alpha(K+1)^2 \leq \frac{1}{2}(K+1)^2 + \frac{\delta}{\Delta t}K^2 + K$ for $K \geq 1$. This is satisfied for $\alpha \leq \frac{1}{4}\left(3 + \frac{\delta}{\Delta t}\right)$, which allows us to write that the FFT is faster than the brute force approach when:

$$(K+1)^2 \log \frac{\delta\,(K+1)}{\Delta t} \leq \frac{1}{4}\left(3 + \frac{\delta}{\Delta t}\right)(K+1)^2$$

which is equivalent to:

$$K \leq \frac{\Delta t}{\delta}\, 2^{\frac{1}{4}\left(3+\frac{\delta}{\Delta t}\right)} - 1$$

and thus a lower bound t reads:

$$t \leq \Delta t\, 2^{\frac{1}{4}\left(3+\frac{\delta}{\Delta t}\right)} - \delta$$

□

This lower bound t is typically a large value in road networks where individual links have large travel-times. Table 30 shows the value of t for some sample values of δ and Δt. The exact value of t can be obtained by computing summation (172) numerically.

	$\Delta t = 0.1$	$\Delta t = 0.2$	$\Delta t = 0.5$	$\Delta t = 1$
$\delta = 90$	$1.51 \cdot 10^{65}$	$4.11 \cdot 10^{31}$	$4.93 \cdot 10^{11}$	$1.6 \cdot 10^{5}$
$\delta = 60$	$4.00 \cdot 10^{42}$	$2.11 \cdot 10^{20}$	$1.50 \cdot 10^{7}$	$9.17 \cdot 10^{2}$
$\delta = 30$	$1.06 \cdot 10^{20}$	$1.08 \cdot 10^{9}$	$4.59 \cdot 10^{2}$	4.57
$\delta = 15$	$5.44 \cdot 10^{8}$	$2.47 \cdot 10^{3}$	2.41	$1.27 \cdot 10^{-1}$

Table 30. Lower bound $\boldsymbol{t = \Delta t \cdot 2^{\frac{1}{4}\left(3+\frac{\delta}{\Delta t}\right)} - \delta}$ (minutes) for which the FFT approach is faster than the brute force approach for the computation of the convolution product in the main algorithm. Valuesof $\boldsymbol{\delta}$ and $\boldsymbol{\Delta t}$ are given in seconds.

22.4.2. Acceleration of Algorithm 29 with localization

As shown in Section 22.4.1, the runtime of the FFT based solution is a function of δ and decreases as the value of δ increases. The value of δ that is used in the algorithm is bounded by the minimum realizable travel-time across the entire network. However, in general, road networks are heterogeneous and contain a large range of minimum realizable travel-times. This section presents an optimization that can significantly improve the runtime of the algorithm by exploiting the disparity of these local δ values.

Proposition 31. Let β_{ij} be the minimum realizable travel-time on link (i,j) with $\delta_{ij} = \beta_{ij} - \epsilon$ $(0 < \epsilon < \beta_{ij})$ and τ_i be the budget up to which the cumulative distribution function $u_i(\cdot)$ has been computed for node i. For correctness, the invariant

$$\tau_i \leq \min_j(\tau_j + \delta_{ij})\ \forall (i,j) \in A \tag{173}$$

must be satisfied throughout the execution of the algorithm.

Proof. Assume that this invariant can be violated. Then, it is possible to compute the cumulative distribution function $u_i(\cdot)$ at some node i such that $\tau_i > \min_j(\tau_j + \delta_{ij})$, which in turn means that $\tau_i - \tau_j > \delta_{ij}$ forat least one node j. This implies that $\exists$ t' such that $u_i(t')$ was computed using the product of a downstream cumulative distribution function $u_j(t' - \omega)$ and $p_{ij}(\omega)$, where $u_j(t' - \omega)$ is unknown because $\tau_j < t' - \Delta_{ij}$ and $p_{ij}(\omega) > 0$ because $\omega > \delta_{ij}$. This value of the cumulative distribution function $u_i(t')$ is undefined and the SOTA algorithm fails. Therefore, for correctness the invariant should not be violated. □

When computing the cumulative density function $u_i(\cdot)$ using local δ_{ij} values, the growth of τ_i is different across the nodes i, unlike in our original algorithm (Algorithm 29) where the τ_i grow at the constant uniform rate δ. Furthermore, when $u_i(\cdot)$ is updated asynchronously using the invariant $\tau_i \leq \min_j(\tau_j + \delta_{ij}), (i,j) \in A$, the order in which the nodes are updated impacts the runtime of the algorithm, as illustrated in Example 32.

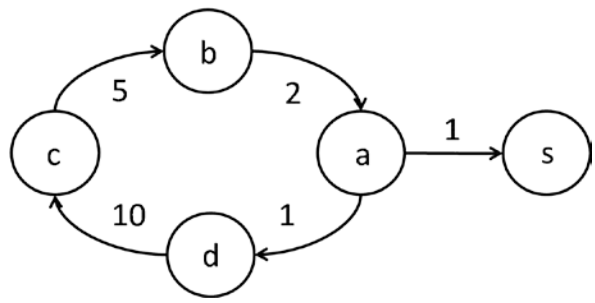


Figure 4.1. Example of a simple loop. The δ value for each link is given along the link.

Example 32. To illustrate how the order in which the nodes are updated impacts the runtime of Algorithm 29, consider the network in Figure 4.1. The value of τ_i and the computation time depends on the order in which the nodes are considered. In the worst case, as a lower bound, we assume that $u_i(\cdot)$ is updated based on the values of its constraint nodes in the previous iteration. Table 31 shows the sequence of updates for four iterations when using the constraints τ_i from the previous iteration. Notice that the update pattern is cyclic every four iterations. The highest speedup is achieved when the nodes in the loop are considered in topological order. Table 32 shows the sequence of updates when the nodes are considered in the topological order (a, b, c, d). As seen in Table 32, the τ_i value for each node i can be incremented by the length of the shortest loop node i belongs to when the nodes are updated in this order. The topological order can be determined easily in this simple example, but such an ordering is unlikely to exist in realistic transportation networks. Table 33 shows the sequence of updates when the nodes are considered in the order (d, c, b, a). It can be clearly seen that this ordering is much more inefficient than the ordering (a, b, c, d). Furthermore, without the local δ optimization, the algorithm can only update u_i by one step at each iteration, since the minimum δ_i value is 1 in this example. This simple example shows how local δ optimization can provide large improvements in the runtime.

Iter.	a	b	c	d
1	1	2	5	10
2	11	3	7	15
3	16	13	8	17
4	18	18	18	18

Table 31. τ_i values when computing u_i constrained on previous iteration.

Iter.	a	b	c	d
1	1	3	8	18
2	19	21	26	36
3	37	39	44	54
4	55	57	62	72

Table 32. τ_i values when computing u_i in the order (a, b, c, d).

Iter.	d	c	b	a
1	10	5	2	11
2	15	7	13	16
3	17	18	18	18
4	28	23	20	29

Table 33. τ_i values when computing u_i in the order (d, c, b, a).

Given that the runtime of the SOTA algorithm depends on the update order, we would like to find an optimal ordering that minimizes the runtime of the algorithm. The first step in finding such an optimal ordering is to formalize the runtime of the FFT SOTA algorithm.

Definition 33. The computation time of the cumulative density function $u_i(\cdot)$ can be minimized by finding the ordering that solves the following optimization problem.

$$\text{minimize}_{(\tau_i^{k_i}, K_i)} \tag{174}$$

$$\begin{aligned}
\text{subject to} \quad & \tau_i^{k_i} \leq \tau_j^{k_j} + \delta_{ij} && \forall \tau_i^{k_i}, \tau_j^{k_j}\ s.t.\ (i,j) \in A, \\
& && C(i,k_i) < C(j,k_j+1) \\
& \tau_r^{K_r} \geq T && \\
& \tau_s^1 \geq T && \\
& \tau_i^1 \geq \Delta t && \forall i \in N, i \neq s \\
& \tau_i^{k+1} > \tau_i^k && \forall i \in N
\end{aligned}$$

where $\tau_i^{k_i}$ is the budget up to which $u_i(\cdot)$ has been computed in the k_i^{th} iteration of computing $u_i(\cdot)$, $C(\cdot,\cdot)$ is an index on the order in which nodes are updated such that $C(i,k_i)$ denotes when node i is updated for the k_i^{th} time and K_i is the total number of iterations required for node i.

The optimal order in which $u_i(\cdot)$ is computed might result in updating some set of nodes multiple times before updating another set of nodes.

Algorithm 34. Optimal order for updating $\boldsymbol{u}(\cdot)$

Step 0. Initialization.
$\tau(i) = 0, \forall i \in N, i \neq r, i \neq s$ {where r is the origin and s is the destination}
$\tau(s) = \infty, \tau(r) = T$
$\psi := \{(r, \tau(r))\}$ {ψ is a priority queue data structure}
$\chi := \{(r, \tau(r))\}$ {χ is a stack data structure}
Step 1. Update
$(v, \theta) =$ ExtractMax(ψ)
$\tau(v) := \theta$
Push$(\chi, (v, \tau(v)))$
$\pi :=$ Children(v)
For $k := 1$ to size(π)
If $((\pi[k] \neq s)$ and $(\tau(v) - \delta_{v\pi[k]}) > 0)$
$\tau :=$ max(Extract$(\psi, \pi[k]), \tau(v) - \delta_{v\pi[k]})$
Insert$(\psi, (\pi[k], \tau))$
Step 2. Termination
If $\psi := \emptyset$ stop;
Otherwise go to Step 1.

Proposition 34. The ordering that gives the optimal solution to the optimization problem (174) can be obtained using Algorithm 34 in $O(\frac{mT}{\Delta t}\log(n))$ time, where n and m are respectively the number of nodes and links in the network, Δt is the time discretization interval and T is the time budget.

Proof. Algorithm 34 begins at the termination condition of the SOTA problem, the source node being updated to the budget, and recursively builds (in reverse order) the optimal sequence of updates that allow the source node to be updated to the budget. Thus, the algorithm is initialized with the terminal condition of $\tau_r = T$. This is the initial constraint of the optimal ordering algorithm. Reaching this condition must be preceded by all the downstream nodes $j \in \pi_r$ of the source node$r$ being updated to at least $\tau_r - \delta_{rj}$, since the correctness of the algorithm requires the invariant in equation (173),

$$\tau_i \leq \min_j(\tau_j + \delta_{ij})\ \forall(i,j) \in A \tag{175}$$

to hold. Therefore, the initial constraint $\tau_r = T$ is relaxed by adding these new constraints to the constraint list ψ. At the same time, we also add $\tau_r = T$ to the optimal order stack χ. This will be the final update in the optimal ordering, since we are building the list from the last update to the first.

Once we have a set of new constraints, we need to decide which node to relax and how far to update τ_i. The contribution from a given node i to the objective function of the optimization problem (174) is minimized when the τ_i value for that node is minimized as much as possible. Furthermore, lowering the τ_i value for a node reduces the new constraints introduced when relaxing that node. Therefore, an optimal update should reduce the τ_i value of a node as much as possible such that invariant (173) is not violated.

The next step is to determine which constraint from the constraint list ψ to relax first. We need to show that that the order of relaxation guarantees that the algorithm will not introduce any new constraints that violate any updates done in previous relaxations. Picking the node i with the largest constraint τ_i in ψ guarantees this, since δ_{ij} is strictly positive and the new constraints $\tau_j\ (\forall j \in \pi_i)$ that are added satisfy the condition $\tau_j \leq \tau_i - \delta_{ij}$, which implies that node i cannot have a new constraint that is greater than its current constraint τ_i at any future point of the algorithm. Therefore, correctness is preserved by relaxing the node i with the largest constraint $\widehat{a}u_i$ in ψ and setting its value to $\tau_i - \delta_{ij}$. Node i is then added to the optimal order stack χ. The new constraints introduced by setting node i to this value are then added to ψ, if $\tau_i - \delta_{ij} > 0$ and $j \neq s$. It is unnecessary to add constraints if these conditions are not satisfied, since $u_i(t) = 0\ (\forall\, i \in N, t \leq 0)$ and $u_s(t) = 1\ (\forall\, t \geq 0)$. This process is performed recursively until the list ψ is empty. The process is guaranteed to terminate because the values of new constraints that are added when relaxing an existing constraint are monotonically decreasing.

The complexity of Algorithm 34 is $O(\frac{mT}{\Delta t}\log(n))$. The *Extract* and *Insert* operations of the Algorithm 34 can be replaced with a single *IncreaseKey* operation, which runs in $O(log(n))$

time (see [294,114] for details). The *IncreaseKey* operation will increase the key of a given node if the new key is greater than its existing value. This is exactly what the *Extract* and *Insert* operations are used for. The pseudo-code for Algorithm 34 uses the *Extract* and *Insert* operations to improve readability. The *ExtractMax* operation can be performed in constant time. Therefore, each iteration of the algorithm takes $O(log(n))$ time. In the worst case, each link might need to be updated $\frac{T}{\Delta t}$ times. Repeating this over the m links of the network, we obtain a complexity of $O(\frac{mT}{\Delta t}\log(n))$. □

The optimal order of updates (node and value) that computes the cumulative distribution function $u_r(T)$ of the origin r most efficiently is stored in the stack χ at the termination of the algorithm. Algorithm 34 works by taking the source node r and the time budget to which it needs to be updated T, and then recursively updating the set of constraints that need to be satisfied before $u_r(T)$ can be computed. At the first iteration, the source and its terminal value in the algorithm (the budget) are added to the stack, and the constraints that are required for updating the source to that value are stored in the heap ψ. At any given iteration, the largest value in the heap is extracted and added to the stack, since it is the most constrained node in the current working set. We leave further discussion of how the algorithm works to the proof of correctness in Appendix B.

22.4.3. Search space pruning by elimination of infeasible paths

The algorithm requires computing the cumulative distribution function $u_i(t)$ for every node in the graph. This can be prohibitively expensive even in reasonably sized road networks. Therefore, we need to constrain the search space of our algorithm. In this section, the algorithm is extended by adding a pruning algorithm that eliminates infeasible paths by removing unnecessary nodes during a preprocessing step. Consider an instantiation of the SOTA problem with an origin node r, destination node s and a travel-time budget of T. Since every link (i,j) in the network has a minimum realizable travel-time β_{ij} (as defined in Section 1), it follows that every path in the network must have a minimum realizable travel-time as well. For any path P_{ik}, let the minimum realizable path travel-time be α_{ik}. The value of α_{ik} can be found by running a standard deterministic shortest path algorithm such as Dijkstra's algorithm on the network with the link weights being the minimum realizable link travel-time corresponding to each link.

Proposition 35. Consider some arbitrary node i in the network. Let α_{ri} and α_{is} be respectively the minimum realizable travel-times from the origin to node i and from node i to the destination.

- If $\alpha_{ri} + \alpha_{is} > T$, we can safely remove this node from the network and ignore it when solving the SOTA problem.
- The cumulative distribution function $u_i(\cdot)$ only needs to be computed for the time interval $\alpha_{is} \leq t \leq T - \alpha_{ri}$.

Proof.

1. If $\alpha_{ri} + \alpha_{is} > T$, the minimum realizable travel-time from the origin to the destination

through node i is greater than the travel-time budget. Therefore, no feasible path exists through node i.

2. The minimum realizable travel-time from the origin to node i is α_{ri}. Therefore, no path in the dynamic programming recursion will query $u_i(t)$ for $t > T - \alpha_{ri}$. The minimum realizable travel-time from node i to the destinationis α_{is}. Therefore, $u_i(t)$ is zero for $t < \alpha_{is}$.

□

By performing an all destinations shortest path computation from the source and an all sources shortest path computation from the destination, we can significantly prune both the size of the network required when solving the SOTA problem and the time interval for which the cumulative distribution function $u_i(\cdot)$ needs to be computed for each node. This pruning algorithm is inspired by the Reach heuristic ([167]) for deterministic shortest paths. For a graph with n nodes and m links, the time complexity of Dijkstra's algorithm is $O(m + n \log n)$ ([114]), which is dominated by the complexity of the SOTA algorithm. Thus, the cost of pruning is negligible compared to the complexity of the SOTA algorithm. Furthermore, state of the art shortest path algorithms can run in near constant time (see [48,159]) when the minimum realizable travel-time does not vary with time. It should be noted that this is a very conservative pruning algorithm and that further runtime reductions can be achieved using more aggressive pruning methods, which we are currently exploring.

22.5. Implementation of the algorithm in the Mobile Millennium system

In this section, the performances of the routing algorithms introduced in this chapter are tested on two specific traffic estimates from the *Mobile Millennium* traffic information system:

- Sample-based representation of the velocity map on the Bay Area highway network (Figure 5.1): this output is produced by the *v-CTM EnKF* algorithm described in Chapter 14 that fuses loop detectors counts, radars speed and spatially sampled probe speeds into a partial differential equation flow model coupled with an ensemble Kalman filter estimation algorithm. This estimate is updated every 30 seconds and has a space resolution of approximately 400 meters. Link travel-time distributions representing model uncertainty on the travel-time at the mean speed can be directly computed from this output and used by our routing algorithm.
- First two moments representation of link travel-times on the San Francisco arterial network (Figure 5.2): this output is produced by a machine learning algorithm described in Chapters 18 and 19 using a mixture of real-time and historical probe generated travel-times. The link travel-time distributions represent individual commuters travel-time and can be directly used by our routing algorithm.

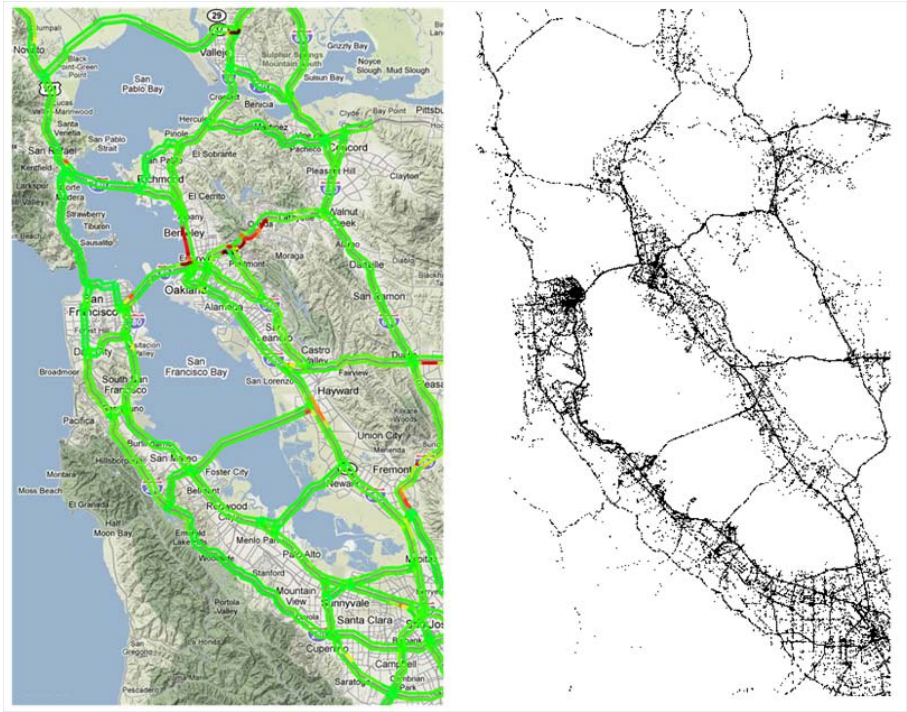


Figure 5.1. Best viewed in color. Velocity estimates on the Bay Area highway network. Right: Cumulated probe speed measurements collected on July 29th by the *Mobile Millennium* system. Left: Velocity estimates on July 29th at 9 pm; most of the network is in free flow, active bottlenecks correspond to red spots.

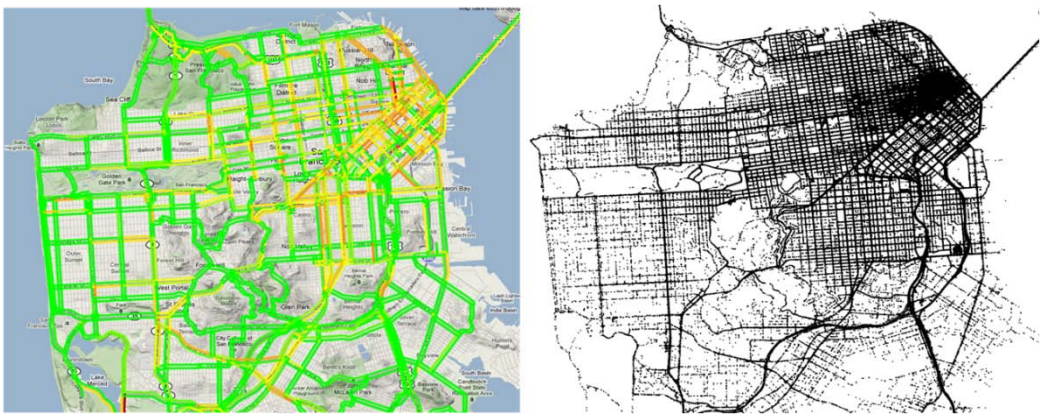


Figure 5.2. Best viewed in color. Travel-time estimates on the San Francisco arterial network. Right: Cumulated probe location measurements collected on July 29th by the *Mobile Millennium* system. Left: Travel-time estimates on July 29th at 8 pm; green links correspond to free flow travel-times and red spots correspond to congestion.

In the following sections, we illustrate the practical applicability of these algorithms in real-time traffic information systems. We show that the novel optimization techniques developed in this work improve the tractability of the problem, compared to existing solutions, and make the SOTA problem feasible to solve in an operational setting.

22.5.1. Numerical results

The algorithms are tested on the arterial and highway road networks described in the previous section. The highway network from Figure 5.1 is a relatively sparse network with short loops at ramps and intersections, and large loops covering the entire network. This network contains 3639 nodes and 4164 links. The San Francisco arterial network of Figure 5.2 is a dense network with a large number of loops with varying lengths, with 1069 nodes and 2644 links.

The algorithm is coded in Java and executed on a Windows 7 PC with a 2.67Ghz Dual Core Intel Itanium processor and 4GB of RAM. We use the open source Java libraries JTransforms ([335]) and SSJ ([222]) for FFT computations and manipulating probability distributions. Time-varying link travel-time distributions are obtained a-posteriori from the traffic estimation models described above. Both travel-time models assume that the link travel-times are independent with respect to each other.

We consider the following performance metrics:

- *Runtime*: computation time for the different variants of our algorithm compared to similar existing routing algorithms and specific runtime improvement provided by the speed up techniques introduced in this work.
- *On-time arrival guarantee*: sampling the *Mobile Millennium* traffic estimates enable the generation of realistic user travel-time realizations. The probability of arriving on time for both the SOTA policy and least expected travel-time path are compared. The performance of the SOTA algorithm under different traffic conditions and route types is also analyzed.
- *En route re-routing*: performance of the algorithm on a real test case on which the ability of the routing module to provide adaptive route choices depending on traffic conditions is presented.

Runtime performance

The runtime of the algorithm is a critical factor in its usability for real-time routing applications. The lack of routing choices that incorporate reliability guarantees, in any of the commonly used routing applications, is in part due to the intractability of executing them efficiently. In this section, we show that the algorithms in this chapter have the potential to bridge this gap.

As shown in the complexity analysis from Section 1, the algorithm introduced in this chapter is linear in the size of the network and pseudo-polynomial in the ratio $T/\Delta t$, where T is the time budget of the user, and $eltat$ is the discretization interval of time. It was argued that this algorithm performs better than the existing solution to the SOTA problem, in theory, for most practical routing problems. In this section we present empirical results to validate this claim. Figure 5.3 shows the actual run-times (in CPU time) for two sample origin-destination (OD) pairs, when computing the optimal policy over a range of travel-time budgets.

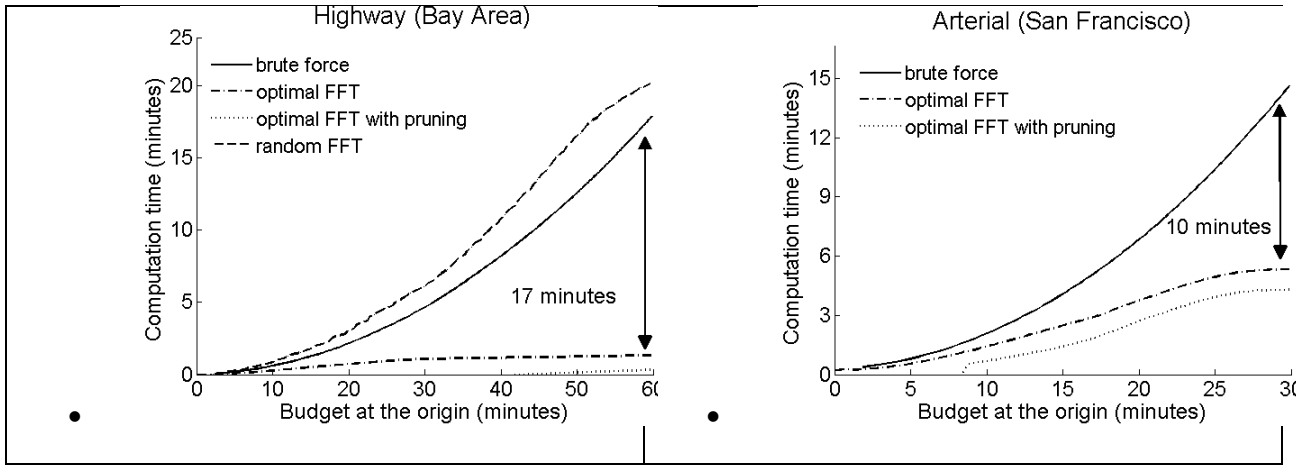


Figure 5.3. Illustration of the tractability of the problem. Comparison of run-times (CPU time) for the brute force convolution (solid line), randomly ordered FFT (dashed line), optimally ordered FFT (dot-dash line) and optimally ordered FFT with pruning (dotted line): Left: Highway network Runtime for computing the optimal policy from Berkeley to Palo Alto. The time discretization (Δt) is 0.5 seconds. Right: Arterial network Runtime for computing the optimal policy for a route from the Financial District (Columbus and Kearny) to the Golden Gate Park (Lincoln and 9th). The time discretization (Δt) is 0.2 seconds.

As illustrated in Figure 5.3, the FFT based algorithm with optimal ordering performs significantly better than the brute force approach in both networks (17 minute gain for a 1 hour policy on the highway network and 10 minute gain for a 30 minute policy on the arterial network) and makes the SOTA problem more tractable for real-time applications. Our algorithm performs much better on the highway network, where most of the loops have large minimum travel-times and allow the cumulative distribution function $u_i(\cdot)$ to be updated in larger increments (as explained in Section 22.4.2), making the convolution product more efficient. The FFT algorithm with a random update order performs quite poorly, especially in the case of the arterial network, where it takes approximately 70 minutes to compute a 30 minute policy.[22] This observation agrees with the lower bound in Proposition 30 and the example values in Table 30, since the relative efficiency of the FFT algorithm increases exponentially with the travel-time of the minimum length loop for each node.

Comparison with classical routing algorithms

Most common routing algorithms rely solely on the knowledge of the travel-time as a deterministic quantity when generating optimal route choices. This deterministic travel-time can be inferred for instance from the speed limitations on the network, from historical realized travel-times (e.g. average, worst case), or from a real-time deterministic output of a traffic information system (e.g. mean travel-time, median travel-time).

[22]The computation time for the FFT algorithm with a random update order is not displayed in Figure 5.3, since it takes much longer than the other algorithms, to improve the readability of the plot.

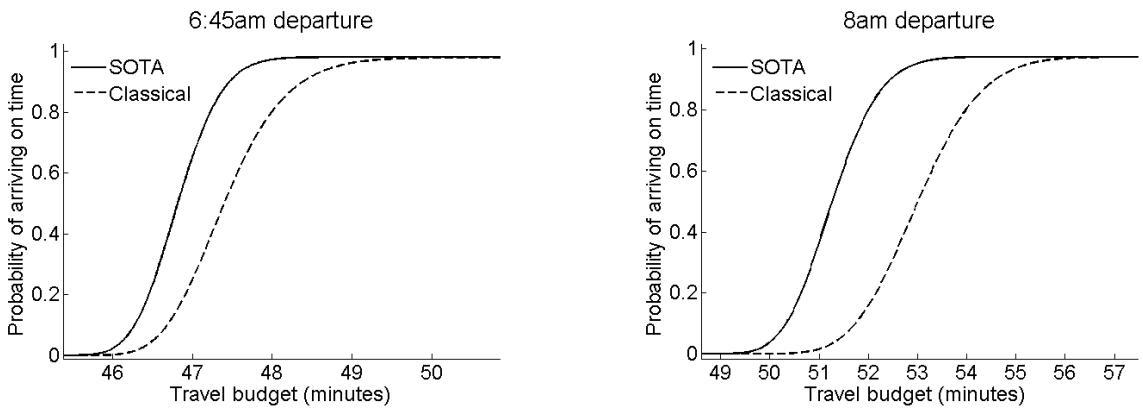


Figure 5.4. July 29th: Probability of arriving on time at Palo Alto when departing from Berkeley: Left: Departure at 6:45 am. The SOTA policy (solid line) provides a higher probability of arriving on time than the choice of the LET route (dashed line). The distance-based route and the speed limit based route are the same as the LET route at this time (Highway I-880). Right: Departure at 8:00 am. The SOTA policy and the LET route provide the same probability of arriving on time(solid line). The speed limit based route and the distance based route (dashed line) are inferior for this criterion.

We compare the performance of the SOTA algorithm to these classical methods on both the highway and arterial networks defined above. First we instantiate the case of a commuter traveling on the highway network from Berkeley (latitude: 37.87201, longitude: -122.3056) to Palo Alto (latitude: 37.4436, longitude: -122.1176), on July 29th, on two departure times, $6{:}45$ am and $8{:}00$ am. This is a typical Bay Area commute experienced by a large population of the San Francisco Bay Area every day. Different optimal routes are possible; for instance the route with minimum expected travel-time (LET route), the route which minimizes the travel-time at the speed limit (speed limit based route), the route with the shortest distance (distance based route), the route which maximizes the probability of arriving on time (SOTA route). We generate optimal routes for all of the above strategies using traffic estimates from the *Mobile Millennium* system. The LET and SOTA routes are computed using the time-dependent implementations of these algorithms. Once the routes are determined, we compute the travel-time distributions for each of these routes a posteriori (this is computed by performing a convolution of the individual link travel-time distributions for the links of each route) and determine the probability of arriving within the budget range of 0 to 60 minutes. Figure 5.4 presents the probability of arriving on time for each of the routing strategies during the budget range.

As traffic conditions vary, the time-dependent SOTA and LET routes change accordingly, while the speed limit based route and the distance based route are static. When departing at $6{:}45$ am, the maximal point wise difference between the SOTA route and the LET route is around 0.4, corresponding to a budget of about 47 minutes. For this budget, the commuter has a 0.65 probability of arriving on time on the SOTA route and a 0.25 probability of arriving on time on the LET route. Naturally, for both the SOTA and LET solutions, the risk of not making the destination on time increases as the budget decreases. However, as illustrated in Figure 5.4, the SOTA route always provides a higher probability of arriving on time. Furthermore, the SOTA algorithm can provide the user with the probability of on time arrival (i.e. the risk level) for any range of time budgets the user is interested in when the policy is computed, which allows the user to determine whether the risk is acceptable or not and act accordingly.

One may note that the morning congestion build up is visible in Figure 5.4 since the sharp increase in the cumulative distributions between the left subfigure (6:45 am) and right subfigure (8:00 am) evolves from around 47 minutes to around 52 minutes in this time period. The increase in the area between the SOTA cumulative probability (solid line from Figure 5.4) and a second choice route (dashed line from Figure 5.4) during this time period illustrates that the SOTA policy can dominate classical optimal routes by a higher margin in congestion and non-stationary phases when there is a high uncertainty on the realized travel-time.

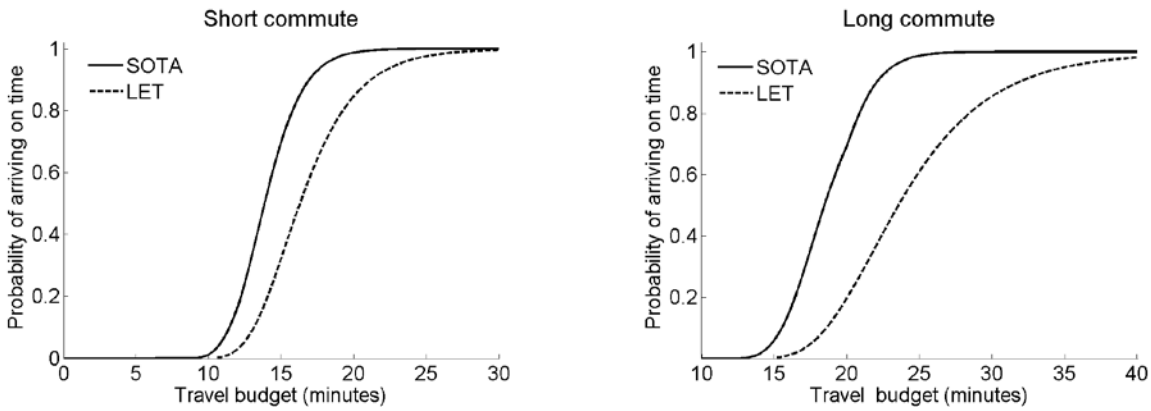


Figure 5.5. February 1st: Probability of arriving on time at Left: Fulton and 2nd, and Right: Lincoln and 9th when departing from the Financial District (Columbus and Kearny) at 8:50 pm. As the graphs imply, the commute to Lincoln and 9th is a longer route than Fulton and 2nd. The relative benefit of using a SOTA policy increases with the route length, since the longer route contains more route choices in the arterial network.

We also compare the performance of the SOTA algorithm on the San Francisco arterial network for two routes starting in the Financial District (Columbus and Kearny) and ending at 1) Lincoln and 9th, and 2) Fulton and 2nd. As seen in Figure 5.5, the maximal point wise difference between the SOTA route and the LET route for the first example is around 0.4, corresponding to a budget of about 15 minutes, where the commuter has a 0.75 probability of arriving on time on the SOTA route and a 0.35 probability of arriving on time on the LET route. For the second example, the maximal point wise difference is around 0.5, corresponding to a budget of about 22 minutes, where the commuter has a 0.89 probability of arriving on time on the SOTA route and a 0.39 probability of arriving on time on the LET route. The relative benefit of using a SOTA policy increases with the length of a route, since the longer route contains more route choices (with varying cumulative distribution functions) in the arterial network. In the highway network examples from Figure 5.4, the SOTA policy dominates the LET path for only a 3-5 minute window of the travel-time budget. However, in the arterial network examples, the SOTA policy dominates the LET path for approximately a 10-15 minute window, even though the route lengths were shorter. The reason for this disparity is not limited to these specific examples and is due to the inherent differences of the two networks. The highway network has a limited number of reasonable travel choices from Berkeley to Palo Alto and relatively low variance of the travel-time distributions. Whereas, the arterial network has a large number of route options and highly variable traffic conditions due to the uncertainty introduced by pedestrians, stop signs, traffic lights etc. This results in routes with many distinct cumulative distribution functions and leads to an improved SOTA policy, since the SOTA policy is the upper envelope of all these

distinct cumulative distributions functions as illustrated below.

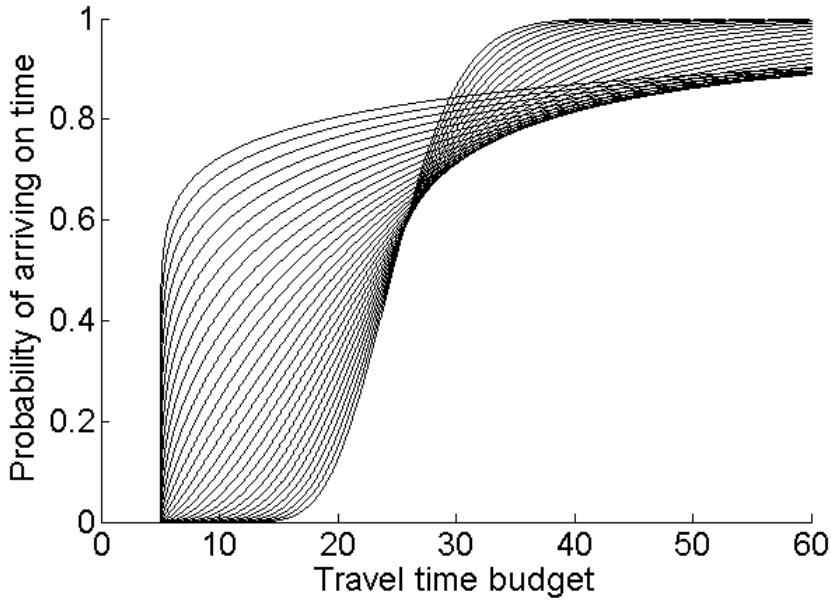


Figure 5.6. A family of travel-times distributions modeled using 30 shifted gamma distributions, each with the same mean travel time of 25 minutes and a minimum travel time of 5 minutes. The shape parameter of the distributions ranges from 4 to 0.13and the scale parameter of the distributions ranges from 5 to 150. The SOTA policy will be the upper envelope of all the curves. The LET path could be any of the curves and in the worst case even be the path that minimizes the probability of arriving on time for a given budget.

To further illustrate the benefits of the SOTA algorithm when the travel-time distributions are very heterogeneous, we consider the very simple example of two nodes connected by 30 different links each gamma distributed with a mean of 25 minutes, but having different shape and scale parameters. As illustrated in Figure 5.6, the travel-time distributions for the links are vastly different even though they have the same expected travel-time. A LET routing algorithm could pick any of these links as the optimal solution and in the worst case pick the path that is the worst option for the travel-time budget. On the other hand, the SOTA algorithm picks the best path for a given time-budget, which graphically corresponds to the upper envelope of all the curves. As illustrated by this simple example, the SOTA algorithm has the potential for being relatively more superior to a LET path when the number of travel choices increases and their travel-time distributions are not similar.

Test case: evening rush commute within the city of San Francisco

In this section, we illustrate the adaptive nature of the SOTA algorithm presented in this paper. The output of the algorithm is a policy which accounts for the stochastic nature of link travel-times. Given a budget T, the optimal policy computation encompasses the design of a decision process at each possible intersection of the network; the choice of the optimal route to take from this node depends on the remaining budget.

Figure 5.7. Commute from point A to point B: two drivers depart from point A at $8{:}50$ pm on February 1st with a budget of 20 minutes to reach point B. They are routed by the SOTA module. Because their realized travel times differ, their recommended routes differ. The first driver is suggested to turn left at point C, whereas the second driver is suggested to drive straight.

Here we consider two drivers commuting from point A to point B (see Figure 5.7) on February 1st. They depart from point A at $8{:}50$ pm and desire to reach point B before $9{:}10$ pm; i.e. their travel budget is 20 minutes. We assume that both drivers are equipped with a mobile device on which the output of the SOTA algorithm is available. At each intersection, they follow the turn directions given by the optimal policy. For this test case we sample the drivers travel-time from the output of the real-time arterial traffic estimation module [183] from the *Mobile Millennium* system.

Because of different driving behaviors, external factors, link travel-time stochasticity, both drivers will experience different travel-times during their commute. The strength of the SOTA algorithm is that the optimal route choice given by the algorithm is given at every intersection in function of the remaining budget. As illustrated in Table 35, the optimal route to take from point C includes a left turn for low values of the remaining budget. Because the second driver experienced a larger travel-time on the path from point A to point B, he is advised to take a left turn and to follow a path with more variability, and thus higher risk, which may be more appropriate to his situation. The first driver continues straight at point C.

Remaining budget b at point C	Turn direction from East
$b \leq 12$ minutes	Take a left turn
$b \geq 12$ minutes	Continue straight

Table 35. The optimal policy at point C routes on different paths depending on the value of the remaining travel budget with respect to 12 minutes. The probability of arriving on time at B when remaining budget at C is 12 minutes is 0.56.

22.6. Conclusions

This chapter considers the reliable routing problem of maximizing the probability of on time arrival, which is a computationally difficult problem, and presents algorithmic methods that improve the tractability of the problem over existing methods. First,we prove the existence of a single iteration convergence algorithm to the continuous time version of the problem. The update property of this solution in conjunction with an optimal ordering process and the use of the Fast Fourier Transform is then used to construct an efficient algorithm for computing a discrete approximation to the problem. The correctness of this algorithm when extended to time-varying and correlated link travel-times is also shown. Finally, the theoretical results are validated by implementing the algorithm in the *Mobile Millennium* traffic information system. Numerical results show that the algorithm provides a significant reduction in the computation time over existing methods, especially in highway networks. Our goal is toprovide the theoretical basis for a tractable implementation of adaptive routing with reliability guarantees in an operational setting.

23. A general phase transition model for vehicular traffic

Motivated by the need to explore alternative methods for data assimilation, this chapter describes a second-order vehicular traffic flow model. One strength of this model is that it structured in a way that makes possible the fusion of both density information and velocity data. In addition it has potential to reproduce traffic flow more accurately than existing models.

Recall that first-order models assume that flow behavior is ruled by first-order differential equations. Although they account for variability of speed along a section of travel, first-order models neglect the effects of acceleration and deceleration. In such a framework, fusion of both density and velocity data rely on an averaged and approximate fundamental diagram.

The specific capability of this second-order model to integrate velocity measurements (through proper treatment of the second state variable of the problem) is a significant advantage over any first-order model for which the density-flux relation is single valued. While the proper use of this key feature for data assimilation is still an open problem, the work described in this chapter opens the door to promising outcomes for highway traffic state estimation.

This chapter develops an extension of the Colombo phase transition model. The congestion phase is described by a two-dimensional zone defined around a standard fundamental diagram. General criteria to build such a set-valued fundamental diagram are enumerated, and instantiated on several standard fluxes with different concavity properties. The solution to the Riemann problem in the presence of phase transitions is obtained through the design of a Riemann solver, which enables the construction of the solution of the Cauchy problem using wavefront tracking. The free-flow phase is described using a Newell-Daganzo fundamental diagram, which allows for a more tractable definition of phase transition compared to the original Colombo phase transition model. The accuracy of the numerical solution obtained by a modified Godunov scheme is assessed on benchmark scenarios for the different flux functions constructed.

23.1. Background

First order scalar models of traffic. Hydrodynamic models of traffic go back to the 1950's

with the work of Lighthill, Whitham and Richards [226,285], who built the first model of the evolution of vehicle density on the highway using a first order scalar hyperbolic *partial differential equation* (PDE) referred to as the LWR PDE. Their model relies on the knowledge of an empirically measured *flux function*, also called the *fundamental diagram* in transportation engineering, for which measurements go back to 1935 with the pioneering work of Greenshields [173]. Numerous other flux functions have since been proposed in the hope of capturing effects of congestion more accurately, in particular: Greenberg [171], Underwood [324], Newell-Daganzo [126,256], and Papageorgiou [331]. The existence and uniqueness of an *entropy* solution to the *Cauchy problem* [295] for the class of scalar conservation laws to which the LWR PDE belongs go back to the work of Oleinik [263] and Kruzhkov [215], (see also the seminal article of Glimm [164]), which was extended later to the *initial-boundary value problem* [69], and specifically instantiated for the scalar case with a concave flux function in [149], in particular for traffic in [309]. Numerical solutions of the LWR PDE go back to the seminal *Godunov scheme* [166,223], which was shown to converge to the entropy solution of the first order hyperbolic PDE (in particular the LWR PDE). In the transportation engineering community, the Godunov scheme in the case of a triangular flux is known under the name of *Cell Transmission Model* (CTM), which was brought to the field by Daganzo in 1995 [126,127] (see [220] for the general case), and is one of the most used discrete traffic flow models in the literature today [88,124,202,227,254,266,328].

Set-valued fundamental diagrams. The assumption of a Greenshields fundamental diagram or a triangular *fundamental diagram*, which significantly simplifies the analysis of the model algebraically, led to the aforementioned theoretical developments. Yet, experimental data clearly indicates that while the free flow part of a fundamental diagram can be approximated fairly accurately by a straight line, the congested regime is set valued, and can hardly be characterized by a single curve [327]. An approach to model the set-valuedness of the congested part of the fundamental diagram consists in using a second equation coupled with the mass conservation equation (i.e. the LWR PDE model). Such models go back to Payne [270] and Whitham [336] and generated significant research efforts, but led to models with inherent weaknesses pointed out by del Castillo [90] and Daganzo [123]. These weaknesses were ultimately addressed in several responses [62,266,351], leading to sustained research in this field.

Motivation for a new model. Despite the existing research, modeling issues remain in most 2×2 models of traffic available today. For instance, the *Aw-Rascle* model [62] can introduce vanishing velocities below jam density, which is not a classical assumption in traffic theory [156]. In agreement with the remarks from Kerner [211,212] affirming that traffic flow presents three different behaviors, *free-flow*, *wide moving jams*, and *synchronized flow*, Colombo proposed a 2×2 phase transition model [111,112] which considers *congestion* and *free-flow* in traffic as two different phases, governed by distinct evolutionary laws (see also [165] for a phase transition version of the Aw-Rascle model). The well-posedness of this model was proved in [113] using *wavefront tracking* techniques [80,193]. In the phase transition model, the evolution of the parameters is governed by two distinct dynamics; in *free-flow*, the Colombo phase transition model is a classical first order model (LWR PDE), whereas in *congestion* a similar equation governs the evolution of an additional state variable, the *linearized momentum* q. The motivation for an extension of the 2×2 phase transition model comes from the following items, which are addressed by the class of models presented in this chapter:

1. *Phases gap*. The phase transition model introduced by Colombo in [111] uses a Greenshields flux function to describe *free-flow*, which despite its simple analytical expression yields a fundamental diagram which is not connected and thus a complex definition of the solution to the Riemann problem between two different phases. We solve this problem by introducing a Newell-Daganzo flux function for *free-flow*, which creates a non-empty intersection between the congested phase and the *free-flow* phase, called *metastable phase*. It alleviates the inconvenience of having to use a shock-like phase transition in many cases of the Riemann problem between two different phases.
2. *Definition of a general class of set-valued fundamental diagrams*. The work achieved in [112] enables the definition of a set-valued fundamental diagram for the expression of the velocity function introduced. However, experimental data shows that several types of fundamental diagram exist, with different congested domain shapes. In this chapter we provide a method to build an arbitrary set-valued fundamental diagram which in a special case corresponds to the fundamental diagram introduced in [111]. This enables one to define a custom-made set-valued fundamental diagram.

Organization The rest of the chapter is organized as follows. Section 23.2 presents the fundamental features of the Colombo phase transition model [112], which serves as the basis for the present work. In Section 23.3, we introduce the modifications to the Colombo phase transition model, and introduce the notion of *standard state* which provides the basis for the construction of a class of 2×2 traffic models. We also assess general conditions which enable us to extend the results obtained for the original Colombo phase transition model to these new models. Finally, this section presents a modified *Godunov scheme* which can be used to solve the equations numerically. The two following sections instantiate the constructed class of models for two specific flux functions, which are the Newell-Daganzo (affine) flux function (Section § 4) and the Greenshields (parabolic concave) flux function (Section 23.5). Each of these sections includes a discussion of the choice of parameters needed for each of the models, the solution to the Riemann problem, a description of the specific properties of the model, and a validation of the numerical results using a benchmark test. Finally, Section 23.6 presents some concluding remarks.

23.2. The Colombo phase transition model

The original Colombo phase transition model [111,112] is a set of two coupled PDEs respectively valid in a *free-flow* regime and *congested* regime:

$$\begin{cases} \partial_t \rho + \partial_x(\rho\, v_f(\rho)) = 0 & \text{in free} - \text{flow } (\Omega_f) \\ \begin{cases} \partial_t \rho + \partial_x(\rho\, v_c(\rho, q)) = 0 \\ \partial_t q + \partial_x((q - q^*)\, v_c(\rho, q)) = 0 \end{cases} & \text{in congestion } (\Omega_c) \end{cases} \tag{176}$$

where the state variables ρ and q denote respectively the density and the *linearized momentum* [112]. Ω_f and Ω_c are the respective domains of validity of the free-flow and congested equations of the model and are explicited below. The term q^* is a characteristic

parameter of the road under consideration. An empirical relation expresses the velocity v as a function of density in free-flow: $v := v_f(\rho)$, and as a function of density and linearized momentum in congestion: $v := v_c(\rho, q)$. Following usual choices for traffic applications [155], the functions below are used:

$$v_f(\rho) = \left(1 - \frac{\rho}{R}\right) V \text{ and } v_c(\rho, q) = \left(1 - \frac{\rho}{R}\right) \frac{q}{\rho}$$

where R is the maximal density or *jam density* and V is the maximal *free-flow speed*. The relation for free-flow is the *Greenshields* model [173] mentioned earlier while the second relation has been introduced in [111]. Since Ω_c has to be an invariant domain [295] for the congested dynamics from system (176), and according to the definition of v, the free-flow and congested domains are defined as follows:

$$\begin{cases} \Omega_f = \left\{(\rho, q) \in [0, R] \times [0, +\infty[\,, v_f(\rho) \geq V_{f-}, q = \rho\, V\right\} \\ \Omega_c = \left\{(\rho, q) \in [0, R] \times [0, +\infty[\,, v_c(\rho, q) \leq V_{c+}\,, \dfrac{Q^- - q^*}{R} \leq \dfrac{q - q^*}{\rho} \leq \dfrac{Q^+ - q^*}{R}\right\} \end{cases}$$

where V_{f-} is the minimal velocity in free-flow and V_{c+} is the maximal velocity in congestion such that $V_{c+} < V_{f-} < V$. R is the maximal density and Q^- and Q^+ are respectively the minimal and maximal values for q. The fundamental diagram in (ρ, q) coordinates and in $(\rho, \rho\, v)$ coordinates is presented in Figure 2.1.

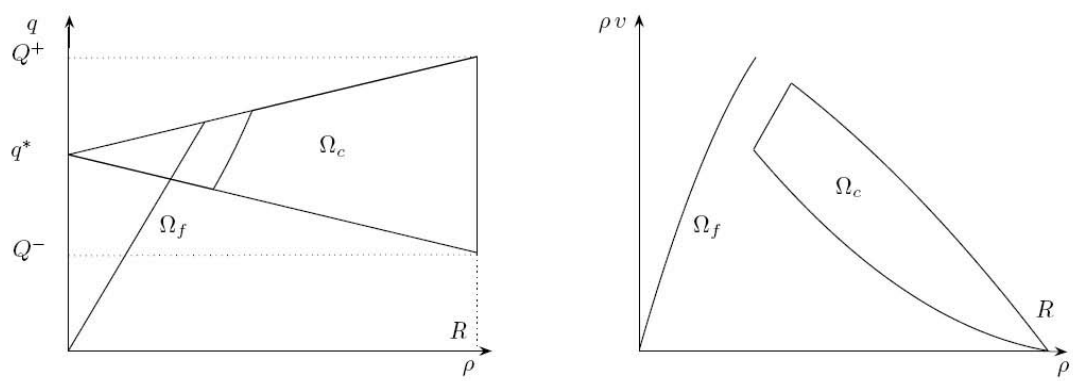


Figure 2.1. Colombo phase transition model. Left: Fundamental diagram in state space coordinates (ρ, q). Right: Fundamental diagram in density flux coordinates $(\rho, \rho\, v)$.

Remark 36. The congested part of system (176) is strictly hyperbolic if and only if the two eigenvalues of its Jacobian are real and distinct for all states $(\rho, q) \in \Omega_c$.

Remark 37. The 1-Lax curves are straight lines going through $(0, q^*)$ in (ρ, q) coordinates which means that along these curves shocks and rarefactions exist and coincide [316]. One must

note that the 1-Lax field is not *genuinely non-linear* (GNL). Indeed the 1-Lax curves are *linearly degenerate* (LD) for $q = q^*$ and GNL otherwise with rarefaction waves propagating in different directions relatively to the eigenvectors depending on the sign of $q - q^*$. The 2-Lax curves, which are straight lines going through the origin in $(\rho, \rho\, v)$ coordinates, are always LD.

23.3. Extension of the Colombo phase transition model

The approach developed by Colombo provides a fundamental diagram which is thick in congestion (Figure 2.1), and thus can model clouds of points observed experimentally (Figure 3.1).

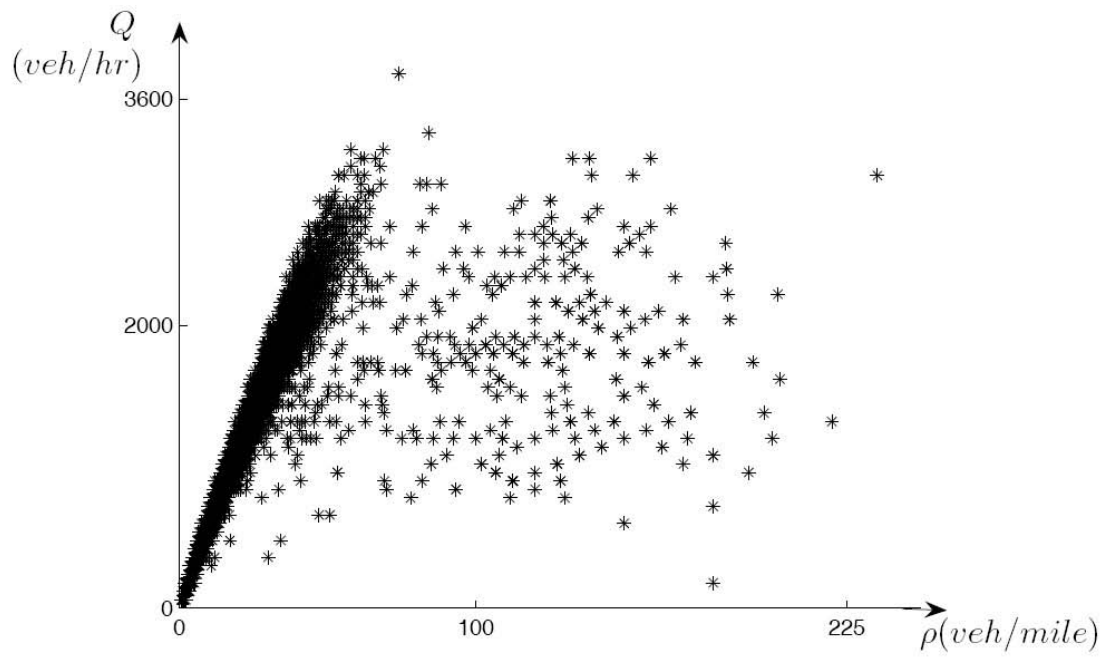


Figure 3.1. Fundamental diagram in density flux coordinates from a street in Rome. In congestion (high densities) the flux is multi-valued. Count C and velocity v were recorded every minute during one week. Flux Q was computed from the count. Density ρ was computed from flux and velocity according to the expression $Q = \rho\, v$ (see [75] **for an extensive analysis of this dataset).**

We propose to extend this approach by considering the second equation in congestion as modeling a perturbation [351,352]. The *standard state* (Definition 38) would be the usual one-dimensional fundamental diagram, with dynamics described by the conservation of mass. Perturbations can move the system off standard state, leading the diagram to span a two-dimensional area in congestion. A single-valued map is able to describe the free-flow mode, which is therefore completely described by the free-flow standard state.

Definition 38. We call *standard state* the set of states described by a one dimensional

fundamental diagram and the classical LWR PDE. In the following we respectively refer to the standard velocity and standard flux as the velocity and flux at the standard state.

In this section we present analytical requirements on the velocity function in congestion which, given the work done in [112], enable us to construct a 2×2 phase transition model. These models provide support for a physically correct, mathematically well-posed initial-boundary value problem which can model traffic phenomena where the density and the flow are independent quantities in congestion, allowing for multiple values of the flow for a given value of the density. Our framework allows to define the two dimensional zone span by the congestion phase according to the reality of the local traffic nature, which is not always possible with the original Colombo phase transition model.

23.3.1. Analysis of the standard state

We consider the density variable ρ to belong to the interval $[0, R]$ where R is the maximal density. Given the *critical density* [23] σ in $(0, R]$, we define the standard velocity $v^s(\cdot)$ on $[0, R]$ by:

$$v^s(\rho) := \begin{cases} V & for\ \rho \in [0, \sigma] \\ v_c^s(\rho) & for\ \rho \in [\sigma, R] \end{cases}$$

where V is the free-flow speed and $v_c^s(\cdot)$ is in $C^\infty((\sigma, R), \mathbb{R}^+)$. It is important to note that $v_c^s(\cdot)$ is a function of ρ only, as it is the case for the classical fundamental diagram. The standard flux $Q^s(\cdot)$ is thus defined on $[0, R]$ by:

$$Q^s(\rho) := \rho\, v^s(\rho) = \begin{cases} Q_f(\rho) := \rho\, V & for\ \rho \in [0, \sigma] \\ Q_c^s(\rho) := \rho\, v_c^s(\rho) & for\ \rho \in [\sigma, R] \end{cases}$$

In agreement with traffic flow features, the congested standard flux $Q_c^s(\rho)$ must satisfy the following requirements (which are consistent with the ones given in [89]).

1. *Flux vanishes at the maximal density*: $Q_c^s(R) = 0$.

This condition encodes the physical situation in which the jam density has been reached. The corresponding velocity and flux of vehicles on the highway is zero.

2. *Flux is a decreasing function of density in congestion*: $dQ_c^s(\rho)/d\rho \leq 0$.

This is required as a defining property of congestion. It implies that $dv_c^s(\rho)/d\rho \leq 0$.

3. *Continuity of the flux at the critical density*: $Q_c^s(\sigma) = Q_f(\sigma)$.

[23] Density for which the flux is maximal at the standard state. At this density the system switches between free-flow and congestion.

Even if some models account for a discontinuous flux at capacity, the *capacity drop* phenomenon [212], we assume, following most of the transportation community, that the flux at the standard state is a continuous function of density.

4. *Concavity of the flux in congestion*: $Q_c^s(\cdot)$.

The flux function at the standard state $Q_c^s(\cdot)$ must be concave on $[\sigma, \sigma_i]$ and convex on $[\sigma_i, R]$ where σ_i is in $(\sigma, R]$. Given the experimental datasets obtained for congestion (Figure 3.1), it is not clear in practice if the standard flux is concave or convex in congestion. The assumption made here is motivated in Remark 51.

Remark 39. In this chapter we instantiate the general model proposed on the most common standard flux functions, i.e. linear or concave, but the framework developed here applies to flux functions with changing concavity such as the Li flux function [320], although it yields a significantly more complex analysis.

23.3.2. Analysis of the perturbation

Model outline

In this section we introduce a perturbation q to the standard velocity in congestion.

Definition 40. The perturbed velocity function $v_c(\cdot,\cdot)$ is defined on Ω_c by:

$$v_c(\rho, q) = v_c^s(\rho)\,(1 + q) \tag{177}$$

where $v_c^s(\cdot) \in C^\infty((\sigma_-, R), \mathbb{R}^+)$ is the congested standard velocity function.

The standard state corresponds to $q = 0$, and the evolution of (ρ, q) is described similarly to the classical Colombo phase transition model [112] by:

$$\begin{cases} \partial_t \rho + \partial_x(\rho\, v) = 0 & \text{in free} - \text{flow} \\ \begin{cases} \partial_t \rho + \partial_x(\rho\, v) = 0 \\ \partial_t q + \partial_x(q\, v) = 0 \end{cases} & \text{in congestion} \end{cases} \tag{178}$$

with the following expression of the velocity:

$$v = \begin{cases} v_f(\rho) := V & \text{in free} - \text{flow} \\ v_c(\rho, q) & \text{in congestion.} \end{cases} \tag{179}$$

The perturbed velocity function defines the velocity in congestion whereas a Newell-Daganzo function describes the velocity in free-flow. The analytical expression of the free-flow and congested domains as explicited in (180) is motivated by the analysis conducted in Table 36 and the necessity for these domains to be invariants [295] for the dynamics (178) in order to have a

well-defined Riemann solver [321].

$$\begin{cases} \Omega_f = \{(\rho,q) \mid (\rho,q) \in [0,R] \times [0,+\infty[\,, v_c(\rho,q) = V\,, & 0 \le \rho \le \sigma_+\} \\ \Omega_c = \left\{(\rho,q) \mid (\rho,q) \in [0,R] \times [0,+\infty[\,, v_c(\rho,q) < V\,, & \frac{q_-}{R} \le \frac{q}{\rho} \le \frac{q_+}{R}\right\} \end{cases} \tag{180}$$

$\sigma_\pm$ is defined by $v_c(\sigma_\pm, \sigma_\pm\, q_\pm/R) = V$ and we assume that $V > 0$ and $q_- \le 0 \le q_+$. A definition of the complete set of parameters can be found in Section § 23.3.3 (See also Figure 3.2 for an illustration in the Newell-Daganzo case.).

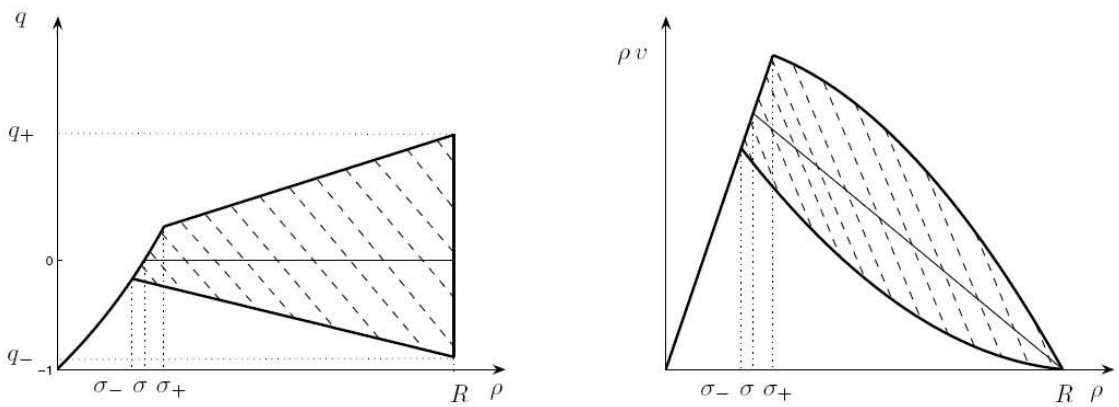


Figure 3.2. Newell-Daganzo standard flux function. Left: Fundamental diagram in state space coordinates. Right: Fundamental diagram in flux-density coordinates. The standard state is the usual triangular diagram. The congestion phase is two-dimensional (striped domain).

Definition 41. The set $\{(\rho,q) \mid v_c(\rho,q) = V,\ \ \sigma_- \le \rho \le \sigma_+\}$ defines the meta-stable phase. This phase defines transition states between the congestion phase and the free-flow phase.

Remark 42. The left boundary of the congested domain is a convex curve in (ρ,q) coordinates (in Figure 2.1 for the Colombo phase transition model as in Figure 3.2 for the new model derived). Thus Ω_c is not convex in (ρ,q) coordinates.

The analysis of the congestion phase of the model (178) is outlined in Table 36.

Eigenvalues	$\lambda_1(\rho,q) = \rho\,(1+q)\partial_\rho v_c^s(\rho) + v_c^s(\rho)\,(1+2\,q)$	$\lambda_2(\rho,q) = v_c^s(\rho)\,(1+q)$
Eigenvectors	$r_1 = \begin{pmatrix}\rho\\ q\end{pmatrix}$	$r_2 = \begin{pmatrix} v_c^s(\rho) \\ -(1+q)\,\partial_\rho v_c^s(\rho)\end{pmatrix}$
Nature of the Lax curves	$\nabla\lambda_1.r_1 = \rho^2\,(1+q)\partial^2_{\rho\rho} v_c^s(\rho) + 2\,\rho\,(1+2\,q)\,\partial_\rho v_c^s(\rho) + 2\,q\,v_c^s(\rho)$	$\nabla\lambda_2.r_2 = 0$
Riemann-invariants	q/ρ	$v_c^s(\rho)\,(1+q)$

Table 36. Congestion phase: algebraic properties of the general phase transition model.

Physical and mathematical considerations

Physical interpretation and mathematical conditions translate into the following conditions:

Condition 43. Positivity of speed. In order to maintain positivity of $v_c(\cdot,\cdot)$ on the congested domain, one must have:

$$\forall\, q \in [q_-, q_+] \quad 1 + q > 0 \tag{181}$$

which is satisfied if and only if $q_- > -1$.

Condition 44. Strict hyperbolicity of the congested system. In order for the congested part of (178) to be strictly hyperbolic, one must have:

$$\forall\, (\rho, q) \in \Omega_c \quad \lambda_1(\rho, q), \lambda_2(\rho, q) \in \mathbb{R}\ and\ \lambda_1(\rho, q) \neq \lambda_2(\rho, q).$$

Given the expression of the eigenvalues outlined in Table 36, and modulo a rearrangement, this yields:

$$\forall\, (\rho, q) \in \Omega_c \quad \rho\, \partial_\rho v_c^s(\rho) + q\big(v_c^s(\rho) + \rho\, \partial_\rho v_c^s(\rho)\big) \neq 0. \tag{182}$$

Since $v_c^s(\cdot)$ is positive and $\rho\, v_c^s(\cdot)$ is a decreasing function of ρ, this can always be satisfied for small enough values of q, and when instantiated for specific expressions of $v_c^s(\cdot)$, will result in a bound on the perturbation q.

Condition 45. Shape of Lax curves. For modeling consistency, we require the 1-Lax curves to be LD or to have no more than one inflexion point (σ_i, q_i). In the latter case they should be concave for $\rho \leq \sigma_i$ and convex for $\rho \geq \sigma_i$. Since$\nabla\lambda_1.r_1$ is the second derivative of the 1-Lax curve with respect to ρ, this condition can be enforced, for any (ρ, q) in the congested domain, by checking the sign of the expression:

$$\nabla\lambda_1.r_1 = \rho\,(2\,\partial_\rho v_c^s(\rho) + \rho\,\partial_{\rho\rho}^2 v_c^s(\rho)) + q\,(2\,v_c^s + 4\,\rho\,\partial_\rho v_c^s(\rho) + \rho^2\,\partial_{\rho\rho}^2 v_c^s(\rho)) \tag{183}$$

which has the sign of the first term for q small enough. So if $2\,\partial_\rho v_c^s(\rho) + \rho\,\partial_{\rho\rho}^2 v_c^s(\rho) > 0$ the rarefaction waves go right in the (ρ, q) or $(\rho, \rho\, v)$ plane. When $v_c^s(\cdot)$ is such that $2\,\partial_\rho v_c^s(\rho) + \rho\,\partial_{\rho\rho}^2 v_c^s(\rho) = 0$ the heading of rarefaction waves changes with the sign of q (it is the case for the original phase transition model), and in this case the 1-curves are LD for $q = 0$.

This condition consists in ensuring that expression (183) is either identically zero (LD curve), or has no more than one zero and is an increasing function of the density.

Remark 46. One may note that condition 44 on the strict hyperbolicity of the system is satisfied whenever condition 43 on the positivity of speed is satisfied. Indeed equation (182) can be re-written as $\forall(\rho, q) \in \Omega_c \quad \rho\,\partial_\rho v_c^s(\rho) + q\partial_\rho Q_c^s(\rho) \neq 0$, which since the first term is negative,

is equivalent to $\forall(\rho,q)\in\Omega_c \quad \rho\,\partial_\rho v_c^s(\rho)+q\partial_\rho Q_c^s(\rho)<0$. For non-zero values of $\partial_\rho Q_c^s(\rho)$, it yields $q>-\rho\,\partial_\rho v_c^s(\rho)/\partial_\rho Q_c^s(\rho)=-1+v_c^s(\rho)/\partial_\rho Q_c^s(\rho)$ which is always satisfied when $q_-\geq -1$, because the second term of the right hand side is negative.

Remark 47. In this model, traffic is anisotropic in the sense that no wave travels faster than vehicles ($\lambda_1(\rho,q)<\lambda_2(\rho,q)=v_c(\rho,q)$). The speed of vehicles is always positive and they stop only at maximal density.

23.3.3. Definition of parameters

Several parameters are used in the proposed model, which we summarize below:

1. The free-flow speed V.
2. The maximal density R.
3. The critical density σ at standard state.
4. The critical density for the lower bound of the diagram σ_-.
5. The critical density for the upper bound of the diagram σ_+.

These parameters can be identified from experimental data, and enable the definition of the parameters q_- and q_+. Figure 3.2 graphically summarizes the definition of the parameters chosen. One must note that the constraints on q_-,q_+ detailed in (181)-(182)-(183) translate into constraints on σ_-,σ_+, which cannot be freely chosen.

23.3.4. Cauchy problem

In this section we define a solution to the Cauchy problem for the system (178). Following [112], we use a definition derived from [80].

Definition 48. Given T in $\mathbb{R}_+$, u_0 in $L^1(\mathbb{R};\Omega_f\cup\Omega_c)\cap BV(\mathbb{R};\Omega_f\cup\Omega_c)$, an admissible solution to the corresponding Cauchy problem for (178) is a function $u(\cdot,\cdot)$ in $L^1([0,T]\times\mathbb{R};\Omega_f\cup\Omega_c)\cap BV([0,T]\times\mathbb{R};\Omega_f\cup\Omega_c)$ such that the following holds.

1. For all t in $[0,T]$, $t\mapsto u(t,.)$ is Lipshitz continuous with respect to the L^1 norm.

2. For all functions φ in $C_c^1([0,T]\times\mathbb{R}\mapsto\mathbb{R})$ with compact support contained in $u^{-1}(\Omega_f)$:

$$\int_0^T\int_{\mathbb{R}}\big(u(t,x)\,\partial_t\varphi(t,x)+Q_f(u(t,x))\partial_x\varphi(t,x)\big)\,dxdt+\int_{\mathbb{R}}u_0\,(x)\,\varphi(0,x)dx=0.$$

3. For all functions φ in $C_c^1([0,T]\times\mathbb{R}\mapsto\mathbb{R}^2)$ with compact support contained in $u^{-1}(\Omega_c)$:

$$\int_0^T \int_{\mathbb{R}} (u(t,x)\,\partial_t\varphi(t,x) + Q_c(u(t,x))\partial_x\varphi(t,x))\,dxdt + \int_{\mathbb{R}} u_0\,(x)\,\varphi(0,x)dx = 0.$$

4. The set of points (t,x) for which there is a change of phase is the union of a finite number of Lipschitz curves $p_i\colon [0,T) \mapsto \mathbb{R}$ such that if $\exists i \neq j$ and $\exists \tau \in [0,T]$ such that $p_i(\tau) = p_j(\tau)$ then$\forall t \in [\tau,T]$ we have $p_i(t) = p_j(t)$.
5. For all points (t,x) where there is a change of phase, let $\Lambda = \dot{p}_\iota(t^+)$, and introducing the left and right flow at (t,x):

$$F^l = \begin{cases} \rho(t,x^-)\,V & if\ \rho(t,x^-) \in \Omega_f \\ \rho(t,x^-)\,v_c\big(\rho(t,x^-), q(t,x^-)\big) & if\ \rho(t,x^-) \in \Omega_c \end{cases}$$

$$F^r = \begin{cases} \rho(t,x^+)\,V & if\ \rho(t,x^+) \in \Omega_f \\ \rho(t,x^+)\,v_c(\rho(t,x^+), q(t,x^+)) & if\ \rho(t,x^+) \in \Omega_c \end{cases}$$

the following relation must be satisfied:

$$\Lambda \cdot (\rho(t,x_+) - \rho(t,x_-)) = F_r - F_l. \tag{184}$$

Remark 49. This definition matches the standard Lax entropy solution for an initial condition with values in Ω_f or Ω_c. Equation (184) is a Rankine-Hugoniot relation needed to ensure mass conservation at the phase transition.

Theorem 50. Let Ω_f and Ω_c be defined by (180), $v_c(\cdot,\cdot)$ be defined by (177). If condition 44 is satisfied, then for all $u_0 \in L^1(\mathbb{R};\Omega_f \cup \Omega_c) \cap BV(\mathbb{R}; ega_f \cup \Omega_c)$ the corresponding Cauchy problem for (178) has an admissible solution, (see definition 48) $u(\cdot,\cdot)$ such that $u(t,\cdot) \in BV(\mathbb{R};\Omega_f \cup \Omega_c)$ for all $t \in [0,T]$.

Proof. A solution is constructed through a standard wavefront tracking procedure by iteratively gluing together the solution to Riemann problems corresponding to piecewise constant approximations of the solution. Measuring total variation along the trajectories of these solutions allows to conclude on the convergence of the sequence of successive approximations. The interested reader is referred to [80] for more details on wavefront tracking techniques and to [112,113] for more insights on proofs of existence for systems of conservation laws with phase transition. □

23.3.5. Model properties

The main differences between the original Colombo model [112] and the class of models introduced in this chapter result from the following design choices:

Choice of $q^* = 0$.

This is a change of variable which has several consequences. Related computations are more

readable. The congested standard state is $q = 0$. According to (177), the meaning of the perturbation q is also more intuitive. Positive values of q correspond to elements of flow moving at a greater speed than the standard speed for this density, and negative values of q correspond to slower elements of flow. In the traffic context, this can be understood as groups of driver characterized by their degree of agressivity, q, which leads them to drive faster or slower than the standard driver.

Newell-Daganzo flux function in free-flow.

This allows the free-flow and congested domain of the fundamental diagram proposed in the present work to be connected and to define a metastable phase, as illustrated in Figure 3.3. This yields a well-posed Riemann problem which can be solved in a simple way (see Remark 2 of [112]). Moreover, the derived models need less parameters and thus are easier to calibrate. Finally, it is consistent with the fact that a gap between phases is not observed in experimental data, see Figure 3.1.

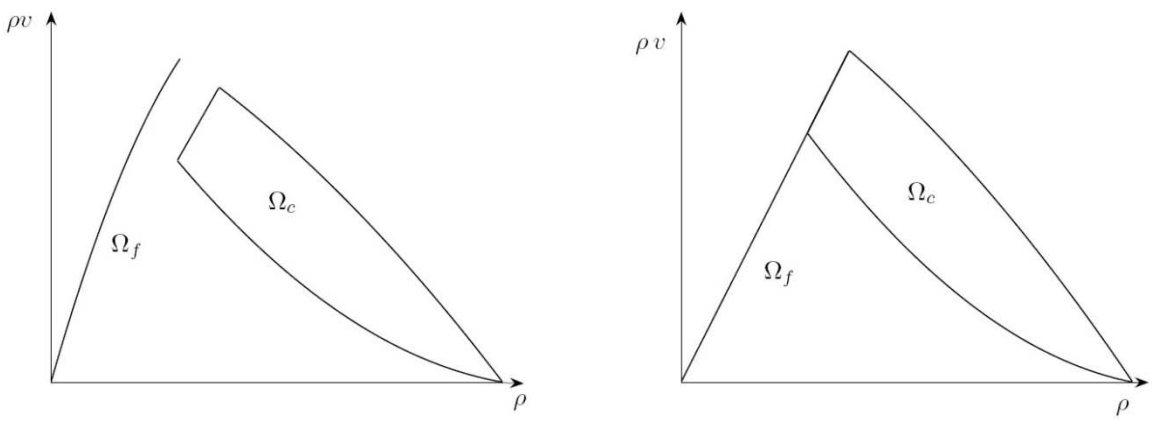


Figure 3.3. Different free-flow phases. Left: Fundamental diagram from the original Colombo phase transition model. Right: Fundamental diagram of the derived model in the particular case of a Newell-Daganzo standard state flux in the congestion phase.

The expression of the function v_c is not fully specified.

This allows us to customize the model depending on the features observed in practice. As explained in Remark 51 below, the concavity of the 1-Lax curves is related to driving behavior. In the class of models we introduce, since $v_c(\cdot,\cdot)$ is not fully specified, in the limit of conditions 43-44-45, it is possible to define the perturbed phase transition model which corresponds to the observed driver agressivity.

Remark 51. A physical interpretation can be given to the concavity of the flux function. In congestion, when the density increases toward the maximal density, the velocity decreases toward zero. This yields a decreasing slope of the flux function in congestion. The way in which drivers velocity decreases impacts the concavity of the flux, as per the expression of the second derivative of the standard flux function, $d^2 Q_c^s(\rho)/d\rho^2 = \rho\, d^2 v_c^s(\rho)/d\rho^2 + 2\, dv_c^s(\rho)/d\rho$.

1. If for a given density increase, the drivers reduce their speeds more at high densities than at low densities (modeling aggressive drivers who wait until high density to reduce speed), then the velocity function is concave and the flux function is concave.
2. If the drivers reduce their speeds less at high densities than at low densities (modeling careful drivers who anticipate and reduce their speed early), then the velocity function is convex, and the flux function may be convex.
3. An affine flux is given by a velocity function which satisfies $\rho\, d^2 v_c^s(\rho)/d\rho^2 + 2\, d v_c^s(\rho)/d\rho = 0$.

23.3.6. Numerics

Because of the non-convexity of the domain $\Omega_f \cup \Omega_c$ (illustrated in Figure 3.2), using the classical Godunov scheme [224] is not feasible due to the projection step of the scheme. We propose to use a modified version of the scheme (see [91]) which mimics the two steps of the classical Godunov scheme and adds a final sampling step.

1. The Riemann problems are solved on a regular time space mesh. When two space-consecutive cells do not belong to the same phase, the position of the phase transition at the next time step is computed.
2. The solutions are averaged on the domains defined by the position of the phase transitions arising from Riemann problems at neighboring cells (Figure 3.4).
3. A sampling method is used to determine the value of the solution in each cell of the regular mesh.

This process answers the issues of the classical Godunov scheme with non-convex domains. Numerical results have shown that it gives accurate results on benchmark tests (we refer to [91] for more details on the test cases used).

Let us note Δt the time discretization and Δx the space discretization satisfying the *Courant-Friedrichs-Lewy* (CFL) condition [224]. We call $x_j = j\,\Delta x$ for $j \in \mathbb{Z}$ and $t_n = n\,\Delta t$ for $n \in \mathbb{N}$. We call $x_{j-1/2} = x_j - \Delta x/2$ and we define a cell $C_j^n = \{t_n\} \times [x_{j-1/2}, x_{j+1/2}[$ which has a length Δx. We call u_j^n the value of $u := (\rho, q)$ at (t_n, x_j), and, by extension, in C_j^n. The speed of the phase transition between each pair of cells (C_j^n, C_{j+1}^n) is noted $\nu_{j+1/2}^n$ ($\nu_{j+1/2}^n$ equals zero if u_j^n and u_{j+1}^n belongs to the same phase). If we call $\overline{x}_{j-1/2}^{n+1} = x_{j-1/2} + \nu_{j-1/2}^n\,\Delta t$ we can define cell $\overline{C}_j^{n+1}$ as $\overline{C}_j^{n+1} = \{t^{n+1}\} \times [\overline{x}_{j-1/2}^{n+1}, \overline{x}_{j+1/2}^{n+1}[$ which has a length $\Delta\overline{x}_j^n = \overline{x}_{j+1/2}^{n+1} - \bar{x}_{j-1/2}^{n+1}$, as shown in Figure 3.4.

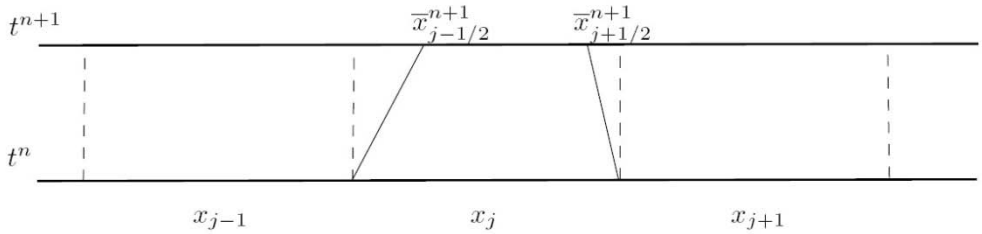


Figure 3.4. Phase transitions enter cell C_j^n from both sides.

The solution to the Riemann problem between cells C_j^n is averaged on cells $\overline{C}_j^{n+1}$, which by construction enclose states which are either free-flowing or congested, according to the modified Godunov scheme. We define:

1. $u_R(v_{j-1/2}^{n,+}, u_{j-1}^n, u_j^n)$ as the solution to the Riemann problem between u_{j-1}^n and u_j^n, at $\frac{x-x_{j-1/2}}{t-t^n} = v_{j-1/2}^n$, and calculated at the right of the cell boundary.
2. $g\left(v_{j+1/2}^{n,-}, u_j^n, u_{j+1}^n\right) := f\left(u_R(v_{j+1/2}^{n,-}, u_j^n, u_{j+1}^n)\right)$ with $f(\rho, q) = (\rho\, v, q\, v)$ and the definition of v from (179), as the numerical flux between cells C_j^n and C_{j+1}^n, at $\frac{x-x_{j+1/2}}{t-t^n} = v_{j+1/2}^n$, and calculated at the left of the cell boundary.

The averaging step of the modified Godunov scheme reads:

$$\Delta\overline{x}_j^n\, \overline{u}_j^{n+1} = \Delta x\, u_j^n - \Delta t\, \left(g\left(v_{j+1/2}^{n,-}, u_j^n, u_{j+1}^n\right) - v_{j+1/2}^n\, u_R\left(v_{j+1/2}^{n,-}, u_j^n, u_{j+1}^n\right)\right)$$
$$+\Delta t\, \left(g\left(v_{j-1/2}^{n,+}, u_{j-1}^n, u_j^n\right) - v_{j-1/2}^n\, u_R\left(v_{j-1/2}^{n,+}, u_{j-1}^n, u_j^n\right)\right).$$

One can notice that when there is no phase transition, $v_{j-1/2}^n = v_{j+1/2}^n = 0$, $\Delta x = \Delta\overline{x}_j^n$ and we obtain the classical Godunov scheme. The last step is the sampling phase to define the solutions on the cells C_j^{n+1}. Following [91], for cell C_j^{n+1} we randomly pick a value between $\overline{u}_{j-1}^{n+1}$, $\overline{u}_j^{n+1}$ and $\overline{u}_{j+1}^{n+1}$ according to their rate of presence in cell C_j^{n+1}. This is done using the Van der Corput sequence $(a_n)_{n\in\mathbb{N}}$ (185) which is a low-discrepancy sequence in the interval [0,1]:

$$u_j^{n+1} = \begin{cases} \overline{u}_{j-1}^{n+1} & if\ a_n \in]0, \max\left(\frac{\Delta t}{\Delta\overline{x}_j^n}\, v_{j-1/2}^n, 0\right)] \\ \overline{u}_j^{n+1} & if\ a_n \in]\max\left(\frac{\Delta t}{\Delta\overline{x}_j^n}\, v_{j-1/2}^n, 0\right), 1 + \min\left(\frac{\Delta t}{\Delta\overline{x}_j^n}\, v_{j+1/2}^n, 0\right)[\\ \overline{u}_{j+1}^{n+1} & if\ a_n \in [1 + \min\left(\frac{\Delta t}{\Delta\overline{x}_j^n}\, v_{j+1/2}^n, 0\right), 1[\end{cases} \tag{185}$$

Remark 52. In the general case the congested domain Ω_c is not convex in (ρ, q) coordinates due to the convexity of the metastable border of the domain as illustrated on Figure 3.2. It is therefore needed to add a projection step as a fourth step to the modified Godunov scheme. The projection (ρ_p, q_p) of state (ρ, q) is defined as the solution in the metastable phase of the system:

$$\begin{cases} \dfrac{q_p}{\rho_p} = \dfrac{q}{\rho} \\ v_c(\rho_p, q_p) = V \end{cases}$$

The error metric chosen to assess the numerical accuracy of the scheme is the $C^0(\mathbb{R}, L^1(\mathbb{R}, \mathbb{R}^2))$ relative error between the computed solution and the analytical solution. We call u and u_c the exact and computed solutions respectively. For the computational domain $[x_0, x_1]$, the error at T is computed as follows:

$$E(T) = \frac{sup_{t\in[0,T]} \int_{x_0}^{x_1} \| u(t,x) - u_c(t,x) \|_1 \, dx}{sup_{t\in[0,T]} \int_{x_0}^{x_1} \| u(t,x) \|_1 \, dx}.$$

23.4. The Newell-Daganzo phase transition model

In this section, we use a Newell-Daganzo velocity function for congestion, i.e. a velocity function for which the flux is affine with respect to the density. We instantiate the corresponding phase transition model for this flux function and derive a corresponding Riemann solver, which we implement and test on a benchmark case.

23.4.1. Analysis

We propose to use the following standard velocity function:

$$v_c^s(\rho) = \frac{V\,\sigma}{R-\sigma}\left(\frac{R}{\rho} - 1\right),$$

which is clearly the unique function yielding an affine flux, and satisfying the requirements from Section § 23.3.1, on the vanishing point, trend, continuity and concavity property of the standard flux.

For a perturbed state, the velocity function reads:

$$\begin{cases} v_f(\rho) = V & \text{for } (\rho, q) \in \Omega_f \\ v_c(\rho, q) = \dfrac{V\,\sigma}{R-\sigma}\,(\dfrac{R}{\rho} - 1)\,(1+q) & \text{for } (\rho, q) \in \Omega_c \end{cases} \tag{186}$$

where Ω_f and Ω_c are defined by (180). The corresponding fundamental diagram is shown in Figure 3.2. The standard flux is affine with the density, but the 1-Lax curves outside the standard state are either convex or concave in $(\rho, \rho\, v)$ coordinates depending on the sign of the perturbation.

Remark 53. Note that the expression of the velocity in Figure 3.2 is given by (186), depends on the phase, and is therefore set-valued for $\rho > \sigma_-$ which is the lowest density value at which congestion can arise.

The conditions from Section § 23.3.2 to have positive speed and strict hyperbolicity of the congested part of the system (178) reduce to:

$$q_- > -1.$$

23.4.2. Solution to the Riemann problem

Following [112], we construct the solution to the Riemann problem for the system (178) with the velocity function defined by (186) and the initial datum:

$$(\rho, q)(0, x) = \begin{cases} (\rho_l, q_l) & \text{if } x < 0 \\ (\rho_r, q_r) & \text{if } x > 0. \end{cases}$$

We note u the vector (ρ, q). We define u_m by the solution in Ω_c of the system:

$$\begin{cases} \dfrac{q_m}{\rho_m} = \dfrac{q_l}{\rho_l} \\ v_c(u_m) = v_c(u_r) \end{cases} \tag{187}$$

which yields a quadratic polynomial in ρ_m. We address the general case where the solution u_m of system (187) can coincide with u_l or u_r.

Case 1: $u_l \in \Omega_f$ and $u_r \in \Omega_f$

For all values of (ρ_l, ρ_r) the solution consists of a contact discontinuity from u_l to u_r.

Case2: $u_l \in \Omega_c$ and $u_r \in \Omega_c$

(a) If $q_l > 0$ and $v_c(u_r) \geq v_c(u_l)$ the solution consists of a 1-rarefaction wave from u_l to u_m and a 2-contact discontinuity from u_m to u_r.
(b) If $q_l > 0$ and $v_c(u_l) > v_c(u_r)$ the solution consists of a shock wave from u_l to u_m and a 2-contact discontinuity from u_m to u_r.
(c) If $q_l = 0$ the solution consists of a 1-contact discontinuity from u_l to u_m and a 2-contact discontinuity from u_m to u_r.

(d) If $0 > q_l$ and $v_c(u_r) > v_c(u_l)$ the solution consists of a shock wave from u_l to u_m and a 2-contact discontinuity from u_m to u_r.
(e) If $0 > q_l$ and $v_c(u_l) \geq v_c(u_r)$ the solution consists of a 1-rarefaction wave from u_l to u_m and a 2-contact discontinuity from u_m to u_r.

Case 3: $u_l \in \Omega_c$ and $u_r \in \Omega_f$

(a) If $0 > q_l$ the solution consists of a shock wave from u_l to u_m and of a contact-discontinuity from u_m to u_r.
(b) If $q_l = 0$ the solution consists of a 1-contact discontinuity from u_l to u_m and of a contact-discontinuity from u_m to u_r.
(c) If $q_l > 0$ the solution consists of a 1-rarefaction wave from u_l to u_m and of a contact-discontinuity from u_m to u_r.

Case 4: $u_l \in \Omega_f$ and $u_r \in \Omega_c$ Let u_{m-} be defined by the solution in Ω_c of the system:

$$\begin{cases} \dfrac{q_{m-}}{\rho_{m-}} = \dfrac{q_-}{R} \\ v_c(u_{m-}) = v_c(u_r) \end{cases}$$

and let $\Lambda(u_l, u_{m-})$ be the Rankine-Hugoniot phase transition speed between u_l and u_{m-} defined by equation (184).

(a) If $\Lambda(u_l, u_{m-}) \geq \lambda_1(u_{m-})$ the solution consists of a phase transition from u_l to u_{m-} and of a 2-contact discontinuity from u_{m-} to u_r.
(b) If $\Lambda(u_l, u_{m-}) < \lambda_1(u_{m-})$ let u_p be defined by the solution in Ω_c of the system:

$$\begin{cases} \dfrac{q_p}{\rho_p} = \dfrac{q_-}{R} \\ \Lambda(u_l, u_p) = \lambda_1(u_p). \end{cases}$$

The solution consists of a phase transition from u_l to u_p, of a 1-rarefaction wave from u_p to u_{m-}, and of a 2-contact discontinuity from u_{m-} to u_r.

23.4.3. Model properties

The properties of the Newell-Daganzo model can be abstracted from the definition of the Riemann solver in previous Section.

The nature of the Lax curves in congestion is the same for the original Colombo model and the Newell-Daganzo phase transition model (see Figure 3.3). Thus the solution for each model differs only when a free-flow state is involved. Three differences appear in that case:

1. For a given density corresponding to the free-flow phase, the associated velocity

differs in general between the two models.

2. Within the free-flow phase, only contact discontinuity can arise in the Newell Daganzo phase transition model, whereas rarefaction waves and shockwaves can arise in the original Colombo model.

3. A transition from congestion to free-flow always involves a shock-like phase transition in the Colombo model (and thus the solution is composed of three waves in general), whereas the transition occurs through a metastable state in the Newell-Daganzo phase transition model, and involves only a "congestion to metastable" wave and a "metastable to free-flow" wave.

These properties are illustrated in the next Section on a Riemann problem.

As in the original Colombo phase transition model [112], the 1-Lax curves are LD for $q = 0$, and the direction of the rarefaction waves changes according to the sign of q. This yields interesting modeling capabilities, but requires the Riemann solver to be more complex than the one described in the following Section.

Remark 54. As illustrated on Figure 3.2 the flux is linear in congestion at the standard state as per the Newell-Daganzo flux function. Remark 51 states that this shape models neutral drivers (aggressivity-wise). When the traffic is above the standard state, meaning that the velocity is higher that what it is for the same density at the standard state, the 1-Lax curves are concave in $(\rho, \rho\, v)$ coordinates, meaning that the drivers are more aggressive. So such a fundamental diagram shape seems to be in accordance with the intuition, that for a given density, the most aggressive drivers tend to have a greater speed. This is symmetrically true for less aggressive drivers, also accounted for by this model.

23.4.4. Benchmark test

In this section we compare the numerical solution given by the modified Godunov scheme with the analytical solution to a Riemann problem. We use the phase transition model (178) in the Newell-Daganzo case (186) with the following choice of parameters: $V = 45$, $R = 1000$, $\sigma_- = 190$, $\sigma = 220$, $\sigma_+ = 270$. The benchmark test is a phase transition from congestion to free-flow with the following left and right states:

1. $u_l = (800, -0.1)$ which corresponds to congestion below standard state with $\rho = 800$ and $v = 2.9$.
2. $u_r = (100)$ which corresponds to a free-flow state with $\rho = 100$ and $v = 45$.

This configuration gives rise to a shock wave between u_l and a congested state u_m followed by a contact discontinuity between u_m and u_r (Riemann case 3, first subcase), as shown in Figure 4.1.

We also present the solution given by the original Colombo model with the following parameters: $V_{c+} = 45$, $V_{f-} = 57$, $V = 67$, $q^* = 0$, $Q^- = -0.88$ and $Q^+ = 1.15$. The congested phases in the two models are identical with this choice of parameters. One may note

that because the fundamental diagram in free-flow differs between the original Colombo model and the Newell-Daganzo phase transition model (see Figure 3.3), the speed corresponding to the right initial state in the Riemann problem is greater in the Colombo model.

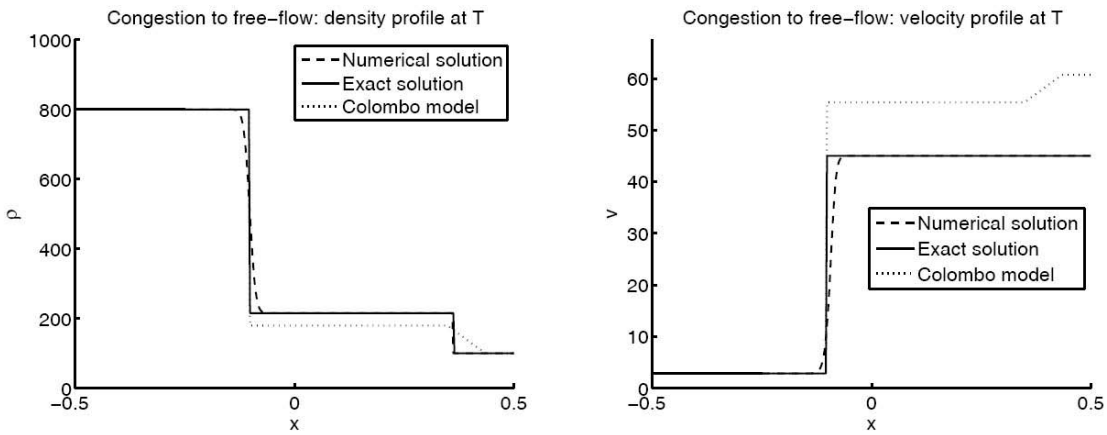


Figure 4.1. Exact solution (continuous line), computed solution (dashed line), and exact solution for the Colombo model (dotted line) for density (left) and speed (right). Between the two initial states, for the class of models presented here, appears a state $u_m = (215.4, -0.03)$ which corresponds to the intersection of the 1-Lax curve going through u_l with the metastable phase. In this graph $T = 0.4$ and $\Delta x = 0.0013$.

The solutions to the Riemann problem for each model differ on several points. First the intermediary state u_m belongs to the metastable phase in the Newell-Daganzo model whereas it belongs to the free-flow phase for the Colombo model. Second the wavefrom the intermediary state u_m to the right state u_r is a rarefaction wave in the Colombo model, as illustrated in Figure 4.1, whereas it is a contact discontinuity in the Newell-Daganzo phase transition model.

The values of the error $E(T)$, as described in Section § 23.3.6 for $T = 4$, (a typical time for which all interactions have moved out of the computational domain) are outlined in Table 37.

Cell #	$E(T)$
50	$5.8\ 10^{-04}$
100	$2.0\ 10^{-04}$
200	$6.4\ 10^{-05}$
400	$2.0\ 10^{-05}$

Table 37. Numerical error: relative error between exact solution and the modified Godunov scheme solution for the benchmark described above, for different discretizations.

23.5. The Greenshields phase transition model

In this section we use a Greenshields model to describe the velocity function in congestion, i.e. we use a concave quadratic flux function. We present the standard and perturbed flux functions, derive the corresponding Riemann solver which we test on a benchmark case, and describe the properties of the Greenshields phase transition model.

23.5.1. Analysis

We use a quadratic relation to describe the congestion standard state, which for physical considerations needs to satisfy the requirements from Section § 23.3.1. This leads us to choose the flux as a quadratic function of the form:

$$\rho\, v_c^s(\rho) = (\rho - R)\,(a\,\rho + b)$$

such that the vanishing condition at $\rho = R$ is satisfied. Continuity at the critical density σ yields:

$$b = \frac{\sigma\, V}{\sigma - R} - a\,\sigma$$

so the flux at the standard state reads:

$$\rho\, v_c^s(\rho) = (\rho - R)\left(a\,(\rho - \sigma) + \frac{\sigma\, V}{\sigma - R}\right)$$

with a variation interval for a defined by the second and third conditions of Section § 23.3.1 as:

$$a \in \left[-\frac{\sigma\, V}{(\sigma - R)^2}, 0\right[.$$

Note that for the specific case in which $R = 2\,\sigma$ and a is defined by the fact that the derivative of the flux equals zero at σ (which reads $a = -\sigma\, V/(\sigma - R)^2$), we obtain the classical Greenshields flux.

Following the general form given in system (179), we write the perturbed velocity function as:

$$\begin{cases} v_f(\rho) = V & \text{for } (\rho, q) \in \Omega_f \\ v_c(\rho, q) = \left(1 - \dfrac{R}{\rho}\right)\left(a\,(\rho - \sigma) + \dfrac{\sigma\, V}{\sigma - R}\right)(1 + q) & \text{for } (\rho, q) \in \Omega_c \end{cases} \tag{188}$$

with $a \in \left[-\frac{\sigma\, V}{(\sigma - R)^2}, 0\right[$, and where Ω_f and Ω_c are defined by (180). The corresponding fundamental diagram is presented in Figure 5.1.

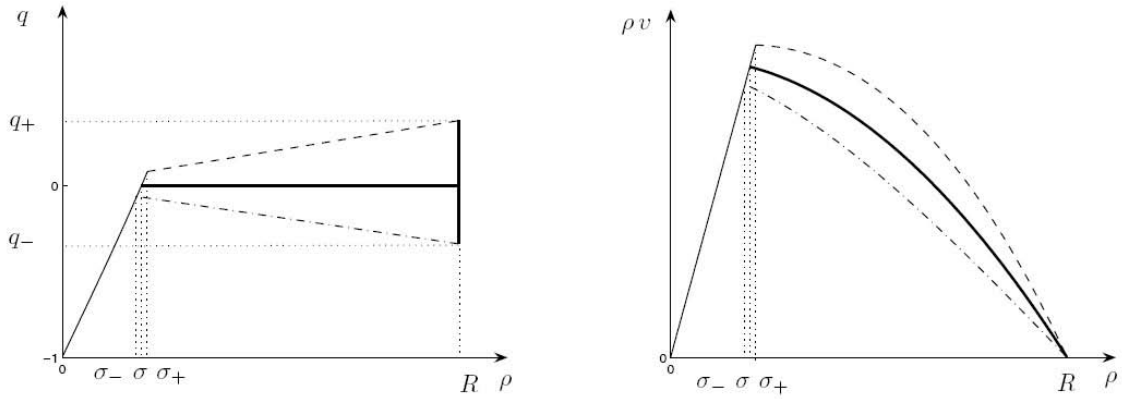


Figure 5.1. Phase transition model with a Greenshields standard state. Left: State-space coordinates. Right: Flux-density coordinates. Thin solid line: Free-flow. Bold solid line: Congestion standard state. Thin dashed line: Upper bound of congestion. Thin dot-dashed line: Lower bound of congestion. The standard flux is concave, and all the 1-Lax curves are concave in $(\boldsymbol{\rho}, \boldsymbol{\rho}\, \boldsymbol{v})$ coordinates. In $(\boldsymbol{\rho}, \boldsymbol{q})$ coordinates the free-flow phase is not a straight line but has a very light convexity.

Remark 55. The expression of the velocity function given by system (188) enables a set-valued velocity function for $\rho > \sigma_-$. For a given density the variable velocity can take several values. The lower bound of the congestion phaseis concave, unlike for the model presented in Section § 23.4. This feature may be more appropriate for usual experimental datasets.

The requirements from Section § 23.3.2 here reduce to:

$$q_- > -\frac{a\, R}{\dfrac{\sigma\, V}{\sigma - R} + a\,(2\, R - \sigma)}.$$

While in the Newell-Daganzo phase transition model the bound on the perturbation is given by the fact that the speed had to be positive, here the bound is given by the requirement on the constant concavity of the 1-Lax curves.

Remark 56. The lower bound on the perturbation is an increasing function of the parameter a, so this parameter should be chosen as small as possible to allow for more freedom, namely $a_{\min} = -\sigma\, V/(\sigma - R)^2$ which yields the lowest lower bound bound $q_-^{min} = R/(2\,\sigma - 3\, R)$.

23.5.2. Solution to the Riemann problem

We consider the Riemann problem for system (178) with the velocity function from equation (188) and the initial datum:

$$(\rho, q)(0, x) = \begin{cases} (\rho_l, q_l) & \text{if } x < 0 \\ (\rho_r, q_r) & \text{if } x > 0. \end{cases} \tag{189}$$

We follow the method used in [112] to construct the solution. We define u_m by the solution in Ω_c of the system:

$$\begin{cases} \frac{q_m}{\rho_m} = \frac{q_l}{\rho_l} \\ v_c(u_m) = v_c(u_r) \end{cases} \tag{190}$$

which yields a quadratic polynomial in ρ_m with one root in $[0, R]$. In the general case, the solution u_m of the system (190) can be equal to u_l or u_r.

Case1: $u_l \in \Omega_f$ and $u_r \in \Omega_f$ For all values of (ρ_l, ρ_r) the solution consists of a contact discontinuity from u_l to u_r.

Case 2: $u_l \in \Omega_c$ and $u_r \in \Omega_c$

(a) If $v_c(u_r) \geq v_c(u_l)$ the solution consists of a 1-rarefaction wave from u_l to u_m and a 2-contact discontinuity from u_m to u_r.
(b) If $v_c(u_l) > v_c(u_r)$ the solution consists of a shock wave from u_l to u_m and a 2-contact discontinuity from u_m to u_r.

Case 3: $u_l \in \Omega_c$ and $u_r \in \Omega_f$ The solution consists of a 1-rarefaction wave from u_l to u_m and of a contact-discontinuity from u_m to u_r.

Case 4: $u_l \in \Omega_f$ and $u_r \in \Omega_c$ Let u_{m-} be defined by the solution in Ω_c of the system:

$$\begin{cases} \frac{q_{m-}}{\rho_{m-}} = \frac{q_-}{R} \\ v_c(u_{m-}) = v_c(u_r). \end{cases}$$

The solution consists of a phase transition from u_l to u_{m-} and of a 2-contact discontinuity from u_{m-} to u_r.

Remark 57. The analysis in the case of a convex standard flux function, which we do not address here, is closely related to this case, modulo the sign of the parameter a and the concavity of the 1-Lax curves.

23.5.3. Model properties

The structure of the solution to the Riemann problem presented in previous section explains the distinction with the original phase transition model:

1. Since the 1-Lax curves are concave, within the congestion phase, shock waves occur only from a low density on the left to a high density on the right. This is similar to classical

traffic models with concave flux.

2. The concavity of the 1-Lax curves yields simple transitions from a free-flow state to a congested state. These phase transitions are composed of a shock-like phase transition followed by a contact discontinuity, whereas a rarefaction wave can appear between the two in the original phase transition model or in the Newell-Daganzo phase transition model.
3. Similarly to the Newell-Daganzo phase transition model, within the free-flow phase, the Greenshield phase transition model exhibits only contact discontinuities.

Another consequence of the fact that the 1-Lax curves are concave is that the Riemann solver is much simpler than in the Newell-Daganzo case, with only five different types of solutions, compared to the Newell-Daganzo case which has eleven different types of solutions.

Remark 58. According to Remark 51 this flux function models aggressive drivers only, who drive along concave 1-Lax curves. In practice, it is able to model a class of clouds of points observed experimentally where the congested domain has a concave lower border in $(\rho, \rho\, v)$ coordinates.

23.5.4. Benchmark test

In this section we compare the numerical results given by the modified Godunov scheme on a benchmark test with its analytical solution. We use the phase transition model (178) in the Greenshields case (188) with the following choice of parameters: $V = 45$, $R = 1000$, $\sigma_- = 190$, $\sigma = 200$, $\sigma_+ = 215$. We choose $a = -0.01$. The resulting values for the extrema of the perturbation are $q_- = -0.34$ and $q_+ = 0.44$. The benchmark test is a phase transition from free-flow tocongestion, with the following left and right states:

1. $u_l = (180)$ which corresponds to a free-flow state with $\rho = 180$ and $v = 45$.
2. $u_r = (900{,}0.2)$ which corresponds to a congested situation above standard state with $\rho = 900$ and $v = 2.4$.

This configuration gives rise to a phase transition between u_l and a congested state u_m followed by a 2-contact discontinuity between u_m and u_r (Riemann case 4) which is illustrated in Figure 5.2.

We also present the solution to the Riemann problem for the original Colombo model with parameters: $V_{c+} = 45$, $V_{f-} = 57$, $V = 67$, $q^* = 0$, $Q^- = -0.32$ and $Q^+ = 0.44$. The speed in free-flow differs between the two models. The phase transition speed is negative for both models but is greater in the case of the Greenshields phase transition model which models more aggressive drivers which have a higher flux in congestion for the same density value. The second wave has the same speed in the two models.

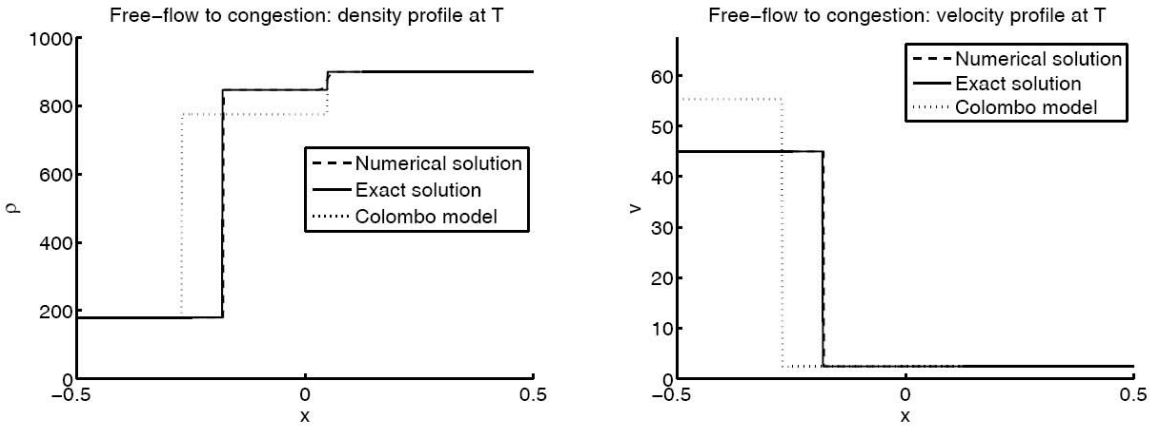


Figure 5.2. Exact solution (continuous line), computed solution (dashed line), and solution to the Colombo model (dotted line) for density (left) and speed (right). Between the two initial states appears a state $u_m = (847.4, -0.24)$ which corresponds to theintersection of the lower bound of the diagram in congestion with the 2-Lax curve going through u_r. In this graph $T = 1$ and $\Delta x = 0.0013$.

Table 38 summarizes the values of the error $E(T)$, as defined in Section § 52, for different sizes of the discretization step, at $T = 4$.

Cell #	$E(T)$
50	$3.1\ 10^{-04}$
100	$7.8\ 10^{-05}$
200	$2.1\ 10^{-05}$
400	$5.4\ 10^{-06}$

Table 38. Relative error between exact solution and numerical solution for the test case explicitly described above, for different numbers of space cells.

23.6. Conclusion

This chapter reviewed the fundamental features of the Colombo phase transition model and proposed to build upon it a class of models in which the fundamental diagram is set-valued in the congested regime. The notion of standard state which provides the basis for the construction of the 2×2 phase transition models was introduced. General conditions which enable the extension of the original Colombo phase transition model to this new class of 2×2 phase transition models were investigated. A modified Godunov scheme which can be applied to models with non-convex state-space was used to solve these equations numerically. The model was instantiated for two specific flux functions, which include the Newell-Daganzo flux function (affine) and the Greenshields flux function (quadratic concave). A discussion of the choice of parameters needed for each of the models was conducted. The solution to the Riemann problem

was derived, and a validation of the numerical results using benchmark tests was conducted. Open questions for this model include the capability of the model to accurately reproduce traffic features experimentally measured on highways. Experimental validations of the model should reveal its capabilities of reproducing traffic flow more accurately than existing models. In addition, the specific potential of the model to integrate velocity measurements (through proper treatment of the second state variable of the problem) is a significant advantage of this model over any first order model for which the density-flux relation is single valued. The proper use of this key feature for data assimilation is also an open problem, which could have very promising outcomes for highway traffic state estimation.

Part VI

Framework and Applications for Hamilton-Jacobi PDEs

24. Hamilton-Jacobi PDEs: A new framework

The integration of mobile data into traffic flow models posed significant challenges that required the development of a new modeling framework capable of integrating mobile measurements (from phones). The following chapters thus present a new computational method for solving the Hamilton Jacobi (HJ) partial differential equation (PDE), as well as a new convex optimization-based estimation framework based on this computational method. This new framework is motivated by the many challenges brought by the proliferation mobile data. For example, mathematical models in general cannot handle data that is prescribed at moving locations. As a result, new kinds of models need to be defined in which to handle mobile data. One proper class of models is one in which the 'labels' of vehicles are incorporated into the formulation, and the model is thus capable of 'following the trajectories' of vehicles. This class of models can be developed using viscosity theory and viability methods. Such models can be used to integrate loop detector data, as well as any other trajectory based or mobile probe data (but not travel time based data), into the flow estimation. As explained in the next series of chapters, these models require specific mathematical treatment that once completed enable a number of problems to be solved using standard tools in viability theory and optimal control.

Three main application areas for this framework have been selected to illustrate its utility. The first application is that of finding bounds on traffic values of interest such as initial number of vehicles on a road, or vehicle travel times. A second application involves consistency problems such as detecting the subtle errors caused by incorrectly mapped loop detectors. This consistency issue may become increasingly important in the future for the purpose of detecting spoofing attacks on traffic information systems that rely on probe data. For example, one can imagine cases in which malicious data introduced to the system could induce the estimation process to predict a traffic jam when in fact the reality is in free flow. The third application concerns privacy analysis, and determining when a vehicle could be re-identified based on the available data.

24.1. Partial differential equation models of large scale infrastructure systems

Large scale infrastructure systems, such as transportation networks, networked water channels,

or air transportation networks are *distributed parameter systems*, that is, their state is usually described by a function of space and time, in contrast to a finite dimensional vector. Another way to think about this would be as an "infinite dimensional" vector. A common mathematical tool for modeling such systems is *partial differential equations* (PDEs). They provide an efficient way of representing physical phenomena in a mathematically compact manner, which integrates the distributed features of the systems of interest [139].

Among PDEs, a specific class stands out, *conservation laws* [224,80], which model phenomena in which a balance equation governs the physics (for example mass balance, momentum balance, charge balance, etc.). Water channels for instance can be modeled using the *Saint-Venant PDE* [230], obtained from the conservation of water mass and momentum. Examples of applications of such models can be found in [115,277,231,115]. Ground [226,285] and air transportation networks [310] can both be modeled by the *Lighthill-Whitham-Richards* (LWR) PDE, which is based on the conservation of vehicles. Alternatively, traffic flow can also be modeled using *second order models* [112,62,266,76,351], which are non-scalar conservation laws. All these PDEs and others used to describe distributed parameter systems are not necessarily conservation laws however. In structural engineering for instance, beam deformation can be modeled by the *Euler-Bernoulli beam* PDE [213], which is not a conservation law. In electrical engineering, the *Telegraph equation* [148] can be used to model wave propagation in telecommunication lines, and is also not a conservation law.

24.2. Control and estimation of partial differential equations

24.2.1. Filtering based methods

State estimation and control for PDE-based systems is more complex than for their *ordinary differential equation* (ODE) based counterparts, because of the distributed nature of the state.

The tools available for estimating [121] and controlling [213] the states of an ODE can be extended to systems modeled by a PDE, for instance using variations of *Kalman Filtering* (KF), originally derived for systems modeled by linear ODEs [67]. *Extended Kalman Filtering* (EKF) [49] is a modification of Kalman filtering for nonlinear systems. EKF techniques have been applied to water channels state estimation problems in [141], and in traffic flow estimation problems in [331,49] for instance.

The EKF can however perform poorly for specific nonlinear systems, for which *Monte Carlo* techniques are a possible alternative. For example, when the dynamics exhibits nonsmoothness or nondifferentiability, EKF is known to have problems [339]. Monte Carlo methods involve estimating the current probable value of the state, computing the state evolution, and comparing it against new measurement data to obtain a current estimate. By their nature, Monte-Carlo based methods can apply to any model, albeit with some computational cost penalty. *Ensemble Kalman Filtering* (EnKF) [143] is a Monte-Carlo based method that can be used for systems modeled by nonlinear PDEs, for instance the LWR [339] PDE, without approximating the model around the current estimate as done in EKF. Other examples of application of EnKF include Shallow Water Equations [322], or meteorology [196]. The EnKF samples the possiblecurrent states of the system according to a probability distribution, computes the evolution of these samples, and

combine these evolutions with new measurements to obtain the best estimate of the state. The *Mobile Millennium* system [14] is an example of operational implementation of the EnKF for traffic flow modeling using the LWR PDE. More generally, the state of distributed parameter systems can be estimated using *Particle Filtering* (PF), which can be used for general nonlinear systems, albeit with a higher computational cost [98].

24.2.2. Other methods

Backstepping methods [213] are control design methods that can be applied to some classes of nonlinear systems. They involve designing a controller for a known-stable system and "back out" new controllers that progressively stabilize each outer supersystem.

The theory of *differential flatness*, which was originally developed in [147], consists in a parametrization of the trajectories of a system by one of its outputs, called the "flat output" [260,46]. It can be used to control the state of water channels [278] for instance.

Lyapunov methods [250] are based the extension of the Lyapunov theory for ODE-based systems to the PDE case. Similarly to ODE-based systems, they involve the use of a *Lyapunov function* associated with the state of the system, and which is either bounded or decreasing.

Machine learning methods [198] in contrast rely on experimental datasets to learn how the state evolves. One of the main focuses of machine learning methods is to automatically learn to recognize specific patterns using statistical methods [47]. Machine learning methods can be applied to very different problems, including estimation problems [183] on systems modeled by PDEs.

Finally, spectral methods [343,100] use modal decomposition techniques to transform dynamic constraints into static constraints in the frequency domain, and subsequently obtain a static inverse modeling problem, which is easier to solve.

One of the major difficulties arising when dealing with sensing problems on systems modeled by PDEs is the integration of the model constraints into the estimation problem. The PDEs investigated here are nonlinear. Their solutions can be nonsmooth and even discontinuous, which makes the model constraints difficult to derive. One of the contributions of this work is to express the model constraints as convex inequalities, which are both explicit and computationally tractable.

24.3. Hamilton-Jacobi equations

In one dimensional systems (for example to model the highway network), hyperbolic scalar conservation laws have a direct counterpart in *Hamilton-Jacobi* (HJ) theory [139], which is the subject of the present work. HJ PDEs [68] have a particular importance in optimal control, and more generally in variational problems, for which they were originally derived.

Because of their structure, the solutions to a given HJ PDE satisfy the HJ PDE in a generalized sense, and are thus called *weak solutions*. Several classes of weak solutions to HJ PDEs exist.

Historically, *viscosity solutions* [117,116] were the first class of weak solutions identified for HJ PDEs. They were initially discovered by taking the limit of the solutions to a modified HJ PDE in which a viscosity term is added, when the value of this term converges to zero, leading to the term of "vanishing viscosity", initially used to describe them. Viscosity solutions are continuous, but not necessarily differentiable everywhere. *Barron-Jensen/Frankowska* (BJ-F) solutions [70,152] generalize the concept of viscosity solutions by allowing the solution to be discontinuous. A third concept of solutions is sometimes used, so called "nonsmooth solutions", based on nonsmooth analysis [103].

HJ PDEs also integrated the framework of *differential games* [140,86,87], which model problems containing two actors, a pursuer and an evader, with conflicting goals. They can for instance be used to solve aircraft safety problems teMBT05 by computing the set in which an evader aircraft is always safe from a pursuer aircraft that attempts to collide with it.

The solutions used in the present work are obtained using a *Lax-Hopf* [80] formula, which expresses the solution at any given point as a minimization (or maximization) problem.

24.4. Numerical analysis for Hamilton-Jacobi equations

The solutions to HJ PDEs (and their conservation laws counterparts) can be computed numerically using various methods, relying either on the structure of the PDE (finite difference schemes), the structure of their solutions (wave-front tracking methods), a different expression of the problem (level set methods), or the Lax-Hopf formula (dynamic programming, Lax-Hopf algorithm). The most basic numerical schemes that can be thought of are *finite difference schemes*, such as the Godunov scheme [166], or the Lax-Friedrichs method [210]. Finite difference methods require the approximation of the PDE as a finite difference equation on a computational grid. The finite difference equation is then solved numerically. Finite difference schemes compute approximate solutions, and are often subject to stability conditions, such as the *Courant-Friedrichs-Levy* (CFL) condition, which constrains the computational grid [224].

Level set methods [247,246] rely on finite difference schemes to numerically approximate the solution with subgrid accuracy and avoid their high cost of grid refinement. They can be extended in some cases by *fast marching methods* [296], which are computationally efficient (but have specific restrictions in their possible applications).

Wave-front tracking methods [80,122] in contrast rely on the structure of the mathematical solutions to hyperbolic conservation laws, which feature *shockwaves* and *expansion waves*. Wave-front tracking methods are event-based numerical methods that compute the location of these waves, and thus derive the expression of the solution everywhere because of its structure.

Finally, the Lax-Hopf formula used in the present work can be solved numerically to compute the solution as a minimization problem. Possible solution methods include *dynamic programming* [139,128] or the Lax-Hopf algorithm derived in Chapter 25, adapted from [59].

24.5. Scientific Contributions

The following chapters introduce a new grid-free solution procedure known as the *Lax-Hopf algorithm* for solving Hamilton-Jacobi equations and their associated scalar conservation laws. This numerical scheme exhibits two main benefits with respect to standard first-order schemes. Firstly, the solutions computed using the Lax-Hopf algorithm are exact, *i.e.* do not exhibit error aside from the error due to numerical accuracy of the numerical software used to compute them. Secondly, the solution can be computed at any time without requiring intermediate computations, unlike (first-order) finite difference schemes which have to do so because of the *Courant-Friedrich-Lewy* (CFL) conditions.

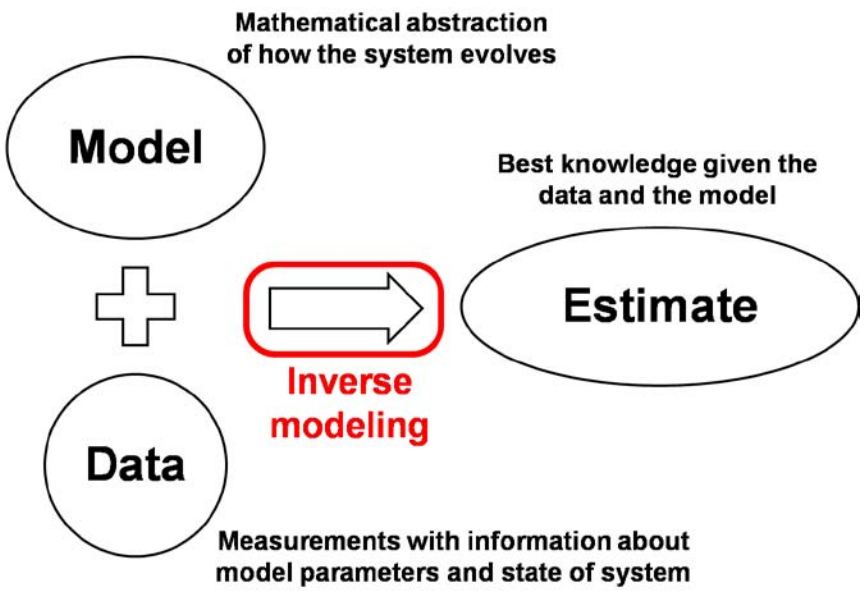


Figure 5.1. Illustration of the state estimation procedure.

A convex-optimization based framework has been constructed for computing solutions to various estimation problems on systems modeled by HJ PDEs. For this, we first establish the relationship between the physics of the problem and the value of the initial boundary or internal conditions which are required to solve the PDE. However, it is in general impossible given our measurement data to establish the value of the initial, boundary and internal conditions univocally, because of sensor errors, coefficients that cannot be measured and constants of integration that are unknown.

The measurement data constrains the possible values that the initial, boundary and internal conditions can take. Similarly the HJ PDE model also constrains the possible values that the initial, boundary and internal conditions can take. While the derivation of the data constraints is usually easy if we know how the sensors perform, deriving these model constraints is very difficult in general because of the nonlinearity of the model and the nonsmoothness of its solutions.

In our formulation, model constraints can be reduced to a set of convex inequalities, which is a desirable property. Estimation problems associated with convex objectives and constraints are usually tractable, even if the dimensionality of the problem (the number of unknown coefficients to estimate) is very high.

The new estimation framework has been implemented as illustrated in Figure 5.1 for solving various transportation engineering problems using experimental traffic data. The same framework can be used for very different problems, such as estimation problems (for instance travel time estimation), sensor fault detection problems, or user privacy analysis. All these problems are posed as *Linear Programs* (LPs), a particular class of convex-optimization problems for which numerous solvers exist [78].

25. Hamilton-Jacobi PDEs: Fast and exact semi-analytic schemes

This chapter is organized as follows. We first construct the Lax-Hopf algorithm introduced as the backbone of our method, and present some of its benefits and applications in chapter 25. For this, section 25.1 introduces the HJ PDE model to be investigated. Section 25.2 presents the notion of *value condition*, which encompasses the traditional concepts of initial and boundary as well as a new concept of internal conditions. We then derive a possible method for solving the HJ PDE using the control framework of *Viability Theory* in section 25.3. This method enables us to define a *Lax-Hopf formula*, which characterizes the solution. In section 25.4, we describe the mathematical properties of the solution, derived from the structure of the Lax-Hopf formula. In particular, the inf-morphism property enables us to decompose a complex problem involving multiple initial, boundary and internal conditions into more tractable subproblems. We then show in section 25.5 that the subproblems, namely the problems of computing the solutions associated with affine initial, boundary and internal conditions can be solved exactly and explicitly. Using these solutions and the inf-morphism property derived earlier, we build in section 25.6 a semi-analytic numerical scheme for solving the HJ PDE exactly and without requiring a computational grid. We also show in section 25.7 that a similar numerical scheme can be used to solve the corresponding scalar conservation laws. Numerical illustrations and a comparison with standard first-order numerical schemes are performed in section 8.

25.1. Macroscopic highway traffic modeling

25.1.1. State of the art

Traffic flow models can be separated into at least two distinct classes, depending on the scale at which they describe traffic. *Microscopic models* such as the *car following model* [255], describe traffic at the individual vehicle level as a flow of particles. Their objective is to provide a relationship between the velocity of a given vehicle and its environment. In contrast, *macroscopic models* [174,226,285] describe traffic flow as a continuous medium and are related to fluid mechanics models. In the present work, we focus on the *Lighthill-Whitham-Richards* [226,285] (LWR) model, which is a first order macroscopic

flow model. Owing to its simplicity and its robustness, the LWR model and its related *cell transmission model* [126,127] are commonly used in transportation engineering [327,128,49,339]. Note that macroscopic models are not necessarily first order models, see for instance [76]. Traffic flow can also be described at an intermediate scale using *mesoscopic models* [84]. Mesoscopic models follow methods of statistical mechanics, and express the solution using an integro-differential equation such as the *Boltzmann equation* [97].

Similarly to other large scale infrastructure systems such as the water channel network, the highway transportation network is a very complex graph containing highway sections connected by junctions or splits. In this framework, we do not consider the effects of the network and solely focus on the description of traffic flow on a highway section. Extending this framework to the whole transportation network [81,155] requires the computation of boundary conditions of each highway section, and is out of the scope of this treatment. It is still a somewhat open problem which will require the generalization of weak boundary conditions [69,149], commonly used in traffic engineering [309,339,182].

25.1.2. First order scalar conservation laws

We define the physical (and computational) domain as the one-dimensional set $X := [\xi, \chi] \subset \mathbb{R}$, where ξ represents the *upstream boundary* and χ represents the *downstream boundary* of the domain. The upstream and downstream boundaries represent the locations at which traffic enters and exits the road section respectively.

Two macroscopic functions are used to describe the state of traffic flow on the highway section: the *density function* and *flow function*, defined as follows. The density $\rho(t,x)$ corresponds to the number of vehicles per unit distance at location x and time t. The flow $q(t,x)$ is defined as the number of vehicles that cross the point x per unit time, at time t. Both functions are related by a conservation equation expressing the fact that vehicles do not appear or disappear inside the highway section:

$$\frac{\partial \rho(t,x)}{\partial t} + \frac{\partial q(t,x)}{\partial x} = 0 \tag{191}$$

Equation (191) alone cannot be solved since it involves two different functions. In order to compute the evolution of $\rho(\cdot,\cdot)$ and $q(\cdot,\cdot)$, one needs an additional equation relating these two functions. Greenshields [174] was one of the first to identify a direct relationship between density and flow of the form $q(\cdot,\cdot) = \psi(\rho(\cdot,\cdot))$, where $\psi(\cdot)$ is a function identified since as *Fundamental Diagram* [273]. The fundamental diagram translates the fact that drivers adapt their speed to the density of vehicles that surround them. Adding this relationship into equation (191) yields a first order scalar conservation law involving the density function, known as *Lighthill-Whitham-Richards* [226,285] PDE:

$$\frac{\partial \rho(t,x)}{\partial t} + \frac{\partial \psi(\rho(t,x))}{\partial x} = 0 \tag{192}$$

25.1.3. Hamilton Jacobi equations with concave Hamiltonians

Instead of describing traffic flow in terms of a density function [224,309], a possible alternate formulation known as the *Moskowitz function* uses a Hamilton-Jacobi equation for describing the evolution of an integral of the function $\rho(\cdot,\cdot)$ [107,59,104,105]. The Moskowitz function is physically defined as follows.

Definition 1. [Moskowitz function] Let consecutive integer labels be assigned to vehicles entering the highway at location $x = \xi$. The Moskowitz function $\boldsymbol{M}(\cdot,\cdot)$ is a continuous function satisfying $\lfloor \boldsymbol{M}(t,x) \rfloor = n$ where n is the label of the vehicle located in x at time t [128,129,252]. Hence, $\boldsymbol{M}(t,x)$ represents the label of the vehicle located at x at time t, counted from the reference point $(0,\xi)$ correspondingto the vehicle numbered 0.

The properties of the Moskowitz function have been extensively studied, for instance in the famous Newell trilogy [256]. The formal link between the density function $\rho(\cdot,\cdot)$, the flow function $q(\cdot,\cdot)$ and the Moskowitz function $\boldsymbol{M}(\cdot,\cdot)$ is given by:

$$\boldsymbol{M}(t_2,x_2) - \boldsymbol{M}(t_1,x_1) = \int_{x_1}^{x_2} -\rho(t_1,x)dx + \int_{t_1}^{t_2} q\,(t,x_2)dt \tag{193}$$

Conversely, the flow and density functions $q(\cdot,\cdot)$ and $\rho(\cdot,\cdot)$ are related to the spatial and temporal derivatives of the Moskowitz function $\boldsymbol{M}(\cdot,\cdot)$:

$$q(t,x) = \frac{\partial \boldsymbol{M}(t,x)}{\partial t} \quad \rho(t,x) = -\frac{\partial \boldsymbol{M}(t,x)}{\partial x} \tag{194}$$

The Moskowitz function $\boldsymbol{M}(\cdot,\cdot)$ solves the following equation, obtained by combining (194) and the LWR PDE (192):

$$\frac{\partial \boldsymbol{M}(t,x)}{\partial t} - \psi\left(-\frac{\partial \boldsymbol{M}(t,x)}{\partial x}\right) = 0 \tag{195}$$

Equation (195) is an *Hamilton-Jacobi* (HJ) PDE [116,59]. In the context of HJ PDEs, the parameter $\psi(\cdot)$ is known as *Hamiltonian*, while it is known as fundamental diagram in the context of the LWR PDE (192) and traffic engineering [126].

25.1.4. Hamiltonian

The LWR PDE (192) and its associated HJ PDE (195) are both characterized by a Hamiltonian $\psi(\cdot)$, which describes the relationship between density and flow. For low densities, the average velocity of traffic $v(\cdot,\cdot) = \frac{q(\cdot,\cdot)}{\rho(\cdot,\cdot)}$ is close to maximal velocity allowed on the road section, denoted by $v^{\flat}$. As the density increases, traffic velocity progressively drops and vanishes for the maximal density ω that the highway section can contain and known as *jam density*. Hence, the Hamiltonian $\psi(\cdot)$ satisfies the following properties:

- $\lim_{\rho\to 0} \frac{\psi(\rho)}{\rho} = v^{\flat}$
- the function $\rho \to \frac{\psi(\rho)}{\rho}$ is decreasing
- $\psi(\omega) = 0$

An example of flow-density plot using experimental data from the *Performance Measurement System* (PeMS) [22] is shown in Figure 1.1.

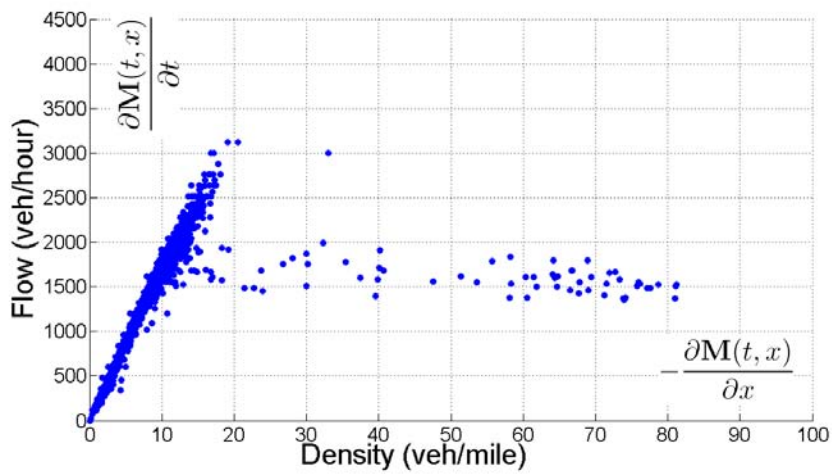


Figure 1.1. Illustration of the flow-density relationship.

The horizontal axis represents the density of vehicles, while the vertical axis corresponds to the flow of vehicles. Each point of this plot corresponds to a simultaneous measurement of flow and density at a fixed location, using an inductive loop detector [22].

For mathematical reasons, the Hamiltonian is often assumed to be either concave or convex [116,59] in the HJ PDE theory, though this requirement is not dictated by the physics of the problem. In the present work, we assume *once and for all* that the Hamiltonian is a concave and upper semicontinuous function defined on $[0,\omega]$, where ω is called *jam density* and that $\psi(0) = \psi(\omega) = 0$. We also assume that $\psi(\cdot)$ satisfies $\psi'(0) = v^{\flat}$ and $\psi'(\omega) = -v^{\#}$,

where $\nu^\flat > 0$ and $\nu^\# > 0$, which implicitly assumes that $\psi(\cdot)$ is differentiable at 0 and ω. However, we do not assume that $\psi(\cdot)$ is differentiable on $]0, \omega[$ and construct our analysis for this general set of concave $\psi(\cdot)$ functions.

Different choices of Hamiltonians satisfying these properties are possible, including the two examples presented below.

Example 2. [Greenshields Hamiltonian] [174,52]. One of the first Hamiltonian identified in the context of traffic-flow modeling is the *Greenshields Hamiltonian* [174], defined by:

$$\forall\ \rho \in \mathbb{R},\quad \psi(\rho) := \frac{\nu}{\omega}\rho(\omega - \rho) \tag{196}$$

where ω and ν are model parameters, respectively referred to as *jam density* and *free flow velocity* in the transportation literature. Note that the Greenshields Hamiltonian depends only on two parameters, which makes it compact and easy to calibrate. The Greenshields Hamiltonian is however not used very often in practice, since it predicts unrealistically high maximal flows.

Another example of Hamiltonian is the *Trapezoidal Hamiltonian*, widely used in traffic flow modeling [126].

Example 3. [Trapezoidal Hamiltonian] [126,127,311]. The trapezoidal Hamiltonian is commonly used to model the hybrid nature of traffic flow propagation:

$$\psi(\rho) = \begin{cases} \nu^\flat \rho & \text{if} \quad \rho \le \gamma^\flat \\ \delta & \text{if} \quad \rho \in [\gamma^\flat, \gamma^\#] \\ \nu^\#(\omega - \rho) & \text{if} \quad \rho \ge \gamma^\# \end{cases}$$

where $\nu^\flat$, $\nu^\#$, ω, δ, $\gamma^\flat$ and $\gamma^\#$ are constants and satisfy the following relations: $\delta \le \frac{\omega\nu^\flat\nu^\#}{\nu^\flat + \nu^\#}$ (called capacity in the transportation engineering literature), $\gamma^\flat := \frac{\delta}{\nu^\flat}$ (called lower critical density in the transportation engineering literature) and $\gamma^\# := \frac{\nu^\#\omega - \delta}{\nu^\#}$ (called upper critical density in the transportation engineering literature). When $\gamma^\flat = \gamma^\#$, the Hamiltonian is triangular, as used in the applications of Chapter 27.

The Greenshields and trapezoidal Hamiltonians are illustrated in Figure 1.2.

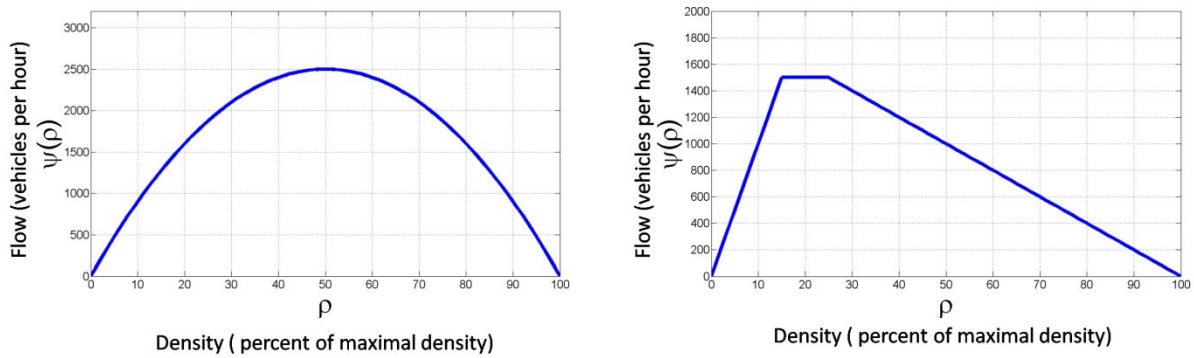


Figure 1.2. Illustration of the Greenshields and trapezoidal Hamiltonians.

Numerical values are represented in the context of transportation, *i.e.* the variable ρ is homogeneous to the vehicle density (in percent of the maximal density). The Hamiltonian $\psi(\rho)$ is represented in vehicles per hour. **Left:** representation of a Greenshields Hamiltonian. **Right:** representation of a trapezoidal Hamiltonian.

Solving the HJ PDE (195) requires the definition of *value conditions*, which we now define.

25.2. Value conditions

25.2.1. General definition

Value conditions encompass the traditional concepts of initial, boundary and internal conditions and are defined as follows.

Definition 1. [Value condition] A value condition $\boldsymbol{c}(\cdot,\cdot)$ is a lower semicontinuous function defined on a subset of $[0, t_{\max}] \times X$.

By convention, a value condition $\boldsymbol{c}(\cdot,\cdot)$ as defined in definition 1 satisfies $\boldsymbol{c}(t,x) = +\infty$ if $(t,x) \notin \mathrm{Dom}(\boldsymbol{c})$. The domain of definition of a value condition represents the subset of the space timedomain $\mathbb{R}_+ \times X$ in which we want the value condition to apply. Different types of value condition exist, including the traditional initial, upstream and downstream boundary conditions [59,126]. More complex value conditions do exist however. Internal conditions consist in value condition whose domains of definition are connected and of empty interior [128,221]. Hybrid conditions [107] are the most general type of value condition, but are out of the scope of this work.

25.2.2. Initial, boundary and internal conditions

Initial, boundary are common in problems involving PDEs. Internal conditions are specific to the problem introduced in the present work, though it applies to numerous other fields. These value conditions are defined as follows.

Definition 2. [Initial condition] An *initial condition* is a value condition $\boldsymbol{c}(\cdot,\cdot)$ defined on $\mathrm{Dom}(\boldsymbol{c}) := \{0\} \times X$.

Note that the traditional *Cauchy problem* consists in finding the solution to (195) associated with a value condition defined on $\{0\} \times \mathbb{R}$, *i.e.* an initial condition defined on an infinite spatial domain.

In contrast, the upstream and downstream boundary conditions are related to the value of the state on the boundaries of the physical domain.

Definition 3. [Upstream and downstream boundary conditions] An *upstream boundary condition* is a value condition $\boldsymbol{c}(\cdot,\cdot)$ defined on the set $\mathrm{Dom}(\boldsymbol{c}) := [0, t_{max}] \times \{\xi\}$. A *downstream boundary condition* is a value condition $\boldsymbol{c}(\cdot,\cdot)$ defined on $\mathrm{Dom}(\boldsymbol{c}) := [0, t_{max}] \times \{\chi\}$.

Note that the traditional *mixed Initial-Boundary conditions problem* [309] consists in finding the solution to (195) associated with a value condition defined on $\{0\} \times X \cup \mathbb{R}_+ \times \{\xi\} \cup \mathbb{R}_+ \times \{\chi\}$, *i.e.* an initial condition, an upstream boundary condition and a downstream boundary condition defined on an infinite temporal domain.

Note that the initial, upstream and downstream boundary conditions are all defined at the boundary of the computational domain $[0, t_{max}] \times X$. Since probe measurements originate from the interior of the computational domain, a specific type of value condition, known as *internal condition* has to be defined as follows.

Definition 4. [Internal condition] An *internal condition* is a value condition $\boldsymbol{c}(\cdot,\cdot)$ defined on a domain of the form $\mathrm{Dom}(\boldsymbol{c}) := \{(t, x_v(t)),\ t \in \mathrm{Dom}(x_v)\}$, where $x_v(\cdot)$ is a function of $[0, t_{max}]$.

In definition 4, the function $x_v(\cdot)$ represents the *velocity function* associated with the internal condition. The set $\{(t, x_v(t)),\ t \in \mathrm{Dom}(x_v)\}$ is the *trajectory* associated with the internal condition.

Note that in the applications of this framework, measurement data alone is not sufficient to define the value conditions unambiguously, since some of coefficients used to build these value conditions are impossible to measure, or are not perfectly known due to measurement errors.

We now present a characterization of the solutions to the HJ PDE (195) associated with the value conditions defined earlier. This characterization uses *Viability theory*, an area of optimal control studying the evolution of dynamical systems evolving under state constraints [57,58] known as *viability constraints*.

25.3. Viability formulation of the solution

25.3.1. Barron-Jensen/Frankowska solutions

As mentioned earlier, several classes of solutions to HJ PDEs exist. *Viscosity solutions* [116] to HJ PDEs are continuous functions. The specific type of solutions to (195) that we consider in the present work is the *Barron-Jensen/Frankowska* (B-J/F) solutions [70,152]. B-J/F solutions extend the concept of viscosity solutions by allowing the solution to be lower semicontinuous. Note that both concepts are identical for mixed initial-boundary conditions problems involving Lipschitz-continuous initial and boundary conditions [152].

The B-J/F solutions to (195) can be derived using the control framework of Viability theory [57], presented in the following section.

25.3.2. Viability characterization of Barron-Jensen/Frankowska solutions

We now introduce some tools used in the context of viability theory [57,58], which are essential building blocks for the work presented here.

Definition 1. [57,58] **[Capture basin]** Given a dynamical system F and two sets $\mathcal{K}$ (called the constraint set) and $\mathcal{C}$ (called the target set) satisfying $\mathcal{C} \subset \mathcal{K}$, the capture basin $\mathrm{Capt}_F(\mathcal{K}, \mathcal{C})$ is the subset of states of $\mathcal{K}$ from which there exists at least one evolution solution to F reaching the target $\mathcal{C}$ in finite time while remaining in $\mathcal{K}$.

Note that the capture basin associated with a given dynamical system, constraint and target set can be numerically computed using the *Capture Basin Algorithm* [86,87,290]. In order to properly define the dynamical system used to construct B-J/F solutions to (195), we first need to define a convex transform $\varphi^*(\cdot)$ of the Hamiltonian $\psi(\cdot)$ as follows.

Definition 2. **[Convex transform]** Given a concave and upper semicontinuous function $\psi(\cdot)$ with domain $\mathrm{Dom}(\psi)$, we define the convex transform $\varphi^*(\cdot)$ of $\psi(\cdot)$ as follows:

$$\varphi^*(u) := \sup_{p \in \mathrm{Dom}(\psi)} [\, p \cdot u + \psi(p)] \tag{197}$$

The inverse transform of a convex and lower semicontinuous function $\varphi^*(\cdot)$ is defined [59] by:

$$\psi(p) := \inf_{u \in \mathrm{Dom}(\varphi^*)} [\, \varphi^*(u) - p \cdot u] \tag{198}$$

Note that equation (197) in definition 2 differs from the traditional definition of the Legendre-Fenchel transform by a sign change.

The function $\varphi^*(\cdot)$ defined by (197) is convex as the pointwise supremum of affine functions [78,288] and is defined on the interval $\mathrm{Dom}(\varphi^*) := [-\nu^\flat, \nu^\#]$. Since $\varphi^*(\cdot)$ is convex, it is subdifferentiable [78] on $[-\nu^\flat, \nu^\#]$ and its subderivative satisfies the Legendre-Fenchel inversion formula [59]:

$$u \in -\partial_+\psi(\rho) \quad \text{if and only if} \quad \rho \in \partial_-\varphi^*(u) \tag{199}$$

in which, following [78], we use the following definition of the subderivative $\partial_-(\cdot)$ and the superderivative $\partial_+(\cdot)$:

$$v \in \partial_-\mathcal{F}(x_0) \;\text{ if and only if }\; \forall x \in \mathrm{Dom}(\mathcal{F}),\;\; \mathcal{F}(x) \geq \mathcal{F}(x_0) + v(x - x_0) \tag{200}$$

$$v \in \partial_+\mathcal{F}(x_0) \;\text{ if and only if }\; \forall x \in \mathrm{Dom}(\mathcal{F}),\;\; \mathcal{F}(x) \leq \mathcal{F}(x_0) + v(x - x_0) \tag{201}$$

Note that any convex (respectively concave) function $\mathcal{F}(\cdot)$ is subdifferentiable (respectively superdifferentiable) on its domain of definition [78].

The convex transform satisfies $\varphi^*(-\nu^\flat) := \sup_{p\in\mathrm{Dom}(\psi)}[-p\nu^\flat + \psi(p)] = 0$ since $\psi(\cdot)$ is concave and satisfies $\psi'(0) = \nu^\flat$. In addition, it is positive by (197) since $\psi(0) = 0$ and $0 \in [0, \omega]$.

The convex transforms associated with Greenshields and trapezoidal Hamiltonians defined in section 25.1.4 are represented in Figure 3.1.

The convex transform $\varphi^*(\cdot)$ enables the definition of an auxiliary dynamical system, which will be used to characterize the solutions to (195) as capture basins.

Definition 3. [Auxiliary dynamical system] We define an auxiliary dynamical system F associated with the HJ PDE (195):

$$F := \begin{cases} \tau'(t) &= -1 \\ x'(t) &= u(t) \\ y'(t) &= -\varphi^*(u(t)) \end{cases} \qquad \text{where } u(t) \in \mathrm{Dom}(\varphi^*) \tag{202}$$

The function $u(\cdot)$ is called *auxiliary control* of the dynamical system F.

The dynamical system (202) is both Marchaud and Lipschitz [59]. To be rigorous, we have to mention *once and for all* that the controls $u(\cdot)$ are measurable integrable functions with values in $\mathrm{Dom}(\varphi^*)$, and thus, ranging $L^1(0,+\infty;\mathrm{Dom}(\varphi^*))$ and that the above system of differential equations is valid for almost all $t \geq 0$. We illustrate the auxiliary dynamical system in Figure 3.2.

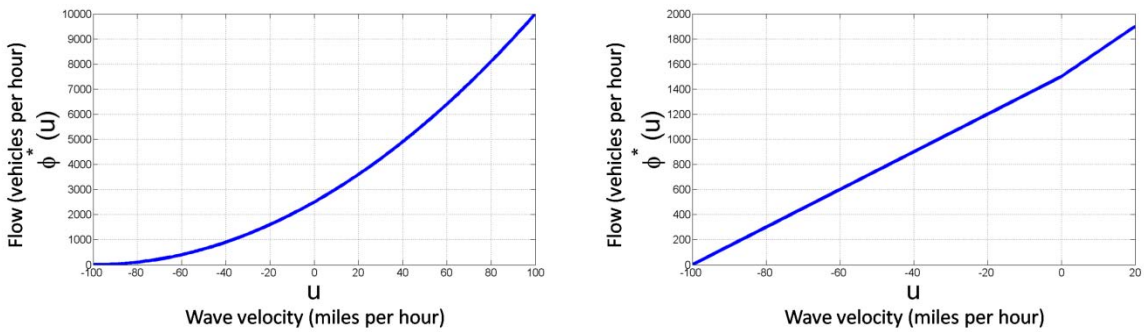


Figure 3.1. Illustration of the convex transforms associated with the Greenshields and trapezoidal Hamiltonians.

Left: representation of the function φ^* associated with a Greenshields Hamiltonian. **Right:** representation of the function $\varphi^*(\cdot)$ associated with a trapezoidal Hamiltonian.

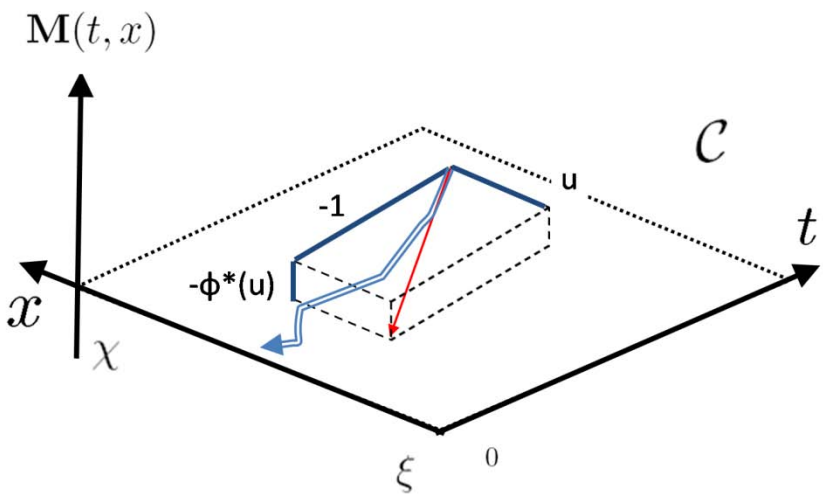


Figure 3.2. Illustration of the auxiliary dynamical system used to construct the solutions to the HJ PDE.

The auxiliary dynamical system (202) is illustrated by a box. The compound line represents a possible evolution of this dynamical system.

The environment set $\mathcal{K}$ is defined in epigraphical form as $\mathcal{K} := \mathcal{E}pi(\boldsymbol{b}(\cdot,\cdot))$, where $\boldsymbol{b}(\cdot,\cdot)$ is a lower semicontinuous function. $\boldsymbol{b}(\cdot,\cdot)$ represents a lower bound that we impose on thesolution to the HJ PDE (195). The problem of finding a solution to (195) under lower bound constraints is extensively studied in [59]. In the present work, we do not impose a lower bound on the solution and thus choose the following environment set:

Definition 4. [Environment set] We define the environment $\mathcal{K}$ as $\mathcal{K} := \mathbb{R}_+ \times [\xi, \chi] \times \mathbb{R}$.

The target set is also defined in epigraphical form as $\mathcal{C} := \mathcal{E}pi(\boldsymbol{c}(\cdot,\cdot))$, where $\boldsymbol{c}(\cdot,\cdot)$ represents an upper bound that we impose on the solution to the HJ PDE (195).

Definition 5. [Target set] Let a value condition $\boldsymbol{c}(\cdot,\cdot)$ be given. The epigraphical target set associated with $\boldsymbol{c}(\cdot,\cdot)$ is defined as $\mathcal{C} := \mathcal{E}pi(\boldsymbol{c})$.

Note that the target set $\mathcal{C} = \mathcal{E}pi(\boldsymbol{c})$ associated with a value condition $\boldsymbol{c}(\cdot,\cdot)$ is closed, since it is the epigraph of a lower semicontinuous function.

Using the above definitions of auxiliary dynamical system F, environment set $\mathcal{K}$ and target set $\mathcal{C}$, we can now represent the capture basin $\mathrm{Capt}_F(\mathcal{K}, \mathcal{C})$ as in definition 1. We illustrate the construction of $\mathrm{Capt}_F(\mathcal{K}, \mathcal{C})$ in Figure 3.3.

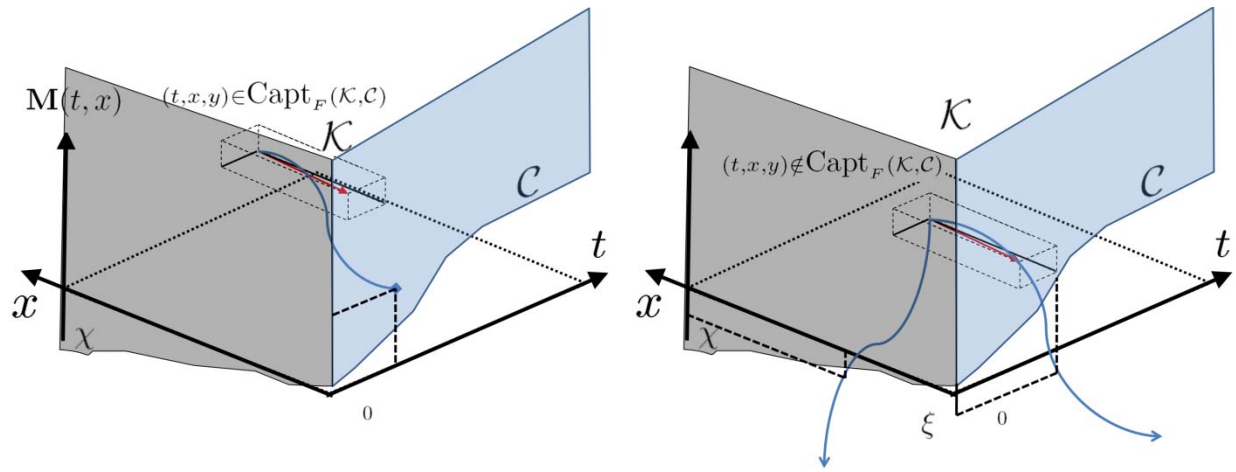


Figure 3.3. Illustration of a capture basin associated with an epigraphical target.

Left: element (t,x,y) of the capture basin $\mathrm{Capt}_F(\mathcal{K},\mathcal{C})$: there exists an evolution starting from (t,x,y) and reaching $\mathcal{C}$ in finite time while remaining in $\mathcal{K}:=\mathbb{R}_+\times X\times\mathbb{R}$. **Right:** element (t,x,y) not belonging to the capture basin $\mathrm{Capt}_F(\mathcal{K},\mathcal{C})$: all evolutions starting from (t,x,y) exit the set $\mathcal{K}$ before reaching $\mathcal{C}$ (only two evolutions are represented for clarity).

Definition 6. [Viability episolution] The *viability episolution* $\boldsymbol{M_c}(\cdot,\cdot)$ associated with the epigraphical target $\mathcal{C}=\boldsymbol{c}(\cdot,\cdot)$ is defined by

$$\boldsymbol{M_c}(t,x):=\inf_{(t,x,y)\in\mathrm{Capt}_F(\mathcal{K},\mathcal{C})} y \tag{203}$$

Remark 1. The capture basin $\mathrm{Capt}_F(\mathcal{K},\mathcal{C})$ of a target $\mathcal{C}$ viable in the environment $\mathcal{K}$ is the subset of initial states (t,x,y) for which there exists a measurable control $u(\cdot)$ such that its associated evolution

$$s\mapsto\left(t-s,x+\int_0^s u(\tau)d\tau,y-\int_0^s\varphi^*(u(\tau))d\tau\right) \tag{204}$$

is viable in $\mathcal{K}$ (*i.e.* remains in $\mathcal{K}$ at all times) until it reaches the target $\mathcal{C}$ in finite time.

Remark 2. The capture basin $\mathrm{Capt}_F(\mathcal{K},\mathcal{C})$ is actually the epigraph of the function $\boldsymbol{M_c}(\cdot,\cdot)$ defined by (203). Indeed, let $(t,x,y)\in\mathrm{Capt}_F(\mathcal{K},\mathcal{C})$. We thus have that (204) is an evolution viable in $\mathcal{K}$ reaching $\mathcal{C}$ in finite time. Since $\mathcal{K}$ and $\mathcal{C}$ are epigraphs, we have for any $y'\geq y$ that the evolution

$$s\mapsto\left(t-s,x+\int_0^s u(\tau)d\tau,y'-\int_0^s\varphi^*(u(\tau))d\tau\right) \tag{205}$$

also remains in $\mathcal{K}$ at all times until it reaches the target $\mathcal{C}$. Hence, (t, x, y') also belongs to $\mathrm{Capt}_F(\mathcal{K}, \mathcal{C})$, which proves that $\mathrm{Capt}_F(\mathcal{K}, \mathcal{C}) = \mathcal{E}pi(\boldsymbol{M_c})$.

We illustrate the viability episolution associated with a given value condition in Figure 3.4.

We represent on the same figure a target $\mathcal{C}$ and its associated viability episolution $\boldsymbol{M_c}(\cdot,\cdot)$. The episolution is the lower boundary of the capture basin $\mathrm{Capt}_F(\mathcal{K}, \mathcal{C})$, shaded in this figure.

The viability episolution $\boldsymbol{M_c}(\cdot,\cdot)$ defined by equation (203) is shown in theorem 7 to be a B-J/F solution to equation (195). If furthermore $\boldsymbol{M_c}(\cdot,\cdot)$ is differentiable, it is a classical solution to equation (195).

The work [59] defines the B/J-F solution in hypographical form for a function $\boldsymbol{N}(\cdot,\cdot)$ satisfying an inhomogeneous HJ PDE:

$$\frac{\partial \boldsymbol{N}(t,x)}{\partial t} + \psi\left(\frac{\partial \boldsymbol{N}(t,x)}{\partial x}\right) = \psi(v(t)) \tag{206}$$

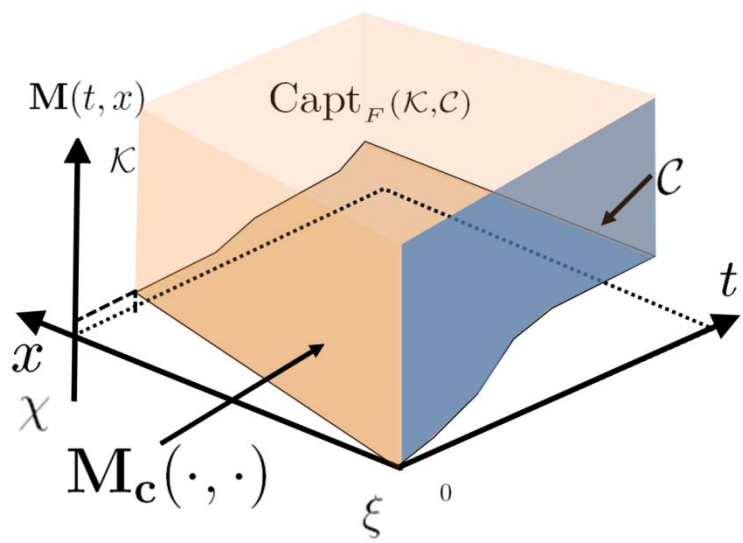


Figure 3.4. Illustration of a viability episolution.

The following theorem is identical to the main existence and uniqueness theorem of [59] modulo the variable change $\boldsymbol{M}(t,x) = -\boldsymbol{N}(t,x) + \int_0^t \psi\,(v(u))du$, the translation of hypographs into epigraphs and the corresponding change on epi/hypo derivatives and differentials.

Theorem 7. [Barron-Jensen/Frankowska solution] [59] For any lower semicontinuous value condition $\boldsymbol{c}_i$, the associated solution $\boldsymbol{M}_{\boldsymbol{c}_i}$ is the **unique** lower semicontinuous function lower than $\boldsymbol{c}_i$ satisfying:

$$\begin{cases}(i) & \forall(t,x)\in \mathrm{Dom}(\boldsymbol{M}_{\boldsymbol{c}_i})\backslash\mathrm{Dom}(\boldsymbol{c}_i),\ \forall(p_t,p_x)\in d_-\boldsymbol{M}_{\boldsymbol{c}_i}(t,x),\quad p_t-\psi(-p_x)\ =\ 0\\ (ii) & \forall(t,x)\in \mathrm{Dom}(\boldsymbol{M}_{\boldsymbol{c}_i})\backslash\mathrm{Dom}(\boldsymbol{c}_i),\ \forall(p_t,p_x)\in(\mathrm{Dom}(D_\uparrow\boldsymbol{M}_{\boldsymbol{c}_i}(t,x)))^+,\ p_t-\sigma(\mathrm{Dom}(\varphi^*),p_x)=0\end{cases} \tag{207}$$

where the epiderivative $D_\uparrow$ is defined by its epigraph:

$$\mathcal{E}p(D_\uparrow\boldsymbol{M}_{\boldsymbol{c}_i}(t,x))\ :=\ T_{\mathcal{E}p(\boldsymbol{M}_{\boldsymbol{c}_i})}(t,x,\boldsymbol{M}_{\boldsymbol{c}_i}(t,x)) \tag{208}$$

where in the formulae (208) and (207) $T_{\mathcal{Z}}(z)$ represents the contingent cone to $\mathcal{Z}$ at z (see [61]), $\sigma(\cdot,\cdot)$ is the support function (see [57,61,60]), the $+$ superscript denotes the normal cone (see [59]) and where the subdifferential d_- of a function $u: X\to\mathbb{R}\cup\{+\infty\}$ is defined by $d_-u(x)=\{p\in X^*|\forall v\in X,\ \ <p,v>\leq D_\uparrow u(x)(v)\}$.

Theorem 7 ensures that $\boldsymbol{M}_{\boldsymbol{c}_i}$ is a solution to the HJ PDE (195) in the B-J/F sense. In particular, since $d_-\boldsymbol{M}_{\boldsymbol{c}_i}(t,x)=\{(\frac{\partial\boldsymbol{M}_{\boldsymbol{c}_i}(t,x)}{\partial t},\frac{\partial\boldsymbol{M}_{\boldsymbol{c}_i}(t,x)}{\partial x})\}$ whenever $\boldsymbol{M}_{\boldsymbol{c}_i}(t,x)$ is differentiable, equation (207) implies the following property:

$$\forall(t,x)\in\mathrm{Dom}(\boldsymbol{M}_{\boldsymbol{c}_i})\backslash\mathrm{Dom}(\boldsymbol{c}_i)\text{ such that }\boldsymbol{M}_{\boldsymbol{c}_i}\text{ is differentiable, }\frac{\partial\boldsymbol{M}_{\boldsymbol{c}_i}(t,x)}{\partial t}-\psi\left(-\frac{\partial\boldsymbol{M}_{\boldsymbol{c}_i}(t,x)}{\partial_x}\right)=0 \tag{209}$$

The construction of the B-J/F solution to (195) as a capture basin enables the definition of a *Lax-Hopf formula*.

25.3.3. The Lax-Hopf formula

The viability episolution $\boldsymbol{M}_{\boldsymbol{c}}(\cdot,\cdot)$ associated with a general value condition $\boldsymbol{c}(\cdot,\cdot)$ can be computed using the following generalized Lax-Hopf formula. The classical Lax-Hopf formulae can be found in [59] for initial and upstream boundary conditions.

Theorem 8. [Generalized Lax Hopf formula] The viability episolution $\boldsymbol{M}_{\boldsymbol{c}}(\cdot,\cdot)$ associated with a target $\mathcal{C}:=\mathcal{E}pi(\boldsymbol{c})$, for a given lower semicontinuous function $\boldsymbol{c}(\cdot,\cdot)$ and defined by equation (203) can be expressed as:

$$\boldsymbol{M}_{\boldsymbol{c}}(t,x)=\inf_{(u,T)\in\mathrm{Dom}(\varphi^*)\times\mathbb{R}_+}(\boldsymbol{c}(t-T,x+Tu)+T\varphi^*(u)) \tag{210}$$

Proof — We fix $(t,x)\in\mathbb{R}_+\times X$ and define R as the set of elements $(u(\cdot),T,y)$ belonging

to $L^1(0,\infty;\mathrm{Dom}(\varphi^*)) \times \mathbb{R}_+ \times \mathbb{R}$ and satisfying viability property (211):

$$\forall s \in [0,T] \left(t-s, x + \int_0^s u(\tau)d\tau, y - \int_0^s \varphi^*(u(\tau))d\tau \right) \in \mathcal{K} \tag{211}$$

Equations (203) and (204) thus imply the following formula:

$$\boldsymbol{M_c}(t,x) = \inf_{(u(\cdot),T,y)\in R \text{ such that } \left(t-T, x+\int_0^T u(\tau)d\tau, y-\int_0^T \varphi^*(u(\tau))d\tau\right)\in \mathcal{E}pi(\boldsymbol{c})} y \tag{212}$$

Since the graph of the value condition $\boldsymbol{c}(\cdot,\cdot)$ (denoted $\mathrm{Graph}(\boldsymbol{c})$) is the lower boundary of $\mathcal{E}pi(\boldsymbol{c})$, we have that

$$\left.\begin{array}{l} \left(t-T, x + \int_0^T u(\tau)d\tau, y - \int_0^T \varphi^*(u(\tau))d\tau\right) \in \mathcal{E}pi(\boldsymbol{c}) \\ \text{and } \left(t-T, x + \int_0^T u(\tau)d\tau, z - \int_0^T \varphi^*(u(\tau))d\tau\right) \in \mathrm{Graph}(\boldsymbol{c}) \end{array}\right\} \Rightarrow z \leq y \tag{213}$$

Hence, we can (without any further assumption) write equation (212) as:

$$\boldsymbol{M_c}(t,x) = \inf_{(u(\cdot),T,y)\in R \text{ such that } \left(t-T, x+\int_0^T u(\tau)d\tau, y-\int_0^T \varphi^*(u(\tau))d\tau\right)\in \mathrm{Graph}(\boldsymbol{c})} y \tag{214}$$

Since $\boldsymbol{c}$ is infinite outside of its domain of definition and given the definition of $\mathrm{Graph}(\boldsymbol{c})$, equation (214) can be expressed as follows:

$$\boldsymbol{M_c}(t,x) = \inf_{(u(\cdot),T,y)\in R} \left[\boldsymbol{c}\left(t-T, x + \int_0^T u(\tau)d\tau\right) + \int_0^T \varphi^*(u(\tau))d\tau \right] \tag{215}$$

We consider a fixed element $(u(\cdot),T,y) \in R$ and define the following constant control function $\hat{u}$ on the time interval $[0,T]$ as:

$$\hat{u} := \frac{1}{T}\int_0^T u(\tau)d\tau \tag{216}$$

The control function $\hat{u}$ is the average value of the control function $u(\cdot)$ on the time interval $[0,T]$. Note that by convexity of $\mathcal{K}$, $(\hat{u},T,y) \in R$ if $(u(\cdot),T,y) \in R$. In the following, we slightly abuse the notation by calling $\hat{u}(\cdot)$ the constant function $t \to \hat{u}$.

We define $y(u(\cdot),T)$ and $y(\hat{u}(\cdot),T)$ respectively as the values of the term minimized in (215) obtained for the control functions $u(\cdot)$ and $\hat{u}(\cdot)$ and for the capture time T:

$$\begin{cases} y(u(\cdot),T) = \boldsymbol{c}(t-T, x + \int_0^T u\,(\tau)d\tau) + \int_0^T \varphi^*\,(u(\tau))d\tau \\ y(\hat{u}(\cdot),T) = \boldsymbol{c}(t-T, x + T\hat{u}) + T\varphi^*(\hat{u}) \end{cases} \tag{217}$$

Since φ^* is convex and lower semicontinuous, Jensen's inequality implies

$$\varphi^*\left(\frac{1}{T}\int_0^T u\,(\tau)d\tau\right) \leq \frac{1}{T}\int_0^T \varphi^*\,(u(\tau))d\tau \tag{218}$$

and thus, since $\hat{u}T = \int_0^T u\,(\tau)d\tau$

$$y(\hat{u}(\cdot),T) \leq y(u(\cdot),T) \tag{219}$$

Equation (219) thus implies that one can replace the search of the infimum over the class of measurable functions $u(\cdot)$ by the search of the infimum over the set of constant functions $\hat{u}(\cdot)$.

Hence, we can write equation (215) as:

$$\boldsymbol{M_c}(t,x) = \inf_{(u,T)\in \mathrm{Dom}(\varphi^*)\times\mathbb{R}_+} (\boldsymbol{c}(t-T, x+Tu) + T\varphi^*(u)) \tag{220}$$

which enables us to restrict ourselves to the set of constant controls and completes the proof. ■

Remark 3. Given a constant control function u, the coefficient T used for the minimization in equation (210) can be restricted to the elements of the set $S_{\boldsymbol{c}}(t,x,u)$ defined by formula (221):

$$S_{\boldsymbol{c}}(t,x,u) := \{s \in \mathbb{R}_+ \text{ such that } (t-s, x+su) \in \mathrm{Dom}(\boldsymbol{c})\} \tag{221}$$

Indeed, when $T \notin S_{\boldsymbol{c}}(t,x,u)$, $\boldsymbol{c}(t-T, x+Tu)$ is infinite.

We could also alternatively define for any (t,x) the set $R_{\boldsymbol{c}}(t,x)$ as $R_{\boldsymbol{c}}(t,x) := \{(u,T) \in \mathrm{Dom}(\varphi^*) \times \mathbb{R}_+$ s. t. $(t-T, x+Tu) \in \mathrm{Dom}(\boldsymbol{c})\}$. Note that the coefficients (u,T) used for the minimization in equation (210) can also be restricted to the elements of $R_{\boldsymbol{c}}$ (when $(u,T) \notin R_{\boldsymbol{c}}(t,x)$, $\boldsymbol{c}(t-T, x+Tu)$ is infinite).

Remark 4. When $\forall u \in \mathrm{Dom}(\varphi^*)$, $S_{\boldsymbol{c}}(t,x,u) = \emptyset$, equation (220) involves a minimization on an empty set and $\boldsymbol{M}_{\boldsymbol{c}}(t,x)$ is infinite.

Remark 5. Since $\boldsymbol{c}(t-T, x+Tu) = +\infty$ when $(t-T, x+Tu) \notin \mathrm{Dom}(\boldsymbol{c})$, we can write equation (220) as:

$$\boldsymbol{M}_{\boldsymbol{c}}(t,x) = \inf_{\{(u,T)\in \mathrm{Dom}(\varphi^*)\times\mathbb{R}_+ \text{ such that } T\in S_{\boldsymbol{c}}(t,x,u)\}} (\boldsymbol{c}(t-T, x+Tu) + T\varphi^*(u)) \tag{222}$$

or alternatively as:

$$\boldsymbol{M}_{\boldsymbol{c}}(t,x) = \inf_{\{(u,T)\in R_{\boldsymbol{c}}(t,x)\}} (\boldsymbol{c}(t-T, x+Tu) + T\varphi^*(u)) \tag{223}$$

Specific forms of the Lax-Hopf formula (210) associated with affine initial, boundary and internal conditions are presented in section 25.5.

25.4. Properties of the Barron-Jensen/Frankowska solutions to Hamilton-Jacobi equations

25.4.1. Domain of definition

Proposition 1. [Domain of definition] For a given value condition $\boldsymbol{c}(\cdot,\cdot)$, the domain of definition of $\boldsymbol{M}_{\boldsymbol{c}}(\cdot,\cdot)$, also called *domain of influence of* $\boldsymbol{c}(\cdot,\cdot)$, is defined by the following formula:

$$\mathrm{Dom}(\boldsymbol{M}_{\boldsymbol{c}}) = \bigcup_{(t,x)\in \mathrm{Dom}(\boldsymbol{c})} \left(\bigcup_{T\in\mathbb{R}_+} \{t+T\} \times [x - \nu^{\#}T, x + \nu^{\flat}T] \right) \tag{224}$$

Proof — The generalized Lax Hopf formula (210) implies that

$$\begin{aligned}\mathrm{Dom}(\boldsymbol{M_c}) = \quad & \{(t,x) \in \mathbb{R}_+ \times X \text{ such that } \exists (T,u) \in \mathbb{R}_+ \times \mathrm{Dom}(\varphi^*) \\ & \text{and } (t-T, x+Tu) \in \mathrm{Dom}(\boldsymbol{c})\}\end{aligned}$$

Equation (224) is derived from the previous formula, observing that u ranges in $\mathrm{Dom}(\varphi^*) := [-\nu^\flat, \nu^\#]$. ■

Remark 6. The domain of influence of $\boldsymbol{c}(\cdot,\cdot)$ is the union of the cones originating at $(t,x) \in \mathrm{Dom}(\boldsymbol{c})$ and limited by the minimal $-\nu^\flat$ and maximal $\nu^\#$ slopes of the Hamiltonian. This property is illustrated in Figure 4.1.

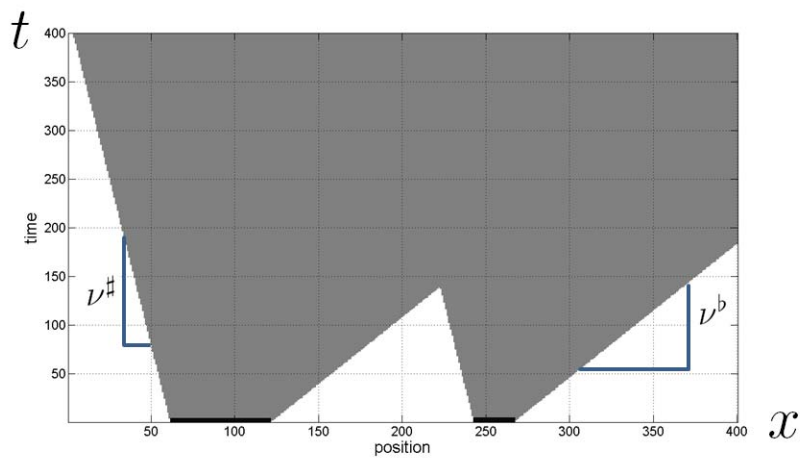


Figure 4.1. Illustration of the domain of influence of a value condition.

define a value condition $\boldsymbol{c}(\cdot)$ on a domain represented by two black segments at $t = 0$. The domain of influence of $\boldsymbol{c}(\cdot,\cdot)$ is highlighted in gray.

25.4.2. The inf-morphism property

It is well known [57,58,59] that for a given environment $\mathcal{K}$, the capture basin of a finite union of targets is the union of the capture basins of these targets.

$$\mathrm{Capt}_F\left(\mathcal{K}, \bigcup_{i\in I} \mathcal{C}_i\right) = \bigcup_{i\in I} \mathrm{Capt}_F\left(\mathcal{K}, \mathcal{C}_i\right) \tag{225}$$

This property can be translated in epigraphical form as follows.

Proposition 2. [Inf-morphism property] [59] Let $\boldsymbol{c}_i$ (i belongs to a finite set I) be a family of functions whose epigraphs are the targets $\mathcal{C}_i$. Since the epigraph of the minimum of the functions $\boldsymbol{c}_i$ is the union of the epigraphs of the functions $\boldsymbol{c}_i$, the target $\mathcal{C} := \bigcup_{i\in I} \mathcal{C}_i$ is the epigraph of the function $\boldsymbol{c} := \min_{i\in I} \boldsymbol{c}_i$. Hence, equation (225) implies the following property,

known as *inf-morphism property*:

$$\forall\ t \geq 0,\ x \in X,\quad \boldsymbol{M}_{\boldsymbol{c}}(t,x)\ =\ \min_{i\in I} \boldsymbol{M}_{\boldsymbol{c}_i}(t,x) \tag{226}$$

Remark 7. The inf-morphism property enables us to decompose a complex problem into more tractable subproblems. For instance, a piecewise affine initial condition can be decomposed as the minimum of a finite number of affine initial conditions. Hence, the solution associated with a piecewise affine initial condition is the minimum of a finite number of solutions associated with affine initial conditions.

The inf-morphism property is illustrated in Figure 4.2.

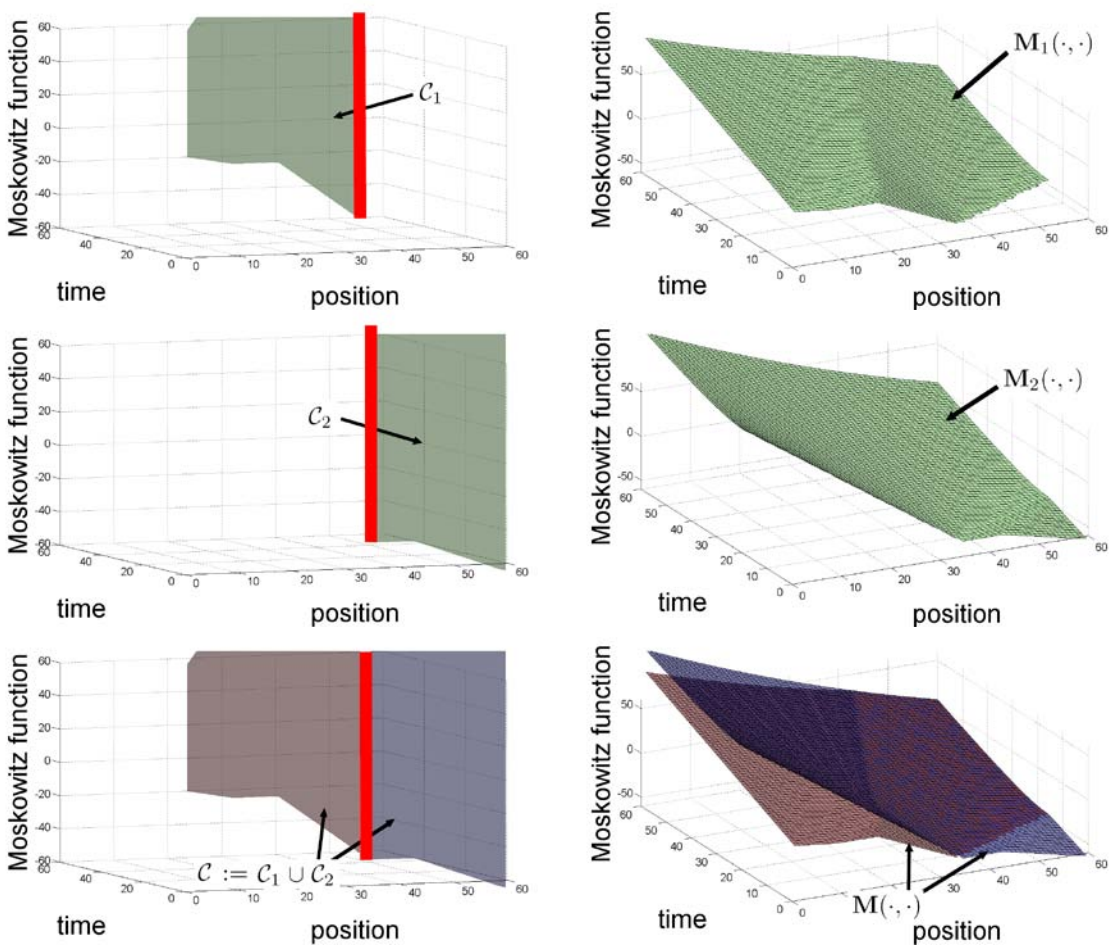


Figure 4.2. Illustration of the inf-morphism property.

Top: representation of the target $\mathcal{C}_1 := \mathcal{E}pi(\boldsymbol{c}_1)$ **(left)**, representation of the corresponding

episolution $\boldsymbol{M}_1(\cdot,\cdot)$ **(right)**. **Center:** representation of the target $\mathcal{C}_2 := \mathcal{E}pi(\boldsymbol{c}_2)$ **(left)**, representation of the corresponding episolution $\boldsymbol{M}_2(\cdot,\cdot)$ **(right)**. **Bottom:** Representation of the target $\mathcal{C} := \mathcal{C}_1 \cup \mathcal{C}_2$ **(left)**. The episolution $\boldsymbol{M}(\cdot,\cdot)$ associated with the target $\mathcal{C}$ **(right)** is the minimum of the episolutions $\boldsymbol{M}_1(\cdot,\cdot)$ and $\boldsymbol{M}_2(\cdot,\cdot)$ associated with $\mathcal{C}_1$ and $\mathcal{C}_2$.

Note that in the context of traffic flow engineering, this property was identified but not proved mathematically by Newell in [256].

25.4.3. Convexity property of the solutions associated with convex value conditions

We now show an important convexity property of the solution to (195) associated with convex value conditions. Let a convex value condition $\boldsymbol{c}(\cdot,\cdot)$ be defined defined on a compact and nonempty domain $\mathrm{Dom}(\boldsymbol{c}) \subset [0, t_{max}] \times X$. Since $\boldsymbol{c}(\cdot,\cdot)$ is convex and defined on a compact set, it is bounded below. In order to prove the convexity of the solution $\boldsymbol{M}_{\boldsymbol{c}}(\cdot,\cdot)$ associated with $\boldsymbol{c}(\cdot,\cdot)$, we first need to define a variable change as follows.

Definition 3. [Variable change for the auxiliary control] We define a new variable v as $v = Tu$ and define the cone $\mathcal{D} := \left\{[-v^{\flat}t, v^{\#}t] \times \{t\} \mid t \in \mathbb{R}_+\right\}$.

Note that definition 3 implies that $(u,T) \in \mathrm{Dom}(\varphi^*) \times \mathbb{R}_+$ if and only if $(v,T) \in \mathcal{D}$. We now define an auxiliary objective function $f(\cdot,\cdot,\cdot,\cdot)$, which is the argument of the Lax-Hopf formula (210) with the variable change $v = Tu$.

Definition 4. [Auxiliary objective function] We define the function $f(\cdot,\cdot,\cdot,\cdot)$ as:

$$\forall (t,x,v,T) \in \mathbb{R}_+ \times X \times \mathcal{D},$$
$$f(t,x,v,T) := \boldsymbol{c}(t-T, x+v) + T\varphi^*\left(\frac{v}{T}\right)$$

Note that $f(\cdot,\cdot,\cdot,\cdot)$ is bounded below since the value condition $\boldsymbol{c}(\cdot,\cdot)$ is bounded below and the function $\varphi^*(\cdot)$ is positive. By definition of $f(\cdot,\cdot,\cdot,\cdot)$, we can rewrite equation (210) as:

$$\boldsymbol{M}_{\boldsymbol{c}}(t,x) = \inf_{(v,T)\in\mathcal{D}} f\,(t,x,v,T) \tag{227}$$

Equation (227) implies:

$$\mathcal{E}pi(\boldsymbol{M}_{\boldsymbol{c}}) = \{(t,x,y) \mid \exists (v,T) \in \mathcal{D} \ \text{ s.t. } \ (t,x,v,T,y) \in \mathcal{E}pi(f)\} \tag{228}$$

The above variable change enables us to prove the following convexity property.

Proposition 5. [Convexity property of solutions associated with convex value conditions] The solution $\boldsymbol{M_c}(\cdot,\cdot)$ associated with a convex value condition $\boldsymbol{c}(\cdot,\cdot)$ is convex.

Proof — Since $\varphi^*(\cdot)$ is convex, its associated *perspective function* $(v,T) \to T\varphi^*(\frac{v}{T})$ is also convex [78] for $T > 0$. Since the function $(t,x,v,T) \to (t-T, x+v)$ is affine and $\boldsymbol{c}(\cdot,\cdot)$ is convex, the function $(t,x,v,T) \to \boldsymbol{c}(t-T, x+v)$ is convex [288,78]. Hence the function $f(\cdot,\cdot,\cdot,\cdot)$ is convex as the sum of two convex functions.

Since the function $f(\cdot,\cdot,\cdot,\cdot)$ is convex, its epigraph $\mathcal{E}pi(f)$ is also convex. Since the set $\mathcal{E}pi(\boldsymbol{c})$ is nonempty, the epigraph of $\boldsymbol{M_c}(\cdot,\cdot)$ is nonempty by the inclusion $\mathcal{E}pi(\boldsymbol{c}) \subset \mathrm{Capt}_F(\mathcal{K}, \mathcal{E}pi(\boldsymbol{c})) := \mathcal{E}pi(\boldsymbol{M_c})$ (see [57] for a proof of this property).

Hence, equation (228) implies that the epigraph of $\boldsymbol{M_c}$ is convex, since it is the projection of a convex set on a subspace [288,78]. ■

In particular, proposition 5 implies that the solutions associated with affine initial, boundary and internal conditions are convex. We now prove that they can also be computed explicitly for general concave Hamiltonians.

25.5. Analytic solutions associated with affine initial, boundary and internal conditions

In this section, we compute the solutions associated with affine initial, boundary and internal conditions analytically. For each type of value condition, we follow the procedure outlined below.

- Write the Lax-Hopf formula associated with the corresponding affine value condition. Because of the structure of the affine value condition, we can compute the set $S_{\boldsymbol{c}}(t,x,u)$ defined by (221) explicitly, which enables us to express (222) as a minimization over a single variable.
- Write the minimization problem associated with this instantiation of the Lax-Hopf formula as a convex optimization problem (convex objective and convex constraints).
- Analytically find a minimizer of the convex optimization problem, using subderivatives (200).

25.5.1. Analytic Lax-Hopf formula associated with an affine initial condition

Definition 1. [Affine initial condition] We consider the following affine initial condition $\mathcal{M}_{0,i}(0,x)$, where i is an integer:

$$\mathcal{M}_{0,i}(0,x) = \begin{cases} a_i x + b_i & \text{if } \ x \in [\overline{\alpha}_i, \overline{\alpha}_{i+1}] \\ +\infty & \text{otherwise} \end{cases} \tag{229}$$

The following formula expresses the Lax-Hopf formula (210) for the specific initial condition (229).

Proposition 2. [Lax-Hopf formula for an affine initial condition] The Lax-Hopf formula associated with the initial condition (229) can be expressed as:

$$\boldsymbol{M}_{\mathcal{M}_{0,i}}(t,x) \ = \inf_{u \in \mathrm{Dom}(\varphi^*) \cap \left[\frac{\overline{\alpha}_i - x}{t}, \frac{\overline{\alpha}_{i+1} - x}{t}\right]} \left(a_i(x+tu) + b_i + t\varphi^*(u)\right), \quad \forall (t,x) \in \mathbb{R}_+^* \times X \tag{230}$$

and

$$\forall x \times X, \quad \boldsymbol{M}_{\mathcal{M}_{0,i}}(0,x) \ = \inf_{(T,u) \in \mathrm{Dom}(\varphi^*) \times [0,0]} \left(a_i(x+0u) + b_i + 0\varphi^*(u)\right) \ = \ a_i x + b_i \tag{231}$$

Proof — The Lax-Hopf formula associated with an initial condition reads:

$$\boldsymbol{M}_{\mathcal{M}_{0,i}}(t,x) \ = \inf_{(T,u) \in \mathrm{Dom}(\varphi^*) \times [0,t] \text{ such that } (x+Tu) \in [\overline{\alpha}_i, \overline{\alpha}_{i+1}] \text{ and } t-T=0} \left(a_i(x+tu) + b_i + T\varphi^*(u)\right) \tag{232}$$

This formula is valid for all $(t,x) \in \mathbb{R}_+ \times X$. Since $t - T = 0$, we have $T = t$. Since $t > 0$, the condition $(x + tu) \in [\overline{\alpha}_i, \overline{\alpha}_{i+1}]$ is equivalent to $u \in [\frac{\overline{\alpha}_i - x}{t}, \frac{\overline{\alpha}_{i+1} - x}{t}]$, which in turn implies equation (230). ■

The domain of definition of the solution can be explicitly characterized as follows.

Proposition 3. [Domain of influence of an affine initial condition] The domain of definition of $\boldsymbol{M}_{\mathcal{M}_{0,i}}(\cdot,\cdot)$ is given by the following formula:

$$\mathrm{Dom}(\boldsymbol{M}_{\mathcal{M}_{0,i}}) = \left\{(t,x) \in \mathbb{R}_+^* \times X \text{ such that } \overline{\alpha}_i - \nu^{\#} t \leq x \leq \overline{\alpha}_{i+1} + \nu^{\flat} t\right\} \tag{233}$$

Proof — The Lax-Hopf formula (230) implies:

$$\mathrm{Dom}(\boldsymbol{M}_{\mathcal{M}_{0,i}}) := \left\{(t,x) \in \mathbb{R}_+^* \times X \text{such that } \exists u \in \mathrm{Dom}(\varphi^*) \cap \left[\frac{\overline{\alpha}_i - x}{t}, \frac{\overline{\alpha}_{i+1} - x}{t}\right]\right\}$$

Equation (233) is obtained using the above formula and noting that $\mathrm{Dom}(\varphi^*) = [-\nu^\flat, \nu^\#]$. ■

The solution can be computed analytically by minimizing an auxiliary function, which we now define.

Definition 4. [Auxiliary objective function] For all $(a_i, b_i, t, x) \in \mathbb{R}^2 \times \mathrm{Dom}(\boldsymbol{M}_{\mathcal{M}_{0,i}})$, we define an objective function $\zeta_{a_i,b_i,t,x}(\cdot)$ by the following formula:

$$\forall u \in \mathrm{Dom}(\varphi^*), \quad \zeta_{a_i,b_i,t,x}(u) := a_i(x + tu) + b_i + t\varphi^*(u) \tag{234}$$

Given this definition, equation (230) becomes:

$$\forall (t,x) \in \mathbb{R}_+^* \times X, \ \boldsymbol{M}_{\mathcal{M}_{0,i}}(t,x) = \inf_{u \in \mathrm{Dom}(\varphi^*) \cap \left[\frac{\overline{\alpha}_i - x}{t}, \frac{\overline{\alpha}_{i+1} - x}{t}\right]} \zeta_{a_i,b_i,t,x}(u) \tag{235}$$

The function $\zeta_{a_i,b_i,t,x}(\cdot)$ is convex as the sum of two convex functions and thus subdifferentiable on $\mathrm{Dom}(\varphi^*)$ in the sense of (200). The subderivative of $\zeta_{a_i,b_i,t,x}(\cdot)$ is given by:

$$\begin{aligned} &\forall u \in \mathrm{Dom}(\varphi^*), \ \partial_- \zeta_{a_i,b_i,t,x}(u) = \{w \mid \exists v \in \partial_- \varphi^*(u), \quad w = a_i t + vt\} \\ &:= t \cdot (\{a_i\} + \partial_- \varphi^*(u)) \end{aligned} \tag{236}$$

with a slight abuse of notation for the summation of the two sets in the second equality. This last expression can now be used to analytically compute the minimizer.

Proposition 5. [Explicit minimization of $\zeta_{a_i,b_i,t,x}(\cdot)$] We now assume that a_i in the value condition $\mathcal{M}_{0,i}$ given by (229) satisfies the condition $-a_i \in \mathrm{Dom}(\psi) := [0, \omega]$. Since $\psi(\cdot)$ is concave, it is also superdifferentiable on its domain of definition and thus $\forall \rho \in [0, \omega], \quad \partial_+ \psi(\rho) \neq \emptyset$.

Let $u_0(a_i)$ be an element of $-\partial_+ \psi(-a_i) \neq \emptyset$. Note that the Legendre-Fenchel inversion formula (199) implies that $u_0(a_i) \in \mathrm{Dom}(\varphi^*)$ and $-a_i \in \partial_- \varphi^*(u_0(a_i))$. Using this definition of $u_0(a_i)$, the function $\zeta_{a_i,b_i,t,x}(\cdot)$ has the following minimizer over

$\mathrm{Dom}(\varphi^*) \cap [\frac{\overline{\alpha}_i - x}{t}, \frac{\overline{\alpha}_{i+1} - x}{t}]$:

$$\begin{cases} u = u_0(a_i) & \text{if } \; u_0(a_i) \in [\dfrac{\overline{\alpha}_i - x}{t}, \dfrac{\overline{\alpha}_{i+1} - x}{t}] \\ u = \dfrac{\overline{\alpha}_i - x}{t} & \text{if } \; u_0(a_i) \leq \dfrac{\overline{\alpha}_i - x}{t} \\ u = \dfrac{\overline{\alpha}_{i+1} - x}{t} & \text{if } \; u_0(a_i) \geq \dfrac{\overline{\alpha}_{i+1} - x}{t} \end{cases} \tag{237}$$

Proof — The function $\zeta_{a_i,b_i,t,x}(u)$ is minimal for a given $u \in \mathrm{Dom}(\varphi^*)$ if and only if $0 \in \partial_- \zeta_{a_i,b_i,t,x}(u)$ by [78]. By equation (236), this happens if and only if for this u, $-a_i \in \partial_- \varphi^*(u)$. Using the Legendre-Fenchel inversion formula (199), we can rewrite $u := u_0(a_i) \in -\partial_+ \psi(-a_i)$ as $-a_i \in \partial_- \varphi^*(u_0(a_i))$ and thus $u_0(a_i)$ minimizes $\zeta_{a_i,b_i,t,x}(\cdot)$ over $\mathrm{Dom}(\varphi^*)$. Hence, since $\zeta_{a_i,b_i,t,x}(\cdot)$ is convex, $\zeta_{a_i,b_i,t,x}(u)$ is decreasing for $u \leq u_0(a_i)$ and increasing for $u \geq u_0(a_i)$, which implies equation (237). ■

Proposition 6. [Computation of $\boldsymbol{M}_{\mathcal{M}_{0,i}}(\cdot,\cdot)$] Let $u_0(a_i)$ be defined as in proposition 5. For all $(t,x) \in \mathrm{Dom}(\boldsymbol{M}_{\mathcal{M}_{0,i}})$, the expression $\boldsymbol{M}_{\mathcal{M}_{0,i}}(t,x)$ can be computed using the following formula:

$$\boldsymbol{M}_{\mathcal{M}_{0,i}}(t,x) = \begin{cases} (i) & t\psi(-a_i) + a_i x + b_i \\ & \text{if } \; u_0(a_i) \in [\dfrac{\overline{\alpha}_i - x}{t}, \dfrac{\overline{\alpha}_{i+1} - x}{t}] \\ (ii) & a_i \overline{\alpha}_i + b_i + t\varphi^*(\dfrac{\overline{\alpha}_i - x}{t}) \\ & \text{if } \; u_0(a_i) \leq \dfrac{\overline{\alpha}_i - x}{t} \\ (iii) & a_i \overline{\alpha}_{i+1} + b_i + t\varphi^*(\dfrac{\overline{\alpha}_{i+1} - x}{t}) \\ & \text{if } \; u_0(a_i) \geq \dfrac{\overline{\alpha}_{i+1} - x}{t} \end{cases} \tag{238}$$

Proof — The cases (ii) and (iii) of equation (238) are trivially obtained by combining equations (230) and (237). Since the function $-\psi(\cdot)$ is convex, it is identical [78] to its Fenchel biconjugate:

$$\forall \rho \in [0,\omega], \quad \psi(\rho) = \inf_{u \in \mathrm{Dom}(\varphi^*)} (-\rho u + \varphi^*(u))$$

The function $g: u \to a_i u + \varphi^*(u)$ is convex and thus subdifferentiable on $\mathrm{Dom}(\varphi^*)$. By definition of $u_0(a_i)$, $0 \in \partial_- g(u_0(a_i))$. This last property implies that $u_0(a_i)$ minimizes $g(\cdot)$ over $\mathrm{Dom}(\varphi^*)$ and thus that $\psi(-a_i) = a_i u_0(a_i) + \varphi^*(u_0(a_i))$. Hence, the case (i) of equation (238) is obtained by combining equations (230), (237) and the property $\psi(-a_i) = a_i u_0(a_i) + \varphi^*(u_0(a_i))$. ■

Figure 5.1 illustrates the different domains of equation (238) for the solution associated with an affine initial condition defined by equation (229).

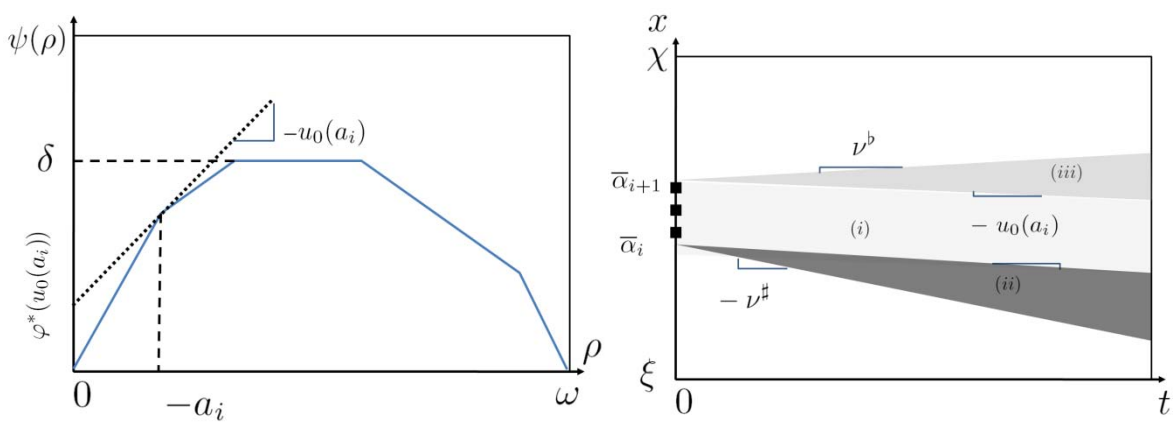


Figure 5.1. Construction of the solution associated with an affine initial condition.

Left: Illustration of the construction of a $u_0(a_i)$ from the knowledge of a_i. The transform $\varphi^*(u_0(a_i))$ corresponds to the value intercepted on the vertical axis by the tangent line of slope $-u_0(a_i)$ to the graph of ψ in $-a_i$. **Right:** The (t, x) domain of the solution corresponding to the affine initial condition (229) can be separated in three different areas. The domain highlighted in light gray corresponds to the case (i) in equation (238). The domain highlighted in medium gray corresponds to the case (iii) and the remaining domain in dark gray corresponds to the case (ii). The domain of the initial condition is represented by a dashed line.

25.5.2. Analytic Lax-Hopf formula associated with an affine upstream boundary condition

Definition 7. [Affine upstream boundary condition] We consider the following upstream boundary condition $\gamma_j(t, \xi)$ of $\gamma_j(t, x)$:

$$\gamma_j(t, \xi) = \begin{cases} c_j t + d_j & \text{if } t \in [\overline{\gamma}_j, \overline{\gamma}_{j+1}] \\ +\infty & \text{otherwise} \end{cases} \tag{239}$$

In the following derivation, we consider that $x > \xi$. We also assume that the value

condition (239) satisfies the condition $c_j \in \mathrm{Im}(\psi) = [0, \delta]$.

Proposition 8. [Lax-Hopf formula for an affine upstream boundary condition] The Lax-Hopf formula (240) associated with the upstream boundary condition (239) can be expressed as:

$$\boldsymbol{M}_{\gamma_j}(t,x) \;=\; \inf_{T \in \left[-\frac{\xi - x}{\nu^{\flat}}, +\infty\right[\cap \left[t - \overline{\gamma}_{j+1}, t - \overline{\gamma}_j\right]} \left(c_j(t-T) + d_j + T\varphi^*\left(\frac{\xi - x}{T}\right)\right) \;\; \forall (t,x) \in \mathbb{R}_+^* \times X \backslash \{\xi\} \tag{240}$$

Proof — The Lax-Hopf formula (222) associated with the upstream boundary condition reads:

$$\boldsymbol{M}_{\gamma_j}(t,x) \;=\; \inf_{(u,T) \in \mathrm{Dom}(\varphi^*) \times \mathbb{R}_+ \text{ such that } x + Tu = \xi \text{ and } \overline{\gamma}_j \leq t - T \leq \overline{\gamma}_{j+1}} \left(c_j(t-T) + d_j + T\varphi^*(u)\right) \tag{241}$$

We define the variable change $T := \frac{\xi - x}{u} > 0$, which represents the capture time using the control u (see [104]). Since $T = \frac{\xi - x}{u} > 0$ and $x > \xi$, we have $u < 0$. The constraint $u \in \mathrm{Dom}(\varphi^*) := [-\nu^{\flat}, \nu^{harp}]$ thus implies $T \in [-\frac{\xi - x}{\nu^{\flat}}, +\infty[$. The additional constraint $t - \frac{\xi - x}{u} \in [\overline{\gamma}_j, \overline{\gamma}_{j+1}]$ results from the definition of $\gamma_j(\cdot,\cdot)$ and implies $T \in [t - \overline{\gamma}_{j+1}, t - \overline{\gamma}_j]$, which yields equation (240). ■

Proposition 9. [Domain of influence of an affine upstream boundary condition] The domain of definition of $\boldsymbol{M}_{\gamma_j}(\cdot,\cdot)$ is given by the following formula:

$$\mathrm{Dom}(\boldsymbol{M}_{\gamma_j}) = \left\{(t,x) \in \mathbb{R}_+ \times X \;\; \text{such that} \;\; x \leq \xi + \nu^{\flat}(t - \overline{\gamma}_j)\right\} \tag{242}$$

Proof — The Lax-Hopf formula (241) implies:

$$\mathrm{Dom}(\boldsymbol{M}_{\gamma_j}) := \left\{(t,x) \in \mathbb{R}_+ \times X \;\; \text{such that} \;\; \exists T \in \left[-\frac{\xi - x}{\nu^{\flat}}, +\infty\right[\cap \left[t - \overline{\gamma}_{j+1}, t - \overline{\gamma}_j\right]\right\}$$

Hence, $(t,x) \in \mathrm{Dom}(\boldsymbol{M}_{\gamma_j})$ if and only if $-\frac{\xi - x}{\nu^{\flat}} \leq t - \overline{\gamma}_j$, which in turn implies equation (242). ■

Definition 10. [Auxiliary objective function] For all $(t,x) \in \mathrm{Dom}(\boldsymbol{M}_{\gamma_j})$, we define an objective function $\eta_{c_j,d_j,t,x}(\cdot)$ by the following formula:

$$\forall T \in \mathbb{R}_+^* \quad \eta_{c_j,d_j,t,x}(T) := c_j(t-T) + d_j + T\varphi^*\left(\frac{\xi - x}{T}\right) \tag{243}$$

Given this definition, equation (240) becomes:

$$\boldsymbol{M}_{\gamma_j}(t,x) \;=\; \inf_{T \in \left[-\frac{\xi - x}{\nu^\flat}, +\infty\right[\cap \left[t-\overline{\gamma}_{j+1}, t-\overline{\gamma}_j\right]} \eta_{c_j,d_j,t,x}(T) \tag{244}$$

Since $\varphi^*(\cdot)$ is convex, its associated perspective function $T \to T\varphi^*(\frac{\xi-x}{T})$ is also convex [78] for $T > 0$. Hence the function $\eta_{c_j,d_j,t,x}(\cdot)$ is convex as the sum of two convex functions. The subderivative of $\eta_{c_j,d_j,t,x}(\cdot)$ is given by:

$$\begin{aligned}
&\forall T \in [-\frac{\xi - x}{\nu^\flat}, +\infty[, \quad \partial_- \eta_{c_j,d_j,t,x}(T) \\
&= \left\{ w \mid \exists v \in \partial_- \varphi^*(\frac{\xi - x}{T}), \quad w = -c_j + \varphi^*(\frac{\xi - x}{T}) - \frac{\xi - x}{T} v \right\} \\
&:= \left\{ -c_j + \varphi^*(\frac{\xi - x}{T}) \right\} - \frac{\xi - x}{T} \partial_- \varphi^*(\frac{\xi - x}{T})
\end{aligned} \tag{245}$$

with a slight abuse of notation for the second line as previously.

Definition 11. [Density associated with c_j] Recalling that $\mathrm{Im}(\psi) := [0,\delta]$, we define ρ_c as:

$$\rho_c = \inf_{\rho \in [0,\omega] \text{ such that } \psi(\rho) = \delta} \rho$$

Since $c_j \in \mathrm{Im}(\psi) = [0,\delta]$, there exists $\rho_j \in [0,\rho_c]$ such that $\psi(\rho_j) = c_j$. Note that since $\psi(\cdot)$ is concave and $\delta > 0$, $\psi(\cdot)$ is increasing on $[0,\rho_c]$ and thus $\partial_+\psi(\rho_j) \cap \mathbb{R}_+bb\neq \emptyset$.

- Let $u_0(\rho_j)$ be an element of $-\partial_+\psi(\rho_j) \cap \mathbb{R}_- \neq \emptyset$.
- Let $T_0(\rho_j, x)$ be defined as

$$T_0(\rho_j, x) := \begin{cases} \dfrac{\xi - x}{u_0(\rho_j)} & \text{if } u_0(\rho_j) \neq 0 \\ +\infty & \text{if } u_0(\rho_j) = 0 \end{cases} \tag{246}$$

We have by the Legendre-Fenchel inversion formula that $u_0(\rho_j) \in \mathrm{Dom}(\varphi^*)$ and $\rho_j \in \partial_- \varphi^*(u_0(\rho_j))$.

Proposition 12. [Explicit minimization of $\eta_{c_j,d_j,t,x}(\cdot)$] Let $T_0(\rho_j, x)$ be given by definition 11. For all $(t,x) \in \mathrm{Dom}(\boldsymbol{M}_{\gamma_j})$, the function $\eta_{c_j,d_j,t,x}(\cdot)$ has the following minimizer over $[-\frac{\xi - x}{\nu^\flat}, \infty[\cap [t - \overline{\gamma}_{j+1}, t - \overline{\gamma}_j]$:

$$\begin{cases} T_0(\rho_j, x) & \text{if } T_0(\rho_j, x) \in [t - \overline{\gamma}_{j+1}, t - \overline{\gamma}_j] \\ t - \overline{\gamma}_j & \text{if } t - \overline{\gamma}_j \leq T_0(\rho_j, x) \\ t - \overline{\gamma}_{j+1} & \text{if } T_0(\rho_j, x) \leq t - \overline{\gamma}_{j+1} \end{cases} \tag{247}$$

Proof — The function $\eta_{c_j,d_j,t,x}(\cdot)$ is minimal for a given $T > 0$ if and only if $0 \in \partial_- \eta_{c_j,d_j,t,x}(T)$ by [78]. Since $u_0(\rho_j) \in -\partial_+ \psi(\rho_j) \cap \mathbb{R}_-$, we have by the Legendre-Fenchel inversion formula that $\rho_j \in \partial_- \varphi^*(u_0(\rho_j))$. This last formula implies $0 \in \partial_-\big(\varphi^*(\cdot) - \cdot\, \rho_j\big)\big(u_0(\rho_j)\big)$ and thus that:

$$\psi(\rho_j) = \inf_{u \in \mathrm{Dom}(\varphi^*)} \big[\varphi^*(u) - \rho_j u\big] = \varphi^*(u_0(\rho_j)) - \rho_j u_0(\rho_j)$$

The property $c_j = \psi(\rho_j)$ implies that $-c_j + \varphi^*\big(u_0(\rho_j)\big) = \rho_j u_0(\rho_j)$ by the previous formula. Equation (245) thus implies:

$$\begin{aligned} \partial_- \eta_{c_j,d_j,t,x}\left(\frac{\xi - x}{u_0(\rho_j)}\right) &= \big\{w \mid \exists v \in \partial_- \varphi^*\big(u_0(\rho_j)\big),\ \ w = \rho_j u_0(\rho_j) - u_0(\rho_j) v\big\} \\ &:= \big\{\rho_j u_0(\rho_j)\big\} - u_0(\rho_j) \partial_- \varphi^*\big(u_0(\rho_j)\big) \end{aligned} \tag{248}$$

Since $\rho_j \in \partial_- \varphi^*\big(u_0(\rho_j)\big)$, this last property implies that $0 \in \partial_- \eta_{c_j,d_j,t,x}(\frac{\xi - x}{u_0(\rho_j)})$. Hence, $T_0(\rho_j, x) := \frac{\xi - x}{u_0(\rho_j)}$ minimizes the convex function $\eta_{c_j,d_j,t,x}(\cdot)$ over $\mathbb{R}_+^*$.

Since $\eta_{c_j,d_j,t,x}(\cdot)$ is convex, it is decreasing for $T \leq T_0(\rho_j, x)$ and increasing for $T \geq T_0(\rho_j, x)$. The values of the capture time T which minimize $\eta_{c_j,d_j,t,x}(T)$ over $[-\frac{\xi - x}{\nu^\flat}, \infty[\cap [t - \overline{\gamma}_{j+1}, t - \overline{\gamma}_j]$ are thus given by equation (247). Note that the property $u_0(\rho_j) \in [-\nu^\flat, 0]$ implies $-\frac{\xi - x}{\nu^\flat} \leq T_0(\rho_j, x)$. Note also that since $(t,x) \in \mathrm{Dom}(\boldsymbol{M}_{\gamma_j})$, we have $-\frac{\xi - x}{\nu^\flat} \leq t - \overline{\gamma}_j$. ■

Proposition 13. [Computation of $\boldsymbol{M}_{\gamma_j}(\cdot,\cdot)$] For all $(t,x) \in \mathrm{Dom}(\boldsymbol{M}_{\gamma_j})$, the solution $\boldsymbol{M}_{\gamma_j}(t,x)$ can be computed using the following formula:

$$\boldsymbol{M}_{\gamma_j}(t,x) = \begin{cases} (i) & t\psi(\rho_j) + \rho_j(\xi - x) + d_j & \text{if } T_0(\rho_j, x) \in \left[t - \overline{\gamma}_{j+1}, t - \overline{\gamma}_j\right] \\ (ii) & \psi(\rho_j)\overline{\gamma}_j + d_j + \left(t - \overline{\gamma}_j\right)\varphi^*\left(\dfrac{\xi - x}{t - \overline{\gamma}_j}\right) & \text{if } t - \overline{\gamma}_j \leq T_0(\rho_j, x) \\ (iii) & \psi(\rho_j)\overline{\gamma}_{j+1} + d_j + \left(t - \overline{\gamma}_{j+1}\right)\varphi^*\left(\dfrac{\xi - x}{t - \overline{\gamma}_{j+1}}\right) & \text{if } T_0(\rho_j, x) \leq t - \overline{\gamma}_{j+1} \end{cases} \tag{249}$$

Proof — The cases (ii) and (iii) in equation (249) are trivially obtained by combining equations (244) and (247). The case (i) in equation (249) is obtained by combining (243), (247) and observing that $\varphi^*\left(\frac{\xi - x}{T_0(\rho_j, x)}\right) = \psi(\rho_j) + \frac{\xi - x}{T_0(\rho_j, x)}\rho_j$. ■

Remark 8. Equation (249) can also be obtained from equation (273), observing that the affine upstream boundary condition (239) can be viewed as an affine internal condition of the form (263), where:

$$\begin{cases} \overline{\delta}_l = \overline{\gamma}_j \\ \overline{\delta}_{l+1} = \overline{\gamma}_{j+1} \\ x_l = \xi \\ v_l = 0 \\ g_l = c_j \\ h_l = c_j\overline{\gamma}_j + d_j \end{cases} \tag{250}$$

Figure 5.2 illustrates the different domains of equation (249) for the solution associated with an affine upstream condition defined by equation (239).

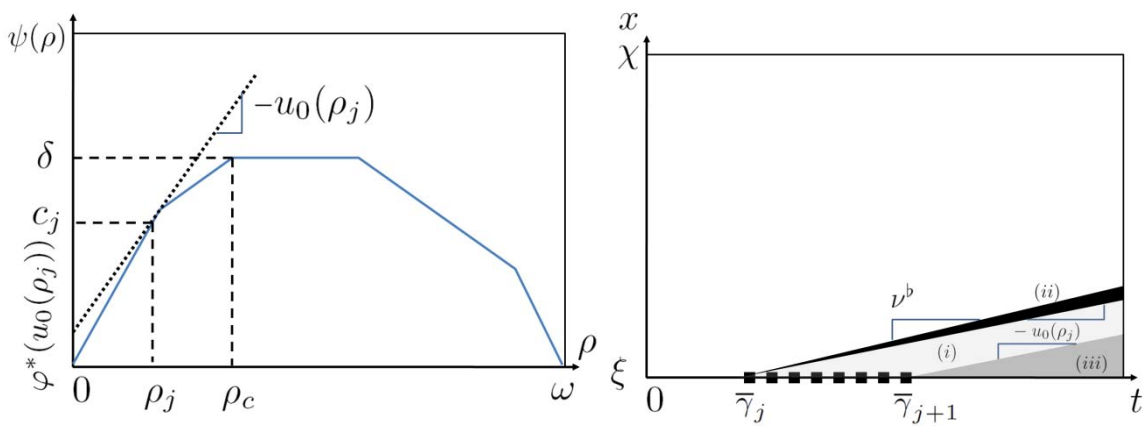


Figure 5.2. Construction of the solution associated with an affine upstream boundary condition.

Left: Illustration of the construction of a $u_0(\rho_j)$ from a known c_j. The transform $\varphi^*(u_0(\rho_j))$ corresponds to the value intercepted on the vertical axis by the tangent line of slope $-u_0(\rho_j)$ to the graph of ψ in ρ_j. **Right:** The (t,x) domain of the solution corresponding to the affine upstream boundary condition (239) can be separated in three different areas. The domain highlighted in light gray corresponds to the case (i) in equation (249). The domain highlighted in dark gray corresponds to the case (ii) and the remaining domain in medium gray corresponds to the case(iii). The domain of the upstream boundary condition is represented by a dashed line.

25.5.3. Analytic Lax-Hopf formula associated with an affine downstream boundary condition

Definition 14. [Affine downstream boundary condition] We consider the following downstream boundary condition $\beta_k(t,\chi)$ of $\beta(t,x)$:

$$\beta_k(t,x) = \begin{cases} e_k t + f_k & \text{if } \ t \in [\overline{\beta}_k, \overline{\beta}_{k+1}] \\ +\infty & \text{otherwise} \end{cases} \tag{251}$$

In the following computation, we consider that $x < \chi$. We also assume that the value condition (251) satisfies the condition $e_k \in \mathrm{Im}(\psi) = [0,\delta]$.

Proposition 15. [Lax-Hopf formula for an affine downstream boundary condition] The Lax-Hopf formula (252) associated with the downstream boundary condition (251) can be expressed as:

$$\boldsymbol{M}_{\beta_k}(t,x) \ = \inf_{T \in \left[\frac{\chi - x}{\nu^\#}, +\infty\right[\cap [t-\overline{\beta}_{k+1}, t-\overline{\beta}_k]} \left(e_k(t-T) + f_k + T\varphi^*\left(\frac{\chi - x}{T}\right)\right) \tag{252}$$

$$\forall (t,x) \in \mathbb{R}_+^* \times X$$

Proof — The Lax-Hopf formula (222) associated with the affine downstream boundary condition can be written as:

$$\boldsymbol{M}_{\beta_k}(t,x) = \inf_{(u,T)\in \mathrm{Dom}(\varphi^*)\times\mathbb{R}_+ \text{ such that } x+Tu=\chi \text{ and } \overline{\beta}_k \le t-T \le \overline{\beta}_{k+1}} \left(e_k(t-T) + f_k + T\varphi^*(u)\right) \tag{253}$$

We define the variable change $T := \frac{\chi - x}{u}$, which represents the capture time using the control u. Since $T = \frac{\chi - x}{u} > 0$ and $x < \chi$, we have $u > 0$. The constraint $u \in \mathrm{Dom}(\varphi^*) := [-\nu^\flat, \nu^\#]$ implies $T \in [\frac{\chi - x}{\nu^\#}, +\infty[$. The additional constraint $t - \frac{\chi - x}{u} \in [\overline{\beta}_k, \overline{\beta}_{k+1}]$ can be expressed as $T \in [t - \overline{\beta}_{k+1}, t - \overline{\beta}_k]$, which yields equation (252). ■

Proposition 16. [Domain of influence of a downstream boundary condition] The domain of definition of $\boldsymbol{M}_{\beta_k}(\cdot,\cdot)$ is given by the following formula:

$$\mathrm{Dom}(\boldsymbol{M}_{\beta_k}) = \{(t,x) \in \mathbb{R}_+ \times X \text{ such that } x \ge \chi - \nu^\#(t - \overline{\beta}_k)\} \tag{254}$$

Proof — The Lax-Hopf formula (253) implies:

$$\mathrm{Dom}(\boldsymbol{M}_{\beta_k}) := \left\{(t,x) \in \mathbb{R}_+ \times X \text{ such that } \exists T \in \left[\frac{\chi - x}{\nu^\#}, +\infty\right[\cap \left[t - \overline{\beta}_{k+1}, t - \overline{\beta}_k\right]\right\}$$

Hence, $(t,x) \in \mathrm{Dom}(\boldsymbol{M}_{\beta_k})$ if and only if $\frac{\chi - x}{\nu^\#} \le t - \overline{\beta}_k$, which in turn implies equation (254). ■

Definition 17. [Auxiliary objective function] For all $(t,x) \in \mathrm{Dom}(\boldsymbol{M}_{\beta_k})$, we define an objective function $\theta_{e_k,f_k,t,x}(\cdot)$ by:

$$\forall T \in \mathbb{R}_+^* \quad \theta_{e_k,f_k,t,x}(T) := e_k(t-T) + f_k + T\varphi^*\left(\frac{\chi - x}{T}\right) \tag{255}$$

Given this definition, equation (252) becomes:

$$\boldsymbol{M}_{\beta_k}(t,x) \;=\; \inf_{T\in\left[\frac{\chi-x}{\nu^{\#}},+\infty\right[\cap[t-\overline{\beta}_{k+1},t-\overline{\beta}_k]} \theta_{e_k,f_k,t,x}(T) \tag{256}$$

Since $\varphi^*(\cdot)$ is convex, its associated perspective function $T \to T\varphi^*(\frac{\chi-x}{T})$ is also convex [78] for $T > 0$. Hence the function $\theta_{e_k,f_k,t,x}(\cdot)$ is convex as the sum of two convex functions.The subderivative of $\theta_{e_k,f_k,t,x}(\cdot)$ is given by:

$$\begin{aligned}
&\forall T \in [\frac{\chi-x}{\nu^{\#}},+\infty[, \quad \partial_-\theta_{e_k,f_k,t,x}(T) \\
&= \left\{w \mid \exists v \in \partial_-\varphi^*(\frac{\chi-x}{T}),\;\; w = -e_k + \varphi^*(\frac{\chi-x}{T}) - \frac{\chi-x}{T}v\right\} \\
&:= \left\{-e_k + \varphi^*(\tfrac{\chi-x}{T})\right\} - \tfrac{\chi-x}{T}\partial_-\varphi^*(\tfrac{\chi-x}{T})
\end{aligned} \tag{257}$$

with a slight abuse of notation for the second line as in the previous two sections.

Definition 18. [Density associated with e_k] Recalling that $\mathrm{Im}(\psi) = [0,\delta]$, we define ρ_c as:

$$\rho_c = \sup_{\rho\in[0,\omega] \text{ such that } \psi(\rho)=\delta} \rho$$

Since $e_k \in \mathrm{Im}(\psi) = [0,\delta]$, there exists $\rho_k \in [\rho_c,\omega]$ such that $\psi(\rho_k) = e_k$. Note that since $\psi(\cdot)$ is concave and $\delta > 0$, $\psi(\cdot)$ is decreasing on $[\rho_c,\omega]$ and thus $\partial_+\psi(\rho_k) \cap \mathbb{R}_- \neq \emptyset$.

- Let $u_0(\rho_k)$ be an element of $-\partial_+\psi(\rho_k) \cap \mathbb{R}_+$
- Let $T_0(\rho_k,x)$ be defined as

$$T_0(\rho_k,x) := \begin{cases} \dfrac{\xi-x}{u_0(\rho_k)} & \text{if } u_0(\rho_k) \neq 0 \\ +\infty & \text{if } u_0(\rho_k) = 0 \end{cases} \tag{258}$$

We have by the Legendre-Fenchel inversion formula that $\rho_k \in \partial_-\varphi^*(u_0(\rho_k))$, which implies that $u_0(\rho_k) \in \mathrm{Dom}(\varphi^*)$.

Remark 9. Note that the definition of ρ_c differs from the previous section for functions $\psi(\cdot)$ which are not strictly concave. This is sometimes referred as "lower critical density" (section 25.5.2) and "upper critical density" (section 25.5.3), but we have kept the same notation since the two corresponding densities are only intermediate variables in our derivations.

Proposition 19. [Explicit minimization of $\theta_{e_k,f_k,t,x}(\cdot)$] For all $(t,x) \in \mathrm{Dom}(\boldsymbol{M}_{\beta_k})$, the function $\theta_{e_k,f_k,t,x}(\cdot)$ has the following minimizer over $[\frac{\chi-x}{\nu^{\#}},+\infty[\cap[t-\overline{\beta}_{k+1},t-\overline{\beta}_k]$:

$$\begin{cases} T_0(\rho_k, x) & \text{if } \; T_0(\rho_k, x) \in [t - \overline{\beta}_{k+1}, t - \overline{\beta}_k] \\ t - \overline{\beta}_k & \text{if } \; t - \overline{\beta}_k \leq T_0(\rho_k, x) \\ t - \overline{\beta}_{k+1} & \text{if } \; T_0(\rho_k, x) \leq t - \overline{\beta}_{k+1} \end{cases} \tag{259}$$

Proof — The function $\theta_{e_k,f_k,t,x}(\cdot)$ is minimal for a given $T > 0$ if and only if $0 \in \partial_- \theta_{e_k,f_k,t,x}(T)$ by [78]. Since $u_0(\rho_k) \in -\partial_+ \psi(\rho_k)$, we have by the Legendre-Fenchel inversion formula that $\rho_k \in \partial_- \varphi^*(u_0(\rho_k))$. In the exact same way as the previous section, this last formula implies that $0 \in \partial_-(\varphi^*(\cdot) -\cdot\, \rho_k)(u_0(\rho_k))$ and thus that:

$$\psi(\rho_k) = \inf_{u \in \mathrm{Dom}(\varphi^*)} [\varphi^*(u) - \rho_k u] = \varphi^*(u_0(\rho_k)) - \rho_k u_0(\rho_k)$$

The property $e_k = \psi(\rho_k)$ implies that $-e_k + \varphi^*(u_0(\rho_k)) = \rho_k u_0(\rho_k)$. Equation (257) thus implies:

$$\begin{aligned} \partial_- \theta_{e_k,f_k,t,x}\left(\frac{\chi - x}{u_0(\rho_k)}\right) &= \{w \mid \exists v \in \partial_- \varphi^*(u_0(\rho_k)), \;\; w = \rho_k u_0(\rho_k) - u_0(\rho_k) v\} \\ &:= \{\rho_k u_0(\rho_k)\} - u_0(\rho_k) l_- \varphi^*(u_0(\rho_k)) \end{aligned} \tag{260}$$

with the same abuse of notation for the second line.

Since $\rho_k \in \partial_- \varphi^*(u_0(\rho_k))$, this last property implies that $0 \in \partial_- \theta_{e_k,f_k,t,x}(\frac{\chi - x}{u_0(\rho_k)})$. Hence, $T_0(\rho_k, x) := \frac{\chi - x}{u_0(\rho_k)}$ minimizes the convex function $\theta_{e_k,f_k,t,x}(\cdot)$ over $\mathbb{R}_+^*$.

Since $\theta_{e_k,f_k,t,x}(\cdot)$ is convex, it is decreasing for $T < T_0(\rho_k, x)$ and increasing for $T > T_0(\rho_k, x)$. The values of the capture time T which minimize $\theta_{e_k,f_k,t,x}(T)$ over $[\frac{\chi - x}{\nu^\#}, +\infty[\cap [t - \overline{\beta_{k+1}}, t - \overline{\beta}_k]$ are thus given by equation (259). Note that the property $u_0(\rho_k) \in [0, \nu^\#]$ implies $\frac{\chi - x}{\nu^\#} \leq T_0(\rho_k, x)$. Note also that since $(t, x) \in \mathrm{Dom}(\boldsymbol{M}_{\beta_k})$, we have $\frac{\chi - x}{\nu^\#} \leq t - \overline{\beta}_k$. ■

Proposition 20. [Computation of $\boldsymbol{M}_{\beta_k}(\cdot,\cdot)$] For all $(t,x) \in \mathrm{Dom}(\boldsymbol{M}_{\beta_k})$, the expression $\boldsymbol{M}_{\beta_k}(t,x)$ can be computed using the following formula:

$$\boldsymbol{M}_{\beta_k}(t,x) = \begin{cases} (i) & t\psi(\rho_k) + \rho_k(\chi - x) + f_k & \text{if } T_0(\rho_k,x) \in [t-\overline{\beta}_{k+1}, t-\overline{\beta}_k] \\ (ii) & \psi(\rho_k)\overline{\beta}_k + f_k + (t-\overline{\beta}_k)\varphi^*(\dfrac{\chi - x}{t-\overline{\beta}_k}) & \text{if } t-\overline{\beta}_k \leq T_0(\rho_k,x) \\ (iii) & \psi(\rho_k)\overline{\beta}_{k+1} + f_k + (t-\overline{\beta}_{k+1})\varphi^*(\dfrac{\chi - x}{t-\overline{\beta}_{k+1}}) & \text{if } T_0(\rho_k,x) \leq t-\overline{\beta}_{k+1} \end{cases} \tag{261}$$

Proof — The cases (ii) and (iii) in equation (261) are trivially obtained by combining equations (256) and (259). The case (i) in equation (261) is obtained by combining (256), (259) and observing that $\varphi^*\left(\frac{\chi-x}{T_0(\rho_k,x)}\right) = \psi(\rho_k) + \frac{\chi-x}{T_0(\rho_k,x)}\rho_k$. ■

Remark 10. Equation (261) can be obtained from equation (274), observing that the affine downstream boundary condition (251) can be viewed as an affine internal condition of the form (263), where:

$$\begin{cases} \overline{\delta}_l = \overline{\beta}_k \\ \overline{\delta}_{l+1} = \overline{\beta}_{k+1} \\ x_l = \chi \\ v_l = 0 \\ g_l = e_k \\ h_l = e_k\overline{\beta}_k + f_k \end{cases} \tag{262}$$

Figure 5.3 illustrates the different domains of equation (261) for the solution associated with an affine downstream boundary condition defined by equation (251).

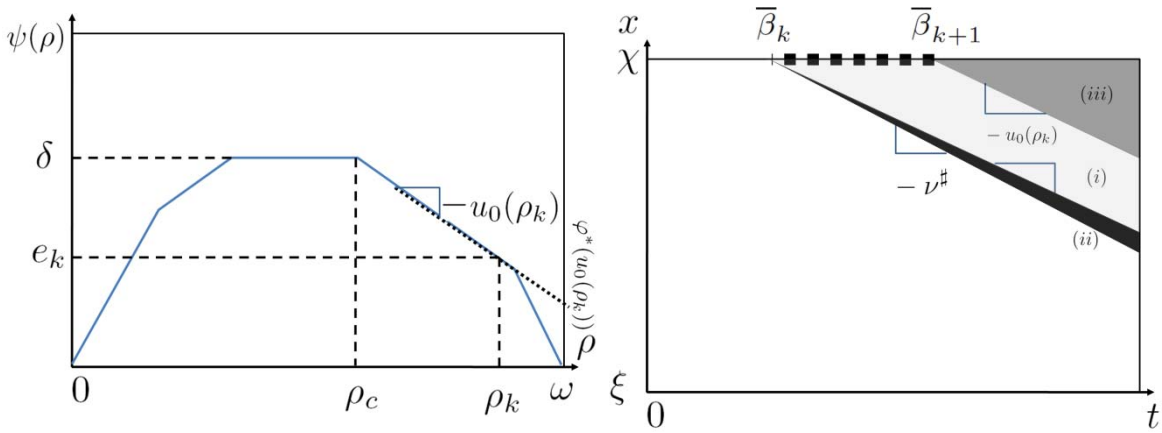


Figure 5.3. Construction of the solution associated with an affine downstream boundary condition.

Left: Illustration of the construction of a $u_0(\rho_k)$ from a known e_k. The transform $\varphi^*(u_0(\rho_k))$ corresponds to the value intercepted on the vertical axis by the tangent line of slope $-u_0(\rho_k)$ to the graph of ψ in ρ_k. **Right:** The (t,x) domain of the solution corresponding to the affine downstream boundary condition (251) can be separated in three different areas. The domain highlighted in light gray corresponds to the case (i) in equation (261). The domain highlighted in dark gray corresponds to the case (ii) and the remaining domain in medium gray corresponds to the case (iii). The domain of the downstream boundary condition is represented by a dashed line.

25.5.4. Analytic Lax-Hopf formula associated with an affine internal condition

The previous section explained how to compute the solution to affine initial and boundary conditions. We now treat the problem of internal conditions using a similar approach. As will appear in this section, the algebra involved in doing this mathematical construction is more involved than the previous case.

Definition 21. [Affine internal condition] We consider the following affine internal condition $\mu_l(\cdot,\cdot)$, where l is an integer:

$$\mu_l(t,x) = \begin{cases} g_l(t-\overline{\delta}_l) + h_l & \text{if } \; x = x_l + v_l(t-\overline{\delta}_l) \\ & \text{and } \; t \in [\overline{\delta}_l, \overline{\delta}_{l+1}] \\ +\infty & \text{otherwise} \end{cases} \tag{263}$$

For the computation of the corresponding solution $\boldsymbol{M}_{\mu_l}(t,x)$, we assume that (t,x) satisfy $x \neq x_l + v_l(t-\overline{\delta}_l)$. In addition, we assume that the constants g_l and v_l satisfy $0 \leq g_l \leq \varphi^*(-v_l)$.

Proposition 22. [Lax-Hopf formula for affine internal condition] The Lax-Hopf formula (264) associated with the internal boundary condition (263) can be expressed as:

$$\boldsymbol{M}_{\mu_l}(t,x) = \inf_{T\in\mathbb{R}_+\cap[t-\overline{\delta}_{l+1},t-\overline{\delta}_l]} \left(g_l(t-T-\overline{\delta}_l) + h_l + T\varphi^*\left(\frac{x_l + v_l(t-\overline{\delta}_l - T) - x}{T}\right)\right) \tag{264}$$

Proof — The Lax-Hopf formula (222) associated with an affine internal condition reads:

$$\boldsymbol{M}_{\mu_l}(t,x) = \inf_{(u,T)\in\mathrm{Dom}(\varphi^*)\times\mathbb{R}_+ \text{ such that } x+Tu=x_l+v_l(t-T-\overline{\delta}_l) \text{ and } \overline{\delta}_l\le t-T\le\overline{\delta}_{l+1}} \left(g_l\left(t-T-\overline{\delta}_l\right) + h_l + T\varphi^*(u)\right) \tag{265}$$

Since $x + Tu = x_l + v_l(t - T - \overline{\delta}_l)$, we have $u = \frac{x_l+v_l(t-T-\overline{\delta}_l)-x}{T}$. In addition, the constraint $\overline{\delta}_l \le t - T \le \overline{\delta}_{l+1}$ can be written as $T \in [t - \overline{\delta}_{l+1}, t - \overline{\delta}_l]$, which yields (264). ■

The solution to the affine internal condition has a domain of definition, which can be computed analytically as follows.

Proposition 23. [Domain of influence of an affine internal condition] The domain of definition of $\boldsymbol{M}_{\mu_l}(\cdot,\cdot)$ is given by the following formula:

$$\mathrm{Dom}(\boldsymbol{M}_{\mu_l}) = \{(t,x) \in \mathbb{R}_+ \times X \text{ such that } t \ge \overline{\delta}_l \text{ and } x_l - \nu^{\#}(t-\overline{\delta}_l) \le x \le x_l + \nu^{\flat}(t-\overline{\delta}_l)\} \tag{266}$$

Proof — The Lax-Hopf formula (264) implies:

$$\mathrm{Dom}(\boldsymbol{M}_{\mu_l}) := \Big\{(t,x) \in \mathbb{R}_+ \times X \text{ s.t. } \exists T \in \mathbb{R}_+^* \cap [t-\overline{\delta}_{l+1}, t-\overline{\delta}_l] \text{ and } \frac{x_l + v_l(t-\overline{a}_l - T) - x}{T} \in \mathrm{Dom}(\varphi^*)\Big\}$$

Since $T > 0$, the condition $\frac{x_l+v_l(t-\overline{\delta}_l-T)-x}{T} \in \mathrm{Dom}(\varphi^*) = [-\nu^{\flat}, \nu^{\#}]$ is equivalent to $T \geq \frac{x_l+v_l(t-\overline{\delta}_l)-x}{\nu^{\#}+v_l}$ and $T \geq \frac{x_l+v_l(t-\overline{\delta}_l)-x}{-\nu^{\flat}+v_l}$. Hence, $(t,x) \in \mathrm{Dom}(\varphi^*)$ if and only if the set $\mathbb{R}_+^* \cap [t-\overline{\delta}_{l+1}, t-\overline{\delta}_l] \cap [\frac{x_l+v_l(t-\overline{\delta}_l)-x}{\nu^{\#}+v_l}, +\infty[\cap [\frac{x_l+v_l(t-\overline{\delta}_l)-x}{-nu^{\flat}+v_l}, +\infty[$ is not empty, which implies

$$\max\left(0, \frac{x_l+v_l(t-\overline{\delta}_l)-x}{-\nu^{\flat}+v_l}, \frac{x_l+v_l(t-\overline{\delta}_l)-x}{\nu^{\#}+v_l}\right) \leq t-\overline{\delta}_l$$

This last inequality implies equation (266). ■

The method followed next also makes use of an auxiliary objective function, which is later used to explicitly find the minimizer.

Definition 24. [Auxiliary objective function] For all $(t,x) \in \mathrm{Dom}(\boldsymbol{M}_{\mu_l})$, we define the function $\kappa_{\overline{\delta}_l,g_l,h_l,x_l,v_l,t,x}(\cdot)$ as:

$$\forall T \in \mathbb{R}_+^*, \quad \kappa_{\overline{\delta}_l,g_l,h_l,x_l,v_l,t,x}(T) := \left(g_l(t-T-\overline{\delta}_l)+h_l+T\varphi^*\left(\frac{x_l+v_l(t-\overline{\delta}_l-T)-x}{T}\right)\right) \tag{267}$$

Given this definition, equation (264) becomes:

$$\boldsymbol{M}_{\mu_l}(t,x) = \inf_{T\in\left[\max\left(0,\frac{x_l+v_l(t-\overline{\delta}_l)-x}{-\nu^{\flat}+v_l},\frac{x_l+v_l(t-\overline{\delta}_l)-x}{\nu^{\#}+v_l},t-\overline{\delta}_{l+1}\right),t-\overline{\delta}_l\right]} \kappa_{\overline{\delta}_l,g_l,h_l,x_l,v_l,t,x}(T) \tag{268}$$

Since $\varphi^*(\cdot)$ is convex, the function $h: u \to \varphi^*(u-v_l)$ is convex and its associated perspective function $T \to Th(\frac{x_l+v_l(t-\overline{\delta}_l)-x}{T})$ is also convex for $T > 0$ by [78]. Hence thefunction $\kappa_{\overline{\delta}_l,g_l,h_l,x_l,v_l,t,x}(\cdot)$ is convex as the sum of two convex functions. The subderivative of $\kappa_{\overline{\delta}_l,g_l,h_l,x_l,v_l,t,x}(\cdot)$ is given by:

$$\forall T \in \left[\max\left(0,\frac{x_l+v_l(t-\overline{\delta}_l)-x}{-\nu^{\flat}+v_l},\frac{x_l+v_l(t-\overline{\delta}_l)-x}{\nu^{\#}+v_l},t-\overline{\delta}_{l+1}\right),t-\overline{\delta}_l\right], \quad \partial_-\kappa_{\overline{\delta}_l,g_l,h_l,x_l,v_l,t,x}(T) =$$
$$\left\{w \mid \exists v \in \partial_-\varphi^*\left(\frac{x_l+v_l(t-\overline{\delta}_l)-x}{T}-v_l\right), w = -g_l+\varphi^*\left(\frac{x_l+v_l(t-\overline{\delta}_l)-x}{T}-v_l\right)-\frac{x_l+v_l(t-\overline{\delta}_l)-x}{T}v\right\} \tag{269}$$

which can be written using a slight abuse of notation as:

$$\partial_{-}\kappa_{\overline{\delta}_l,g_l,h_l,x_l,v_l,t,x}(T) := \quad -g_l + \varphi^*\left(\frac{x_l + v_l(t-\overline{\delta}_l) - x}{T} - v_l\right) - \frac{x_l + v_l(t-\overline{\delta}_l) - x}{T}\partial_{-}\varphi^*\left(\frac{x_l + v_l(t-\overline{\delta}_l) - x}{T} - v_l\right) \tag{270}$$

Because of the higher complexity of this case, we need to define intermediate quantities used in the explicit minimization.

Definition 25. [Densities associated with v_l and g_l]

- We define the function $f_{v_l}(\cdot)$ as $f_{v_l}: \rho \to \psi(\rho) - \rho v_l$. The function f_{v_l} is concave as the sum of concave functions and attains its maximum value $\varphi^*(-v_l)$ (by definition of the function $\varphi^*(.)$) for a given $\rho := \rho_l$.
- Note that since $v_l \in [0, v^\flat[$, the function f_{v_l} satisfies $f_{v_l}(0) = 0$ and $f_{v_l}(\omega) \leq 0$. By assumption, we also have $g_l \leq \varphi^*(-v_l)$ and since $f_{v_l}(\cdot)$ is concave and continuous, there exist two solutions $\rho_1(v_l, g_l) \in [0, \rho_l]$ and $\rho_2(v_l, g_l) \in [\rho_l, \omega]$ such that $f_{v_l}(\rho_p(v_l, g_l)) = g_l$ for $p \in \{1,2\}$ (see Figure 5.4).
- For $p \in \{1,2\}$, we also define $u_p(v_l, g_l)$ as elements of $-\partial_{+}\psi(\rho_p(v_l, g_l))$. Note that since f_{v_l} is concave, it is increasing on $[0, \rho_l]$ and decreasing on $[\rho_l, \omega]$, which implies that $u_1(v_l, g_l) \leq -v_l$ and $u_2(v_l, g_l) \geq -v_l$. Note also that the Legendre-Fenchel inversion formula implies that $u_p(v_l, g_l) \in \mathrm{Dom}(\varphi^*)$ for $p \in \{1,2\}$.

Definition 26. [Capture times associated with $u_p(v_l, g_l)$, for $p \in \{1, 2\}$]

- We define $T_p(t, x, v_l, g_l)$ for $p \in \{1,2\}$ as:

$$T_p(t, x, v_l, g_l) := \begin{cases} \dfrac{x_l + v_l(t-\overline{\delta}_l) - x}{u_p(v_l, g_l) + v_l} & \text{if} \quad u_p(v_l, g_l) \neq -v_l \\ +\infty & \text{if} \quad u_p(v_l, g_l) = -v_l \end{cases} \tag{271}$$

- The definition of $T_p(\cdot,\cdot,\cdot,\cdot)$ implies that $T_1(t, x, v_l, g_l) \geq 0$ if and only if $x_l + v_l(t-\overline{\delta}_l) - x \leq 0$ and that $T_2(t, x, v_l, g_l) \geq 0$ if and only if $x_l + v_l(t-\overline{\delta}_l) - x \geq 0$.
- Note also that since $u_p(v_l, g_l) \in [-v^\flat, v^\#]$, we have

$T_1(t,x,v_l,g_l) \geq \frac{x_l+v_l(t-\overline{\delta}_l)-x}{-\nu^{\flat}+v_l}$ when $x_l + v_l(t-\overline{\delta}_l) - x \leq 0$ and $T_2(t,x,v_l,g_l) \geq \frac{x_l+v_l(t-\overline{\delta}_l)-x}{\nu^{\#}+v_l}$ when $x_l + v_l(t-\overline{\delta}_l) - x \geq 0$.

The previous definitions can now be used to compute the explicit minimizer.

Proposition 27. [Explicit minimization of $\kappa_{\overline{\delta}_l,g_l,h_l,x_l,v_l,t,x}(\cdot)$] For all $(t,x) \in \mathrm{Dom}(\boldsymbol{M}_{\mu_l})$, the function $\kappa_{\overline{\delta}_l,g_l,h_l,x_l,v_l,t,x}(\cdot)$ has the following minimizer over

$\left[\max\left(0, \frac{x_l+v_l(t-\overline{\delta}_l)-x}{-\nu^{\flat}+v_l}, \frac{x_l+v_l(t-\overline{\delta}_l)-x}{\nu^{\#}+v_l}, t-\overline{\delta}_{l+1}\right), t-\overline{\delta}_l\right]$:

$$\begin{cases} (i) & T_1(t,x,v_l,g_l) & \text{if } x_l + v_l(t-\overline{\delta}_l) - x \leq 0 \\ & & \text{and } T_1(t,x,v_l,g_l) \in \left[t-\overline{\delta}_{l+1}, t-\overline{\delta}_l\right] \\ (ii) & t-\overline{\delta}_l & \text{if } x_l + v_l(t-\overline{\delta}_l) - x \leq 0 \\ & & \text{and } T_1(t,x,v_l,g_l) \geq t-\overline{\delta}_l \\ (iii) & t-\overline{\delta}_{l+1} & \text{if } x_l + v_l(t-\overline{\delta}_l) - x \leq 0 \\ & & \text{and } T_1(t,x,v_l,g_l) \leq t-\overline{\delta}_{l+1} \\ (iv) & T_2(t,x,v_l,g_l) & \text{if } x_l + v_l(t-\overline{\delta}_l) - x \geq 0 \\ & & \text{and } T_2(t,x,v_l,g_l) \in \left[t-\overline{\delta}_{l+1}, t-\overline{\delta}_l\right] \\ (v) & t-\overline{\delta}_l & \text{if } x_l + v_l(t-\overline{\delta}_l) - x \geq 0 \\ & & \text{and } T_2(t,x,v_l,g_l) \geq t-\overline{\delta}_l \\ (vi) & t-\overline{\delta}_{l+1} & \text{if } x_l + v_l(t-\overline{\delta}_l) - x \geq 0 \\ & & \text{and } T_2(t,x,v_l,g_l) \leq t-\overline{\delta}_{l+1} \end{cases} \tag{272}$$

Proof — The function $\kappa_{\overline{\delta}_l,g_l,h_l,x_l,v_l,t,x}(T)$ is minimal for a given $T > 0$ if and only if $0 \in \partial_{-}\kappa_{\overline{\delta}_l,g_l,h_l,x_l,v_l,t,x}(T)$. Since $u_p(v_l,g_l) \in -\partial_{+}\psi(\rho_p(v_l,g_l))$ for $p \in \{1,2\}$, we have by the Legendre-Fenchel inversion formula that $\rho_p(v_l,g_l) \in \partial_{-}\varphi^*(u_p(v_l,g_l))$. This last formula implies that $0 \in \partial_{-}\left(\varphi^*(\cdot) -\cdot\, \rho_p(v_l,g_l)\right)\left(u_p(v_l,g_l)\right)$ and thus that:

$$\begin{aligned} \psi(\rho_p(v_l,g_l)) &= \inf_{u \in \mathrm{Dom}(\varphi^*)} \left[\varphi^*(u) - \rho_p(v_l,g_l)u\right] \\ &= \varphi^*(u_p(v_l,g_l)) - \rho_p(v_l,g_l)u_p(v_l,g_l) \end{aligned}$$

Since we consider only positive capture times T, we have to consider two situations:

- If $x_l + v_l(t - \overline{\delta}_l) - x \leq 0$, we have that $T_2(t, x, v_l, g_l) \leq 0$ and $T_1(t, x, v_l, g_l) \geq 0$. The relations $\psi(\rho_1(v_l, g_l)) - \rho_1(v_l, g_l)v_l = g_l$, $\rho_1(v_l, g_l) \in \partial_- \varphi^*(u_1(v_l, g_l))$ and $\psi(\rho_1(v_l, g_l)) = \varphi^*(u_1(v_l, g_l)) - \rho_1(v_l, g_l)u_1(v_l, g_l)$ imply that

  $-g_l + \varphi^*(u_1(v_l, g_l)) - (u_1(v_l, g_l) + v_l)\rho_1(v_l, g_l) = 0$. Hence, using our definition of $T_1(t, x, v_l, g_l) := \frac{x_l + v_l(t - \overline{\delta}_l) - x}{u_1(v_l, g_l) + v_l}$, we have that:

$$0 = -g_l + \varphi^* \left(\frac{x_l + v_l(t - \overline{\delta}_l) - x}{T_1(t, x, v_l, g_l)} - v_l \right) - \frac{x_l + v_l(t - \overline{\delta}_l) - x}{T_1(t, x, v_l, g_l)} \rho_1(v_l, g_l)$$

  Using equation (269), we have $0 \in \partial_- \kappa_{\overline{\delta}_l, g_l, h_l, x_l, v_l, t, x}(T_1(t, x, v_l, g_l))$ and thus $T_1(t, x, v_l, g_l)$ minimizes $\kappa_{\overline{\delta}_l, g_l, h_l, x_l, v_l, t, x}(T)$ for positive times T.

  The cases (i), (ii) and (iii) in equation (272) are obtained using the convexity of $\kappa_{\overline{\delta}_l, g_l, h_l, x_l, v_l, t, x}(\cdot)$. Note that in our situation, definition 26 implies that $T_1(t, x, v_l, g_l) \geq \frac{x_l + v_l(t - \overline{\delta}_l) - x}{-\nu^\flat + v_l}$.

  Since $x_l + v_l(t - \overline{\delta}_l) - x \leq 0$, we also have that $\frac{x_l + v_l(t - \overline{\delta}_l) - x}{\nu^\# + v_l} \leq 0$. Hence, we have that:

$$T_1(t, x, v_l, g_l) \geq \max \left(0, \frac{x_l + v_l(t - \overline{\delta}_l) - x}{-\nu^\flat + v_l}, \frac{x_l + v_l(t - \delta_l) - x}{\nu^\# + v_l} \right)$$

  Hence, the condition $T_1(t, x, v_l, g_l) \geq \max \left(0, \frac{x_l + v_l(t - \overline{\delta}_l) - x}{-\nu^\flat + v_l}, \frac{x_l + v_l(t - \overline{\delta}_l) - x}{\nu^\# + v_l}, t - \overline{\delta}_{l+1} \right)$ is satisfied if and only if $T_1(t, x, v_l, g_l) \geq t - \overline{\delta}_{l+1}$. Note also that if the previous condition is not satisfied, then $\max \left(0, \frac{x_l + v_l(t - \overline{\delta}_l) - x}{-\nu^\flat + v_l}, \frac{x_l + v_l(t - \overline{\delta}_l) - x}{\nu^\# + v_l}, t - \overline{\delta}_{l+1} ght) = t - \overline{\delta}_{l+1}$.

- If $x_l + v_l(t - \overline{\delta}_l) - x \geq 0$, we have that $T_1(t, x, v_l, g_l) \leq 0$ and $T_2(t, x, v_l, g_l) \geq 0$. The relations $\psi(\rho_1(v_l, g_l)) - \rho_1(v_l, g_l)v_l = g_l$, $\rho_2(v_l, g_l) \in \partial_- \varphi^*(u_2(v_l, g_l))$ and $si(\rho_2(v_l, g_l)) = \varphi^*(u_2(v_l, g_l)) - \rho_2(v_l, g_l)u_2(v_l, g_l)$ imply that $-g_l + \varphi^*(u_1(v_l, g_l)) - (u_1(v_l, g_l) + v_l)\rho_1(v_l, g_l) = 0$. Hence, using our definition of $T_2(t, x, v_l, g_l) := \frac{x_l + v_l(t - \overline{\delta}_l) - x}{u_2(v_l, g_l) + v_l}$, we have that:

$$0 = -g_l + \varphi^* \left(\frac{x_l + v_l(t - \overline{\delta}_l) - x}{T_2(t, x, v_l, g_l)} - v_l \right) - \frac{x_l + v_l(t - \overline{\delta}_l) - x}{T_2(t, x, v_l, g_l)} \rho_2(v_l, g_l)$$

  Using equation (269), we have $0 \in artial_- \kappa_{\overline{\delta}_l, g_l, h_l, x_l, v_l, t, x}(T_2(t, x, v_l, g_l))$ and thus $T_2(t, x, v_l, g_l)$ minimizes $\kappa_{\overline{\delta}_l, g_l, h_l, x_l, v_l, t, x}(T)$ for positive times T.

The cases (iv), (v) and (vi) in equation (272) are obtained using the convexity of $\kappa_{\overline{\delta}_l,g_l,h_l,x_l,v_l,t,x}(\cdot)$. Note that in our situation, definition 26 implies that $T_2(t,x,v_l,g_l) \geq \frac{x_l+v_l(t-\overline{\delta}_l)-x}{\nu^{\#}+v_l}$.

Since $x_l + v_l(t-\overline{\delta}_l) - x \geq 0$, we also have that $\frac{x_l+v_l(t-\overline{\delta}_l)-x}{-\nu^{\flat}+v_l} \leq 0$. Hence, we have that:

$$T_2(t,x,v_l,g_l) \geq \max\left(0, \frac{x_l + v_l(t-\overline{\delta}_l) - x}{-\nu^{\flat} + v_l}, \frac{x_l + v_l(t-\overline{\delta}_l) - x}{\nu^{\#} + v_l}\right)$$

Hence, the condition $T_2(t,x,v_l,g_l) \geq \max\left(0, \frac{x_l+v_l(t-\overline{\delta}_l)-x}{-\nu^{\flat}+v_l}, \frac{x_l+v_l(t-\overline{\delta}_l)-x}{\nu^{\#}+v_l}, t-\overline{\delta}_{l+1}\right)$ is satisfied if and only if $T_2(t,x,v_l,g_l) \geq t-\overline{\delta}_{l+1}$. Note also that if the previous condition is not satisfied, then

$$\max\left(0, \frac{x_l+v_l(t-\overline{\delta}_l)-x}{-\nu^{\flat}+v_l}, \frac{x_l+v_l(t-\overline{\delta}_l)-x}{\nu^{\#}+v_l}, t-\overline{\delta}_{l+1}\right) = t-\overline{\delta}_{l+1}.$$

Once the minimizer is computed, it can be used to find the explicit expression of the value function.

Proposition 28. [Computation of $\boldsymbol{M}_{\mu_l}(\cdot,\cdot)$] For all $(t,x) \in \mathrm{Dom}(\boldsymbol{M}_{\mu_l})$, the expression $\boldsymbol{M}_{\mu_l}(t,x)$ can be computed using the following formulae:

$$\boldsymbol{M}_{\mu_l}(t,x) = \begin{cases} (i) & \psi(\rho_1(v_l,g_l))(t-\overline{\delta}_l) + (x_l - x)\rho_1(v_l,g_l) + h_l \\ & \text{if } x_l + v_l(t-\overline{\delta}_l) \leq x \\ & \text{and } T_1(t,x,v_l,g_l) \in [t-\overline{\delta}_{l+1}, t-\overline{\delta}_l] \\ (ii) & \psi(\rho_2(v_l,g_l))(t-\overline{\delta}_l) + (x_l - x)\rho_2(v_l,g_l) + h_l \\ & \text{if } x_l + v_l(t-\overline{\delta}_l) \geq x \\ & \text{and } T_2(t,x,v_l,g_l) \in [t-\overline{\delta}_{l+1}, t-\overline{\delta}_l] \end{cases} \tag{273}$$

and

$$\boldsymbol{M}_{\mu_l}(t,x) = \begin{cases} (iii) & h_l + (t-\overline{\delta}_l)\varphi^*\left(\dfrac{x_l - x}{t-\overline{\delta}_l}\right) \\ & \text{if } \; x_l + v_l(t-\overline{\delta}_l) \le x \;\text{ and }\; T_1(t,x,v_l,g_l) \ge t-\overline{\delta}_l \\ & \text{or if } \; x_l + v_l(t-\overline{\delta}_l) \ge x \;\text{ and }\; T_2(t,x,v_l,g_l) \ge t-\overline{\delta}_l \\ (iv) & g_l(\overline{\delta}_{l+1}-\overline{\delta}_l) + h_l + (t-\overline{\delta}_{l+1})\varphi^*\left(\dfrac{x_l + v_l(\overline{\delta}_{l+1}-\overline{\delta}_l) - x}{t-\overline{\delta}_{l+1}}\right) \\ & \text{if } \; x_l + v_l(t-\overline{\delta}_l) \le x \;\text{ and }\; T_1(t,x,v_l,g_l) \le t-\overline{\delta}_{l+1} \\ & \text{or if } \; x_l + v_l(t-\overline{\delta}_l) \ge x \;\text{ and }\; T_2(t,x,v_l,g_l) \le t-\overline{\delta}_{l+1} \end{cases} \tag{274}$$

Proof — The cases (iii) and (iv) in equation (274) are trivially obtained by combining equations (268) and (272).

The cases (i) and (ii) in equation (273) are also obtained by combining equations (268) and (272). By combining the formula $\varphi^*\left(\frac{x_l+v_l(t-\overline{\delta}_l-T_p(t,x,v_l,g_l))-x}{T_p(t,x,v_l,g_l)}\right) = \psi(\rho_p(v_l,g_l)) + \frac{x_l+v_l(t-\overline{\delta}_l-T_p(t,x,v_l,g_l))-x}{T_p(t,x,v_l,g_l)}\rho_p(v_l,g_l)$ and the definition of $\rho_p(v_l,g_l)$, we have:

$$-g_l + \varphi^*\left(\frac{x_l + v_l(t-\overline{\delta}_l - T_p(t,x,v_l,g_l)) - x}{T_p(t,x,v_l,g_l)}\right) = \rho_p(v_l,g_l)\frac{x_l + v_l(t-\overline{\delta}_l) - x}{T_p(t,x,v_l,g_l)}$$

Using again the definition of $\rho_p(v_l,g_l)$, we have $g_l - \rho_p(v_l,g_l)v_l = \psi(\rho_p(v_l,g_l))$, which after some algebra leads to the cases (i) and (ii) in equation (273). ■

Figure 5.4 illustrates the domains of equation (273) and (274), for the solution to an affine internal condition defined by equation (263).

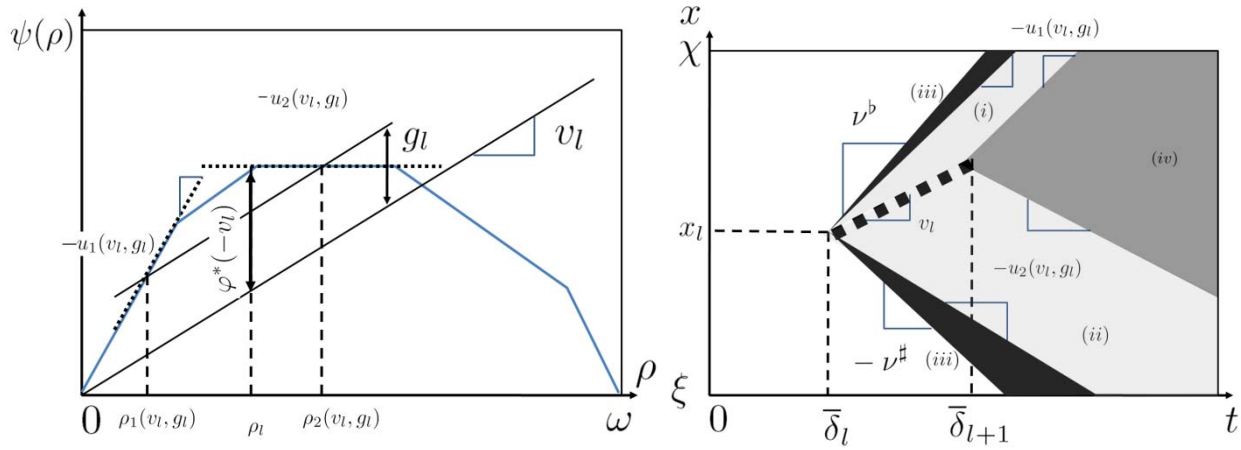


Figure 5.4. Construction of the solution associated with an affine internal condition.

Left: Illustration of the construction of a $u_1(v_l, g_l)$ and $u_2(v_l, g_l)$ from known v_l and g_l. **Right:** The (t, x) domain of the solution corresponding to the affine internal condition (263) can be separated in four different areas. The domains highlighted in dark gray correspond to the case (iii) in equation (274). The domains highlighted in light gray correspond to the cases (i) and (ii) in equation (273). The remaining domain in medium gray corresponds to the case (iv) in (274). The domain of the internal condition is represented by a dashed line.

25.6. Extension to piecewise affine initial, boundary and internal conditions

25.6.1. Semi-analytic solutions

Using the inf-morphism property (226), we can express the solution associated with piecewise affine initial, boundary and internal conditions as a *semi-analytic formula*. A semi-analytic formula is defined here as an operation (in the present case a minimization) on a finite set of closed-form expression functions.

In order to establish the Lax-Hopf algorithm, we first need to establish the following result.

Proposition 1. [Decomposition of piecewise affine functions] Let $\boldsymbol{c}(\cdot,\cdot)$ be a piecewise affine value condition defined on a finite number of segments of $\mathbb{R}^2$. There exists a finite number of affine functions $\boldsymbol{c}_i(\cdot,\cdot)$, $i \in I$ of $\mathbb{R}^2$ such that:

$$\boldsymbol{c}(\cdot,\cdot) = \min_{i \in I} \boldsymbol{c}_i\,(\cdot,\cdot) \tag{275}$$

Proof — Since $\boldsymbol{c}(\cdot,\cdot)$ is a piecewise affine function defined on segments of $\mathbb{R}^2$, there exist $x_{\min,i}$, $x_{\max,i}$, $t_{\min,i}$, $t_{\max,i}$, $s_{x,i}$, $s_{t,i}$, $s_{p,i}$ such that

$$\boldsymbol{c}(t,x) = \begin{cases} s_{p,i} + s_{x,i}x + s_{t,i}t & \text{if } \exists i \in I \text{ and } \alpha \in [0,1] \text{ such that } x = x_{\min,i} + \alpha(x_{\max,i} - x_{\min,i}) \\ & \text{and } t = t_{\min,i} + \alpha(t_{\max,i} - t_{n,i}) \\ +\infty & \text{otherwise} \end{cases} \tag{276}$$

Let us define the affine functions $\boldsymbol{c}_i(\cdot,\cdot)$ for $i \in I$ as follows:

$$\boldsymbol{c}_i(t,x) = \begin{cases} s_{p,i} + s_{x,i}x + s_{t,i}t & \text{if } \exists \alpha \in [0,1] \text{ such that } x = x_{\min,i} + \alpha(x_{\max,i} - x_{\min,i}) \\ & \text{and } t = t_{\min,i} + \alpha(t_{\max,i} - t_{\min,i}) \\ +\infty & \text{otherwise} \end{cases} \tag{277}$$

This definition trivially implies (275), which completes the proof. ■

Proposition 1 implies that a set of piecewise affine initial, boundary and internal conditions can be decomposed as the minimum of a finite number of affine initial, boundary and internal conditions. In addition, the solutions associated with affine initial, boundary and internal conditions have a closed form expression by equations (238), (249), (261), (273) and (274). Hence, using the inf-morphism property (226), we can compute the solution associated with a set of piecewise affine initial, boundary and internal conditions semi-analytically. The corresponding *Lax-Hopf algorithm* is presented in the following section.

25.6.2. Lax Hopf algorithm

We now present a specific instantiation of the Lax-Hopf algorithm for a rectangular grid and for the mixed initial-boundary-internal conditions problem. Note that any grid can be used, since each point of the solution is computed using the coefficients of the initial, boundary and internal conditions only. The space and time steps are denoted by Δx and Δt respectively. The grid is defined by the set $\mathcal{G} := \{1, \dots, n_t\} \times \{1, \dots, n_x\}$, where n_t and n_x are positive integers.

LAX-HOPF ALGORITHM FOR THE MOSKOWITZ FUNCTION

INITIALIZATION
$\boldsymbol{M}(M,N) \longleftarrow +\infty \quad \forall (M,N) \in \mathcal{G}$ [Output matrix containing the Moskowitz function]
MAIN LOOP
For $\boldsymbol{M} := \mathbf{0}$ **to** $n_{\boldsymbol{t}}$ **do** [time iteration]
  For $\boldsymbol{N} := \mathbf{0}$ **to** $n_{\boldsymbol{x}}$ **do** [space iteration]
    Definition of $x_N := N\Delta x + \xi$ and $t_M :=$ $M\Delta t$ [space and time grid definition]
    For $i \in I$ **do** [iteration on the set of initial conditions]
      Compute $\boldsymbol{M}_{\mathcal{M}_{0_i}}(t_M, x_N)$ [using equation (238)]
      If $\boldsymbol{M}_{\mathcal{M}_{0,i}}(t_M, x_N) \leq \boldsymbol{M}(M,N)$ **then** $\boldsymbol{M}(M,N) = \boldsymbol{M}_{\mathcal{M}_{0,i}}(t_M, x_N)$
    For $j \in J$ **do** [iteration on the set of upstream boundary conditions]
      Compute $\boldsymbol{M}_{\gamma_j}(t_M, x_N)$ [using equation (249)]
      If $\boldsymbol{M}_{\gamma_j}(t_M, x_N) \leq \boldsymbol{M}(M,N)$ **then** $\boldsymbol{M}(M,N) = \boldsymbol{M}_{\gamma_j}(t_M, x_N)$
    For $k \in K$ **do** [iteration on the set of downstream boundary conditions]
      Compute $\boldsymbol{M}_{\beta_k}(t_M, x_N)$ [using equation (261)]
      If $\boldsymbol{M}_{\beta_k}(t_M, x_N) \leq \boldsymbol{M}(M,N)$ **then** $\boldsymbol{M}(M,N) = \boldsymbol{M}_{\beta_k}(t_M, x_N)$
    For $l \in L$ **do** [iteration on the set of internal conditions]
      Compute $\boldsymbol{M}_{\mu_l}(t_M, x_N)$ [using equations (273) and (274)]
      If $\boldsymbol{M}_{\mu_l}(t_M, x_N) \leq \boldsymbol{M}(M,N)$ **then** $\boldsymbol{M}(M,N) = \boldsymbol{M}_{\mu_l}(t_M, x_N)$
RETURN $\boldsymbol{M}(M,N)$

The quantity $\boldsymbol{M}(M,N)$ represents the exact value of the Moskowitz function at (t_M, x_N), up to machine accuracy.

25.7. Extension to scalar conservation laws

In this section, we extend the Lax-Hopf algorithm for solving scalar conservation laws, related to scalar HJ PDEs by a variable change. Indeed, as mentioned in section 25.1, the derivatives of a function modeled by a HJ PDE satisfy a scalar conservation law themselves.

25.7.1. Spatial derivatives of the solutions to affine initial, boundary and internal conditions

The solutions $\boldsymbol{M}_{\mathcal{M}_{0_i}}(\cdot,\cdot)$, $\gamma_j(\cdot,\cdot)$, $\beta_k(\cdot,\cdot)$ and $\boldsymbol{M}_{\mu_l}(\cdot,\cdot)$ are convex since they are associated with convex target functions defined on a compact subset of $\mathbb{R}_+ \times X$. Hence, these functions are differentiable almost everywhere on their domains of definition. The spatial derivatives of the above functions can be computed (whenever $\varphi^*(\cdot)$ is differentiable and using $\varphi^{*\prime}(\cdot)$ as the

notation for the derivative of $\varphi^*(\cdot)$) explicitly as:

$$\frac{\partial \boldsymbol{M}_{\mathcal{M}_{0_i}}(t,x)}{\partial x} = \begin{cases} a_i & \text{if } u_0(a_i) \in]\frac{\overline{\alpha}_i - x}{t}, \frac{\overline{\alpha}_{i+1} - x}{t}[\\ -\varphi^{*\prime}(\frac{\overline{\alpha}_i - x}{t}) & \text{if } u_0(a_i) < \frac{\overline{\alpha}_i - x}{t} \\ -\varphi^{*\prime}(\frac{\overline{\alpha}_{i+1} - x}{t}) & \text{if } u_0(a_i) > \frac{\overline{\alpha}_{i+1} - x}{t} \end{cases} \tag{278}$$

In the previous formula, $u_0(a_i)$ is an element of $-\partial_+\psi(-a_i)$.

$$\frac{\partial \boldsymbol{M}_{\gamma_j}(t,x)}{\partial x} = \begin{cases} -\rho_j & \text{if } T_0(\rho_j, x) \in [t - \overline{\beta}_{j+1}, t - \overline{\beta}_j] \\ -\varphi^{*\prime}(\frac{\xi - x}{t - \overline{\beta}_j}) & \text{if } t - \overline{\beta}_j < T_0(\rho_j, x) \\ -\varphi^{*\prime}(\frac{\xi - x}{t - \overline{\beta}_{j+1}}) & \text{if } T_0(\rho_j, x) < t - \overline{\beta}_{j+1} \end{cases} \tag{279}$$

In the previous formula, ρ_j and T_0 are computed by definition 11.

$$\frac{\partial \boldsymbol{M}\beta_k(t,x)}{\partial x} = \begin{cases} -\rho_k & \text{if } T_0(\rho_k, x) \in]t - \overline{\gamma}_{k+1}, t - \overline{\gamma}_k[\\ -\varphi^{*\prime}(\frac{\chi - x}{t - \overline{\gamma}_k}) & \text{if } t - \overline{\gamma}_k < T_0(\rho_k, x) \\ -\varphi^{*\prime}(\frac{\chi - x}{t - \overline{\gamma}_{k+1}}) & \text{if } T_0(\rho_k, x) < t - \overline{\gamma}_{k+1} \end{cases} \tag{280}$$

In the previous formula, ρ_k and T_0 are computed by definition 18.

$$\frac{\partial \boldsymbol{M}_{\mu_l}(t,x)}{\partial x} = \begin{cases} -\rho_1(v_l, g_l) & \text{if } x_l + v_l(t-\overline{\delta}_l) < x \\ & \text{and } T_1(t,x,v_l,g_l) \in \left]t-\overline{\delta}_{l+1}, t-\overline{\delta}_l\right[\\ -\rho_2(v_l, g_l) & \text{if } x_l + v_l(t-\overline{\delta}_l) > x \\ & \text{and } T_2(t,x,v_l,g_l) \in \left]t-\overline{\delta}_{l+1}, t-\overline{\delta}_l\right[\\ -\varphi^{*\prime}\left(\dfrac{x_l - x}{t-\overline{\delta}_l}\right) & \text{if } x_l + v_l(t-\overline{\delta}_l) < x \\ & \text{and } T_1(t,x,v_l,g_l) > t-\overline{\delta}_l \\ & \text{or if } x_l + v_l(t-\overline{\delta}_l) > x \\ & \text{and } T_2(t,x,v_l,g_l) > t-\overline{\delta}_l \\ -\varphi^{*\prime}\left(\dfrac{x_l + v_l(\overline{\delta}_{l+1} - \overline{\delta}_l) - x}{t-\overline{\delta}_{l+1}}\right) & \text{if; } x_l + v_l(t-\overline{\delta}_l) < x \\ & \text{and } T_1(t,x,v_l,g_l) < t-\overline{\delta}_{l+1} \\ & \text{or if } x_l + v_l(t-\overline{\delta}_l) > x \\ & \text{and } T_2(t,x,v_l,g_l) < t-\overline{\delta}_{l+1} \end{cases} \tag{281}$$

In the previous formula, ρ_1, ρ_2, T_1 and T_2 are computed by definition 26.

25.7.2. Computation of the density function

In order to provide a similar algorithm for the density, we now assume that the Moskowitz function is Lipschitz-continuous on $\mathbb{R}_+ \times X$ in order to define a measurable-integrable density function by equation (282). Note that this assumption is only required for the computation of ρ and not for the computation of the Moskowitz function $\boldsymbol{M}(\cdot,\cdot)$. For instance, the solution to the HJ PDE (195) associated with any Lipschitz-continuous initial, left and downstream boundary condition functions (but not internal conditions) is itself Lipschitz-continuous [116,128,129].

Whenever the Moskowitz function is differentiable in (t,x), we compute the density function $\rho(t,x)$ by:

$$\rho(t,x) = -\frac{\partial \boldsymbol{M}(t,x)}{\partial x} \tag{282}$$

Proposition 1. [Computation of the spatial derivative of $\boldsymbol{M}(\cdot,\cdot)$] Let us consider $(t,x) \in \mathbb{R}_+ \times X$ such that $\boldsymbol{M}(\cdot,\cdot)$ is differentiable at (t,x). Since the Moskowitz function $\boldsymbol{M}(\cdot,\cdot)$ is the minimum of the convex functions $\boldsymbol{M}_{\mathcal{M}_{0,i}}(\cdot,\cdot)$, $\boldsymbol{M}_{\gamma_j}(\cdot,\cdot)$, $\boldsymbol{M}_{\beta_k}(\cdot,\cdot)$ and $\boldsymbol{M}_{\mu_l}(\cdot,\cdot)$ for $(i,j,k,l) \in I \times J \times K \times L$, there exists a solution $\boldsymbol{M}_{\boldsymbol{a}}(\cdot,\cdot)$ associated with a value condition $\boldsymbol{a}(\cdot,\cdot)$ which is equal to the Moskowitz function at (t,x), *i.e.* $\boldsymbol{M}(t,x) = \boldsymbol{M}_a(t,x)$. We assume that $\boldsymbol{M}_{\boldsymbol{a}}(\cdot,\cdot)$ is

differentiable at (t,x). Given these assumptions, we have the following property:

$$\frac{\partial \boldsymbol{M}(t,x)}{\partial x} = \frac{\partial \boldsymbol{M_a}(t,x)}{\partial x} \tag{283}$$

Proof — Let us define the function $g(\cdot,\cdot)$ as $g(\cdot,\cdot) := \boldsymbol{M_a}(\cdot,\cdot) - \boldsymbol{M}(\cdot,\cdot)$. Since $\boldsymbol{M}(\cdot,\cdot)$ and $\boldsymbol{M_a}(\cdot,\cdot)$ are both differentiable at (t,x), $g(\cdot,\cdot)$ is also differentiable at (t,x). By definition of $\boldsymbol{M}(\cdot,\cdot)$, the function $g(\cdot,\cdot)$ is positive and satisfies $g(t,x) = 0$. Hence, (t,x) minimizes $g(\cdot,\cdot)$ and we have $\frac{\partial g(t,x)}{\partial x} = 0$, which implies equation (283). ■

Since $\boldsymbol{M}(\cdot,\cdot)$ is the minimum of convex functions, it is differentiable almost everywhere [78]. Hence, its associated density function $\rho(\cdot,\cdot)$ is defined almost everywhere on $\mathbb{R}_+ \times X$. We use equation (283) to compute the density function $\rho(\cdot,\cdot)$ exactly whenever it is defined using the following algorithm, extending the Lax-Hopf algorithm.

25.7.3. Extension of the Lax-Hopf algorithm for scalar conservation laws

We consider the specific instantiation of the extension of the Lax-Hopf algorithm for a rectangular grid and for the mixed initial-boundary-internal conditions problem. Note again that any grid can be used, since each point of the solution is computed using the coefficients of the initial, boundary and internal conditions only. The space and time steps are denoted by Δx and Δt respectively. The grid is defined by the set $\mathcal{G} := \{1,\dots,n_t\} \times \{1,\dots,n_x\}$, where n_t and n_x are positive integers.

LAX-HOPF ALGORITHM FOR THE DENSITY FUNCTION

INITIALIZATION

$\boldsymbol{M}(M,N) \longleftarrow +\infty \quad \forall (M,N) \in \mathcal{G}$ [Output matrix containing the Moskowitz function]

$\boldsymbol{D}(M,N) \longleftarrow \text{NaN} \quad \forall (M,N) \in \mathcal{G}$ [Output matrix containing the Density function]

MAIN LOOP

For $\boldsymbol{M} := \mathbf{1}$ **to** n_t **do** [time iteration]

  For $\boldsymbol{N} := \mathbf{1}$ **to** n_x **do** [space iteration]

    $x_N := N\Delta x + \xi$ and $t_M := M\Delta t$ [space and time grid definition]

    For $i \in I$ **do** [iteration on the set of initial conditions]

      Computation of $\boldsymbol{M}_{\mathcal{M}_{0,i}}(t_M, x_N)$ [using equation (238)]

      If $\boldsymbol{M}_{\mathcal{M}_{0,i}}(t_M, x_N) \leq \boldsymbol{M}(M,N)$ **then**

      $\boldsymbol{M}(M,N) = \boldsymbol{M}_{\mathcal{M}_{0,i}}(t_M, x_N)$

        If $\boldsymbol{M}_{\mathcal{M}_{0,i}}$ is differentiable at (t_M, x_N)

        then

        $\boldsymbol{D}(M,N) = -\frac{\partial \boldsymbol{M}_{\mathcal{M}_{0,i}}(t_M,x_N)}{\partial x}$ [using equation (278)]

    For $j \in J$ **do** [iteration on the set of upstream boundary conditions]

Computation of $\boldsymbol{M}_{\gamma_j}(t_M, x_N)$ [using equation (249)]

If $\boldsymbol{M}_{\gamma_j}(t_M, x_N) \leq \boldsymbol{M}(M, N)$ **then**

$\boldsymbol{M}(M, N) = \boldsymbol{M}_{\gamma_j}(t_M, x_N)$

If $\boldsymbol{M}_{\gamma_j}$ is differentiable at (t_M, x_N) **then**

$\boldsymbol{D}(M, N) = -\frac{\partial \boldsymbol{M}_{\gamma_j}(t_M, x_N)}{\partial x}$ [using equation (279)] [using equation (279)]

For $k \in K$ **do** [iteration on the set of downstream boundary conditions]

Computation of $\boldsymbol{M}_{\beta_k}(t_M, x_N)$ [using equation (261)]

If $\boldsymbol{M}_{\beta_k}(t_M, x_N) \leq \boldsymbol{M}(M, N)$ **then**

$\boldsymbol{M}(M, N) = \boldsymbol{M}_{\beta_k}(t_M, x_N)$

If $\boldsymbol{M}_{\beta_k}$ is differentiable at (t_M, x_N) **then**

$\boldsymbol{D}(M, N) = -\frac{\partial \boldsymbol{M}_{\beta_k}(t_M, x_N)}{\partial x}$ [using equation (280)]

For $l \in L$ **do** [iteration on the set of internal conditions]

Computation of $\boldsymbol{M}_{\mu_l}(t_M, x_N)$ [using equations (273) and (274)]

If $\boldsymbol{M}_{\mu_l}(t_M, x_N) \leq \boldsymbol{M}(M, N)$ **then**

$\boldsymbol{M}(M, N) = \boldsymbol{M}_{\mu_l}(t_M, x_N)$

If $\boldsymbol{M}_{\mu_l}$ is differentiable at (t_M, x_N) **then**

$\boldsymbol{D}(M, N) = -\frac{\partial \boldsymbol{M}_{\mu_l}(t_M, x_N)}{\partial x}$ [using equation (281)]

RETURN $\boldsymbol{D}(M, N)$

The quantity $\boldsymbol{D}(M, N)$ represents the exact value of the density function associated with the Moskowitz function at (t_M, x_N), up to machine accuracy.

25.8. Numerical examples

25.8.1. Integration of internal conditions into Hamilton-Jacobi equations

In this implementation, we consider a triangular Hamiltonian as defined in example 3, with parameters $\nu^{\flat} = 1$, $\gamma = 1$, $\omega = 6$, $\nu^{\#} = \frac{1}{5}$ and $\delta = \varphi^*(0) = 1$. We also consider piecewise affine initial, upstream, downstream and internal condition functions defined by equation (284):

$$\mathcal{M}_0(t,x) = \begin{cases} a_i x + b_i & \text{if } \ t = 0, \\ & \text{and } \ \exists i \in I \ \text{ such that } \ x \in [\overline{\alpha}_i, \overline{\alpha}_{i+1}] \\ +\infty & \text{otherwise} \end{cases}$$

$$\gamma(t,x) = \begin{cases} c_j t + d_j & \text{if } \ x = \xi \\ & \text{and } \ \exists j \in J \ \text{ such that } \ t \in [\overline{\gamma}_j, \overline{\gamma}_{j+1}] \\ +\infty & \text{otherwise} \end{cases}$$

$$\beta(t,x) = \begin{cases} e_k t + f_k & \text{if } \ x = \chi \\ & \text{and } \ \exists k \in K \ \text{ such that } \ t \in [\overline{\beta}_k, \overline{\beta}_{k+1}] \\ +\infty & \text{otherwise} \end{cases} \tag{284}$$

$$\mu_p(t,x) = \begin{cases} g_{p_l}(t - \overline{\delta_{p_l}}) + h_{p_l} & \text{if } \ \exists l \in L_p \ \text{ such that} \\ & x = v_{p_l}(t - \overline{\delta_{p_l}}) + x_{p_l} \\ & \text{and } \ t \in [\overline{\delta_{p_l}}, \overline{\delta_{p_{l+1}}}] \\ +\infty & \text{otherwise} \end{cases}$$

In this numerical application, we choose the following set of coefficients a_i, b_i, $\overline{\alpha}_i$, c_j, d_j, $\overline{\gamma}_j$, e_k, f_k, $\overline{\beta}_k$:

$$\begin{cases} a := (-1, -7/2, -1/10, -7/5) \\ b := (0, -\frac{25}{2}, -\frac{43}{2}, \frac{9}{2}) \\ \overline{\alpha} := (0,5,10,20,25) \end{cases}$$
$$\begin{cases} c := (1,1/2,4/5,7/10) \\ d := (0, \frac{3}{2}, -\frac{9}{5}, -\frac{3}{10}) \\ \overline{\gamma} := (0,3,11,15,20) \end{cases} \tag{285}$$
$$\begin{cases} e := (0,2/5,0,4/5) \\ f := (-\frac{61}{2}, -\frac{289}{10}, -\frac{221}{10}, -\frac{365}{10}) \\ \overline{\beta} := (0,4,17,18,20) \end{cases}$$

We first compute the solution to equation (195) associated with (285) numerically using the Lax-Hopf algorithm. The results are shown in Figure 8.1.

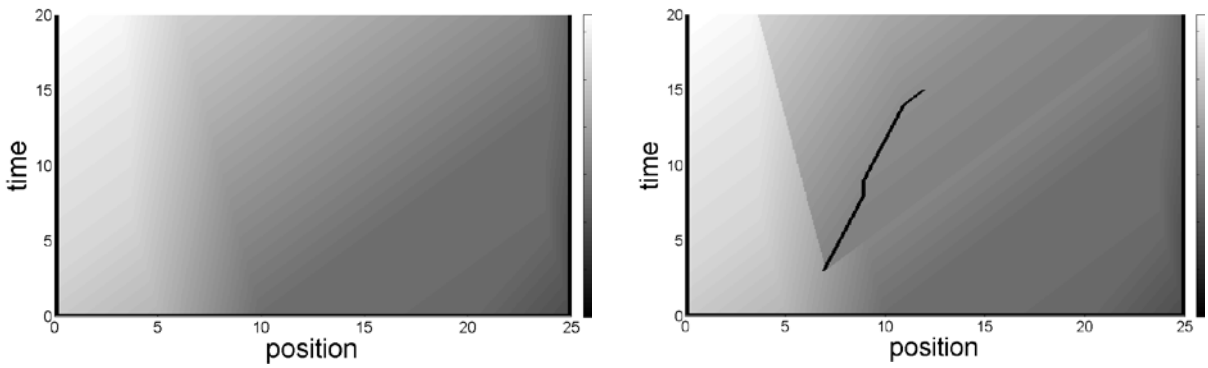


Figure 8.1. Example of integration of an internal condition into the solution of the HJ PDE (195).

Left: Computation of the solution to the mixed initial-boundary conditions problem with parameters listed in (285). **Right:** Computation of the solution to the mixed initial-boundary-internal conditions problem (285) and (286). The initial, boundary and internal conditions are represented by solid lines.

We then incorporate a single internal condition, defined by the following coefficients.

$$\begin{cases} v_1 := (2/5,0,1/2,1/2) \\ g_1 := (1/5,1,1/4,0) \\ h_1 := (-18,-19,-20,-21,-21) \\ \overline{\delta_1} := (3,8,9,14,15) \end{cases} \tag{286}$$

As can be seen in Figure 8.1, the incorporation of the internal condition modifies the value of the solution around it and enables us to add new information.

25.8.2. Numerical validation of the Lax-Hopf algorithm (density function)

We compare the Lax-Hopf algorithm and the Godunov scheme [309,166,144] (and its specific instantiation as the Daganzo cell transmission model [126,127]), which is widely used by the transportation research community.

In this implementation, we consider a (non piecewise affine) Greenshields Hamiltonian defined as in example 2, where $\nu = 1$ and $\rho^* = 4$ (dummy values). We consider the following initial and upstream boundary condition functions:

$$\begin{cases} a := (-2,-4,-1) \\ b := (0,20,-40) \\ \overline{\alpha} := (0,10,20,30) \end{cases} \quad \begin{cases} c := (2) \\ d := (0) \\ \overline{\gamma} := (0,20) \end{cases} \tag{287}$$

These initial and upstream boundary conditions were used previously in the article [309].

We compute the Moskowitz and density functions solution to the initial and upstream boundary conditions problem (287) using the Lax-Hopf algorithm and compare the results with the analytical formula derived in [309]. The results are illustrated in Figure 8.2.

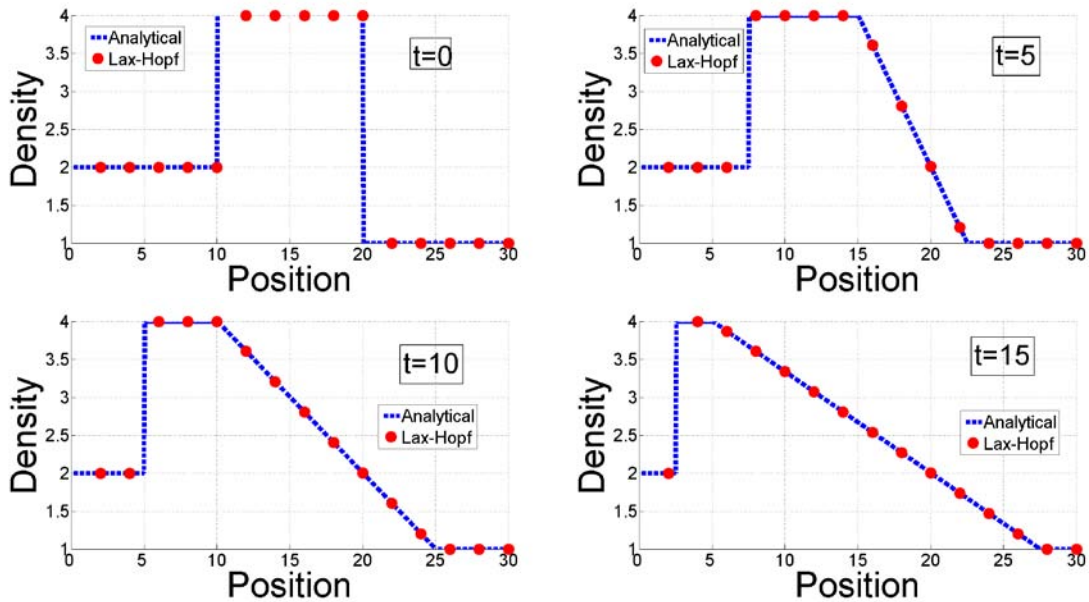


Figure 8.2. Comparison between the Lax-Hopf algorithm and the analytical solution of problem (287).

The solutions at times $t = 0$, $t = 5$, $t = 10$ and $t = 15$ are represented in the upper left, upper right, lower left and lower right subfigures respectively. In each of these subfigures, the analytical solution is represented by a dashed line and the solution yielded by the Lax-Hopf algorithm is represented using dots. The difference between the two solutions is of the order of machine error and thus not visible on these figures.

As can be seen in Figure 8.2, the numerical solution of the LWR PDE using the Lax-Hopf algorithm is identical to the analytical solution computed by the method of characteristics in [71]. In addition to its high accuracy, the Lax-Hopf algorithm is not limited by the *Courant Friedrichs Lewy* (CFL) time step size condition inherent to many finite difference schemes and can thus compute the solution at a given time faster than finite difference schemes, such as the Godunov scheme.

The Godunov scheme is only stable when the CFL condition $v\Delta t \leq \Delta x$ is satisfied, where Δt and Δx represent the discretized time and space steps. We consider the mixed initial-boundary-internal conditions problem (287) as previously and compute the solution at time $t = 15$ using the Godunov scheme and the Lax-Hopf algorithm, for different space

resolutions Δx. The computational times are shown in Figure 8.3. For fairness of the comparison, all algorithms presented here have been implemented in the same programming language (`Matlab`) and run on the same platform (`Thinkpad T61` running `Windows XP`).

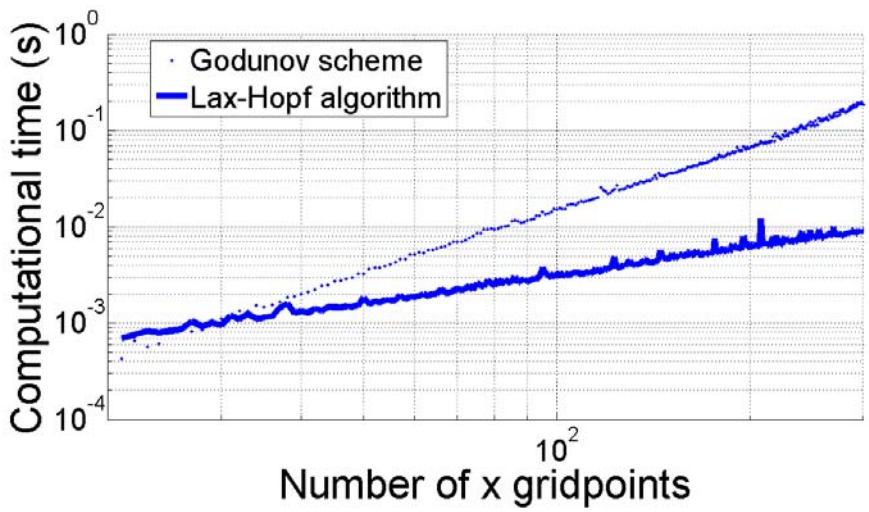


Figure 8.3. Computational time comparison between the Lax-Hopf algorithm and the Godunov scheme (287).

This figure represents the time required to compute the solution of problem (287) at time $t = 15$, using both the Godunov scheme (dots) and the Lax-Hopf algorithm (dashed line).

Figure 8.3 shows that the Lax-Hopf algorithm is significantly faster than the Godunov scheme when high accuracy is required. Indeed, the Lax-Hopf algorithm can compute the solution at time $t = 15$ using only the knowledge of the initial and boundary conditions. In contrast, the Godunov scheme has to compute the solution for each time step Δt, which is upper-constrained by the CFL condition and thus cannot be arbitrary large.

25.8.3. Comparison with standard numerical schemes

The striking difference in terms of computational cost between the Lax-Hopf algorithm and any finite difference scheme, such as the Lax-Friedrichs scheme, is that one does not need intermediate computations for times $M \in \{1, \dots, n_t\}$ to compute the solution at time step n_t. In other words, no iteration is needed to compute the value of the solution at any given time.

Another difference is that the computational cost of the Lax-Hopf algorithm is related to the number of piecewise affine elements in the initial, boundary and internal conditions only. In particular, the computational time required to solve a given problem depends upon its complexity (*i.e.* the total number of piecewise affine elements in the definition of the piecewise affine initial, boundary and internal conditions). In finite difference schemes however, the computational time is independent of the complexity of the problem to solve.

Unlike finite difference schemes, the solution computed using the Lax-Hopf algorithm is (up to machine accuracy) exact. Indeed, the formulae (238), (249), (261), (273) and (274) are closed form and the minimization process used in the Lax-Hopf algorithm yields exact results.

Note that other computational methods such as front tracking methods [80,122,193] can also be used to explicitly compute solutions to conservation laws, from which the HJ PDE (195) is derived. However, the Lax-Hopf algorithm is different, since it can be applied to a general concave Hamiltonian, is not event-based and does not require the explicit computation of shockwaves propagation.

26. Hamilton-Jacobi PDEs: Convex formulations of the model constraints

This chapter presents the derivation of the model constraints as convex inequalities on systems modeled by HJ PDEs. We derive the model constraints in section 26.1 and present some important properties of the model constraints in section 26.2. Using the Lax-Hopf formula, we show that the model constraints are convex, and can be written explicitly. The nature of the model constraints also imply an important monotonicity property with respect to new data, which states that adding new data into the estimation problem can only increase the accuracy of the solution.

26.1. Model constraints for well-posedness

In this section, we investigate the constraints that must apply on a general set of value conditions to ensure that the solution to the HJ PDE satisfies all prescribed value conditions. One of the specificities of the HJ PDE (195) investigated here is the fact that the solution itself may not reflect the value conditions that are imposed on it. Indeed, an arbitrary set of value conditions is said to apply in the *strong sense* if the solution is identical to the set of value conditions (on their respective domains of definition) and in the *weak sense* if at least one of the value conditions does not apply everywhere. In the following, we determine the conditions on the value conditions and on the Hamiltonian $\psi(\cdot)$ that ensure that *all* value conditions apply in the strong sense.

For this, we first have to define a binary relation that characterizes the order between general concave or convex functions.

Definition 1. [Hypographical and epigraphical characterizations of pointwise inequality between functions] Let $\mathcal{H}yp(\cdot)$ denote the hypograph of a function and $\mathcal{E}pi(\cdot)$ its epigraph. Let two concave functions $si_1(\cdot)$ and $\psi_2(\cdot)$ be given. The binary relation of inequality between these functions is defined by:

$$\psi_1(\cdot) \leq \psi_2(\cdot) \Leftrightarrow \mathcal{H}yp(\psi_1) \subset \mathcal{H}yp(\psi_2) \\ \Leftrightarrow \begin{cases} \mathrm{Dom}(\psi_1) \subset \mathrm{Dom}(\psi_2) \quad \text{and} \\ \forall p \in \mathrm{Dom}(i_1), \quad \psi_1(p) \leq \psi_2(p) \end{cases} \tag{288}$$

Let two convex functions $\varphi_1(\cdot)$ and $\varphi_2(\cdot)$ be given. The binary relation of inequality between these functions is defined by:

$$\varphi_1(\cdot) \leq \varphi_2(\cdot) \Leftrightarrow \mathcal{E}pi(\varphi_1) \supset \mathcal{E}pi(\varphi_2) \\ \Leftrightarrow \begin{cases} \mathrm{Dom}(\varphi_1) \supset \mathrm{Dom}(\varphi_2) \quad \text{and} \\ \forall u \in \mathrm{Dom}(varphi_2), \quad \varphi_1(u) \leq \varphi_2(u) \end{cases} \tag{289}$$

We now define the concept of *true state*, *i.e.* the actual value of the Moskowitz function, which plays a particular role in our estimation problem.

Definition 2. [True state] The true state $\overline{\boldsymbol{M}}(\cdot,\cdot)$ represents the state of the system, which could be obtained if measured by errorless sensors covering the entire space-time domain $[0, t_{max}] \times X$.

Some experimental data sets such as the *NGSIM* data [19] enable us to derive the true state function $\overline{\boldsymbol{M}}(\cdot,\cdot)$. An example of true state function $\overline{\boldsymbol{M}}(\cdot,\cdot)$ derived from the NGSIM data is illustrated in Figure 1.1.

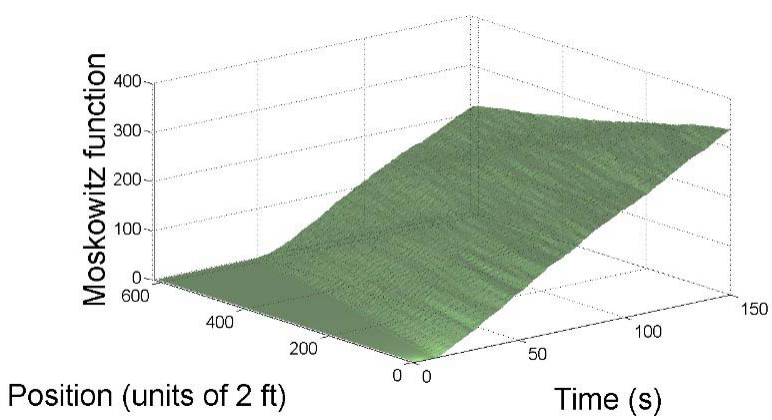


Figure 1.1. NGSIM experimental data.

This figure represents of the experimental Moskowitz surface obtained from the NGSIM data.

In most real time applications, $\overline{\boldsymbol{M}}(\cdot,\cdot)$ is not known, since measuring it requires a sensor

observing the state of traffic on the whole spatial domain and for all times. In the NGSIM dataset [19] for instance, the true state $\overline{\boldsymbol{M}}(\cdot,\cdot)$ was observed by using a camera filming a highway section from above. This data however required post-processing and was not available in real time.

Our data assimilation framework requires the following assumption on the true state function $\overline{\boldsymbol{M}}(\cdot,\cdot)$.

Fact 3. [Mathematical properties of the state] The true state $\overline{\boldsymbol{M}}(\cdot,\cdot)$ is assumed to be Lipschitz-continuous [128,129].

Note that the Lipschitz continuity of $\overline{\boldsymbol{M}}(\cdot,\cdot)$ implies the existence almost everywhere and boundedness of the flow $\frac{\partial \overline{\boldsymbol{M}}(t,x)}{\partial t}$ and the density $-\frac{\partial \overline{\boldsymbol{M}}(t,x)}{\partial x}$. This assumption is true for most physical systems, including highway traffic modeling [128]. Note also that no assumption is made that $\overline{\boldsymbol{M}}(\cdot,\cdot)$ satisfies the HJ PDE (195) exactly, which is in general true for most physical systems (*i.e.* their state does not satisfy a model perfectly).

The true state function enables the definition of corresponding *true value conditions*, which will be later shown to satisfy specific constraints.

Definition 4. [True value condition] Let $\overline{\boldsymbol{M}}(\cdot,\cdot)$ denote the true state of the system. A *true value condition* $\overline{\boldsymbol{c}}(\cdot,\cdot)$ is a function defined on a subset of $[0, t_{max}] \times X$ and satisfying:

$$\overline{\boldsymbol{c}}(t,x) := \begin{cases} \overline{\boldsymbol{M}}(t,x) & \text{if}(t,x) \in \text{Dom}(\overline{\boldsymbol{c}}) \\ +\infty & \text{otherwise} \end{cases} \tag{290}$$

The following property holds:

Proposition 5. [Minimum of true value conditions] Let $\overline{\boldsymbol{c}}_j(\cdot,\cdot)_{j\in J}$ be a finite family of true value conditions, as in definition 4. The minimum $\overline{\boldsymbol{c}}(\cdot,\cdot) := \min_{j\in J}(\overline{\boldsymbol{c}}_j(\cdot,\cdot))$ of the true value conditions $\overline{\boldsymbol{c}}_j(\cdot,\cdot)$ is also a true value condition, whose domain of definition is given by:

$$\text{Dom}(\overline{\boldsymbol{c}}) := \bigcup_{j\in J} \text{D om}(\overline{\boldsymbol{c}}_j) \tag{291}$$

In addition, we have the following property:

$$\forall j \in J, \quad \forall (t,x) \in \text{Dom}(\overline{\boldsymbol{c}}_j), \quad \overline{\boldsymbol{c}}(t,x) = \overline{\boldsymbol{c}}_j(t,x) \tag{292}$$

Proof — The proof of this proposition is straightforward and follows directly from

definition 4. ■

A value condition represents some knowledge of the true state of the system, which is used in conjunction with the HJ PDE (195) to construct an *estimated state* of the system.

Definition 6. [Estimated state] Let a value condition $\boldsymbol{c}(\cdot,\cdot)$ be defined as in definition 1. The estimated state is defined as the solution (210) associated with $\overline{\boldsymbol{c}}(\cdot,\cdot)$ and the Hamiltonian $\psi(\cdot)$ and denoted by $\boldsymbol{M}_{\overline{\boldsymbol{c}},\psi}(\cdot,\cdot)$.

Note the $\psi(\cdot)$ index in the definition above, which as previously indicates that the value of the solution $\boldsymbol{M}_{\boldsymbol{c},\psi}(\cdot,\cdot)$ associated with the value condition $\boldsymbol{c}(\cdot,\cdot)$ depends (implicitly) on the Hamiltonian ofthe HJ PDE. As a consequence of theorem 7, the estimated state $\boldsymbol{M}_{\boldsymbol{c},\psi}(\cdot,\cdot)$ is a solution to (195) in the B-J/F sense. However, the estimated state does not necessarily satisfy the true value condition that we want to impose on it [104,105].

In the following section, we find the conditions on a finite set of value conditions $(\boldsymbol{c}_j(\cdot,\cdot))_{j\in J}$ such that all of these value conditions apply in the strong sense when solving (195). In this case, we say that the value conditions are *compatible with the model*. The corresponding constraints on the value conditions are called *model compatibility constraints*. The value conditions $(\boldsymbol{c}_j(\cdot,\cdot))_{j\in J}$ satisfy the model compatibility constraints if and only if the following equality is true:

$$\forall j \in J,\ \forall (t,x) \in \mathrm{Dom}(\boldsymbol{c}_j),\quad \boldsymbol{M}_{\boldsymbol{c}_j,\psi}(t,x) = \boldsymbol{c}_j(t,x) \tag{293}$$

The following section presents an equivalent formulation of (293), based on the properties of the solution (210), which results in algebraic conditions to be verified for (293) to be satisfied.

26.1.1. Compatibility conditions

Because of the inf-morphism property (226) and the Lax-Hopf formula (210), the equality (293) can be decomposed as a set of inequalities known as *compatibility conditions*, which we now express.

Proposition 7. [Compatibility conditions] Let us define a finite family of value condition functions $\boldsymbol{c}_j(\cdot,\cdot)$, $j \in J$ as in definition 1 and their minimum $\boldsymbol{c}(\cdot,\cdot) := \min_{j\in J} \boldsymbol{c}_j(\cdot,\cdot)$. The estimated state $\boldsymbol{M}_{\boldsymbol{c},\psi}(\cdot,\cdot)$ associated with $\overline{\boldsymbol{c}}(\cdot,\cdot)$ satisfies the property (293) if and only if the following set of inequalities is satisfied:

$$\boldsymbol{M}_{\boldsymbol{c}_i,\psi}(t,x) \geq \boldsymbol{c}_j(t,x),\quad \forall (t,x) \in \mathrm{Dom}(\boldsymbol{c}_j),\quad \forall i \in J,\quad \forall j \in J \tag{294}$$

Proof — Let us first start from (293). By definition of $\boldsymbol{c}(\cdot,\cdot)$, we have that $(t,x) \in \mathrm{Dom}(\boldsymbol{c})$ if

and only if $(t,x) \in \mathrm{Dom}(\boldsymbol{c}_j)$ for some $j \in J$. Hence, using equation (292), we can equivalently rewrite (293) as:

$$\forall j \in J, \quad \forall (t,x) \in \mathrm{Dom}(\boldsymbol{c}_j), \quad \boldsymbol{M}_{\boldsymbol{c},\psi}(t,x) = \boldsymbol{c}_j(t,x) \tag{295}$$

We now prove that (295) implies (294). The inf-morphism property (226) implies that the estimated state $\boldsymbol{M}_{\boldsymbol{c},\psi}(\cdot,\cdot)$ associated with the value condition $\boldsymbol{c}(\cdot,\cdot)$ is the minimum of the estimated states $\boldsymbol{M}_{\boldsymbol{c}_i,\psi}(\cdot,\cdot)$ associated with the value conditions $\boldsymbol{c}_i(\cdot,\cdot)$:

$$\boldsymbol{M}_{\boldsymbol{c},\psi}(t,x) = \min_{i \in J} \ \boldsymbol{M}_{\boldsymbol{c}_i,\psi}(t,x) \tag{296}$$

Hence, the condition (295) implies the constraints (294).

Reciprocally, we prove that (294) implies the equality (295). When (294) is satisfied, equation (296) implies that $\boldsymbol{M}_{\boldsymbol{c},\psi}(t,x) \geq \boldsymbol{c}_j(t,x)$ for all $j \in J$ and for all $(t,x) \in \mathrm{Dom}(\boldsymbol{c}_j)$. The converse inequality is obtained from the Lax-Hopf formula (210):

$$\boldsymbol{M}_{\boldsymbol{c}_j,\psi}(t,x) = \inf_{(u,T) \in \mathrm{Dom}(\varphi^*) \times \mathbb{R}_+} \left(\boldsymbol{c}_j(t-T, x+Tu) + T\varphi^*(u) \right) \tag{297}$$

By taking $T = 0$ and $u \in \mathrm{Dom}(\varphi^*)$ in (297), we have that $\forall j \in J, \ \forall (t,x) \in \mathrm{Dom}(\boldsymbol{c}_j), \ \boldsymbol{M}_{\boldsymbol{c}_j,\psi}(t,x) \leq \boldsymbol{c}_j(t,x)$. By the inf-morphism property, this last inequality implies $\forall j \in J, \ \forall (t,x) \in \mathrm{Dom}(\boldsymbol{c}_j), \ \boldsymbol{M}_{\boldsymbol{c},\psi}(t,x) \leq \boldsymbol{c}_j(t,x)$ which completes the proof. ■

We assumed in this section that $\psi(\cdot)$ was given. In the next section, we define conditions on $\psi(\cdot)$ and the true state $\overline{\boldsymbol{M}}(\cdot,\cdot)$ which ensure that the compatibility conditions (294) associated with true value condition $\overline{\boldsymbol{c}}_j(\cdot,\cdot)$ are automatically satisfied, *i.e.* the equality (293) is satisfied. In general, the true state $\overline{\boldsymbol{M}}(\cdot,\cdot)$ not given, but some of its properties are known. Thus, the following results amount to finding the proper Hamiltonian $\psi(\cdot)$ such that the compatibility conditions (294) are satisfied.

26.1.2. Sufficient conditions on the Hamiltonian for compatibility of true value conditions

While the true state is generally unknown, the properties of its derivatives have been extensively studied in the literature [174,226,285]. Note that by (194), the derivatives of the true state

function represent the true density and true flow functions. We assume that we can measure some values of the derivatives of $\overline{\boldsymbol{M}}(\cdot,\cdot)$ which are representative of the range of physical measurements of the system. Using these measurements, we define a particular class of Hamiltonians as follows.

Proposition 8. [Upper estimate of the Hamiltonian]. For a given true state $\overline{\boldsymbol{M}}(\cdot,\cdot)$, we define the set $B(\overline{\boldsymbol{M}})$ as follows:

$$B(\overline{\boldsymbol{M}}) := \left\{ \left(-\frac{\partial \overline{\boldsymbol{M}}(t,x)}{\partial x}, \frac{\partial \overline{\boldsymbol{M}}(t,x)}{\partial t} \right), \quad (t,x) \in [0, t_{max}] \times X \quad \text{such that} \quad \overline{\boldsymbol{M}}(\cdot,\cdot) \quad \text{is} \quad \text{differentiable} \right\}$$

There exists a concave and upper semicontinuous function $\psi_0(\cdot)$ such that:

$$B(\overline{\boldsymbol{M}}) \subset \mathcal{H}yp(\psi_0) \tag{298}$$

Proof — Recall that the true state is Lipschitz-continuous by assumption. Thus, its derivatives are defined almost everywhere and bounded, which implies the boundedness of $B(\overline{\boldsymbol{M}})$. Hence, we can choose for $\psi_0(t)$ any concave function greater than the upper concave envelope of $B(\overline{\boldsymbol{M}})$. ■

Note that the choice of a function $\psi_0(\cdot)$ compatible with (298) is not unique. An example of choice of $\psi_0(\cdot)$ satisfying (298) is illustrated in Figure 1.2 later.

The conditions (294) are necessarily satisfied for a true value condition $\overline{\boldsymbol{c}}(\cdot,\cdot)$ and for a Hamiltonian $\psi_0(\cdot)$ satisfying (298), as shown in the following proposition.

Proposition 9. [Compatibility property for true value conditions] Let us define a finite set of true value condition functions $\overline{\boldsymbol{c}}_j(\cdot,\cdot)$, $j \in J$ as in definition 4, a concave and upper semicontinuous Hamiltonian $\psi_0(\cdot)$ satisfying (298) and its associated convex transform φ_0^* as in (197). Let us also define the set of solutions $\boldsymbol{M}_{\overline{\boldsymbol{c}}_j,\psi_0}(\cdot,\cdot)$ associated with $\overline{\boldsymbol{c}}_j(\cdot,\cdot)$ as in (210). Given these assumptions, the set of inequalities (294) are satisfied.

Proof — In the present case, the compatibility conditions (294) can be written as:

$$\boldsymbol{M}_{\overline{\boldsymbol{c}}_i,\psi_0}(t,x) \geq \overline{\boldsymbol{c}}_j(t,x), \quad \forall (t,x) \in \mathrm{Dom}(\overline{\boldsymbol{c}}_j), \quad \forall i \in J, \quad \forall j \in J \tag{299}$$

Let us fix $i \in J$, $j \in J$ and $(t,x) \in \mathrm{Dom}(\overline{\boldsymbol{c}}_j)$.

We first express $\boldsymbol{M}_{\overline{\boldsymbol{c}}_i,\psi_0}(t,x)$ in terms of $\overline{\boldsymbol{c}}_i(\cdot,\cdot)$ using the Lax-Hopf formula (210):

$$\boldsymbol{M}_{\overline{\boldsymbol{c}}_i,\psi_0}(t,x) = \inf_{(u,T)\in \mathrm{Dom}(\varphi_0^*)\times\mathbb{R}_+} (\overline{\boldsymbol{c}}_i(t-T,x+Tu) + T\varphi_0^*(u)) \tag{300}$$

Since $(t,x) \in \mathrm{Dom}(\overline{\boldsymbol{c}}_j)$, we have by definition 4 that $\overline{\boldsymbol{c}}_j(t,x) = \overline{\boldsymbol{M}}(t,x)$. Hence, we can write the inequality (299) which we want to prove as:

$$\inf_{(T,u)\in[0,t_{max}]\times \mathrm{Dom}(\varphi_0^*)} (\overline{\boldsymbol{c}}_i(t-T,x+Tu) + T\varphi_0^*(u)) \geq \overline{\boldsymbol{M}}(t,x) \tag{301}$$

By definition 4, we have that $\overline{\boldsymbol{c}}_i(t-T,x+Tu) \geq \overline{\boldsymbol{M}}(t-T,x+Tu)$ for all $(T,u) \in [0,t_{max}] \times \mathrm{Dom}(\varphi_0^*)$. Indeed, $\overline{\boldsymbol{c}}_i(t-T,x+Tu) = \overline{\boldsymbol{M}}(t-T,x+Tu)$ if $(t-T,x+Tu) \in \mathrm{Dom}(\overline{\boldsymbol{c}}_i)$ and that $\overline{\boldsymbol{c}}_i(t-T,x+Tu) = +\infty$ otherwise. Hence, if the equation (302) below is satisfied, then inequality (301) will be automatically true:

$$\inf_{(T,u)\in[0,t_{max}]\times \mathrm{Dom}(\varphi_0^*)} \left(\overline{\boldsymbol{M}}(t-T,x+Tu) + T\varphi_0^*(u)\right) \geq \overline{\boldsymbol{M}}(t,x) \tag{302}$$

We now prove that (302) holds. Since $\overline{\boldsymbol{M}}(\cdot,\cdot)$ is Lipschitz-continuous by assumption and assuming that $\overline{\boldsymbol{M}}(\cdot,\cdot)$ is differentiable almost everywhere on $\{(t-\tau,x+\tau u),\ \ \tau \in [0,T]t\}$, we can write:

$$\overline{\boldsymbol{M}}(t-T,x+Tu) + T\varphi_0^*(u) - \overline{\boldsymbol{M}}(t,x) = \int_0^T \left(-\frac{\partial \overline{\boldsymbol{M}}(t-\tau,x+\tau u)}{\partial t} + u\frac{\partial \overline{\boldsymbol{M}}(t-\tau,x+\tau u)}{\partial x} + \varphi_0^*(u)\right) d\tau \tag{303}$$

Since $\psi_0(\cdot)$ is concave and upper semicontinuous, it is equal to its Legendre-Fenchel biconjugate [59]. Hence, we have that $\psi_0(\rho) = \inf_{u\in \mathrm{Dom}(\varphi_0^*)}(-\rho u + \varphi_0^*(u))$ and thus that $\psi_0(\rho) \leq -\rho u + \varphi_0^*(u)$ for all $u \in \mathrm{Dom}(\varphi_0^*)$. This result enables us to derive the following inequality from equation (303):

$$\overline{\boldsymbol{M}}(t-T,x+Tu) + T\varphi_0^*(u) - \overline{\boldsymbol{M}}(t,x) \geq \int_0^T \left(-\frac{\partial \overline{\boldsymbol{M}}(t-\tau,x+\tau u)}{\partial t} + \psi_0\left(-\frac{\partial \overline{\boldsymbol{M}}(t-\tau,x+\tau u)}{\partial x}\right)\right) d\tau \tag{304}$$

Using (298), we have that $-\frac{\partial \overline{\boldsymbol{M}}(t-\tau,x+\tau u)}{\partial t}+\psi_0\left(-\frac{\partial \overline{\boldsymbol{M}}(t-\tau,x+\tau u)}{\partial x}\right)\geq 0$ for all $(\tau,u)\in[0,T]\times \mathrm{Dom}(\varphi_0^*)$. Since $T>0$, the right hand side of equation (304) is nonnegative, which implies the following inequality:

$$\forall(T,u)\in\mathbb{R}_+\times\mathrm{Dom}(\varphi_0^*),\quad \overline{\boldsymbol{M}}(t-T,x+Tu)+T\varphi_0^*(u)-\overline{\boldsymbol{M}}(t,x)\geq 0 \tag{305}$$

Equation (302) is obtained from equation (305) by taking the infimum over $(T,u)\in\mathbb{R}_+\times \mathrm{Dom}(\varphi_0^*)$, which completes the proof. Note that if $\overline{\boldsymbol{M}}(\cdot,\cdot)$ is not differentiable almost everywhere on the set $\{(t-\tau,x+\tau u),\ \ \tau\in[0,T]\}$, it will be differentiable on the set $\{(t-\tau,x+\delta x+\tau u),\ \ \tau\in[0,T]\}$ for a small δx, by Lipschitz-continuity. Hence, we have that $\overline{M}(t-T,x+\delta x+Tu)+T\varphi_0^*(u)-\overline{\boldsymbol{M}}(t,x+\delta x)\geq 0$, which implies $\overline{\boldsymbol{M}}(t-T,x+\delta x+Tu)+T\varphi_0^*(u)-\overline{\boldsymbol{M}}(t,x+\delta x)\geq 0$ by (Lipschitz) continuity of $\overline{\boldsymbol{M}}(\cdot,\cdot)$. ■

Proposition 9 thus implies that the estimated state $\boldsymbol{M}_{\overline{\boldsymbol{c}},\psi_0}(\cdot,\cdot)$ associated with any true value condition $\overline{\boldsymbol{c}}(\cdot,\cdot)$ satisfies the imposed true value condition when the Hamiltonian $\psi_0(\cdot)$ satisfies (298).

Because of the order-preserving property of (197), the constraints (294) are satisfied for a given Hamiltonian $\psi_1(\cdot)$ only if they are also satisfied for any Hamiltonian $\psi_2(\cdot)$ greater than $\psi_1(\cdot)$, as expressed by the following proposition.

Proposition 10. [Hamiltonian inequality property] Let us define a finite set of true value conditions $\overline{\boldsymbol{c}}_j(\cdot,\cdot)$, $j\in J$ as in definition 4. Let us also define two concave and upper semicontinuous Hamiltonians $\psi_1(\cdot)$ and $\psi_2(\cdot)$, satisfying $\psi_1(\cdot)\leq\psi_2(\cdot)$. The associated solutions $\boldsymbol{M}_{\overline{\boldsymbol{c}}_j,\psi_1}(\cdot,\cdot)$ and $\boldsymbol{M}_{\overline{\boldsymbol{c}}_j,\psi_2}(\cdot,\cdot)$ associated with the true value condition $\overline{\boldsymbol{c}}_j(\cdot,\cdot)$ are defined by (210). We have the following property:

$$\boldsymbol{M}_{\overline{\boldsymbol{c}}_i,\psi_1}(t,x)\geq\overline{\boldsymbol{c}}_j(t,x),\quad\forall(t,x)\in\mathrm{Dom}(\overline{\boldsymbol{c}}_j),\quad\forall i\in J,\quad\forall j\in J \tag{306}$$

implies

$$\boldsymbol{M}_{\overline{\boldsymbol{c}}_i,\psi_2}(t,x)\geq\overline{\boldsymbol{c}}_j(t,x),\quad\forall(t,x)\in\mathrm{Dom}(\overline{\boldsymbol{c}}_j),\quad\forall i\in J,\quad\forall j\in J \tag{307}$$

Proof — The proof of this proposition is a direct consequence of the following property:

$$\boldsymbol{M}_{\overline{\boldsymbol{c}}_i,\psi_1}(t,x)\leq\boldsymbol{M}_{\overline{\boldsymbol{c}}_i,\psi_2}(t,x),\quad\forall(t,x)\in[0,t_{max}]\times X,\quad\forall i\in J \tag{308}$$

Indeed, since $\psi_1(\cdot) \leq \psi_2(\cdot)$, we have that $pu + \psi_1(p) \leq pu + \psi_2(p) \quad \forall (p,u) \in \mathbb{R}^2$. Hence, the convex transforms $\varphi_1^*(\cdot)$ and $\varphi_2^*(\cdot)$ respectively associated with $\psi_1(\cdot)$ and $\psi_2(\cdot)$ satisfy $\varphi_1^*(\cdot) \leq \varphi_2^*(\cdot)$.

Let us fix $(t,x,j) \in [0,t_{max}] \times X \times J$. The solutions $\boldsymbol{M}_{\overline{\boldsymbol{c}}_i,\psi_1}(t,x)$ and $\boldsymbol{M}_{\overline{\boldsymbol{c}}_i,\psi_2}(t,x)$ can be expressed using the following Lax-Hopf formulae:

$$\begin{aligned} \boldsymbol{M}_{\overline{\boldsymbol{c}}_i,\psi_1}(t,x) &= \inf_{(u,T)\in \mathrm{Dom}(\varphi_1^*)\times\mathbb{R}_+} (\overline{\boldsymbol{c}}_i(t-T, x+Tu) + T\varphi_1^*(u)) \\ \boldsymbol{M}_{\overline{\boldsymbol{c}}_i,\psi_2}(t,x) &= \inf_{(u,T)\in \mathrm{Dom}(\varphi_2^*)\times\mathbb{R}_+} (\overline{\boldsymbol{c}}_i(t-T, x+Tu) + T\varphi_2^*(u)) \end{aligned} \tag{309}$$

Since $\varphi_1^*(\cdot) \leq \varphi_2^*(\cdot)$, we have that $\mathrm{Dom}(\varphi_2^*) \subset \mathrm{Dom}(\varphi_1^*)$. Hence, equation (309) implies (308) and thus, (306) implies (307). ■

In consequence, the smallest concave function satisfying (298), illustrated in Figure 1.2 plays a particular role in our problem.

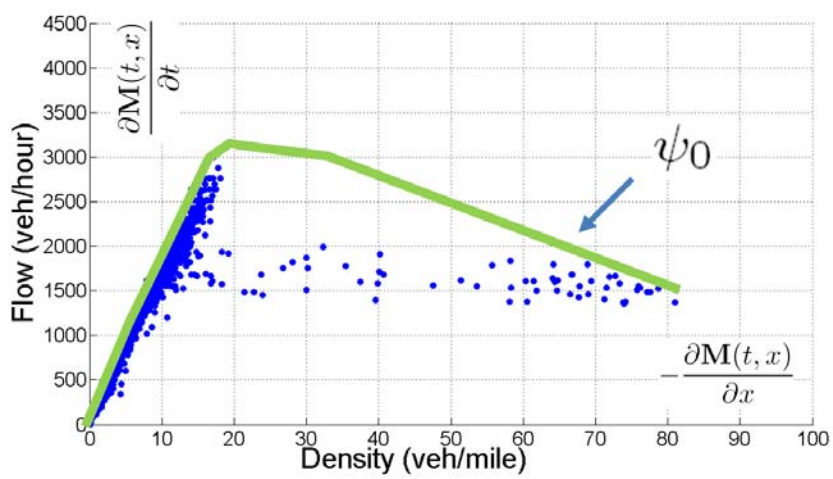


Figure 1.2. Illustration of an upper estimate function $\boldsymbol{\psi_0}(\cdot)$.

In this figure, the horizontal axis represents the density and the vertical axis the flow. The scatter plot represents the values of flow and density obtained from experimental traffic flow data [22]. Each point in this plot is a given $\left(-\frac{\partial \overline{\boldsymbol{M}}(t,x)}{\partial x}, \frac{\partial \overline{\boldsymbol{M}}(t,x)}{\partial t}\right)$ for some $(t,x) \in [0,t_{max}] \times X$. A typical example of upper estimate function $\psi_0(\cdot)$ is the upper concave envelope of the points, represented by a dashed line and which satisfies (298).

Proposition 11. [Smallest concave upper estimate] Let $\overline{\boldsymbol{M}}$ be given and let $B(\overline{\boldsymbol{M}})$ be defined as in proposition 8. Let $\mathcal{C}$ be the set of upper semicontinuous concave functions from $\mathbb{R}$ to $\mathbb{R}$ and let us define the set of functions $\mathcal{A}$ by:

$$\mathcal{A} := \{\psi \in \mathcal{C} \text{ such that } B(\overline{\boldsymbol{M}}) \subset \mathcal{H}yp(\psi)\} \tag{310}$$

Let us define the function $\psi_{\inf}(\cdot)$ as:

$$\mathcal{H}yp(\psi_{inf}) := \bigcap_{\psi \in \mathcal{A}} \mathcal{H}\, yp(\psi) \tag{311}$$

The function $\psi_{inf}(\cdot)$ defined by (311) is the smallest element of $\mathcal{A}$.

Proof — The set $\mathcal{A}$ is not empty by proposition 8 and thus the function $\psi_{\inf}(\cdot)$ defined by (311) exists. We now prove that $\psi_{inf}(\cdot)$ is the smallest element of $\mathcal{A}$. Let $\psi(\cdot) \in \mathcal{A}$. Since $\psi(\cdot)$ is concave, upper semicontinuous and satisfies (298), its hypograph is closed, convex and contains the set $B(\overline{\boldsymbol{M}})$. By (311), the hypograph of $\psi_{inf}(.)$ is thus closed, convex and contains $B(\overline{\boldsymbol{M}})$ since it is the (infinite) intersection of closed and convex sets containing $B(\overline{\boldsymbol{M}})$. Hence, ψ_{inf} is concave, upper semicontinuous and satisfies (298), which implies $\psi_{inf} \in \mathcal{A}$. The function $\psi_{inf}(\cdot)$ is also the smallest element of $\mathcal{A}$, since any element $\psi(\cdot)$ of $\mathcal{A}$ satisfies $\mathcal{H}yp(\psi_{inf}(\cdot)) \subset \mathcal{H}yp(\psi(\cdot))$ by (311). ■

Proposition 12. [Minimal conditions] Let $\mathcal{A}$ be defined as in proposition 11 and let $\psi_{inf}(\cdot)$ be defined as in as in (311). Let $\psi(\cdot) \in \mathcal{A}$ and let us define a finite set of true value conditions $\overline{\boldsymbol{c}}_j(\cdot,\cdot)$, $j \in J$ and their associated solutions $\boldsymbol{M}_{\overline{\boldsymbol{c}}_j,\psi_{inf}}(\cdot,\cdot)$ and $\boldsymbol{M}_{\overline{\boldsymbol{c}}_j,\psi}(\cdot,\cdot)$ as in (210). Given the above definitions, we have the following property:

$$\boldsymbol{M}_{\overline{\boldsymbol{c}}_i,\psi}(t,x) \geq \overline{\boldsymbol{c}}_j(t,x), \quad \forall (t,x) \in \mathrm{Dom}(\overline{\boldsymbol{c}}_j), \quad \forall i \in J, \quad \forall j \in J, \quad \forall \psi(\cdot) \in \mathcal{A} \tag{312}$$

if and only if

$$\boldsymbol{M}_{\overline{\boldsymbol{c}}_i,\psi_{\inf}} \quad (t,x) \geq \overline{\boldsymbol{c}}_j(t,x), \quad \forall (t,x) \in \mathrm{Dom}(\overline{\boldsymbol{c}}_j), \quad \forall i \in J, \quad \forall j \in J \tag{313}$$

Proof — The conditions (312) imply (313), since $\psi_{inf}(\cdot) \in \mathcal{A}$ by proposition 11. Conversely, the conditions (313) imply (312), by proposition 10, remarking that $\mathcal{H}yp(\psi_{inf}) \subset \mathcal{H}yp(\psi)$. ■

Proposition 12 enables the verification of the conditions (312) for a true value condition $\overline{\boldsymbol{c}}(\cdot,\cdot)$ and for all Hamiltonians $\psi(\cdot)$ satisfying (298) using the conditions (313) only.

We now present some important properties of the model compatibility constraints.

26.2. Properties of the model compatibility constraints

26.2.1. Concavity property of the solutions with respect to their coefficients

Because of the Lax-Hopf formula (210), the solutions associated with affine initial, boundary and internal conditions have a concavity property with respect to some of their coefficients, which we now present.

Proposition 1. [Concavity property of the solution associated with an affine initial condition] The solution $\boldsymbol{M}_{\mathcal{M}_{0,i}}(\cdot,\cdot)$ associated with the affine initial condition (229) is a concave function of the coefficients a_i and b_i.

Proof — The Lax-Hopf formula (210) associated with the solution $\boldsymbol{M}_{\mathcal{M}_{0,i}}(\cdot,\cdot)$ can be written as:

$$\boldsymbol{M}_{\mathcal{M}_{0,i}}(t,x) = \inf_{u\in\mathrm{Dom}(\varphi^*) \text{ s. t. } (x+tu)\in[\overline{\alpha}_i,\overline{\beta}_i]} (a_i(x+tu)+b_i+t\varphi^*(u)) \tag{314}$$

Let us fix $(t,x,u)\in[0,t_{max}]\times X\times \mathrm{Dom}(\varphi^*)$. The function $f(\cdot,\cdot)$ defined as $f(a_i,b_i) = a_i(x+tu)+b_i+t\varphi^*(u)$ is concave (indeed, affine). Hence, the solution $\boldsymbol{M}_{\mathcal{M}_{0,i}}(t,x)$ is a concave function of (a_i,b_i), since it is the infimum of concave functions of (a_i,b_i) [78,288]. ■

Proposition 2. [Concavity property of the solution associated with an affine upstream boundary condition] The solution $\boldsymbol{M}_{\gamma_j}(\cdot,\cdot)$ associated with the affine upstream boundary condition (239) is a concave function of the coefficients c_j and d_j.

Proof — The Lax-Hopf formula (210) associated with the solution $\boldsymbol{M}_{\gamma_j}(\cdot,\cdot)$ can be written as:

$$\boldsymbol{M}_{\gamma_j}(t,x) = \inf_{T\in\left[-\frac{\xi-x}{v^\flat},+\infty\right[\cap\left[t-\overline{\gamma}_{j+1},t-\overline{\gamma}_j\right]} \left(c_j(t-T)+d_j+T\varphi^*\left(\frac{\xi-x}{T}\right)\right) \tag{315}$$

Let us fix $(t,x,T)\in[0,t_{max}]\times X\times\left[-\frac{\xi-x}{v^\flat},+\infty\right[\cap\left[t-\overline{\gamma}_{j+1},t-\overline{\gamma}_j\right]$. The function $g(\cdot,\cdot)$ defined as $f(c_j,d_j) = c_j(t-T)+d_j+T\varphi^*\left(\frac{\xi-x}{T}\right)$ is concave (indeed, affine). Hence, the solution $\boldsymbol{M}_{\gamma_j}(t,x)$ is a concave function of (c_j,d_j), since it is the infimum of concave functions of (c_j,d_j) [78,288]. ■

Proposition 3. [Concavity property of the solution associated with an affine downstream boundary condition] The solution $\boldsymbol{M}_{\beta_k}(\cdot,\cdot)$ associated with the affine downstream boundary condition (251) is a concave function of the coefficients e_k and f_k.

Proof — The Lax-Hopf formula (210) associated with the solution $\boldsymbol{M}_{\beta_k}(\cdot,\cdot)$ can be written as:

$$\boldsymbol{M}_{\beta_k}(t,x) = \inf_{T\in\left[\frac{\chi-x}{\nu^{\#}},+\infty\right[\cap\left[t-\overline{\beta}_{k+1},t-\overline{\beta}_k\right]} \left(e_k(t-T)+f_k+T\varphi^*\left(\frac{\chi-x}{T}\right)\right) \tag{316}$$

Let us fix $(t,x,T)\in[0,t_{max}]\times X\times\left[\frac{\chi-x}{\nu^{\#}},+\infty\right[\cap\left[t-\overline{\beta}_{k+1},t-\overline{\beta}_k\right]$. The function $h(\cdot,\cdot)$ defined as $f(e_k,f_k)=e_k(t-T)+f_k+T\varphi^*\left(\frac{\chi-x}{T}\right)$ is concave (indeed, affine). Hence, the solution $\boldsymbol{M}_{\beta_k}(t,x)$ is a concave function of (e_k,f_k), since it is the infimum of concave functions of (e_k,f_k) [78,288]. ■

Proposition 4. [Concavity property of the solution associated with an affine internal condition] The solution $\boldsymbol{M}_{\mu_l}(\cdot,\cdot)$ associated with the internal condition (263) is a concave function of the coefficients g_l and h_l.

Proof — The Lax-Hopf formula (210) associated with the solution $\boldsymbol{M}_{\mu_l}(\cdot,\cdot)$ can be written [104,105] as:

$$\boldsymbol{M}_{\mu_l}(t,x) = \inf_{T\in\mathbb{R}_+\cap[t-\overline{\delta}_l,t-\overline{\gamma}_l]} g_l\,(t-T-\overline{\gamma}_l)+h_l+T\varphi^*\left(\frac{x_l+v_l(t-\overline{\gamma}_l-T)-x}{T}\right) \tag{317}$$

Let us fix $(t,x,T)\in[0,t_{max}]\times X\times\mathbb{R}_+$. The function $d(\cdot,\cdot)$ defined as $d(g_l,h_l):=g_l(t-T-\overline{\gamma}_l)+h_l+T\varphi^*\left(\frac{x_l+v_l(t-\overline{\gamma}_l-T)-x}{T}\right)$ is concave (indeed, affine). Hence, the solution $\boldsymbol{M}_{\mu_l}(t,x)$ is a concave function of (g_l,h_l), since it is the infimum of concave functions [78,288]. ■

26.2.2. Convex formulation of the model compatibility constraints

For the initial, boundary and internal conditions defined by (229), (239), (251) and (263), the model compatibility constraints (294) define constraints on the coefficients a_i, b_i, c_j, d_j, e_k, f_k, g_l, h_l. We now prove that these constraints are convex.

Proposition 5. [Convexity property of the model constraints] Let initial, boundary and internal conditions be defined as in (229), (239), (251) and (263), for $i\in I$, $j\in J$, $k\in K$ and $l\in L$ where I, J, K and L are finite sets. The model compatibility constraints (294) are convex inequalities in the coefficients a_i, b_i, c_j, d_j, e_k, f_k, g_l and h_l.

Proof — The constraints (294) are of the form:

$$\boldsymbol{M}_{\boldsymbol{c}_n}(t,x) \geq \boldsymbol{c}_m(t,x), \quad \forall (t,x) \in \mathrm{Dom}(\boldsymbol{c}_m), \quad \forall n \in N, \quad \forall m \in N \tag{318}$$

Let $(n,m) \in N^2$ and $(t,x) \in \mathrm{Dom}(\boldsymbol{c}_m)$. The quantity $\boldsymbol{c}_m(t,x)$ is an affine function of the coefficients a_i, b_i, c_j, d_j, e_k, f_k, g_l and h_l. In addition, $\boldsymbol{M}_{\boldsymbol{c}_n}(t,x)$ is a concave function of a_i, b_i, c_j, d_j, e_k, f_k, g_l and h_l. The constraint $\boldsymbol{M}_{\boldsymbol{c}_n}(t,x) \geq \boldsymbol{c}_m(t,x)$ can be written as $-\boldsymbol{M}_{\boldsymbol{c}_n}(t,x) + \boldsymbol{c}_m(t,x) \leq 0$ where $-\boldsymbol{M}_{\boldsymbol{c}_n}(t,x) + \boldsymbol{c}_m(t,x)$ is a convex function(as the sum of convex functions) of a_i, b_i, c_j, d_j, e_k, f_k, g_l and h_l. Hence, $\boldsymbol{M}_{\boldsymbol{c}_n}(t,x) \geq \boldsymbol{c}_m(t,x)$ is a convex constraint [78] in a_i, b_i, c_j, d_j, e_k, f_k, g_l and h_l. ■

Proposition 5 states that the inequality constraints (294) define a convex set in the space $(a_i, b_i, c_j, d_j, e_k, f_k, g_l, h_l)$. Using this property, we pose inverse modeling problems as convex optimization programs in chapter 27.

26.2.3. Monotonicity property of the model compatibility conditions

An important property of the model compatibility constraints (294) is their monotonicity with respect to new data, outlined in the following proposition.

Proposition 6. [Monotonicity property] Let a set of affine initial, boundary and internal conditions be defined as in (229), (239), (251) and (263) for $i \in I$, $j \in J$, $k \in K$ and $l \in L$ where I, J, K and L are finite sets. The convex set defined by the inequality constraints (318) is decreasing (in the sense of inclusion) as new initial, boundary or internal conditions are added.

Proof — The model compatibility constraints can be written as:

$$\boldsymbol{M}_{\boldsymbol{c}_n}(t,x) \geq \boldsymbol{c}_m(t,x), \quad \forall (t,x) \in \mathrm{Dom}(\boldsymbol{c}_m), \quad \forall n \in N, \quad \forall m \in N \tag{319}$$

Let $\mathcal{N} \subset \mathbb{R}^{|N|}$ be the convex set defined by (319). We now add a finite number of new value conditions $\boldsymbol{c}_p$, defined for $p \in P$. The model compatibility constraints become:

$$\boldsymbol{M}_{\boldsymbol{c}_n}(t,x) \geq \boldsymbol{c}_m(t,x), \quad \forall (t,x) \in \mathrm{Dom}(\boldsymbol{c}_m), \quad \forall n \in N \cup P, \quad \forall m \in N \cup P \tag{320}$$

Let $\mathcal{M} \subset \mathbb{R}^{|N|+|P|}$ be the convex set defined by (320). Since the constraints (320) imply (319), we the projection of $\mathcal{M}$ on $\mathbb{R}^{|N|}$ is a subset of $\mathcal{N}$, which completes the proof. ■

The above property is very important in practice, since it ensures that the feasible sets decreases in size when new data is added. Hence, the results of the estimation problems derived in the next chapter necessarily improve when new data is added, which is not the case for Monte-Carlo based estimation methods.

27. Hamilton-Jacobi PDEs: Applications of the new framework

In this chapter, we exploit the convexity of the model compatibility constraints (from chapter 26) to solve different estimation problems arising in traffic-flow engineering. For this, we first establish the relationship between measurement data and initial, boundary and internal conditions in section 27.1. We then instantiate the model compatibility constraints explicitly for triangular Hamiltonians in section 27.2. We also derive the corresponding measurement data constraints explicitly (in section 27.3). In section 27.4, we introduce two fundamental convex feasibility problems that can be used to determine if the model and data constraints are compatible and if the measurement data is consistent with the physics of the problem. This is used in section 27.5 to present different estimation problems that can be solved using LPs obtained by direct instantiation of the convex problems derived earlier. We show in particular that some nonconvex estimation problems such as the travel time estimation problem can still be decomposed as a series of LPs and thus are computationally tractable. In section 27.6, we define two important inverse modeling problems for situations in which the data and model constraints are incompatible. These problems are respectively known as *data assimilation* and *data reconciliation*, and are obtained by relaxing model and data constraints respectively. We then proceed to solve different problems of interest for transportation engineering. The examples presented in this chapter involve experimental highway traffic data sets, obtained from the *Performance Measurement System* (PeMS) and the *Mobile Century* experiment in California. Some of the resulting algorithms have been implemented in the *Mobile Millennium* system [14], in particular a sensor fault detection algorithm detailed in section 21.7 that runs in real time, every $30s$ for all highways in northern California.

27.1. Traffic flow measurement data and value conditions

In the context of traffic flow monitoring, measurement data traditionally originates from *Eulerian* (*i.e.* fixed) sensors. This is in contrast with new *Lagrangian* (*i.e.* mobile) sensors, which sense traffic conditions while moving alongside it.

27.1.1. Fixed detector data

Fixed sensors are currently the backbone of traffic monitoring. They measure various quantities related to traffic flow at a fixed location and for all times. Current fixed-sensor technology includes:

- Inductive loop detectors, such as the *Performance Measurement System* PeMS [22] in California and magnetometers [95] are based on measurements of the inductance of an electromagnetic loop. They can measure two quantities: the flow of vehicles above the sensor and the occupancy which can be related to the density of vehicles above the sensor. Dual loop detector arrangements [22], as well as some classes of magnetometers [95] can also directly measure traffic speed.
- Speed radars, which are measuring the doppler shift as well as the time of flight of electromagnetic waves. They can measure three quantities: the flow and density of vehicles around the sensor, as well as the average traffic speed.
- Speed cameras, which are paired with image recognition systems. They usually measure flow, density and average traffic speed.

One of the biggest problems associated with the fixed sensing infrastructure is their deployment and maintenance costs. Since this sensing infrastructure is dedicated (*i.e.* the infrastructure cannot be used to sense other physical phenomena besides traffic flow), these high costs are limiting the deployment of new sensors. In practice, the Departments of Transportation (DOTs) are operating these sensors, and usually have to spend their funding on more urgent issues.

27.1.2. Mobile sensor data

The emergence of new portable computational platforms with communication and sensing capabilities provides the engineering community with unprecedented opportunities for sensing. In the context of traffic flow, cellular phones [339,181] located onboard vehicles and equipped with positioning systems such as the GPS can act as traffic sensors. Other possible mobile sensing systems exist, including *toll tag readers* such as the FasTrak system in California.

All mobile sensing systems pose an additional mathematical and computational challenge with respect to fixed sensors. Since most of the current traffic sensing systems are fixed, most of the estimation techniques for traffic-flow engineering are not specifically designed to incorporate mobile measurements.

In order to estimate the state of traffic based on Eulerian and Lagrangian sensor measurements, we first need to establish the relation between the measurement data and the value conditions that incorporate the measurement constraints into the HJ PDE (25.5).

27.1.3. Experimental setup

In the following sections, we pose different problems arising in transportation engineering as *Linear Programs* (LPs) or series of LPs and test their performance using experimental data from

the Mobile Century [181] experiment.

In all numerical applications, we consider a $3.858\ km$ long spatial domain, located between the PeMS [22] stations 400536 and 400284 on Highway $I-880\ N$ in Hayward, California. The measurement data comes from two sources. The flow data $q_{\text{in}}^{\text{meas}}(\cdot)$ and $q_{\text{out}}^{\text{meas}}(\cdot)$ is generated by the PeMS stations 400536 and 400284 respectively. The probe location and timing data comes from GPS measurements generated by Nokia $N95$ cellphones located onboard probe vehicles. The layout is illustrated in Figure 1.1.

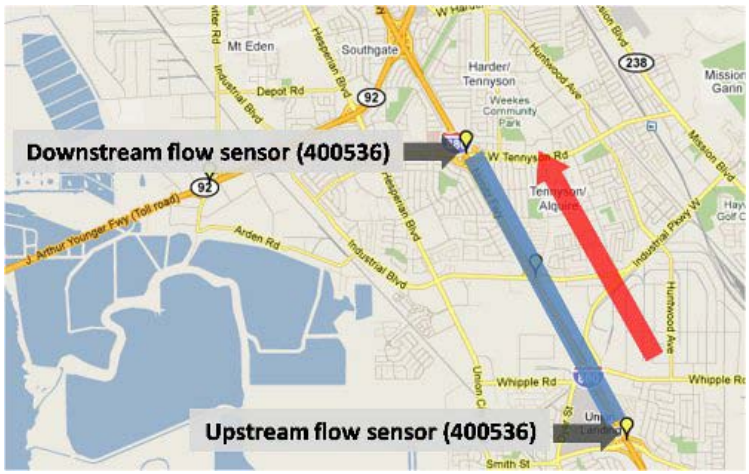


Figure 1.1. Experiment site layout.

The upstream and downstream PeMS stations are delimiting a $3.858\ km$ spatial domain, outlined by a solid line. The direction of traffic flow is represented by an arrow.

The complete experimental setting is described in [181]. The data set used in all numerical applications of this chapter can be freely downloaded from [14].

All LPs have been implemented in `Matlab`, using the package `CVX` [170]. The problems solved in this chapter are relatively tractable: they typically involve thousands of variables and constraints and can be solved numerically in a few seconds on a typical laptop computer.

27.1.4. Link between measurement data and value conditions

In our specific application, the sensor data does not provide the initial condition of the problem, since this would require us instrumenting the entire spatial domain. Fixed traffic sensors traditionally measure the inflow and outflow of vehicles on the spatial domain, which are related to the upstream and downstream boundary conditions. In addition to fixed sensors, mobile sensors onboard vehicles track the vehicle trajectory and thus generate *internal conditions* [104]. The formal link between traffic measurement data and value condition (boundary and internal conditions) blocks is shown in the following definition.

Definition 1. [Affine upstream, downstream and internal conditions] Let us define $\mathbb{N} = \{0, \dots, n_{max}\}$ and $\mathbb{M} = \{0, \dots, m_{max}\}$. For all $n \in \mathbb{N}$ and $m \in \mathbb{M}$, we define the following upstream, downstream and internal conditions:

$$\gamma_n(t,x) = \begin{cases} \sum_{i=0}^{n-1} q_{\text{in}}(i)T + q_{\text{in}}(n)(t - nT) & \text{if } x = \xi \\ & \text{and } t \in [nT, (n+1)T] \\ +\infty & \text{otherwise} \end{cases} \tag{321}$$

$$\beta_n(t,x) = \begin{cases} \sum_{i=0}^{n-1} q_{\text{out}}(i)T + q_{\text{out}}(n)(t - nT) - \Delta & \text{if } x = \chi \text{ and } t \in [nT, (n+1)T] \\ +\infty & \text{otherwise} \end{cases} \tag{322}$$

$$\mu_m(t,x) = \begin{cases} L_m + r_m(t - t_{\min}(m)) & \text{if } x = x_{min}(m) + v^{\text{meas}}(m)(t - t_{\min}(m)) \\ & \text{and } t \in [t_{\min}(m), t_{\max}(m)] \\ +\infty & \text{otherwise} \end{cases} \tag{323}$$

where $v^{\text{meas}}(m) = \frac{x_{max}(m) - x_{min}(m)}{t_{max}(m) - t_{min}(m)}$.

The domains of definitions of these functions are illustrated in Figure 1.2.

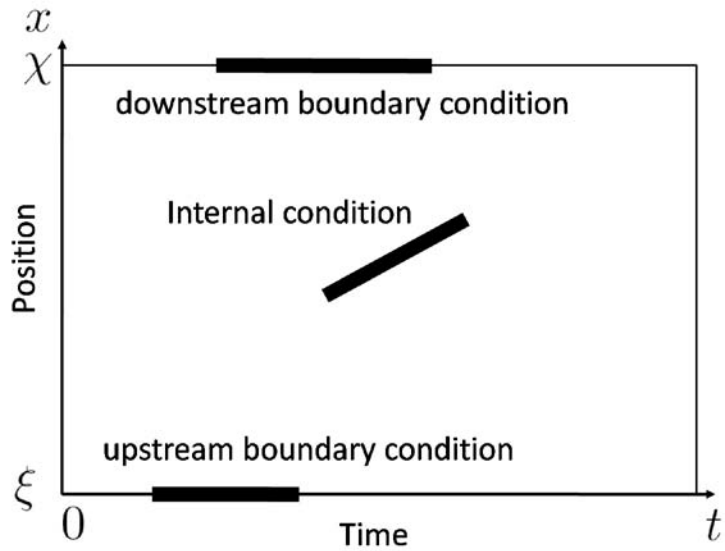


Figure 1.2. Illustration of the domains of the possible value conditions used to construct the solution of the Moskowitz HJ PDE.

The time is represented by the horizontal axis, while the location is represented by the vertical axis. The coefficients ξ and χ represent respectively the upstream and downstream boundaries of the highway segment of interest.

The coefficients in equations (321), (322) and (323) can be physically interpreted as follows:

$$\begin{cases} q_{\text{in}}(n) & \text{average inflow between times nT and (n + 1)T} \\ q_{\text{out}}(n) & \text{average outflow between times nT and (n + 1)T} \\ \Delta & \text{initial number of vehicles on the highway section} \end{cases}$$

$$\begin{cases} t_{\min}(m) & \text{initial time at which the internal condition } m \text{ applies} \\ t_{\max}(m) & \text{final time at which the internal condition } m \text{ applies} \\ x_{\min}(m) & \text{initial location at which the internal condition } m \text{ applies} \\ x_{\max}(m) & \text{final location at which the internal condition } m \text{ applies} \\ v^{\text{meas}}(m) & \text{speed of the internal condition } m \\ L_m & \text{label of the vehicle } m \text{ at time } t_{\min}(m) \\ r_m & \text{rate of change of the label of vehicle } m \end{cases} \tag{324}$$

nT(n+1)T (324)Since the Moskowitz function is increasing in time and decreasing in space, the coefficients of (321), (322) and (323) satisfy the following conditions:

$$\begin{cases} \Delta \geq 0 \quad \text{positivity of the initial number of vehicles} \\ \forall n \in \mathbb{N}, \quad q_{\text{in}}(n) \geq 0 \quad \text{positivity of the inflow} \\ \forall n \in \mathbb{N}, \quad q_{\text{out}}(n) \geq 0 \quad \text{positivity of the outflow} \\ \forall m \in \mathbb{M}, \quad r_m \geq 0 \quad \text{positivity of the passing rate} \end{cases} \tag{325}$$

The monotonicity properties of the Moskowitz function with respect to its variables follow directly from the positivity of flow and density functions (194) and are derived in [256].

Some of the above coefficients can be obtained (with some error) through traffic measurement data. Inductive loop detectors [22] and speed radars located in ξ and χ can measure the inflow $q_{\text{in}}(n)$ and outflow $q_{\text{out}}(n)$ for all time intervals $[nT, (n+1)T]$. The coefficients $t_{\min}(m)$, $t_{\max}(m)$, $x_{\min}(m)$ and $x_{\max}(m)$ can be obtained using vehicle positioning systems, such as GPS-enabled cellphones onboard vehicles [14]. In contrast, the coefficients L_m and r_m cannot be measured using conventional traffic sensors. Similarly, the initial number of vehicles Δ cannot be measured using conventional sensors. Hence, the available measurements do not enable us to define the upstream (321), downstream (322) and internal conditions (323) univocally. In order to estimate the state $\boldsymbol{M}(\cdot,\cdot)$ of the system, one has to estimate the coefficients (324), which are constrained both by the model and the measurement data.

In the applications of this chapter, the coefficients $x_{\min}(m)$, $x_{\max}(m)$, $t_{\min}(m)$ and $t_{\max}(m)$ are measured by are measured by GPS systems, which are very accurate. Hence, we assume that these coefficients are fixed. Giventhis assumption, we define the decision variable of our estimation problems as follows.

Definition 2. The coefficients of the upstream, downstream and internal conditions to be estimated are defined by the following decision variable:

$$y := \left(q_{\text{in}}(1), \dots, q_{\text{in}}(n_{max}), q_{\text{out}}(1), \dots, q_{\text{out}}(n_{max}), L_1, \dots, L_{m_{\max}}, r_1, \dots, r_{m_{max}}\right) \tag{326}$$

27.2. Explicit instantiation of the model compatibility conditions for triangular Hamiltonians

We now instantiate (318) explicitly so it can be applied to traffic flow engineering problems. Following common assumptions in transportation engineering [128,129], we assume that the Hamiltonian $\psi(\cdot)$is a continuous triangular function defined by:

$$\psi(\rho) = \begin{cases} v\rho & \text{if } \rho \le k_c \\ w(\rho - k_m) & \text{otherwise} \end{cases} \tag{327}$$

where v, w, k_c and k_m are model parameters satisfying $vk_c = w(k_c - k_m)$ and representing the *free flow speed* (v), the *critical density* (k_c), the *congestion speed* (w) and the *maximal density* (k_m).

Explicit expression of the solutions to the affine value conditions

In this section, we compute the solutions associated with the value conditions (321), (322) and (323) explicitly using the specific Hamiltonian (327). The results below are the instantiation of equations (249), (261), (273) and (274) for (327).

$$\boldsymbol{M}_{\gamma_n}(t,x) = \begin{cases} +\infty & \text{if } \; t \leq nT + \dfrac{x-\xi}{v} \\ \displaystyle\sum_{i=0}^{n-1} q_{\text{in}}(i)T + q_{\text{in}}(n)(t - \frac{x-\xi}{v} - nT) & \text{if } \; nT + \dfrac{x-\xi}{v} \leq t \\ & \text{and } \; t \leq (n+1)T + \dfrac{x-\xi}{v} \\ \displaystyle\sum_{i=0}^{n} q_{\text{in}}(i)T + k_c v(t - (n+1)T - \frac{x-\xi}{v}) & \text{otherwise} \end{cases}$$

$$\boldsymbol{M}_{\beta_n}(t,x) = \begin{cases} +\infty & \text{if } \; t \leq nT + \dfrac{x-\chi}{w} \\ -\Delta + \displaystyle\sum_{i=0}^{n-1} q_{\text{out}}(i)T + q_{\text{out}}(n)(t - \frac{x-\chi}{w} - nT) & \text{if } \; nT + \dfrac{x-\chi}{w} \leq t \\ & \textit{and } t \leq (n+1)T + \dfrac{x-\chi}{w} \\ -\Delta + \displaystyle\sum_{i=0}^{n} q_{\text{out}}(i)T + k_c v(t - (n+1)T - \frac{x-\chi}{w}) & \text{otherwise} \end{cases}$$

(328)

$$\boldsymbol{M}_{\mu_m}(t,x) = \begin{cases} L_m + r_m\left(t - \dfrac{x - x_{min}(m) - v^{meas}(m)(t - t_{min}(m))}{v - v^{meas}(m)} - t_{\min}(m)\right) \\ \text{if } x \geq x_{\min}(m) + v^{\text{meas}}(m)(t - t_{\min}(m)) \\ \text{and } x \geq x_{\max}(m) + v(t - t_{\max}(m)) \\ \text{and } x \leq x_{\min}(m) + v(t - t_{\min}(m)) \\ L_m + r_m\left(t - \dfrac{x - x_{min}(m) - v^{meas}(m)(t - t_{min}(m))}{w - v^{meas}(m)} - t_{\min}(m)\right) \\ +k_c(v - w)\dfrac{x - x_{min}(m) - v^{meas}(m)(t - t_{min}(m))}{w - v^{meas}(m)} \\ \text{if } x \leq x_{\min}(m) + v^{\text{meas}}(m)(t - t_{\min}(m)) \\ \text{and } x \leq x_{\max}(m) + w(t - t_{\max}(m)) \\ \text{and } x \geq x_{\min}(m) + w(t - t_{\min}(m)) \\ L_m + r_m(t_{\max}(m) - t_{\min}(m)) + (t - t_{\max}(m))k_c\left(v - \dfrac{x - x_{max}(m)}{t - t_{max}(m)}\right) \\ \text{if } x \leq x_{\max}(m) + v(t - t_{\max}(m)) \\ \text{and } x \geq x_{\max}(m) + w(t - t_{\max}(m)) \\ +\infty \text{ otherwise} \end{cases} \tag{329}$$

Explicit instantiation of the model constraints

For the specific boundary and internal conditions (321), (322) and (323), the model compatibility constraints (318) are:

$$\begin{cases} \boldsymbol{M}_{\gamma_n}(t,\xi) \geq \gamma_p(t,\xi) & \forall t \in [pT,(p+1)T], \forall (n,p) \in \mathbb{N}^2 \quad (i) \\ \boldsymbol{M}_{\gamma_n}(t,\chi) \geq \beta_p(t,\chi) & \forall t \in [pT,(p+1)T], \forall (n,p) \in \mathbb{N}^2 \quad (ii) \\ \boldsymbol{M}_{\gamma_n}(t,x) \geq \mu_m(t,x) & \forall (t,x) \in \mathrm{Dom}(\mu_m), \forall n \in \mathbb{N}, \forall m \in \mathbb{M} \quad (iii) \\ \boldsymbol{M}_{\beta_n}(t,\xi) \geq \gamma_p(t,\xi) & \forall t \in [pT,(p+1)T], \forall (n,p) \in \mathbb{N}^2 \quad (iv) \\ \boldsymbol{M}_{\beta_n}(t,\chi) \geq \beta_p(t,\chi) & \forall t \in [pT,(p+1)T], \forall (n,p) \in \mathbb{N}^2 \quad (v) \\ \boldsymbol{M}_{\beta_n}(t,x) \geq \mu_m(t,x) & \forall (t,x) \in \mathrm{Dom}(\mu_m), \forall n \in \mathbb{N}, \forall m \in \mathbb{M} \quad (vi) \\ \boldsymbol{M}_{\mu_m}(t,\xi) \geq \gamma_p(t,\xi) & \forall t \in [pT,(p+1)T], \forall (m,p) \in \mathbb{M} \times \mathbb{N} \quad (vii) \\ \boldsymbol{M}_{\mu_m}(t,\chi) \geq \beta_p(t,\chi) & \forall t \in [pT,(p+1)T], \forall (m,p) \in \mathbb{M} \times \mathbb{N} \quad (viii) \\ \boldsymbol{M}_{\mu_m}(t,x) \geq \mu_p(t,x) & \forall (t,x) \in \mathrm{Dom}(\mu_p), \forall (m,p) \in \mathbb{M}^2 \quad (ix) \end{cases} \tag{330}$$

Although inequalities (330) are a function of the decision variable (326), they cannot necessarily be expressed as linear inequalities (in terms of the decision variable) in general. However, because of the specific structure of the solutions (328) for triangular Hamiltonians, the

inequalities (330) can be rewritten as a finite number of linear inequality constraints, as shown in the following proposition.

Proposition 1. [Model constraints for triangular Hamiltonians] For triangular Hamiltonians defined by (327), the inequality constraints (330) can be expressed as a finite number of inequality constraints:

$$\begin{cases} \boldsymbol{M}_{\gamma_n}(pT,\xi) \geq \gamma_p(pT,\xi) & \forall (n,p) \in \mathbb{N}^2 \quad (i) \\ \boldsymbol{M}_{\gamma_n}(pT,\chi) \geq \beta_p(pT,\chi) & \forall (n,p) \in \mathbb{N}^2 \quad (ii)(a) \\ \boldsymbol{M}_{\gamma_n}(nT + \frac{\chi-\xi}{v},\chi) \geq \beta_p(nT + \frac{\chi-\xi}{v},\chi) & \forall (n,p) \in \mathbb{N}^2 \text{ such that} \\ & nT + \frac{\chi-\xi}{v} \in [pT,(p+1)T] \end{cases} \tag{331}$$

$$\begin{cases} \boldsymbol{M}_{\gamma_n}\big(t_{min}(m),x_{min}(m)\big) \geq \mu_m\big(t_{min}(m),x_{min}(m)\big) & \forall n \in \mathbb{N}, \forall m \in \mathbb{M} \quad (\\ \boldsymbol{M}_{\gamma_n}\big(t_{max}(m),x_{max}(m)\big) \geq \mu_m\big(t_{max}(m),x_{max}(m)\big) & \forall n \in \mathbb{N}, \forall m \in \mathbb{M} \quad (\\ \boldsymbol{M}_{\gamma_n}\big(t_1(m,n),x_1(m,n)\big) \geq \mu_m\big(t_1(m,n),x_1(m,n)\big) & \forall n \in \mathbb{N}, \forall m \in \mathbb{M} \quad \text{su} \\ t_1(m,n) \in [t_{min}(m);t_{max}(m)] & (\end{cases} \tag{332}$$

$$\begin{cases} \boldsymbol{M}_{\beta_n}(pT,\xi) \geq \gamma_p(pT,\xi) & \forall (n,p) \in \mathbb{N}^2 \quad (iv)(a) \\ \boldsymbol{M}_{\beta_n}(nT + \frac{\xi-\chi}{w},\xi) \geq \gamma_p(nT + \frac{\xi-\chi}{w},\xi) & \forall (n,p) \in \mathbb{N}^2 \text{ such that} \\ & nT + \frac{\xi-\chi}{w} \in [pT,(p+1)T] \\ \boldsymbol{M}_{\beta_n}(pT,\chi) \geq \beta_p(pT,\chi) & \forall (n,p) \in \mathbb{N}^2 \quad (v) \end{cases} \tag{333}$$

$$\begin{cases} \boldsymbol{M}_{\beta_n}(t_{min}(m),x_{min}(m)) \geq \mu_m(t_{min}(m),x_{min}(m)) & \forall n \in \mathbb{N}, \forall m \in \mathbb{M} \quad (\\ \boldsymbol{M}_{\beta_n}(t_{max}(m),x_{max}(m)) \geq \mu_m(t_{max}(m),x_{max}(m)) & \forall n \in \mathbb{N}, \forall m \in \mathbb{M} \quad (\\ \boldsymbol{M}_{\beta_n}(t_2(m,n),x_2(m,n)) \geq \mu_m(t_2(m,n),x_2(m,n)) & \forall n \in \mathbb{N}, \forall m \in \mathbb{M} \quad \text{su} \\ t_2(m,n) \in [t_{min}(m);t_{max}(m)] & (vi)(c) \end{cases} \tag{334}$$

$$\begin{cases} \boldsymbol{M}_{\mu_m}(pT,\xi) \geq \gamma_p(pT,\xi) & \forall (m,p) \in \mathbb{M} \times \mathbb{N} \quad (vii)(a) \\ \boldsymbol{M}_{\mu_m}(t_3(m),\xi) \geq \gamma_p(t_3(m),\xi) & \forall (m,p) \in \mathbb{M} \times \mathbb{N} \\ & \text{such that } t_3(m) \in [pT,(p+1)T] \quad (vii) \\ \boldsymbol{M}_{\mu_m}(t_4(m),\xi) \geq \gamma_p(t_4(m),\xi) & \forall (m,p) \in \mathbb{M} \times \mathbb{N} \\ & \text{such that } t_4(m) \in [pT,(p+1)T] \quad (vii) \end{cases} \tag{335}$$

$$\begin{cases} \boldsymbol{M}_{\mu_m}(pT,\chi) \geq \beta_p(pT,\chi) & \forall(m,p) \in \mathbb{M} \times \mathbb{N} \quad (viii)(a) \\ \boldsymbol{M}_{\mu_m}(t_5(m),\chi) \geq \beta_p(t_5(m),\chi) & \forall(m,p) \in \mathbb{M} \times \mathbb{N} \\ & \text{such that } t_5(m) \in [pT,(p+1)T] \quad (viii \\ \boldsymbol{M}_{\mu_m}(t_6(m),\chi) \geq \beta_p(t_6(m),\chi) & \forall(m,p) \in \mathbb{M} \times \mathbb{N} \\ & \text{such that } t_6(m) \in [pT,(p+1)T] \quad (viii \end{cases} \tag{336}$$

$$\begin{cases} \boldsymbol{M}_{\mu_m}(t_{min}(p),x_{min}(p)) \geq \mu_p(t_{min}(p),x_{min}(p)) & \forall(m,p) \in \mathbb{M}^2 \quad (ix)(a) \\ \boldsymbol{M}_{\mu_m}(t_{min}(p),x_{max}(p)) \geq \mu_p(t_{max}(p),x_{\max}(p)) & \forall(m,p) \in \mathbb{M}^2 \quad (ix)(b) \\ \boldsymbol{M}_{\mu_m}(t_7(m,p),x_7(m,p)) \geq \mu_p(t_7(m,p),x_7(m,p)) & \forall(m,p) \in \mathbb{M}^2 \text{ such that} \\ & t_7(m,p) \in [t_{min}(p),t_{max}(p)] \quad (ix)(c) \\ \boldsymbol{M}_{\mu_m}(t_8(m,p),x_8(m,p)) \geq \mu_p(t_8(m,p),x_8(m,p)) & \forall(m,p) \in \mathbb{M}^2 \text{ such that} \\ & t_8(m,p) \in [t_{min}(p),t_{max}(p)] \quad (ix)(d) \\ \boldsymbol{M}_{\mu_m}(t_9(m,p),x_9(m,p)) \geq \mu_p(t_9(m,p),x_9(m,p)) & \forall(m,p) \in \mathbb{M}^2 \text{ such that} \\ & t_9(m,p) \in [t_{min}(p),t_{max}(p)] \quad (ix)(e) \\ \boldsymbol{M}_{\mu_m}(t_{10}(m,p),x_{10}(m,p)) \geq \mu_p(t_{10}(m,p),x_{10}(m,p)) & \forall(m,p) \in \mathbb{M}^2 \text{ such that} \\ & t_{10}(m,p) \in [t_{min}(p),t_{max}(p)] \quad (ix)(f) \\ \boldsymbol{M}_{\mu_m}(t_{11}(m,p),x_{11}(m,p)) \geq \mu_p(t_{11}(m,p),x_{11}(m,p)) & \forall(m,p) \in \mathbb{M}^2 \text{ such that} \\ & t_{11}(m,p) \in [t_{min}(p),t_{max}(p)] \quad (ix)(g) \end{cases} \tag{337}$$

where

$$\begin{cases} t_1(m,n) = \dfrac{nTv - v^{meas}(m)t_{min}(m) + x_{min}(m) - \xi}{v - v^{meas}(m)} \\ x_1(m,n) = v^{\text{meas}}(m)\left(\dfrac{nTv - v^{meas}(m)t_{\min}(m) + x_{min}(m) - \xi}{v - v^{meas}(m)} - t_{min}(m)\right) + x_{min}(m) \\ t_2(m,n) = \dfrac{nTw - v^{meas}(m)t_{min}(m) + x_{\min}(m) - \chi}{w - v^{meas}(m)} \\ x_2(m,n) = v^{\text{meas}}(m)\left(\dfrac{nTw - v^{meas}(m)t_{min}(m) + x_{min}(m) - \chi}{w - v^{meas}(m)} - t_{min}(m)\right) + x_{min}(m) \\ t_3(m) = \dfrac{\xi - x_{min}(m) + wt_{min}(m)}{w} \\ t_4(m) = \dfrac{\xi - x_{min}(m) + wt_{max}(m)}{w} \\ t_5(m) = \dfrac{\chi - x_{min}(m) + vt_{min}(m)}{v} \\ t_6(m) = \dfrac{\chi - x_{max}(m) + vt_{max}(m)}{v} \end{cases} \tag{338}$$

and

$$\begin{cases} t_7(m,p) = \dfrac{x_{\min}(m) - x_{min}(p) + v^{meas}(p)t_{min}(p) - v^{meas}(m)t_{\min}(m)}{v^{meas}(p) - v^{meas}(m)} \\ x_7(m,p) = v^{\text{meas}}(p)\left(\dfrac{x_{\min}(m) - x_{\min}(p) + v^{meas}(p)t_{min}(p) - v^{meas}(m)t_{min}(m)}{v^{meas}(p) - v^{meas}(m)} - t_{\min}(p)\right) + x_{min}(p) \\ t_8(m,p) = \dfrac{x_{max}(m) - x_{min}(p) + v^{meas}(p)t_{min}(p) - vt_{max}(m)}{v^{meas}(p) - v} \\ x_8(m,p) = v^{\text{meas}}(p)\left(\dfrac{x_{max}(m) - x_{min}(p) + v^{meas}(p)t_{min}(p) - vt_{max}(m)}{v^{meas}(p) - v} - t_{\min}(p)\right) + x_{min}(p) \\ t_9(m,p) = \dfrac{x_{\min}(m) - x_{min}(p) + v^{meas}(p)t_{min}(p) - vt_{min}(m)}{v^{meas}(p) - v} \\ x_9(m,p) = v^{\text{meas}}(p)\left(\dfrac{x_{min}(m) - x_{min}(p) + v^{meas}(p)t_{min}(p) - vt_{min}(m)}{v^{meas}(p) - v} - t_{\min}(p)\right) + x_{min}(p) \\ t_{10}(m,p) = \dfrac{x_{max}(m) - x_{min}(p) + v^{meas}(p)t_{min}(p) - vt_{min}(m)}{v^{meas}(p) - w} \\ x_{10}(m,p) = v^{\text{meas}}(p)\left(\dfrac{x_{max}(m) - x_{min}(p) + v^{meas}(p)t_{min}(p) - vt_{max}(m)}{v^{meas}(p) - w} - t_{\min}(p)\right) + x_{min}(p) \\ t_{11}(m,p) = \dfrac{x_{min}(m) - x_{min}(p) + v^{meas}(p)t_{min}(p) - vt_{min}(m)}{v^{meas}(p) - w} \\ x_{11}(m,p) = v^{\text{meas}}(p)\left(\dfrac{x_{min}(m) - x_{min}(p) + v^{meas}(p)t_{min}(p) - vt_{min}(m)}{v^{meas}(p) - w} - t_{\min}(p)\right) + x_{min}(p) \end{cases} \tag{339}$$

Proof — The inequality constraints (330) are of the following form:

$$\boldsymbol{M}_{\boldsymbol{c}_j}(t,x) \geq \boldsymbol{c}_i(t,x), \quad \forall (t,x) \in \mathrm{Dom}(\boldsymbol{c}_i) \tag{340}$$

where $\mathrm{Dom}(\boldsymbol{c}_i)$ is a line segment of $\mathbb{R}^2$, $\boldsymbol{c}_i(\cdot,\cdot)$ is an affine function of the form (321), (322) or (323) and $\boldsymbol{M}_{\boldsymbol{c}_j}(\cdot,\cdot)$ is a piecewise affinefunction of the form (328). Hence,$\boldsymbol{M}_{\boldsymbol{c}_j}(\cdot,\cdot) - \boldsymbol{c}_i(\cdot,\cdot)$ is a piecewise affine function, defined on $\mathrm{Dom}(\boldsymbol{M}_{\boldsymbol{c}_j}) \cap \mathrm{Dom}(\boldsymbol{c}_i)$. Note that $\mathrm{Dom}(\boldsymbol{M}_{\boldsymbol{c}_j})$ is convex by proposition 5 and that $\mathrm{Dom}(\boldsymbol{c}_i)$ is a line segment of $\mathbb{R}^2$. Hence, $\mathrm{Dom}(\boldsymbol{M}_{\boldsymbol{c}_j}) \cap \mathrm{Dom}(\boldsymbol{c}_i)$ is also a line segment of $\mathbb{R}^2$, which can thus be written as $\mathrm{Dom}(\boldsymbol{M}_{\boldsymbol{c}_j}) \cap \mathrm{Dom}(\boldsymbol{c}_i) = \{u + \alpha v, \alpha \in [0,1]\}$ for some $(u, v) \in \mathbb{R}^4$.

Let us define $f(\cdot)$ on $[0,1]$ as $f: \alpha \to \boldsymbol{M}_{\boldsymbol{c}_j}(u + \alpha v)$. With this definition, inequality (340) can be written as:

$$f(\alpha) \geq 0, \quad \forall \alpha \in [0,1] \tag{341}$$

Since $\boldsymbol{M}_{\boldsymbol{c}_j}(\cdot,\cdot) - \boldsymbol{c}_i(\cdot,\cdot)$ is piecewise affine and continuous, so is $f(\cdot)$. Let us define the intervals in which $f(\cdot)$ is affine by $[0, \alpha_1], \dots, [\alpha_p, 1]$. Since $f(\cdot)$ is monotonic on the intervals $[0, \alpha_1], \dots, [\alpha_p, 1]$, inequality (341) is satisfied if and only if $f(0) \geq 0$, $f(\alpha_1) \geq 0$,…,$f(\alpha_p) \geq 0$ and $f(1) \geq 0$, which yields the finite number of inequalities (331), (332), (333), (334), (335), (336) and (337). ■

Since the model inequality constraints (325), (331), (332), (333), (334), (335), (336) and (337) are all linear inequalities in the decision variable y defined by (326), we can write them in a compact form as follows:

$$A_{\mathrm{model}}(\psi) y \leq b_{\mathrm{model}}(\psi) \tag{342}$$

Note that the model constraints are a function of the parameters of the Hamiltonian.

27.3. Data constraints

Similarly to the model constraints shown above, measurement data also restricts the possible values that the coefficients (324) can take. The values of $t_{\min}(\cdot)$, $t_{\max}(\cdot)$, $x_{\min}(\cdot)$, $x_{\max}(\cdot)$, $q_{\mathrm{in}}(\cdot)$ and $q_{\mathrm{out}}(\cdot)$ can be directly measured, though we assume that $t_{\min}(\cdot)$, $t_{\max}(\cdot)$, $x_{\min}(\cdot)$ and $x_{\max}(\cdot)$ are perfectly known. The measured values of $q_{\mathrm{in}}(\cdot)$ and $q_{\mathrm{out}}(\cdot)$ are denoted by $q_{\mathrm{in}}^{\mathrm{meas}}(\cdot)$ and $q_{\mathrm{out}}^{\mathrm{meas}}(\cdot)$ respectively. In the remainder of this chapter, we choose the following error model for $q_{\mathrm{in}}(\cdot)$ and $q_{\mathrm{out}}(\cdot)$:

$$||\frac{q_{in}(\cdot) - q_{in}^{meas}(\cdot)}{q_{in}^{meas}(\cdot)}||_p \leq e_{\max}$$
$$||\frac{q_{out}(\cdot) - q_{out}^{meas}(\cdot)}{q_{out}^{meas}(\cdot)}||_p \leq e_{\max} \tag{343}$$

where $||\cdot||_p$ is the standard L_p norm:

$$||f(\cdot)||_p = (\sum_{n=1}^{n_{max}} |f(n)|^p)^{\frac{1}{p}} \tag{344}$$

Different choices of norm are possible, but all choices of $p \geq 1$ yield convex constraints by convexity of the norm. In particular, the choices $p = 1$ and $p = +\infty$ yield linear constraints, which can be written as:

$$A_{\text{data}} y \leq b_{\text{data}} \quad \text{for } p = 1 \text{ or } p = +\infty \tag{345}$$

The choice $q = 2$ yields quadratic convex constraints, which can be written as:

$$y^T Q(i) y \leq b_{\text{data}}(i), \quad (Q(i) \geq 0), \quad \forall i \in [1, i_{max}] \quad \text{for } p = 2 \tag{346}$$

Note that the error model (343), for $p = +\infty$ is commonly used in practice. It corresponds to a situation in which we assume that the relative error on each measurement of the sensor is bounded by a constant value. In the remainder of this chapter, we assume that the error model yields linear inequalities of the form (345) for simplicity. Note that the results presented below could be trivially extended for quadratic constraints (346), yielding quadratically constrained convex programs.

27.4. Compatibility and consistency problems

We now define two fundamental convex feasibility problems which will play an important role in the subsequent sections.

27.4.1. Data and model compatibility problem

Let y denote the decision variable (326), and let the model and data constraints be defined as in (342) and (345) respectively. The data and model compatibility constraints can be satisfied at the same time if and only if the following problem is feasible:

$$\begin{aligned} &\boldsymbol{Find} \quad y \\ &\boldsymbol{such\ that} \quad \begin{cases} A_{\text{model}}(\psi)y \leq b_{\text{model}}(\psi) \\ A_{\text{data}}y \leq b_{\text{data}} \end{cases} \end{aligned} \tag{347}$$

When the above problem is feasible, one can estimate the minimum (respectively maximum) of a piecewise affine convex (respectively concave) function of the decision variable using a LP. One can thus estimate lower and upper bounds on linear functions of the decision variable using LPs. We apply this property in section 27.5 to estimate upper and lower bounds on linear functions of the decision variable.

In contrast, when (347) is infeasible, no set of value conditions satisfying both the model and data constraints can exist. However, by relaxing alternatively the model or data constraints, one can define [108] two problems of interest, which are the subject of section 27.6. The data reconciliation problem consists in finding the set of value conditions satisfying the model constraints, that is as close as possible (in some norm sense) to satisfy the data constraints. In contrast, the data assimilation problem consists in finding the set of value conditions satisfying the data constraints, that is as close as possible to satisfy the model constraints.

27.4.2. Data consistency problem

The problem (347) must be feasible when the following conditions are satisfied:

1- The Hamiltonian $\psi(\cdot)$ satisfies (298) (that is, the hypograph of the Hamiltonian contains all experimental flow-density values, as in Figure 1.2 of chapter 6.
2- The coefficients $\overline{y}$ associated with the actual value condition satisfy the data constraints (345)

(348)

Indeed, if the conditions (348) are met, the coefficients $\overline{y}$ associated with the true value condition will satisfy (345) and (342) by proposition 9. The problem of checking the feasibility of (347) under the constraints (348) is referred to as *consistency check*. This problem is used in section 7 to detect cyberattacks and in section 27.5 to give guaranteed bounds on some traffic-related quantities.

27.5. Estimation problems

27.5.1. Definition for general functions of traffic-related coefficients

A number of traffic-flow related quantities can be written as linear functions of the decision variable (349) and can be estimated using Linear Programming, as shown in the following proposition.

$$y := \left(q_{\text{in}}(1), \dots, q_{\text{in}}(n_{max}), q_{\text{out}}(1), \dots, q_{\text{out}}(n_{max}), L_1, \dots, L_{m_{max}}, r_1, \dots, r_{m_{max}}\right) \tag{349}$$

Proposition 1. [Estimation of linear functions] Let $f(\cdot)$ be a linear function of (326), defined as $f(y) = c^T y$. The possible values that $f(\cdot)$ can take under the linear model (342) and data (343) constraints (for the L_1 or L_∞ norms) is the interval $[f_{min}, f_{max}]$, where f_{min} and f_{max} are solutions to the following LPs:

$$\begin{aligned} &\boldsymbol{Minimize}(\text{respectively } \boldsymbol{Maximize}) \quad c^T y \\ &\boldsymbol{such\ that} \quad \begin{cases} A_{\text{model}}(\psi) y \le b_{\text{model}}(\psi) \\ A_{\text{data}} y \le b_{\text{data}} \end{cases} \end{aligned} \tag{350}$$

Note that the above estimation problem has a sense only if the compatibility problem (347) is feasible. Not also that when the conditions (348) are satisfied, (347) is feasible and the true value condition $\overline{y}$ is an element of the feasible set. Hence the actual value $f(\overline{y})$ of $f(\cdot)$ is guaranteed to be in the interval $[f_{min}, f_{max}]$.

27.5.2. Lower and upper bounds on traffic coefficients

Estimation of the initial number of vehicles using linear programming

The initial number of vehicles Δ on the highway section can be estimated through Linear Programming. Indeed, Δ appears linearly in the decision variable (326), while the model (342) and data constraints (345) are linear inequalities in (326). Since the feasible set is convex by the constraints of (350), the possible values of Δ such that the model and data constraints are satisfied are $\Delta_{min} \le \Delta \le \Delta_{max}$, where Δ_{min} and Δ_{max} are solutions to the following optimization programs:

$$\begin{aligned} &\boldsymbol{minimize}(\text{respectively } \boldsymbol{maximize}) \quad \Delta \\ &\boldsymbol{suchthat} \quad \begin{cases} A_{\text{model}}(\psi) y \le b_{\text{model}}(\psi) \\ A_{\text{data}} y \le b_{\text{data}} \end{cases} \end{aligned} \tag{351}$$

We illustrate the estimation process in Figure 5.1, in which we show the evolution of the interval $[\Delta_{min}, \Delta_{max}]$ as we increase the quantity of measurement data. In this problem, we consider the spatial domain defined in section 3, between the times 11:40 AM and 12:10 PM. We solve (360) using 60 blocks of upstream boundary conditions (321) and downstream boundary conditions (322), and a variable number of internal conditions (323).

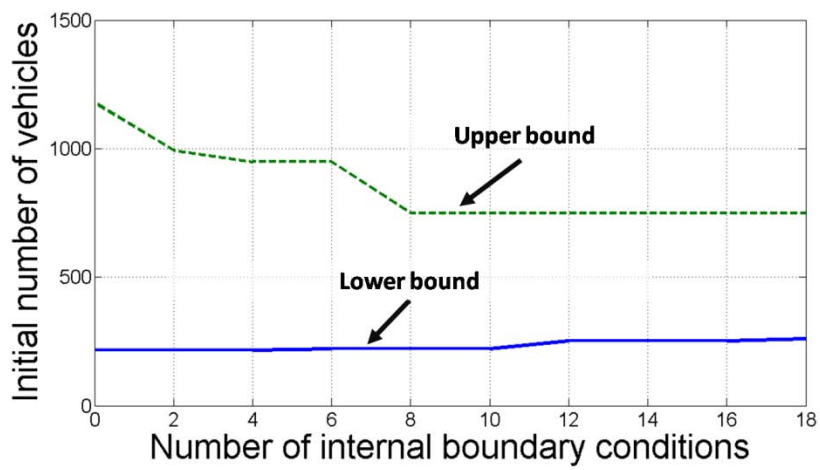


Figure 5.1. Initial number of vehicles estimation using linear programming.

This figure represents the evolution of the upper and lower bounds on the initial number of vehicles Δ as new internal condition data is added into the estimation problem. The horizontal axis represents the number of probe measurement data blocks $\mu_m(\cdot,\cdot)$ as defined in (323). As predicted by proposition 6, the upper bound (dashed) on Δ decreases and the lower bound (solid line) on Δ increases when additional data is added into the estimation problem.

The same framework can also be applied for estimating other functions of the decision variable (326), such as the travel time across the highway section. Unlike the initial number of vehicles, the travel time is a nonlinear and nonconvex function of the decision variable (326), which makes the estimation problem more challenging.

Travel time estimation using convex programming

In order to properly define a travel time function, we first need to assume [256] that no vehicles can pass each other, which implies in particular $r_m = 0$ for all $m \in \mathbb{M}$. In this situation, known in the transportation engineering as *First In First Out* (FIFO), the vehicle trajectories are the isolines of the state function. In order to properly define the travel time function, we also have to assume that the function $\beta(\cdot,\cdot) = \min_{n\in\mathbb{N}} \beta_n(\cdot,\cdot)$ is strictly increasing. Note that by (322), imposing this last condition amounts to impose $q_{\text{out}}(\cdot) > 0$ instead of the inequality $q_{\text{out}}(\cdot) \geq 0$ in (325). With these two assumptions, the travel time can be defined as follows. Let t be

given, and $i = \lfloor \frac{t}{T} \rfloor$. The travel time $\sigma(t)$ is defined as $\tau - t$, where $\gamma_i(t, \xi) = \beta(\tau, \chi)$. Since $\beta(\cdot, \chi)$ is strictly increasing, we can also define the travel time as:

$$\sigma(y, t) = \min_{s \in \mathbb{R}_+ \text{ s. t. } \beta(s,\chi) \geq \gamma_i(t,\xi)} (s - t) \tag{352}$$

or alternatively:

$$\sigma(y, t) = \max_{s \in \mathbb{R}_+ \text{ s. t. } \beta(s,\chi) \leq \gamma_i(t,\xi)} (s - t) \tag{353}$$

Since $\beta_j(s, \chi)$ and $\gamma_i(t, \chi)$ are functions of the decision variable (326), the travel time function $\sigma(\cdot,\cdot)$ hereby defined is a function of the decision variable (326), though not linear. While we cannot estimate the travel time using a LP of the form (350), we can still obtain valuable information on upper and lower bounds of the travel time function using LPs, as outlined in the following proposition.

Proposition 2. [Upper and lower bounds on travel time function] Let us assume that (347) is feasible, that is, the model and data constraints are compatible. Let two times t and τ be given, and let $i = \lfloor \frac{t}{T} \rfloor$ and $j = \lfloor \frac{\tau}{T} \rfloor$. We have that $\tau - t$ is a lower bound on the travel time $\sigma(y, t)$ (under the model and data constraints) if and only if the following problem is infeasible:

$$\begin{aligned} &\boldsymbol{find} \quad y \\ &\boldsymbol{such\ that} \quad \begin{cases} A_{\text{model}}(\psi) y \leq b_{\text{model}}(\psi) \\ A_{\text{data}} y \leq b_{\text{data}} \\ \beta_j(\tau, \chi) - \gamma_i(t, \xi) \geq 0 \end{cases} \end{aligned} \tag{354}$$

Similarly, $\tau - t$ is an upper bound on the travel time $\sigma(y, t)$ under the model and data constraints if and only if the following problem is infeasible:

$$\begin{aligned} &\boldsymbol{find} \quad y \\ &\boldsymbol{such\ that} \quad \begin{cases} A_{\text{model}}(\psi) y \leq b_{\text{model}}(\psi) \\ A_{\text{data}} y \leq b_{\text{data}} \\ \beta_j(\tau, \chi) - \gamma_i(t, \xi) \leq 0 \end{cases} \end{aligned} \tag{355}$$

Proof — We prove that $\tau - t$ is a lower bound on the travel time function if and only if (354) is infeasible. Let us assume that (354) is infeasible. This amounts to saying that $\beta_j(\tau, \chi) <$

$\gamma_i(t,\xi)$ whenever the model and data constraints $A_{\text{model}}(\psi)y \leq b_{\text{model}}(\psi)$ and $A_{\text{data}}y \leq b_{\text{data}}$ are both satisfied. Hence, since $\beta(\tau,\chi) = \beta_j(\tau,\chi)$ by construction, this is equivalent to saying that $\beta(\tau,\chi) < \gamma_i(t,\xi)$ whenever the model and data constraints are both satisfied. By the definition (353) of $\sigma(y,t)$, this is equivalent to $\sigma(y,t) > \tau - t$, whenever y satisfies the model and data constraints, which completes the proof. The proof relative to the upper bound is similar, and involves the definition (352) of $\sigma(y,t)$. ■

Note that the feasibility programs (354) and (355) enable us to compute the largest lower bound $\sigma_{\text{d}}(t)$ and the smallest upper bound $\sigma_{\text{u}}(t)$ on the travel time by trial and error. We illustrate the above results by computing the upper and lower bounds on the travel time function, using the experimental setup of section 27.1.3, between times 11:40 AM and 12:10 PM. For this, we check the feasibility of problems (354) and (355) for $\tau = jT$, and plot in Figure 5.2 respectively the lowest and highest value of jT such that (355) and (354) are respectively infeasible. The lowest value $j_{max}T$ for which (355) is infeasible implies that $\sigma_{\text{u}}(t)$ is in the interval $[(j_{max}-1)T - t, j_{max}T - t]$. Similarly, the highest value $j_{min}T$ for which (354) is infeasible implies that $\sigma_{\text{d}}(t)$ is in the interval $[j_{min}T - t, (j_{min}+1)T - t]$. As stated in proposition 6, the distance between the upper and lower bounds decreases as more data is added into the estimation problem.

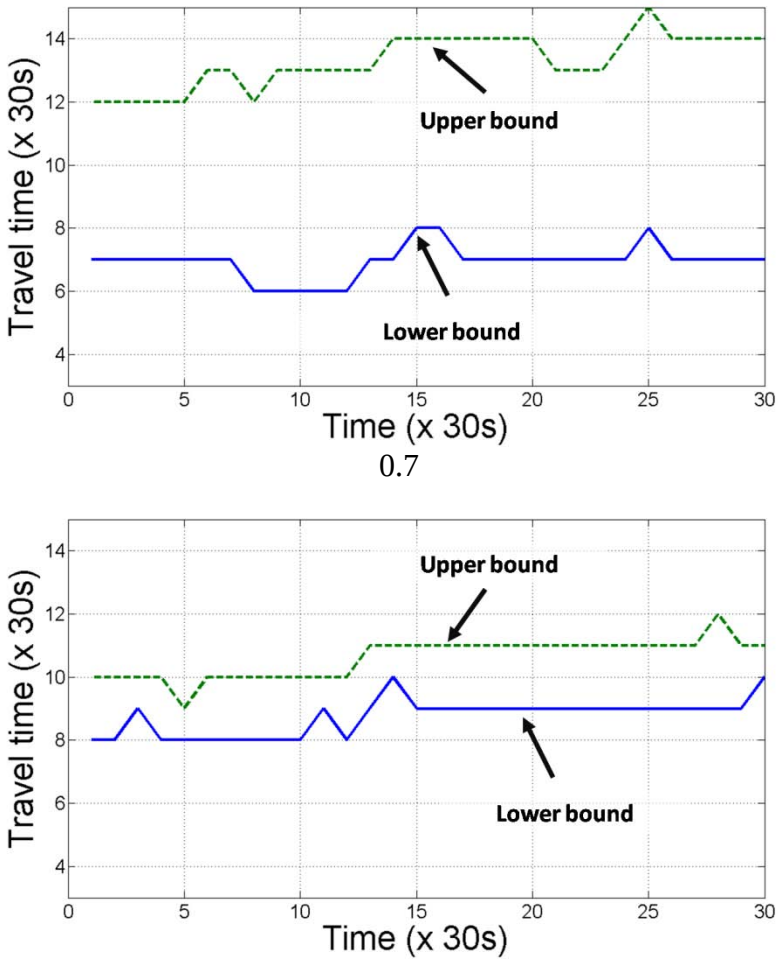
Remark — The largest lower bound (or smallest upper bound) on travel time cannot be directly estimated using convex programming. Indeed, by checking the feasibility of (354) for increasing values of $\tau = nT$, we can find the integer j such that $\sigma_{\text{d}}(t) \in [jT - t, (j+1)T - t]$ (in this situation, (354) is infeasible for $\tau = jT$, and becomes feasible for $\tau = jT + 1$). When such a j is identified, $\sigma_{\text{d}}(t)$ is the solution to the following optimization program:

$$\begin{aligned} &\boldsymbol{minimize} \frac{z}{q_{out}(j)} \\ &\boldsymbol{such\ that} \begin{cases} A_{\text{model}}(\psi)y \leq b_{\text{model}}(\psi) \\ A_{\text{data}}y \leq b_{\text{data}} \\ \beta_j(\frac{z}{q_{out}(j)},\chi) - \gamma_i(t,\xi) \leq 0 \end{cases} \end{aligned} \tag{356}$$

The decision variable of (356) can be written as (y,z), where y is the decision variable defined by (326). The constraints $A_{\text{model}}(\psi)y \leq b_{\text{model}}(\psi)$ and $A_{\text{data}}y \leq b_{\text{data}}$ are both linear in the new decision variable (they indeed depend only upon y). The constraint $\beta_j(\frac{z}{q_{out}(j)},\chi) - \gamma_i(t,\xi) \leq 0$ is also linear, since it can be written as:

$$\begin{aligned} &\sum_{k=0}^{j-1} q_{\text{out}}(k)T + q_{\text{out}}(j)(\frac{z}{q_{out}(j)} - jT) \\ &-\Delta - \sum_{k=0}^{i-1} q_{\text{in}}(k)T - q_{\text{in}}(i)(t - iT) \leq 0 \end{aligned} \tag{357}$$

The choice of $\frac{z}{q_{out}(j)}$ in the objective function is made to enforce the linearity of the constraints (and it plays the role of τ in the previous equations). The objective is however nonconvex, since $(z, q) \to \frac{z}{q}$ is not convex. Problem (356) thus cannot be solved using convex programming, but may still be solved numerically using other optimization methods. ■



0.7

Figure 5.2. Travel time estimation using linear programming.

In this figure, the horizontal axis represents the time, while the vertical axis represents the travel time. The upper and lower bounds on the travel time function are represented by a dashed and solid line respectively. **Top:** In this figure, we consider 60 upstream and downstream boundary conditions blocks and 20 internal condition blocks. **Bottom:** In this figure, we increase the number of internal condition blocks to 45. As can be seen, the corresponding bounds on the travel time function are improved since more data is added into the estimation problem, following proposition 6.

27.5.3. Guaranteed ranges for traffic coefficients estimation

The upper and lower bounds on functions of the decision variable investigated above do not necessarily hold in practice, since the true value $\overline{y}$ of the decision variable (326) may not satisfy the model and data constraints. However, when the conditions (348) hold, the values of the upper and lower bounds are guaranteed. Indeed, when (348) is satisfied, $\overline{y}$ belongs to the set $\{y|A_{\text{model}}(\psi)y \leq b_{\text{model}}(\psi)\} \cap \{A_{\text{data}}y \leq b_{\text{data}}\}$, which implies

$$f(\overline{y}) \geq \begin{array}{ll} \boldsymbol{minimize} & f(y) \\ \boldsymbol{such\ that} & \begin{cases} A_{\text{model}}(\psi)y \leq b_{\text{model}}(\psi) \\ A_{\text{data}}y \leq b_{\text{data}} \end{cases} \end{array} \tag{358}$$

and

$$f(\overline{y}) \leq \begin{array}{ll} \boldsymbol{maximize} & f(y) \\ \boldsymbol{such\ that} & \begin{cases} A_{\text{model}}(\psi)y \leq b_{\text{model}}(\psi) \\ A_{\text{data}}y \leq b_{\text{data}} \end{cases} \end{array} \tag{359}$$

In order to obtain guaranteed bounds in practice, one has to choose the model parameter such that the condition (348) holds. In the context of traffic flow, the typical values of the model parameter are accurately known, and do not vary significantly between experimental sites. They are available from [236] for instance. In order to impose (348) on all practical traffic scenarios, one simply has to overapproximate these values to define the model parameters.

27.6. Data assimilation and data reconciliation problems

27.6.1. Problem definition

In the field of distributed parameters system estimation, the problems of *data assimilation* [143] and *data reconciliation* [120] are closely linked. The data assimilation process consists in finding the value of the state of the system that satisfies the observations, and that is the closest to being a solution to the evolution model. In contrast, the data reconciliation process consists in finding a solution to the evolution model that is the closest to the observations. Given the framework detailed above, the data assimilation and reconciliation problems are related to the solutions of the following convex optimization program.

$$\begin{array}{ll} \boldsymbol{minimize} & ||y_1 - y_2||_q \\ \boldsymbol{suchthat} & \begin{cases} A_{\text{model}}(\psi)y_1 \leq b_{\text{model}}(\psi) \\ A_{\text{data}}y_2 \leq b_{\text{data}} \end{cases} \end{array} \tag{360}$$

In the above optimization program, we have to choose $q = 1$ or $q = +\infty$ to obtain a linear objective. Two situations can arise:

- If the optimal value of (360) is 0, the model and data constraints can be satisfied at the same time. In this situation, the data assimilation and data reconciliation problems coincide in a setting in which data and model are compatible. The solution is not necessarily unique.
- If the optimal value of (360) is nonzero, the optimal solutions y_1^{optimal} and y_2^{optimal} enable us to compute the upstream, downstream and internal conditions respectively associated with the data reconciliation and data assimilation problems. Note that these solutions may not be unique. The value conditions associated with y_1^{optimal} satisfy the model constraints by construction, *i.e.* all upstream boundary, downstream boundary and internal conditions blocks apply in the strong sense [59,69]. They however do not satisfy the data constraints, but are as close as possible in the $||\cdot||_q$ sense to satisfy them. In contrast, the value conditions associated with y_2^{optimal} satisfy the data constraints by construction, but do not satisfy the model constraints (they are as close as possible to satisfy them in the $||\cdot||_q$ sense).

27.6.2. Numerical example

In this application, we consider the spatial domain defined in section 27.1.3, between the times 11:40 AM and 12:05 PM for data collected on February 8th, 2008. We use the following Hamiltonian parameters: $k_c = 0.048\ m^{-1}$, $v = 24.6\ m/s$, $w = -4.5\ m/s$, and a maximal relative error level of $e_{max} = 0.01$. We solve (360) for $q = 1$, using 604 variables and 17415 linear constraints. For this specific application, the optimal value of (360) is $+8.58$, which ensures that the data assimilation and data reconciliation problems are well defined. As mentioned above, y_1^{optimal} and y_2^{optimal} enable us to compute the value conditions associated with the data assimilation and data reconciliation problems. We compute the solutions to (195) associated with these value conditions, and display them in Figure 6.1. The solution to the *data reconciliation problem* at the top of Figure 6.1 satisfies all the boundary and internal conditions that are prescribed on it. The model applies in the strong sense, however the decision variable violates the data constraints (345). In contrast, the upstream and downstream boundary conditions do not apply everywhere in the solution to the *data assimilation problem* (Figure 6.1, center). In the illustrated data assimilation example, the data constraints some internal conditions to be set in a way that is incompatible with the upstream and downstream boundary conditions. This can be seen for instance around time $t = 1100s$: a back propagating wave hits the upstream boundary condition at $x = 11000m$, which prevents it from applying between times $t = 1100s$ and $t = 1400s$.

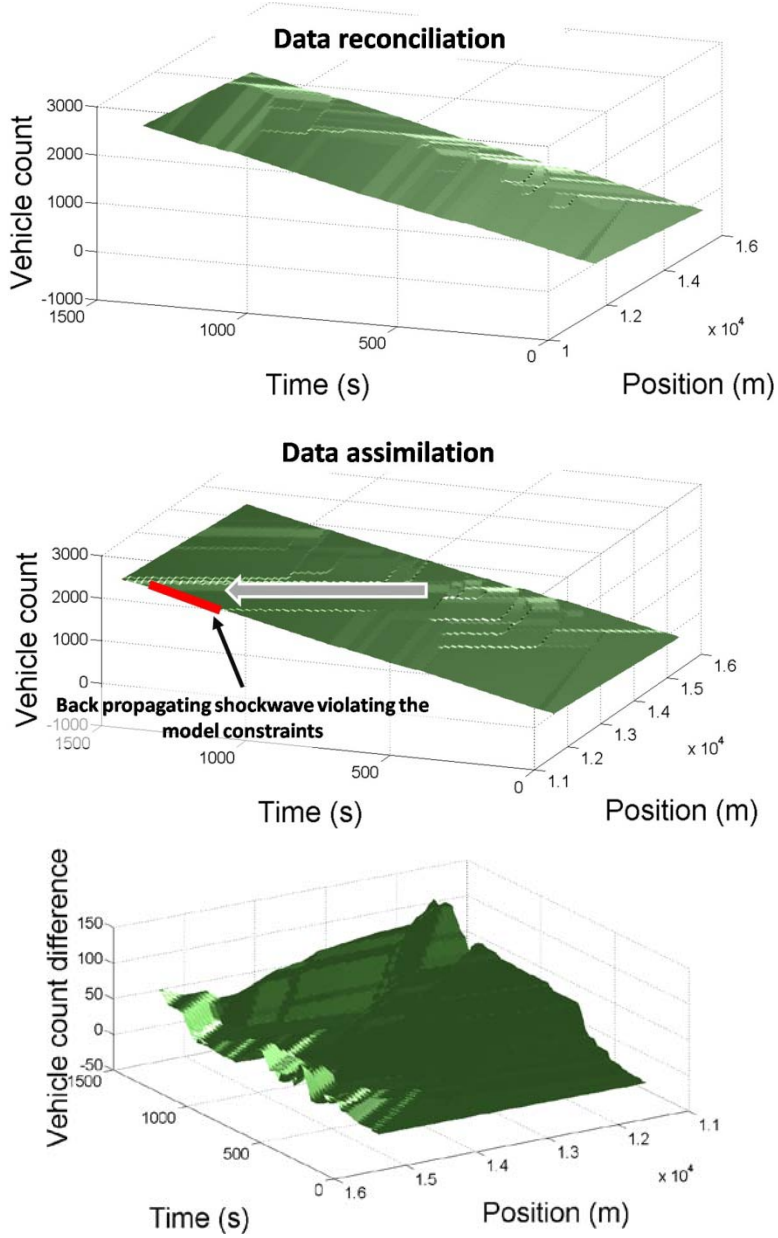


Figure 6.1. Solutions to data assimilation and data reconciliation problems.

Top: Solution to the data reconciliation problem, in which the model constraints are satisfied, but the data constraints are not. **Center:** Solution to the data assimilation problem, in which the data constraints are satisfied, but the model constraints are not. Both problems are solved simultaneously by (360). **Bottom:** Difference (in number of vehicles) between the solution to the data reconciliation problem and the solution to the data assimilation problem.

27.7. Cybersecurity, sensor fault detection and privacy analysis problems

27.7.1. Consistency problems applied to sensor failure detection

The framework developed in section 27.4 can also be applied to detect failures in sensor networks. In this section, we are interested in checking the consistency of the data generated by sensors of the PeMS system [22], which is a network of loop detectors measuring traffic on California highways. The PeMS system is one of the data feeds currently integrated in the *Mobile Millennium* traffic monitoring system [340,14], operated jointly by Nokia and UC Berkeley. One of the main challenges arising when using data from the PeMS system is the automated identification of the mislocated or faulty sensors. Previous approaches such as [216] have successfully implemented sensor fault detection algorithms based on statistical correlation with adjacent sensors. Our approach is different though, since it can guarantee using the PDE model that at least one of the sensors in an array of sensors is failing.

We solve the fault detection problem either by checking the feasibility of the consistency problem (347) using a Hamiltonian satisfying (348) on all pairs of consecutive sensors present on the highway network.

We assume that the maximal allowable error of a PeMS sensor is $e_{max} = 0.3$ in (343). There are multiple sources of uncertainty arising when dealing with loop detectors, such as pavement depth, loop layout, which typically creates maximal errors of this magnitude. We also choose an Hamiltonian $\psi(\cdot)$ satisfying (348) by overapproximating the tabulated values of [236]: $k_c = 0.05\ m^{-1},\ v = 30\ m/s,\ w = -7\ m/s.$

As an application, we consider five consecutive PeMS sensors, labeled 401339, 401714, 401376, 400609 and 400835 respectively, as illustrated in Figure 7.1. For each one of the four adjacent pairs of sensors, we compute the minimal value of the error e_{max} such that the consistency problem (347) is feasible during a one month period at the frequency of one day. The distribution of these results is shown in Figure 7.1. Note that since e_{max} appears linearly in the data constraints, the minimal value of the error e_{max} such that (347) is feasible is also a LP.

Figure 7.1 shows that there is no indication of malfunction for the first and the last pairs of sensors. Note that the success to the minimal error test does not guarantee that a pair of sensors is working, since the actual error of the pair of sensors can be above the maximal allowable error.

The second and third pairs exhibit errors that are higher than 0.3, which indicates a malfunction of the corresponding pairs. Further analysis has shown that the pair $401714 - 400609$ is passing the minimal error test and thus that sensor 401376 is likely incorrectly mapped.

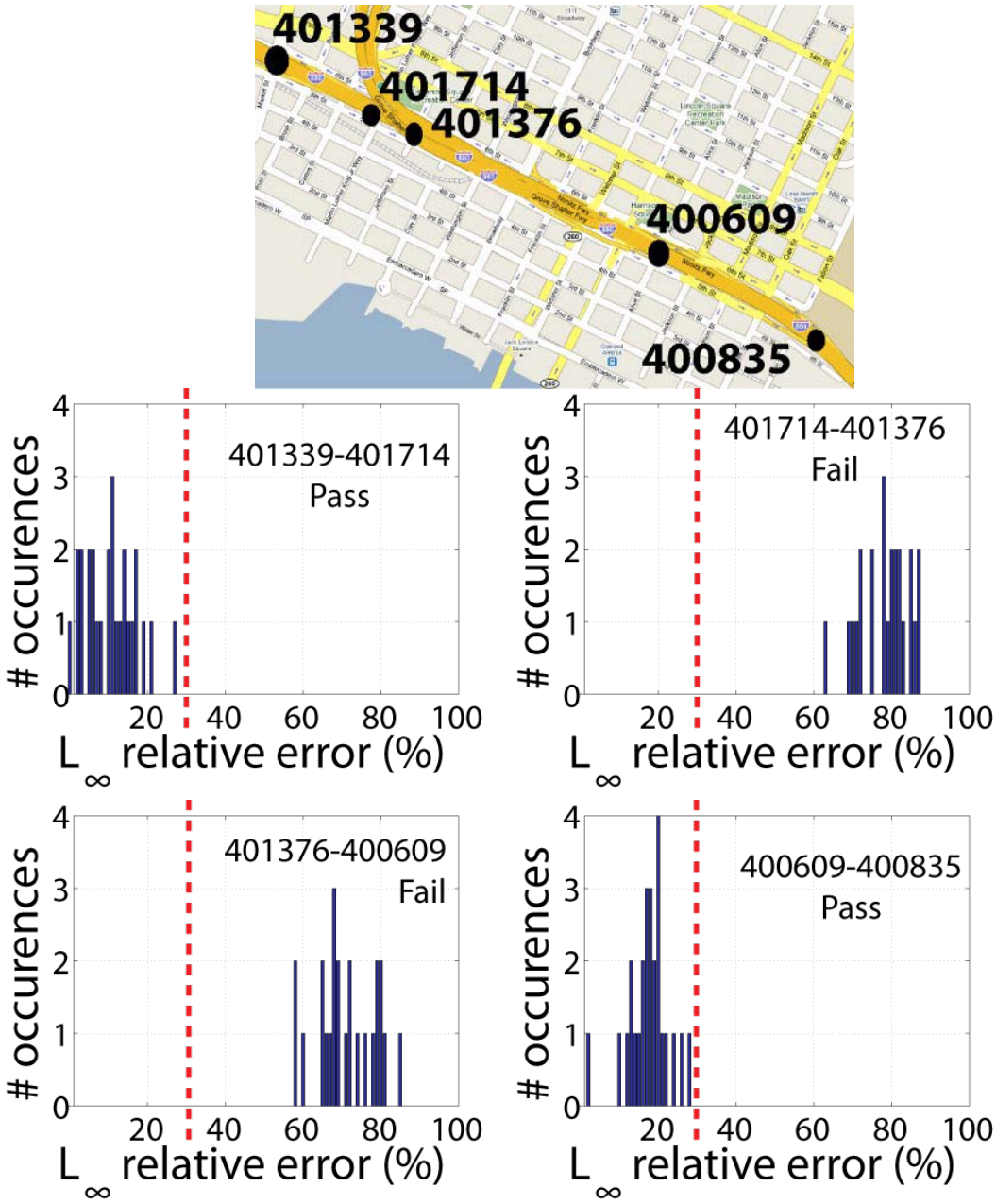


Figure 7.1. Faulty sensor detection.

We consider here the traffic flow on highway I880-S near Oakland, CA. The minimal error estimation problem (a LP) is run each day on a one-month period. The sensors of interests are highlighted in the top figure and their corresponding minimal error distribution over the one-month period is represented in the four bottom figures. **Bottom:** The top left and bottom right subfigures represents the minimal errors of the pair $401339 - 401714$ and $40609 - 400835$. These minimal errors fall in the allowable range. In contrast, the minimal errors of the pair $401714 - 401376$ and $401376 - 400609$ are above the allowable range. This means that there must exist a fault in one of the sensors 401714, 401376, or 400609.

27.7.2. Consistency problems applied to cybersecurity

One type of cyberattack [50] consists in faking sensor data and sending it to the monitoring system as if it was originating from valid sensors. Detecting this form of cyberattack is a complex problem in general. Detecting fake data that follows some pattern (for instance if the faked data is periodic) or that falls out of physically reasonable bounds is easy. However, detecting fake data that is both random and consistent with the expected value of sensor measurements is difficult.

A possible approach for solving this problem is to check if (347) is feasible, under the assumption (348). If (347) is feasible, this implies that the data is consistent with our model and data assumptions. Note that this does not guarantee that no cyberattack occurs. Indeed, an attacker can send fake data in a way that is consistent with the model and error levels assumptions. However, if (347) is infeasible, at least one of the assumptions (348) is false. Either the Hamiltonian does not satisfy (348), or the error model (348) is wrong. Any of these two situations can denote a cyber-attack if the conditions (348) are known to hold. However, note that other phenomena such as sensor failures can also cause (347) to be infeasible. The same framework can be applied to sensor fault detection.

We illustrate the cyberattack detection method (347) by simulating an attacker sending fake random values of $x_{min}(\cdot)$, $x_{max}(\cdot)$, $t_{min}(\cdot)$ and $t_{max}(\cdot)$, leading to the construction of new fake internal conditions using (323). The values of $x_{min}(\cdot)$, $x_{max}(\cdot)$, $t_{min}(\cdot)$ and $t_{max}(\cdot)$ are chosen randomly as follows. The speed $\frac{x_{max}(\cdot)-x_{min}(\cdot)}{t_{max}(\cdot)-t_{min}(\cdot)}$ associated with the internal condition is chosen uniformly in an interval $[v_{min}, v_{max}]$. The coefficients satisfy $x_{\min}(\cdot) \geq \xi$, $x_{max}(\cdot) \leq \chi$, $t_{min}(\cdot) \geq 0$ and $t_{max}(\cdot) \leq n_{max}T$. In the numerical applications, we consider the experimental setup described in section 27.1.3, between times 11:40 AM and 12:00 PM. We use 30 experimental internal conditions (323), 40 experimental upstream boundary conditions (321) and 40 experimental downstream boundary conditions (322). We progressively add fake internal conditions (323) and solve problem (360) for $q = 1$, and for a Hamiltonian satisfying (348). Note that (347) is feasible if and only if the solution to (360) is zero. Thus, the solution to (360) is a measure of the "distance" or incompatibility between data and model. In order to facilitate comparisons and reproduce the results, each result in Figure 7.2 top and bottom was averaged over 10 different choices of fake internal conditions.

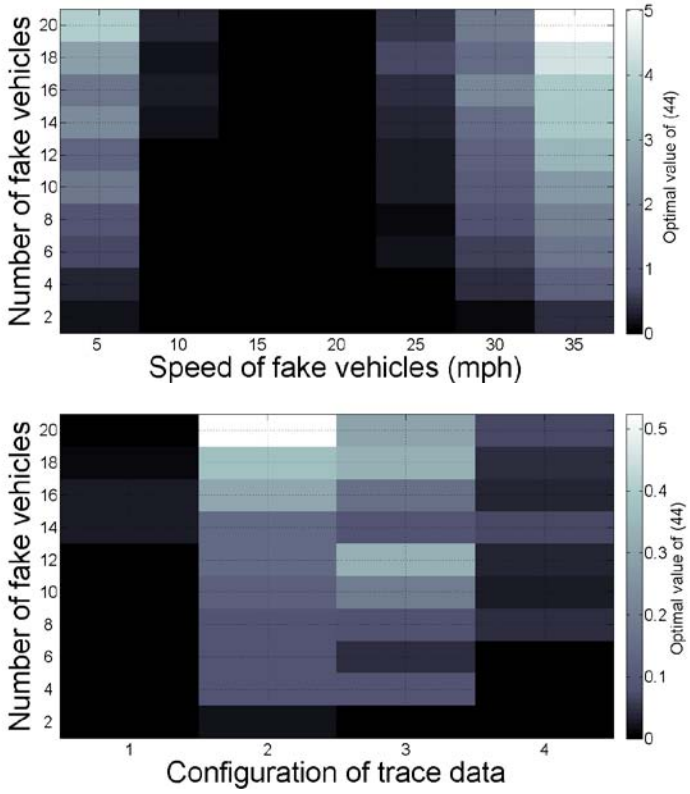


Figure 7.2. Cyberattack detection using linear programming.

In these figures, we represent the solution to (360) as a color map. Low values are represented as dark areas, and correspond to situations in which the "compatibility" between model and data is "good", *i.e.* lower values of $||y_2 - y_i||$ in (347). High values are represented as light-colored areas, and denote a higher degree of incompatibility between the model and data constraints. **Top:** The horizontal axis in this figure represents the lower bound k of the interval [k mph, $k + 5$ mph] in which the fake speed data is drawn. The vertical axis represents the number of fake internal conditions added. For instance, the cell $(15,12)$ corresponds to 12 fake internal conditions for which the speed is in the interval [15 mph, 20 mph]. As can be seen from this figure, the distance increases when more fake measurements are added into the estimation problem, and when they correspond to a speed that is far away from the true average speed (around 20 mph in this application). **Bottom:** This figure illustrates the high sensitivity of the solution to (360) with respect to the available measurement data. In this figure, we consider four different sets of 30 actual internal boundary conditions each. The procedure used for choosing a random subset of the available measurement data is illustrated in Figure 7.3. For each of these subsets (configurations), we add an increasing quantity of fake internal conditions, associated with random speeds ranging in [30 mph, 35 mph]. As can be seen, the results vary dramatically depending on which subset of the available data was chosen, even if the number of fake internal conditions is identical.

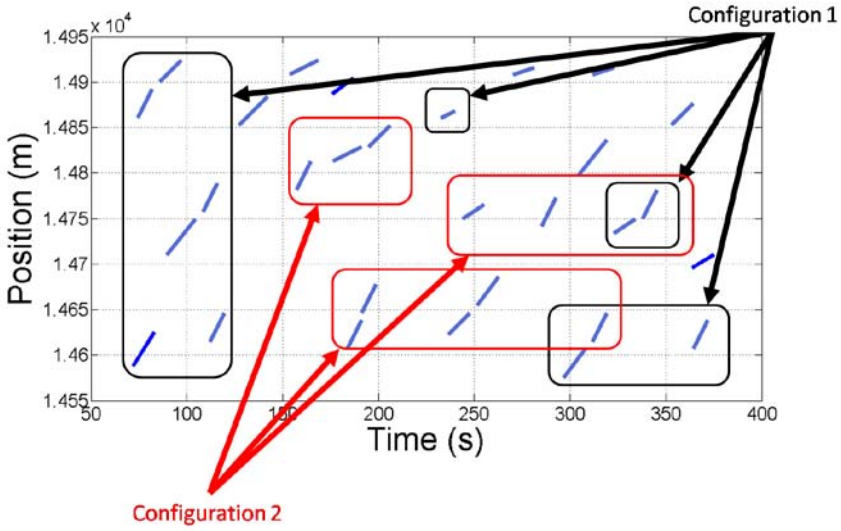


Figure 7.3. Illustration of the choice of a subset of measurement data.

In this figure, each segment represents the domain of an internal condition (323), obtained using experimental data. We illustrate the choice of two subsets of 12 internal conditions among 28 speed measurements, which we call "configuration". The same process applies for Figure 7.2, bottom, with four different configurations involving 30 internal conditions each among 94 available speed measurements.

As illustrated in Figure 7.2 top, adding fake speed measurements increases the incompatibility between data and model. The incompatibility between data and model is 0 when no fake measurements are added, which is consistent with the factthat (348) holds. Note that adding fake speeds does not have a significant impact on the level of incompatibility between data and model when the fake speeds are close to the average speed on the highway section (20 mph in this experiment) which is also consistent with the physics of the problem.

Figure 7.2 bottom shows that the configuration of the measurement data plays a critical role. In this figure, we study the influence of the measurement data on the detection of cyberattacks. For this, we consider four different subsets of 30 internal conditions each, extracted from our measurement data. An example of subset of 12 internal conditions among 28 available measurements is illustrated in Figure 7.3. We fix the fake speeds range to [30 mph, 35 mph], andshow the solution to (347) for these four configurations, represented on the horizontal axis. As can be seen from this figure, depending on the configuration of our measurement data, we can have very different ranges of level of incompatibility between data and model, for identical number of actual measurements, number of fake internal conditions and range of fake speeds. In the configurations $\#1$ and $\#4$, it is very difficult to detect that a spoofing attack occurs, since no change in the optimal value of (360). However, the spoofing attack is easily detected in the configuration $\#2$, even though all configurations contain the same amount of measurement data. These results thus show that it is almostimpossible to determine if detecting a cyberattack is easy based on the amount of measurement data alone.

27.7.3. Privacy analysis problems

Another possible application of the framework defined in section 27.5 is the analysis of user privacy using linear programming. The label of the vehicle represented by the internal condition $\mu_m(\cdot,\cdot)$ defined by (323) is L_m. In practical problems, the same vehicle sends different packets of information, representing different internal conditions (323). To what extent is it possible to "reidentify" one vehicle, *i.e.* to track it by identifying the pieces of data that came from the same vehicle?

Standard methods [190] do not take into account the model constraints: they usually try to reidentify vehicles under the assumption that vehicles maintain a relatively constant speed. While this is true for a large number of traffic scenarios, it does not take into account the underlying model, and can fail if the traffic speeds change significantly through the computational domain.

If we assume that vehicles do not pass each other (this implies $r_m = 0$ for all $m \in \mathbb{M}$), the minimal and maximal number of vehicles between two different internal condition blocks $\mu_i(\cdot,\cdot)$ and $\mu_j(\cdot,\cdot)$ is solution to the following LP:

$$\begin{aligned} &\boldsymbol{minimize}(\text{or } \boldsymbol{maximize}) \quad |L_i - L_j| \\ &\boldsymbol{suchthat} \quad \begin{cases} A_{\text{model}}(\psi)y \leq b_{\text{model}}(\psi) \\ A_{\text{data}}y \leq b_{\text{data}} \\ r_m = 0 \quad \forall m \in \mathbb{M} \end{cases} \end{aligned} \tag{361}$$

The above LP enables us to identify situations in which the privacy of users could be breached. Indeed, when the maximal number of vehicles between $\mu_i(\cdot,\cdot)$ and $\mu_j(\cdot,\cdot)$ is zero, these boundary conditions represent the same vehicle. In contrast, when the minimal number of vehicles between $\mu_i(\cdot,\cdot)$ and $\mu_j(\cdot,\cdot)$ is nonzero, these boundary conditions cannot originate from the same vehicle. We thus have three cases:

- If the optimal value $|L_i - L_j|^{max}$ of the maximization problem (361) is zero, then $\mu_i(\cdot,\cdot)$ and $\mu_j(\cdot,\cdot)$ have been generated by the same vehicle.
- If the optimal value $|L_i - L_j|^{min}$ of the minimization problem (361) is nonzero, then $\mu_i(\cdot,\cdot)$ and $\mu_j(\cdot,\cdot)$ cannot have been generated from the same vehicle.
- Other cases are inconclusive

We show an example of vehicle reidentification in Figure 7.4, using the experimental setup of section 27.1.3.

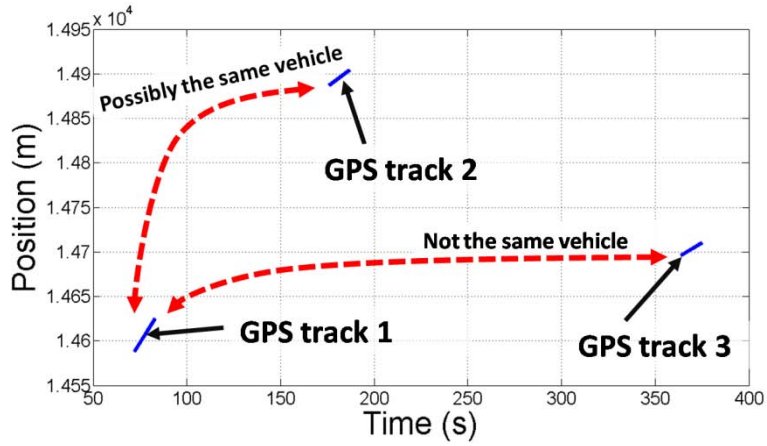


Figure 7.4. Vehicle reidentification using linear programming.

This figure represents the domains of definition of three internal conditions of the form (323). The horizontal axis represents the time, while the vertical axis represents the spatial domain. For this specific problem, the solutions to (361) are as follows: the minimal value of $|L_1 - L_2|$ is zero, the maximal value of $L_1 - L_2$ is 196, and the minimal value of $|L_1 - L_3|$ is 164. This thus guarantees that the block $\#3$ cannot originate from the same vehicle as the block $\#1$. The block $\#2$ can possibly come from the same vehicle as the block $\#1$ (and indeed is), but we have no guarantee of this since the maximal possible value of $L_1 - L_3$ is nonzero.

In practical computations, given an internal condition $\mu_i(\cdot,\cdot)$, there may exist multiple $j \in \mathbb{M}$ such that the solution to the minimization problem (361) is zero. If this happen, we lost track of vehicle i, which can be desirable in the context of traffic flow engineering (see [190] for an analysis of user privacy in mobile traffic sensing systems).

Part VII

Conclusion

28. Recommendations

28.1. Summary of the project

Advanced Traveler Information Systems are a key component of managing transportation growth in urban areas. Yet collecting and disseminating good-quality, real-time traffic information is a costly proposition. Therefore, government agencies at the federal, state and local levels have recognized the need to foster public-private partnerships in this area. This project focused on building on the successes of the recent Mobile Century field experiment and on carrying forward a partnership with cell phone manufacturer and application service provider, Nokia, and the leading map manufacturing company in the US, NAVTEQ, to collect traffic data from GPS-equipped mobile phones throughout the San Francisco Bay Area. The goal of this partnership was to deploy thousands of phones in a short timeframe as part of a Field Operational Test (FOT) nicknamed Mobile Millennium. The traffic data collected from those phones was to be processed into meaningful traveler information and fed back to a variety of channels, including the mobile handsets that generated them in the first place. Mobile Millennium hence served as an early instantiation of a participatory-sensing-based data collection and dissemination system, and was soon followed by numerous industry products of the same nature. One of the fundamental components of the research completed in this work included the protection of individuals participating in this pilot.

This novel approach to traffic data collection was enabled by recent advances in telecommunications and microelectronics technologies, and the growing ubiquity of wireless communications that results from it. Several years after the start of this project, the initial vision that led to the creation of the Mobile Millennium project revealed itself to be accurate: numerous industry based traffic information systems today rely on participatory sensing and GPS enabled smartphones.

Prior to this work, available methods to collect data from cell phones relied on approximate positioning provided by the cellular networks and have shown limited accuracy. Also, no system using this type of technology had managed to take off at a global scale. Given that in today's market, an increasing number of cell phones now ship with built-in GPS receivers, this prototype Location-Based Service is one of the early examples of such systems. Leveraging commercial

cellular networks could drastically cut the ongoing costs of traffic monitoring and expand coverage to thousands of miles of highways and urban arterials for which dedicated sensors are not even considered an option. This project provided a demonstration of the feasibility of this concept.

Under joint sponsorship from the US DOT, the California DOT, Nokia, NAVTEQ, and others who contributed to this effort, the Mobile Millennium project presented here enabled the development of algorithms and an information technology infrastructure to instruct the design of an industrial-grade data collection system. The principal objectives were for this system to feature: (1) online, real-time data processing; (2) privacy-preservation; and (3) data efficiency, i.e. not requiring excessive cellular network load. Mobile Millennium was also a demonstration of GPS mobile probes on an unprecedented scale. Finally, one of the goals of the proposed project was to serve in creating a paradigm shift in traffic data collection, which is already well underway, and will be explained later in this chapter. Indeed, mobile probes deployed as part of a self-sustaining business model will complement the existing fixed sensors that are funded by state and local governments, a paradigm that has now become part of the discussions at the level of the state and federal DOTs.

28.2. Relevance of the research approach

28.2.1. Research agenda in the context of 2008

The potential of cell phones to operate as traffic data collection devices had been considered by the Intelligent Transportation Systems community for several years. Prior to the start of Mobile Millennium, government agencies deployed networks of traffics ensors that were expensive to install and maintain. In the context of 2008, when the project was started, it appeared that leveraging commercial cellular networks could drastically cut the ongoing costs of traffic monitoring and expand coverage to thousands of miles of highways and urban arterials. By comparison, available methods to collect data from cell phones at the time relied on approximate positioning provided by the cellular networks and have shown limited accuracy to date. The option of using such methods was initially discussed with Nokia, and clearly judged not worthwhile, mainly because of the emergence of GPS enabled smartphones.

The use of GPS as a source of valid data for traffic was not initially an uncontroversial choice. In the context of 2008, the number of firm believers in the quality of approximate cellular network information was still high. Three years later, the fact is that dozens of companies have emerged, and based their traffic information systems on GPS based probes, while almost no new company has emerged to base its traffic information system on cellular data (even though this type of data has become available at a broader scale). From this, one may safely conclude that the technology chosen in this project was the correct one.

While the start of this project is officially in 2008, our team comprised of scientists at the University of California, Berkeley (UC Berkeley) and the Nokia Research Center (NRC) in Palo Alto, CA, had started investigating these questions in the Fall of 2006, with a seed grant from the Center for Information Technology Research in the Interest of Society (CITRIS) supporting the UC Berkeley effort. In 2007, the team's efforts were supplemented by the participation of UC

Berkeley's California Center for Innovative Transportation (CCIT), which focuses on the acceleration of research implementation and technology deployment by transportation practitioners. Under sponsorship from the California Department of Transportation (Caltrans) and the US DOT, the group embarked on an ambitious program to develop algorithms and an information technology infrastructure with a goal to demonstrate the capabilities of GPS-equipped cellular phones to operate as mobile traffic probes, and later to instruct the design of an industrial-grade data collection system, the Mobile Millennium project. In the summer of 2007, Nokia brought in a team from Rutgers University with a charter to investigate the privacy aspects of cell phone motion monitoring. This team, led by Professor Marco Gruteser had the mission to lead the agenda of the research to perform the design of the privacy mechanisms of Mobile Century (and later Mobile Millennium). Subsequently, the Berkeley-Nokia-Rutgers team developed, in just over four months, a proof-of-concept prototype that was demonstrated in a day-long field test involving 100 vehicles. The Mobile Century field experiment took place in the San Francisco Bay Area on February 8, 2008. Mobile Century was considered a resounding success by all of its witnesses, who included RITA's administrator Paul Brubaker and the Volpe Center's Gary Ritter. The event generated considerable media coverage in the local and national press, and on broadcast television, which is a tribute to the universal appeal of the concept, i.e. using familiar technology to better negotiate traffic. The Mobile Millennium was thus the logical next step to take in the context of 2008, and after the large success of Mobile Century.

28.2.2. Mobile Millennium, a step towards nextGen ATIS

At the start of the project, expanding the scope and coverage of roadway ATIS was a top-priority of Caltrans. Supporting statements for more and better traveler information across the state of California had come all the way from the Governor's office.

ATIS benefits the transportation system for at least two reasons.

- First, the availability of information enhances the service provided to travelers. Numerous studies reveal that commuters appreciate and value timely information, which reduces their uncertainty and their stress.
- Second, reliable information can arguably enable travelers to make educated choices about their itinerary, departure time, or even transportation mode, with the result of bringing about system self-management. It remains to be established that system self-management can take place on a large scale and can significantly impact network-level operations. However, at a more anecdotal level, information about an accident ahead, or a scheduled ramp closure, certainly influences driver decisions. An additional side benefit of ATIS is that it builds the awareness of the traveling public toward Intelligent Transportation Systems (ITS). Such awareness can translate into political support for ITS projects and enable more improvements in the long term.

One of the main pieces of ATIS content is undoubtedly travel time estimations, which was one of the main focuses of Mobile Millennium: at the end of Mobile Millennium, one of the most valuable contributions (presented earlier in this report) is its traffic estimation engine, capable of providing real-time speed (and thus travel time) estimates. Travel times on selected itineraries represent information that is easy for the traveling public to understand and process. Travel times

can be posted on freeway or arterial CMS and reach a very large audience, as is currently done at dozens of locations in the San Francisco Bay Area and in Southern California. Estimating travel times, either at the present time or into the future, requires large amounts of good quality traffic data, which this project focused on collecting.

Appraising various methods of collecting traffic data and specifically travel times was one of the goals which this project tackled, in agreement with US DOT's and Caltrans' operational objectives. Besides providing the bulk of the content required for ATIS, travel times also represent precious data to Caltrans (and US DOT) as a network operator. While travel times alone may not cover the full extent of the department's traffic data needs, accurate and reliable travel times can be used for both planning and operations purposes.

In the years preceding this project, a number of private industry vendors had approached Caltrans with solutions to collect travel time data on highways and city arterials. Solutions revolved around two basic concepts and trends. The first trend was based on leveraging new technologies that significantly lower the cost of fixed detection. Both in-pavement technologies such as wireless magnetometers from Sensys Networks, Inc. and off-pavement technologies such as radar-based sensors by Speedinfo, Inc. offered much more attractive price points than inductive loops and made it conceivable to augment detection to a level that would yield accurate traffic maps and travel time estimates. An alternative concept was to use so-called mobile traffic probes to measure travel times from actual trips. Mobile traffic probes are essentially vehicles that are tagged and tracked along a corridor. This concept can be implemented by toll collection tags and readers, or by automated license plate readers. In either of those two cases, travel times are collected for preset segments of roadways in between readers. For instance, the San Francisco Bay Area 511 system relies for a large part on data collected from FasTrak readers. This solution came with some additional challenges, in particular the necessity of deploying additional infrastructure to "collect" the data from these readers, which unlike cellular devices, cannot rely on industry and market driven communication infrastructure.

In the decade preceding the launch of Mobile Millennium, cell-phone based technology had gained momentum as a promising avenue, although previous research and field tests conducted until the Mobile Century experiment in 2008 were not conclusive. This technology relied on positioning provided by cellular networks, which is still known to be quite approximate. The introduction of GPS receiver chips into more and more handsets represented a new opportunity at the time the project was launched. The prospect of large numbers of GPS-equipped cell phones reporting position and speed with higher accuracy at regular intervals represents a huge leap forward. Yet its implementation required addressing key questions regarding individual privacy, data ownership, network load, and proper traffic flow estimation techniques.

28.3. Immediate conclusions from the experience

28.3.1. Industry perspective

One of the immediate conclusions from the project from an industry perspective is the paradox of the mobile internet, for which Mobile Millennium was a perfect illustration. The mobile internet is a space in which all the major high tech companies want to be, however, it is still at

this time extremely difficult to monetize. This is particularly true for traffic, which explains the state of the art of industry in this field today.

The history since 2008 is unequivocal: the market has witnessed in the last three years an unprecedented war for domination of the smartphones. It first materialized itself with the emergence of two operating systems, the Apple OS for the iPhone, soon followed by its competitor, the Android. In the process, Symbian, which had previously dominated the market, started to rapidly decrease in importance and relevance (as indicated by the small number of apps available for it). Two of the other OS are today at a crossroads, RIM (Blackberry) and Windows Mobile, which may yet survive in this market, with the appropriate partnerships (for example, by the Nokia-Microsoft alliance started in 2011). The mobile space is a difficult place for new businesses to emerge because most of the major companies want to be there, and they are willing to tolerate that their presence not yield revenue in the short term. In fact numerous start up companies started businesses based on the assumption that one could sell traffic data. However, Google started to give away traffic products (on the Android) as early as 2009, thus directly threatening aftermarket device manufacturers, and other companies whose profit might rely on the traffic data they were collecting, and later potentially selling. As a result, the public (in 2011) expects to have traffic data, traffic information, routing, and turn-by-turn navigation for free (bundled with the functionality of the phone like on the Android Nexus S, and soon other models), the paradigm initially envisioned in 2008 changed. Today, not having traffic information as part of a suite of geolocalization services of location based services is a negative, but having it is not necessarily revenue generating.

When this project was started, the team worked under the assumption that the launch of Mobile Millennium as a pilot, with a corresponding field operational test, would logically lead to the launch of a new product for Nokia, based on the experience learned during the pilot. For various reasons having to do with the low penetration of Nokia phones in the US, and because of the necessity to launch Mobile Millennium on the iPhone for it to be successful, this never happened. This lesson is indicative of the necessity for services, such as the one provided by Mobile Millennium, not only to be revenue generating, but also to fit within a business plan of a company. For example, even though Google Maps (and corresponding traffic and geolocation products) might not generate revenue directly, it might still make sense for a company like Google to continue leading this field (on its own investments), to guarantee a sustained supremacy over integration of all mobile services available on a phone (mail, geolocation, traffic, calendar, search, etc.). In the case of Nokia, such a necessity was not as clear, which explains why Mobile Millennium stopped at the level of the pilot and field operational test. In the mean time, the industry context had changed enough that it did not make sense anymore for Nokia to pursue this endeavor. All the other smaller companies that built their model based on similar concepts are collecting or purchasing mobile data (INRIX, NAVTEQ, Waze, BeatTheTraffic, etc.).

While this rapid change in the industrial context with respect to monetization of the mobile internet (and traffic in particular) could probably not have been anticipated, it provides different opportunities to the DOTs today (which are explained in the later sections of this chapter).

28.3.2. Government perspective

Numerous questions of value to the government were answered during this project. The first question was to know if fusion would work, i.e. if the process of augmenting dedicated infrastructure data (such as loop detectors) with probe data (in particular smartphone data) could work. The success of Mobile Millennium demonstrates that this fusion process is possible, and thus moved the state of the art one step further. Today, several Departments of Transportation (including the US DOT and the California DOT) are conducting investigations on data procurement and the possibility of complementing their own data sources with probe data.

Another interesting fact revealed by this study is the issue of the fragmentation of the market for probe data. One of the consequences of the situation described in Section 1 is that while the forecast of the explosion of GPS enabled smartphones was correct (see Section 1 below), no single player (including Google) today seems to have enough data to cover the entire transportation network adequately. While numerous companies have made significant claims about their coverage (i.e. the proportion of the network they deliver processed information for), it is still unclear today who has adequate vehicle penetration and where. This raises some very important concerns for the government.

- *Unverifiability of the quality of processed data.* As was demonstrated in numerous places in this report, the quality of traffic estimates is contingent on the amount of data. Today, many of the companies who sell processed information, are unwilling to share the raw data sources (or even to share the amount of data) that contribute to this information, thus making the quality of the output unverifiable.
- *Institutional difficulty to aggregate.* Because many of the companies in the business of collecting data or providing traffic information content are competing against each other, it is difficult for them to create partnerships, and thus are(for now) doomed to work with small amounts of data (since they cannot combine their own data sources). One of the key challenges for DOTs to overcome is to break this insularity and to create the proper mechanisms to enable data fusion at a global scale. Having such mechanisms is a prerequisite for high quality DOT products. Thus one of the most important conclusions for the government at the end of this project, is the necessity to create the proper institutional mechanisms to enable a data market at a global scale from which the DOTs can benefit. While collecting data and developing traffic applications for traffic information should mostly stay on the private sector side, the role of making the data available to a community broader than the entities collecting the data belongs to the DOTs, and is now the object of other ongoing efforts (in particular from the procurement angle).

28.3.3. University perspective

One of the major lessons learned from Mobile Millennium is the benefit for both the DOTs and industry to partner with the academic community to launch new technology applications that require significant scientific and technological breakthroughs. The complementarity of the talents within the Nokia-Berkeley team exemplifies the benefits that both parties received from this partnership. It was never in Nokia’s intention to develop a traffic research team to support

the rapid launch of the program. While NAVTEQ (now part of Nokia) did later, it would have been almost impossible to create a critical mass of half a dozen PhD students and researchers to support this effort inside Nokia. On the other hand, the UC Berkeley part of the team had very little knowledge on mobile computing, cloud computing, and real-time systems, which made the interaction with the Nokia team synergistic. Also, UC Berkeley provided the institutional link with the California and US DOT. This role also illustrates the benefits of working with the University. In the context of traffic, given the rich past of the University relationship with the California DOT, Berkeley was the right partner for the California DOT to choose.

The Mobile Millennium project significantly influenced the transportation community at Berkeley. While mobile technology had predominantly resided in communications groups at Berkeley, this project anchored part of it in the Civil and Environmental Engineering Department, through this application (traffic), soon to be followed by other applications (transit, water resources, earthquakes etc.). While it would be excessive to say that the agenda of the Civil and Environmental Engineering Department changed due to the Mobile Millennium project, the image of its research agenda definitely became associated with this project, due to its visibility. Within the CEE Department, this research was hosted by the System Engineering program, which gained significant visibility as well. UC Berkeley was a pioneer in what has now become a trend: CEE Departments today offer increased numbers of Assistant Professor positions to candidate with knowledge and expertise in wireless technologies, cell phones and sensor networks. Just in the last 3 years, MIT, Stanford, University of Michigan, and University of Illinois at Urbana Champaign (the leading institutions in CEE) have each hired at least one, (sometimes more) Assistant Professors with such expertise, several of whom are alumni of the Mobile Millennium project.

The Mobile Millennium project was the first academic project of this scale to demonstrate the impact the mobile internet could have on a traditional field of CEE, in the present case of transportation. It was a precursor of many of the publications available today in the literature in which researchers devise new methods to fuse mobile data with other infrastructure data.

One last finding which was also exemplified by the partnership is the role a University can play in a pilot study or deployment. While it is not in the University's mission to operate and maintain a traffic product, being in charge of a pilot study can be of great use. It was never the intention of Mobile Millennium to have Berkeley operate, manage, and maintain a traffic application on an ongoing basis. In fact, UC Berkeley offers numerous cases of start-up, and spin-off companies taking the technology developed at the University to the next step (for example Berkeley Transportation Systems operates the PeMS system; Sensys developed sensing technology for next generation loops). On the other hand, a University is a great entity for incubating a pilot deployment, along with its imperfections, which might not be compatible with standards of industry. UC Berkeley played that role in the Nokia - Berkeley partnership.

28.4. Advances to research, technology and practice

28.4.1. Post facto context: closing the project in 2011

As of the writing of this report, the world of the mobile internet is very different from when the

project started in 2008. As a reference [42] the numbers of mobile devices and services are astronomical, and exceeded the expectations of the team when the project started. Today, 5.3 billion mobile devices are used worldwide—which represents 77% of the world's population. Smartphones represent 21.8% of all mobile devices. At the present time, despite the enormous success of the iPhone and the Android platforms, Nokia still holds an enormous share in sales of phones. During the project, social network usage exploded: Facebook now tops 629 million registered users with almost 250 million people accessing the site via mobile devices. This usage of mobile devices was not anticipated when the project was started. While the number of geolocalized uses of Facebook or Twitter is still limited, it promises to be a significant source of geolocalized data in the future. The situation is similar in foreign countries, for example, China's version of Facebook, Qzone, is experiencing an enormous growth with 480 million registered users. In the US, Twitter broke the 200 million registered user mark with nearly 40 percent of people tweeting via mobile (which as mentioned before is very promising for GPS data). Geolocalized tweets today are still a minor part of the overall number of tweets. A less known fact is that Hotmail still dominates email, but gmail's usage is increasing fast, leading to the integration of numerous Google products on Android. In the mean time, other geolocalized services and location based services have emerged, which also provide additional sources of probe data (or at least human activity). For example, Yelp is topping 50 million unique visitors per month. Its move to team up with OpenTable earlier this year will only increase its relevancy. Other examples are Foursquare and Gowalla, which are still growing but do not seem to have the potential to reach the numbers of Facebook or Twitter with the current functionalities they have.

The aforementioned situation is thus very different from what was anticipated when this project started in 2008. In particular, maybe with the exception of Google (due to the massive use of its mobile maps client), the scales of usage of mobile applications for traffic (in particular all the companies that have built a business model based on traffic) are much smaller than the examples mentioned above. Thus in the future, while the traffic applications developed specifically for traffic information might provide sources of data in the near future, it is likely that they might soon be taken over by other geolocalized postings that might provide valuable information as well. The growth of probe data is inevitable, and it is clear that in a (near or not so near future), when penetration rates on the highway reach 10% or 20%, the situation of traffic information and traffic operations will change drastically. In the mean time, dedicated traffic applications are among the main sources of data, and their quantities are slowly rising, in a fragmented market described earlier, which will be the basis for future DOT usage of this data.

This project thus closes in a very different technological context than the one it started in. The explosion of smartphones has happened, the market is very fragmented, no single player has enough data for providing the DOTs with the sources they need, and today, creating a successful traffic app is virtually impossible. At the time this report is written, the Apple app store totals already more than 350,000 apps, which makes it almost impossible for any new app to go "viral" as initially envisioned for this work. Note that Mobile Millennium started before the context of the app store for the iPhone, at a time when apps were downloaded from webpages. This drastically different situation emphasizes the fact that in today's fragmented market, fusion will be the next direction to create the proper traffic and transportation tools for the DOTs.

28.4.2. Research

UC Berkeley served as a contractor in the execution of Mobile Millennium, with specific deliverables which were summarized in the previous chapters. The nature of the deliverables led to significant advances in the state of knowledge in transportation engineering, which are summarized below.

Modeling

Because of the specific nature of probe data, it was necessary for the team to work on several aspects of modeling. This was necessitated by the challenges created by the mobile data, and the features they come with. For highways, one of the most important contributions (absolutely crucial for this project) was the creation of a *velocity model* (see Chapters 11 through 14), which could encompass velocity measurements only. This is in contrast to density based models that rely on counts of vehicles. The algorithm that runs at the core of Mobile Millennium is based on this model, which constitutes a significant breakthrough required to make full use of the traffic data available from probe vehicles (phones or otherwise). Subsequently, so-called second order models were developed to encompass the variability in speed distributes potentially recorded by the phones in models. While this type of variability might not be visible from more classical sources of data, such as loop detectors, it is clearly visible from probe data, as was demonstrated earlier in the report. The second order model developed in Chapter 23 also constitutes a significant advancement to the state of the art, and opens the door for numerous technological implementations of this model. Finally, a breakthrough concept of this project was the idea of internal boundary condition modeling, which consists in prescribing the value of a function along the trajectory of a probe vehicle. Chapters 24 through 27 describe Lagrangian formulations of internal boundary conditions in Hamilton Jacobi equations, and thus provide the proper mathematical formulations of integration of such measurements (trajectory based) into the models. This is a significant breakthrough, which enables the integration of mobile data into flow models in a way that keeps the Lagrangian data as such (i.e. it does not need to translate mobile data into discretized static data).

On the arterial side, the models are based on classical queuing theory. As described in Chapter 15, this was the natural framework to start from, as these models have been well established over the years and provide sound descriptions of traffic patterns at traffic lights. However, as was revealed from the data, standard assumptions usually made when dealing with these models are not necessarily satisfied. In particular the standard assumption that cars arrive at locations with a uniform flow has been shown not to be valid in numerous cases investigated; in fact, as was revealed by several of the tests ran as part of Mobile Millennium, cars bunch up, and pass through intersections in platoons, a fact that causes the queuing approach not to be accurate. Thus, the work achieved in this project took a probabilistic route and stochastified the queuing approach traditionally pursued in a deterministic setting. This description enabled the creation of models that can encompass delay distribution in arterial networks, without necessarily knowing the actual length of the queues (in a deterministic setting), but in an expected sense. These models have been implemented in the pilot system that ran for the duration of the Mobile Millennium project (see Chapters 16 through 19).

Privacy

The Mobile Millennium project started at the beginning of an era in which cell phone data are now crowdsourced at massive scales. In 2008, it was not clear what angle the public would take on privacy, and this project thus took a route that focused on privacy protection. The main concept invented (and used) through this project was the concept of *virtual trip lines* (VTLs), which are geographic markers deployed on the map, which trigger phone updates when phones cross them. This approach was natural from a transportation perspective (it represents a "virtual" loop) but somewhat disruptive in the mobile sensing world. On the privacy side, the development of the VTL concept, and the corresponding studies that were conducted on the system (in which VTLs were implemented) was a significant contribution in the world of privacy (see Chapter 20). The concept of an electronic portal had a great future in the world of smartphone based location based services, and Mobile Millennium was the first instantiation of this concept.

As seen in Section 1, the world of 2011 has seen enormous expansion of shared data through participatory sensing, crowdsourcing, and other mechanisms. Also, other social aspects of privacy appeared, with the explosion of social networks such as Facebook, leading people to be more inclined to share more data. Today, while privacy policies are stated explicitly in the agreement forms to be signed when downloading a new app, practices have shifted towards more sharing, less on privacy concerns. For example, in the current Google privacy practices [36], one can read the following sentences: *"Sometimes, we record your phone number. We record your phone number when you send it to us; ask us to remember it; or make a call or send a text message or SMS to or from Google. If you ask us to remember your phone number, we will associate your phone number with your Google Account, or, if you do not have a Google Account, with some other similar account ID. We often generate this account ID based on your device and hardware IDs, so if you change your device or hardware, you will have to re-associate this new device or hardware with your account before we can authenticate you."* The role of the VTLs and corresponding architecture was specifically to make sure that re-identification (in particular through phone numbers) would not be possible within the Mobile Millennium system. In the last three years, history has shown that the general public seems comfortable with sharing this type of data (though a significant portion of the population might not realize that the practice above actually makes them share more than they think they are). The Google privacy practices [36] furthermore state *"Most of the other information we collect for mobile, such as your device and hardware IDs and device type, the request type, your carrier, your carrier user ID, the content of your request, and basic usage stats about your device and use of Google's products and services does not by itself identify you to Google, though it may be unique or consist of or contain information that you consider personal."* Such practices have been studied in the past, and one of the issues with these are potential re-identifications using statistical methods. These issues are stated explicitly, so people have a choice today, and seem to be comfortable with them. Specific to geolocation, the Google policies are also very clear: *"However, if you use an Android-powered device, Google will associate your device id with your Google Account in order to provide services, such as sync functionality for your Google email and contacts. [...] If you use location-enabled products and services, such as Google Maps for mobile, you may be sending us location information. This information may reveal your actual location, such as GPS data, or it may not, such as when you submit a partial address to look at a map of the area."* The implications of this statement are numerous, but one of them is the ability

to track specific individuals through their phone activities, and in particular through the disclosed locations they transmit. Again, a significant portion of the population seems to be comfortable with these practices today.[24] Mobile Millennium provided an alternative architecture in which the typical privacy issues dealing with disclosure of identity, and re-identification would be more protected. In today's technological context, it seems that people do not require such levels of identify protection, at least for now in the United States.

Filtering

Filtering in the context of this section is understood as the process of cleaning data (as opposed to estimation techniques such as Kalman filtering and extensions, presented in the next section). Part of the contributions of Mobile Millennium was to come up with coherent filters that enabled to usage of data that was inherently extremely noisy (see Chapter 6). This is particularly true from the loop detector data, which is notoriously noisy, but also to a certain extent from the probe data, which comes with its own challenges. Many of the algorithmic contributions of Mobile Millennium was to create filters enabling us to clean data from the corresponding sources. These filters were adapted to the different sources of data. One significant filter of interest to this work was the map matching / path inference filter created for low frequency probe data (see Chapter 10). This example of filtering is a perfect illustration of the contributions brought to the field from the work presented in this report. Overall, one of the contributions of this project was to develop the appropriate filtering tools to be able to create a fusion engine that could work at a large scale and in real-time. These filters are important for practical applications, and have significant impact on the practitioner 's world.

Estimation

One of the major contributions of Mobile Millennium was the advancement of the state of the art in estimation. Estimation in control theory means "nowcast" (as opposed to prediction, which means "forecast"). Estimation is sometimes also referred to as inference in machine learning. The models developed were used in practice for estimation, i.e. for "nowcasting" traffic based on the models and the data. As a very brief summary of all chapters that present the work performed on this topic, the process of estimation consists in finding the most likely state of the system (traffic) given a mathematical model (chosen) and data (measured). Most likely in this context can have several different meanings as were presented earlier in the report. One of the contributions of the work was to produce an Ensemble Kalman Filtering approach to traffic, at a scale it had never been implemented before (see Chapter 14). The Ensemble Kalman Filtering approach was particularly appropriate, given the challenges of the models (see in particular the chapters on first and second order models using partial differential equations). The Ensemble Kalman Filtering approach enables one to bypass the difficulties of such models which have inherent nonlinearities and whose solutions have smoothness issues. Demonstrating that the Ensemble Kalman Filtering approach works was a major contribution of this work. In fact, the Ensemble Kalman Filtering based algorithm was turned on in 2008 when the system was started

[24]The example of Google is quoted here because Google is today the most successful provider of integrated mapping / traffic / location based services at a global scale, thus a representative product of the state of the art.

and worked uninterrupted in the live system until the end of the project. Other approaches were also necessary for the system to evolve into a multi-featured traffic information system. In particular, following the work done on Hamilton Jacobi formulations of traffic, algorithms for robust estimation of traffic were developed (see Chapters 26 and 27). These formulations are important because guaranteed bounds are almost more important than giving actual estimates.

In the context of arterials, the estimation algorithms developed here were focused specifically on travel time, since it was very clear from the start of the work that the volume of probes would not be sufficient for reconstruction of the volumes on this part of the network. A convex optimization framework to solve this problem was explored in Chapter 21. Machine learning approaches to traffic were the most heavily studied in this report, which is quite different from the traditional direction taken by the transportation community over the last two decades (see Chapters 18 and 19). Indeed, most of the work in traffic flow modeling (and thus estimation) relies on physics based traffic models (mostly inspired from hydrodynamic theory and variations). Machine learning models are statistically based, and might not take physical models into account at all. Several contributions of this work were specifically to create machine learning models inspired by physics (for example as was mentioned earlier by stochastification of the deterministic traffic flow models). These contributions were necessary, because they enabled the integration of probe data into models (statistical or not). All of these contributions are significant breakthroughs in the field.

Routing

Because the Mobile Millennium project was able to produce statistical descriptions of traffic, it enabled routing applications in which the statistical features developed during the project were used to quantify the uncertainty in the routing procedures. In particular the statistic features of traffic enabled us to provide new solutions to the problem of “arriving on time” and enabling its use in real-time mobile phone applications (see Chapter 22). Optimal routing in transportation networks with highly varying traffic conditions is a challenging problem due to the stochastic nature of travel-times on links of the network, which was resolved partially by our algorithms for arterial traffic estimation. Most common routing algorithms consider the expected value of link travel-time as a sufficient statistic for the problem and produce least expected travel-time paths without consideration of travel-time variability, which was a specific problem investigated in our work. Indeed, in numerous practical settings the reliability of the route is also an important decision factor. Thus, we consider the following optimality criterion: maximizing the probability of arriving on time at a destination given a departure time and a time budget. For these, we provided a new approach that creates routing policies. A routing policy is an adaptive algorithm that determines the optimal solution based on en route travel-times and therefore provides better reliability guarantees than an a-priori solution. This advances the field of travel planning and thus fits well in the contributions of this project performed for traffic information systems. Overall there is still significant work to be done with routing, but it will require the advancement of several other fields, in particular that of traffic forecast.

28.4.3. Technology

This project has drastically advanced technology, in several fields, which include location based

services, real-time algorithms, online algorithms, visualization, with applications to traffic. Mobile Millennium was mainly a technology project though, numerous other aspects were important as well, in particular the privacy aspect). This section summarizes the different contributions made to technology as part of Mobile Millennium.

Location based services

A location-based service (LBS) is an information or entertainment service, accessible with mobile devices through the mobile network and utilizing the ability to make use of the geographical position of the mobile device (this definition is due to Wikipedia). In the present case, Mobile Millennium was one of the early instantiations of location based services for traffic. It enabled the development of expertise at UC Berkeley and within our partners at the DOT to understand better the challenges presented by such services, as well as the opportunities for government to use these services. While at the present time, it is clear that location based services are mostly to be prototyped, developed and deployed by the private sectors, the understanding of data and the information that can be collected from these services is very important in the development of future traffic information systems. Mobile Millennium was key in that respect, as it helped with the understanding of what could be done with the data, which is now a key are of interest at the US DOT and the California DOT.

From a technology perspective, the vision of Mobile Millennium, and the deployment of an early app for traffic was just the beginning of an era that is blossoming extremely fast, with the emergence of dozens of traffic information systems on mobile, for example Google, Waze, INRIX, BeatTheTraffic, and several others. The research leading to this technology was a success. The institutional reasons why one company might be more successful than another in widely deploying a technology have been explained earlier and are important in understanding today's context for traffic monitoring.

Real-time and online algorithms

A key contribution of this work is the development of real-time and online algorithms. Real-time refers to the fact that algorithms can run faster than the physics i.e. in the context of traffic, the algorithm can provide a forecast faster than the time it takes for traffic to reach that state (or to provide a nowcast almost instantaneously). While the private sector has demonstrated the ability of performing such tasks, the corresponding algorithms are almost all unavailable because of IP protection issues. One of the contributions of this work was to provide published algorithms (as outlined above and described in detail in Parts II, III, IV, V, and VI), but also to implement the corresponding algorithms, in order to demonstrate the capabilities of implementing these algorithms and their effectiveness in practice. Online refers to the feature of being able to "absorb" data while the algorithm is running (i.e. not all the data is initially available, but only some of it, while the rest comes in streaming). In the case of traffic, obviously streaming data is going to arrive constantly, and thus the project demonstrated the ability of integrating live feeds into the system. Real-time and online aspects are key at a time when volumes of available data are increasing extremely rapidly, and when processes that handle the data are becoming more and more complex. Part of the work leading to the success of Mobile Millennium was specifically to demonstrate the feasibility of creating such algorithms and implementing them.

Visualization and visual analytics

While it was not the goal of the project to make scientific contributions in the field of visual analytics, the project advanced the field of traffic visualization as well, by creating tools to enable one to view traffic and traffic features (quantitative) and superimpose them on a map. The field of real-time traffic mapping is still relatively new at the timescales of ITS (i.e. less than 10 years old). Mobile Millennium was one of the first projects to map arterial traffic in real time at the scale of California. As part of the work performed for the project, it also developed numerous visual analytics tools necessary to assess the performance of the different algorithms developed as part of this work. Overall the Mobile Millennium project developed a packaged solution to traffic monitoring based on GPS enabled smartphones, which also included backend visualization tools, which constitute an important part of this work, and have advanced the field of traffic visualization (see Chapter 8).

Cloud computing

Between the start of the project in 2008 and the end of the project, cloud computing emerged as a paradigm to make high performance computing more accessible. For example, companies like Amazon today offer EC2 type services which are very useful for scaling applications that are initially small but need to quickly scale in size. This evolution happened during the development of Mobile Millennium, and strongly influenced our designs. In this report, we summarized our experiences scaling up Mobile Millennium, and in particular the machine learning algorithms that constitute the core of the system (see Chapter 10). Although numerous designers of cloud computing systems have evaluated their systems with simple machine learning applications, we found that our real-world application posed several challenges that are not widely studied, including managing large parameter vectors, using memory efficiently, and integrating with the application's existing storage infrastructure. We believe that Mobile Millennium is representative of a wide range of machine learning problems beyond traffic estimation, and thus believe that the lessons learned when scaling Mobile Millennium will have significant impact not only in cloud based traffic applications, but in general applications of machine learning algorithms on the cloud.

28.4.4. Practice

Some of the contributions of Mobile Millennium were also the practical work done to make this project succeed. While this does not necessarily constitute a significant theoretical or academic contribution, this is very helpful to the state of practice.

User recruitment

One of the major contributions of the work for user recruitment was the diversity of the communication forms initially utilized to recruit users. Because most of the recruitment was done in 2008 when the application was launched, the lessons learned would most likely not apply anymore, as the context of the mobile internet has drastically. At the time we targeted the driving public through various broadcast forms (radio, TV, press releases, press conferences), which led to significant enthusiasm of the public, leading to significant internet traffic to our website and

subsequent downloads (see Chapter 4). This was mainly before the era of the app store (iPhone or android) and would probably not apply anymore. Today, with the large number of smartphone apps (over 300,000 just in the iPhone app store), recruitment of users is very difficult for traffic apps, mainly because people need convincing reasons for downloading applications in addition to Google maps, which provides almost all features required by motorists today.

In addition, one of the key issues discovered during the pilot was the difficulty of retaining users. When we started the pilot we managed to achieve significant participation of users, but the numbers inevitably decreased with time. This problem observed during the pilot is even more clear today, given the enormous competition in the sphere of mobile traffic applications. One of the findings of this work was the necessity for traffic applications to have a satisfying user engagement strategy, in order to be able to keep the pool of users engaged.

Practical tests

One of the other contributions of Mobile Millennium was the development of test protocols for algorithmic testing. Mobile Century enabled the development of numerous algorithms and methods to integrate traffic data into highway traffic flow models. Mobile Millennium did the same with arterial traffic. In particular, with tests in New York, San Francisco, and Berkeley (see Chapter 5), the situations encountered for arterial traffic enabled numerous contributions and are a significant step forward. A significant portion of the work currently done in traffic modeling works under very idealized assumptions, which are not necessarily satisfied in practice. The tests performed as part of this work, and the data collected, enabled the team to push the state of practice, by identifying the limitations encountered when working with idealized assumptions.

Deployment

One of the major contributions of Mobile Millennium is *being* an actual deployment (see Chapter 5). Unlike traditional research projects, which are merely a study or an analysis, Mobile Millennium was a field operational test, with all its challenges and issues. Historically, it was (and probably will be) the only such deployment, since the field of GPS enabled smartphone traffic monitoring is now mature, and has moved to the private sector.

Customer service

Finally, one of the lessons of Mobile Millennium was the difficulty of customer service, particularly in the “developer’s hell.” Mobile Millennium was made available to the public on half a dozen phone platforms (and variations): most of the GPS equipped Symbian phones, and some of the Blackberry phones. Each of these phones posed specific challenges, which created the need for helping the public dealing with these applications, at an early age of traffic apps (and apps in general). The team created a forum (see Chapter 4) that was used by the team to provide a form of “customer service,” which also provided a lot of valuable experience. Today, three years after the launch of the program, it is clear that most of the available traffic applications are made available to the public by the private sector, which tries to minimize the level of required customer service, a significant cost. As an early instantiation of such an application, Mobile Millennium uncovered the same issues.

28.4.5. General outreach

While from a contractual perspective, the deliverables represent the obligations of the University with respect to its funders, the research performed as part of Mobile Millennium also had numerous other benefits for the institutions involved which are worth mentioning. Mobile Millennium was funded by a consortium, for which the three main funders were the US DOT, the California DOT, and the National Science Foundation. The National Science Foundation provided some of the funds essential for developing some of the most theoretical contributions of this work, which later on went into the Mobile Millennium system. As such, and with the presence of numerous other (minority) funders, the outreach contributions of Mobile Millennium have to be judged in the context of all of its funders, and in particular with respect to academic institutions.

Mobile Millennium enabled several Ph.D. thesis, which were defended during the project, or will be defended in the future (depending on the start date of the students). The number of Ph.D. students graduated (or to be graduated) for which the research performed contributed to Mobile Millennium is by itself a testimony to the success of the program (in total, about 10 students will have graduated and have been contributors to this program). Some of them later became University Professors themselves in prestigious institutions in the United States (Massachusetts Institute of Technology, University of Illinois at Urbana Champaign), and abroad (La Catolica Pontificad University, Chile, KAUST University, Saudi Arabia). Others were offered positions in industry (IBM, Telenav, NAVTEQ, and many others), which shows the relevance of the work to the private sector.

This project also enabled the training of numerous undergraduate students, and visiting students, who later came to the US to study at major universities such as UC Berkeley. As part of its mission of education, UC Berkeley used the project to motivate young students to embrace the field of transportation and mobile sensing for their career.

Overall, more than 50 academic publications have been or will be published based on the work performed for the Mobile Millennium project, which is enormous, and shows the impact of the work in the field. The project won several prizes, including the TRANNY Award, California Transportation Foundation, 2009, the Best of ITS Award, citation for “Best Innovating Practice,” ITS World Congress, New York, 2008. The PI for the effort won several prestigious prizes as well, including the Presidential Early Career Award for Scientists and Engineers (PECASE) from The White House, 2010, and the CAREER Award, National Science Foundation, 2009. Students involved in the project won several awards as well, including the Leon O. Chua award, traditionally awarded for contributions in nonlinear control, and the Rodney E. Slater Award, ENO Transportation Foundation, 2010. Three of the Mobile Millennium students became Eisenhower Fellows and one became an ENO Fellow. Overall this series of achievements illustrates the impact of this Mobile Millennium on the field.

In parallel to the technical development of the Mobile Millennium system and program, the project supported a lecture series co-funded by Nokia, which had enormous impact in the field, and enabled UC Berkeley to promote cyberphysical systems as an agenda for research. In the Fall of 2008, the California Center for Innovative Transportation, the Center for Information

Technology Research in the Interest of Society, and Nokia presented and hosted the *Distinguished Lecture Series on Cyber-Physical Systems* (CPS).[25] The lectures series was launched during the Fall semester of 2008, to coincide with the launch of Mobile Millennium. It gathered experts in the field of Cyber-physical systems, who each gave their own perspective on this topic, in light of their respective experience in the context of their own research. This enabled the team to maximize the impact of the project in the academic community.

Finally, the work performed for the Mobile Millennium was presented in numerous industry, academic and government venues. More than 50 seminars and conference presentations showcased the contributions of the project, making it visible at a national scale, which demonstrates its success. The project was viewed more than 100 times in the media (TV, radio, newspapers, and technical blogs). It was presented to high profile executives from government and industry and has been prominently featured at UC Berkeley as part of the CITRIS museum on campus.

Overall, the project has been one of the most visible projects on transportation in the academic world in the last decade.

28.5. Lessons Learned

There were many lessons learned through the process of designing, building, launching and operating Mobile Millennium. These lessons constitute valuable knowledge for the agencies interested in using similar technologies, or using the outcomes of similar technologies.

Challenges

The biggest challenge of this program was the difficulty of keeping users engaged. This has at least two causes. First, Mobile Millennium was only available on Nokia (Symbian) and Blackberry phones, which was a reduced part of the market. It thus limited user recruitment and thus user retention. Second, at the time when Mobile Millennium was started, the term "developer's hell" still made sense to a significant degree, i.e. there was, at the time, no single platform dominating the market, thus it was very difficult to have an app "go viral" as one sometimes sees in today's technological context. At the end of the project, with the domination of the iPhone and the Android, the situation is very different, but this could not have been anticipated, as the Android did not yet exist, and the iPhone was just starting its conquest of the market at the time. In addition, there were obvious issues in developing the phone software client for the iPhone during the duration of the project.

[25]Cyber-physical systems are systems that integrate computational processes and physical processes. The tight integration between computation and physics differentiates CPS from traditional embedded systems and is the focus of active research in numerous scientific communities around the world. Mobile Millennium is a Cyber-physical system that is based on participatory sensing. The physics of the system (motion of people in the transportation network) is modeled inside the traffic estimation engine of Mobile Millennium and is coupled to the information flow (sensing, using the cellular communication network).

The necessity of building a cloud based infrastructure has also become a reality, which was revealed by the work. While it was not anticipated at the start of the project, the computational processes developed as part of the system needed an architecture that could support massive computations, and soon, a paradigm shift was needed, to switch from dedicated computational infrastructure to cloud based infrastructure (this is described precisely in the corresponding chapters of this report). This was successfully achieved during the project, but constituted a significant challenges, as a radically new architecture is needed to be able to build such a system (which Mobile Millennium now possesses at the conclusion of the project).

Problems

One significant problem encountered during the project was the driver's distraction laws passed at the federal level, which had direct consequences on the work. First, it became necessary to immediately stop the downloads of the app, in order to comply to federal regulations. Second, because of the interrupted download options, the number of users decreased inevitably. This could not have been anticipated, and these types of regulations need to be taken into account for future work on this topic.

Items to avoid

In the context of 2011, one of the clear conclusions of the Mobile Millennium is that it will be beneficial for the government to leverage the private sector when starting a new app of this sort to collect data, or to perform any other task. In particular, one can already see that Google Maps has surpassed any other mapping system in the capabilities it provides to the public, and thus any mapping system developed by a consortium of the type of Mobile Millennium should leverage the capabilities offered by Google. Thus, in the future, having most of this work outsourced will lead to significant savings for the government (in particular a lot of the Google products are free), and improved quality of service. In the historical context of 2008, it made sense to develop a prototype for the system we developed. A few years later, most of the mapping features we developed are available from the private sector, most likely for free, thus one should avoid re-developing them.

Advice for future research on the same topic or similar topic

In order for a system or service like Mobile Millennium to be effective and attractive to the public, one of the necessary conditions today is to be able to bootstrap it, so from day one it provides value to its customers. In that respect, the Mobile Millennium had all the necessary assets, in particular it provided highway traffic to the driving public from day one (when this data was partially collected by NAVTEQ). We see this trend today in mobile apps providing traffic information: many of them today prefer to provide inaccurate information (or unverifiable information) on some part of the network, rather than no information. While these practices can be discussed, they are unfortunately creating expectations on the part of the consumers, which are thus now becoming a prerequisite for the success of new apps.

28.6. Plans for deploying the research findings

28.6.1. Immediate practical application of the research findings

From a university perspective, Mobile Millennium was a research project, and is thus a complete success as demonstrated by the very high performance on the typical metrics used to rate research projects (see the section on research ,above). In addition, Mobile Millennium provided a significant amount of very valuable research findings, which have practical applications.

The application of new procedures

The Mobile Millennium created a new paradigm, the *Virtual Trip Lines* (VTLs), which now serves as a reference for data collection in a privacy preserving environment. This concept is broader than just for traffic data collection, it instantiates the process and practice of a phone based electronic portal, which has several other applications in transportation, in particular tolling, insurance, etc. The VTL paradigm constitutes a significant advance which was provided by the outcomes of the project.

The project also made very immediate and practical contributions in the field of data fusion. The systematic use of flow models and statistical models for characterizing levels of congestion in the transportation network is another significant advance provided by Mobile Millennium. As such, all algorithms developed are now in the public domain, and are available to practitioners to develop.

Finally, the project also pioneered the field of adaptive routing, which was one of the components of traveler information systems on the agenda of the Mobile Millennium team when the project began. The main contribution in this field was the development of new routing policies that are adaptive, i.e. which are capable of adapting based on traffic. There is still significant amount of work to be done in this field, and the research presented in the chapter on routing is just an early instantiation of this topic.

The issuance of new specifications, standards, or designs

One of the most significant contributions of Mobile Millennium was the questions it raised on the issuance of new specifications, standards or designs. Mobile Millennium was one of the earliest instantiations of probe data collection at massive scales. This data raised questions which are not answered today, but for which Mobile Millennium provided at least a first set of answers. In particular, through the various feeds which were collected and built with our different industry partners (NAVTEQ, Telenav, Cabspotting, and others), it became clear that exchanging GPS data would have to come with specific standards (in particular, time of measurements, delay to be received, specs on error, procedures to handle duplicates, etc.). The findings here are enormous, and will shape the future of traffic information systems. The current and ongoing work at CCIT (and in the community at large) will make extensive use of these findings, and will require more work to continue advancing the practices, required for exchanging data at massive scales.

The privacy practices developed during the Mobile Millennium project are directly applicable and also represent significant contributions for establishing new procedures and guidelines for practice in the field of probe data collection.

Mobile Millennium was also a project that enabled the definition of numerous features which today have become standard in the practitioners' world. In particular, front end design (mapping app on a phone, audio features to enhance the user experience and comply to hands free distraction laws, and a reporting feature for erroneous data). Overall, Mobile Millennium was initially conceived as a program that would be used to enhance user feedback (in particular through direct feedback about the quality of the data, a practice which was new at the time, and interactive).

Mobile Millennium was an app before the rise of the app store, which was also a new practice. It pioneered cloud based traffic systems, by defining an architecture based on services such as EC2 (provided by Amazon), a practice now standard in the private sector for any company that wants to start a service easily or to scale quickly.

28.6.2. Institutional context for deployment of the research findings

The question of determining the proper institutional context for the deployment of the research findings is a deep one. On one hand, the development of a smartphone based traffic client clearly should be done by the private sector. Based on the experience of Mobile Millennium, this is clear, as was explained throughout the report. In fact, in the organization of the work between Nokia and UC Berkeley, the smartphone client development tasks were specifically carved out for the Nokia team.

One of the fundamental questions of this work was to determine the proper model for the government to use the collected data. Should there be a partnership to collect data between government and the proper industry partners? Mobile Millennium provided a first answer to this question, which today is becoming closer to practice. As a prime consumer of the data, the government needs this information for several purposes, which include traffic information, traffic operations, and planning. The data needs to be purchased, as the government should not be responsible for collecting the data at a massive scale (which would be equivalent to deploying a dedicated infrastructure – a paradigm that for its lack of financial resources motivated the Mobile Millennium project). Thus, the proper institutional structures need to be created to collect the data, aggregate it, exchange it, and make it into information directly usable by the government. The proper institutional model for this is tripartite, i.e. (1) data collectors, (2) data aggregator, and (3) data and information consumer (the government). There is no consensus today in the private sector about this model, mostly because numerous entities are interested in providing (1) and (2) at the same time. This strategy leads more quickly toward market domination (but comes with challenges of transparency for both data and back-end systems functionalities). Current work is ongoing to help determine the proper structure to achieve these tasks and demonstrate that it works in practice).

28.6.3. Potential benefits

One of the major achievements of Mobile Millennium was to reveal the scientific value of information provided by probe data. The project demonstrated the contexts in which this data could be used to provide adequate information, if fed properly to traffic models. This was a major technical achievement, which was necessary, given the novelty of the data. The project at this stage did not put a specific value on the savings (expressed in dollar amounts) or give an estimate on the first year savings and the subsequent average annual savings anticipated upon application of the research results. The findings of the study are positive, but while not suitable for immediate application (because the integration of probes into a traffic information system has further consequences which are difficult to quantify without further consideration of all of Caltrans' operations), the report outlines the extent of additional work needed to produce results suitable for deployment, e.g., testing for verification, combining, correlating and interpreting additional research, etc. In the present case, putting a dollar amount on the potential savings that could be realized based on the work achieved so far would amount to a careful investigation of the business plan for fusion of probe data with data collected from dedicated infrastructure. This is out of the scope of Mobile Millennium, as Mobile Millennium was a technology project, with a first goal of demonstrating the technological feasibility of constructing a probe based traffic monitoring system.

There are numerous other benefits from the construction of this system, which include all the research work and its applicability to traffic information systems, operations and planning (see other sections in this chapter), as well as technological breakthroughs that are now in the public domain (from the publications created from this work), and which served as the basis for several software client apps developed by the private sector for routing (which directly benefit California or traffic at large, as these are used by commuters and travelers for their trip planning, and thus are benefiting the traveling public).

28.7. Recommendations for the future

28.7.1. Traffic app developments / industry

It is very likely that in the future, public agencies (state DOTs or the federal DOT, or other) will feel the need for developing traffic applications, or smartphone based applications, to support their internal needs, or the needs of the customers they are serving. For these agencies, the experience acquired during the Mobile Millennium is of considerable value, as it teaches the challenges that such a process (creation of a new app and corresponding system to support it) creates. Among the important features of the development cycle of such a product are the following items:

- *User engagement, recruitment campaign, incentivization.* Because these applications are specifically dedicated to a portion of the public, it is crucial to have a sound strategy for user enrollment (which can be achieved by the proper advertisement, information campaigns, or similar endeavors of the same nature). These are key, and the government can help greatly by providing the appropriate incentivization or information

support.

- *Social gaming.* One of the major findings from social networks over the last two years is the importance of social gaming in the sustaining of user engagement. Mobile Millennium revealed the possibility of having spikes of downloads when the proper advertisement or information campaign were performed. It also showed that sustaining a basis of active users is difficult. In the mean time, numerous other applications (not necessarily linked to traffic) showed that having a game associated with it (for example in which participants compete) is a very effective mechanism to keep users engaged. Other approaches are possible as well, which are key in ensuring a sustained user basis.
- *Execution.* The quality of the final product is also essential in ensuring the success of such a product. Because of the significant number of applications available today, few companies in the private sector have the ability of creating products comparable with the standards imposed by Google Maps (which has integrated this software with all of its other applications available through Google). Thus, the success of similar applications in the future is also contingent on the ability to either provide the same level of integration, or to occupy a different niche in the market, which is not in direct competition with these products.
- *Policy.* The driver distraction laws which were created in the middle of the project showed how policy is important for regulation and adoption of applications like Mobile Millennium. In the present case, the project had to interrupt the download of the traffic applications in order to comply to the distracted driving laws. While this happened after most of the data had been collected, and thus was not a fatal issue to the project, it could have had a dramatic impact overall on the goals assigned initially to the team. In a more general context, it shows how some products (industry or not) can be affected by policy, and is a very important lesson to be taken into account in the development of traffic applications.
- *Car / phone fusion.* The next decade will be the decade of the fusion between the car infrastructure and the phone infrastructure (and thus fusion with web based services). This project was not specifically oriented towards this goal (as it was clear from the start of Mobile Millennium that this project was focused on the development of a phone based application). However, with the technology trends of today, it is clear that in the next decade, the services currently available for the phone will become available through the car infrastructure (for example through the various displays available in the car, or using data available from the car through the IBD2 port). Recommendations for the future in this area include the necessity to consider car integration in the development of new products, to make them “car compatible”.
- *Cloud based services.* It was not anticipated at the beginning of the project that cloud based services would have such an impact on the field. In particular, in 2010 and 2011, it became obvious that the issue of scaling up systems could be solved by the proper cloud environment using services such as the ones provided by Amazon, for example. It is clear that in the future, for private sector entities in the business of developing apps like Mobile Millennium, or even for public agencies in need of scaling up systems, the cloud provides an attractive solution at low cost.

28.7.2. Data fusion, data hybridization, data procurement

From a public agency's perspective, the most valuable achievement performed during Mobile Millennium is probably the demonstration that probe based traffic information generation is possible, and that data fusion (i.e. the process of blending probe data with other types of data using traffic flow models) is an alternative to dedicated infrastructure based solutions for traffic monitoring. While Mobile Millennium only scratched the surface of what will probably be the most significant revolution in traffic monitoring systems in the coming decade, the data collected, and the studies conducted provide the first steps towards this goal, in a field just starting to explode.

Practices in the state DOTs and US DOT will change (and in some cases have already started changing). The technical feasibility of blending data, and potentially to partially replace some of the dedicated infrastructure by probe data, opens the door for a new market, and new products, and also raises numerous questions. From a government perspective, the following questions are now open, and need to be answered in the coming years:

- *Procurement.* How to perform procurement? The procedures to be used by the government to acquire third party data are undefined, and need to be created.
- *Quality and standards.* In order to procure data, one needs proper quality metrics and standards to be defined, in order to rate the data.
- *Market price.* Because of the fragmentation of the market, how to establish a price point at which the data can be exchanged? These are important next steps currently being investigated by some of the state DOTs (including California), and these efforts are essential to ensure the transition of the DOTs to the next phase of their traffic information infrastructure (and more generally, operations infrastructure).

28.7.3. Beyond highway and arterial traffic

Mobile Millennium opened the door of the mobile internet to public agencies and government. It was the first pilot of this kind operated at a large scale with mobile phones under a tripartite partnership comprised of government, Academia and Industry. It focused exclusively on ground traffic (highways and arterials), because of the objectives of the industry partners associated with the effort. Of course, this was only the beginning of a that has since considerably expanded, and is a fertile ground for the government to improve its transportation system. In particular, the following topics are of interest and come with great promises to have significant impacts on mobility at large.

- *Multimodal transport.* The work performed so far has demonstrated the ability to change human behavior, by giving information through cell phone platforms, which enabled commuters to make better decisions and potentially shift mode of transportation. This field is still open, and at the present time no single solution has emerged to enable multimodal transport at a global scale.
- *Social networks.* Social network applications enable behavior change (in particular with respect to commute patterns, or mobility patterns). At the present time,

they have been underexploited for traffic, and are very promising in the future.

- *Parking.* Smart parking applications are already well underway, and will benefit in the future from being connected with other traffic applications, or other transportation applications. Parking information today is becoming available through the deployment of dedicated infrastructure that has the potential of being profitable by the nature of the activity (parking can be charged for), and thus will provide in the coming years a new source of traffic data (mainly related to demand), which will have the potential to provide further improvement to commuter's needs.
- *Integrated transportation services.* From a consumer point of view, one of the roadblocks preventing people from shifting transportation modes from car to alternate modes is the lack of availability of planning tools at global scales. Endeavors such as Google transit provide good first steps in this direction, as the level of integration between the very fragmented underlying networks of transportation is progressively increasing.
- *Inter agency / private sector integration.* Transportation at large is also very fragmented, as many of the networks, fleets, services, roads, and highways are operated and owned by different entities. The progressive integration of these will provide numerous benefits for passengers. The mobile internet will be of great help in achieving this unification (which obviously needs to go beyond just information).

Mobile Millennium was a first step of transportation towards the mobile internet. While it has become mature, with the emergence of numerous traffic applications that are smartphone based, it opened the door for numerous other breakthroughs at the levels of operators, apps, industry and government. The decade of the mobile internet has just begun, and it will continue to impact transportation significantly.

Part VIII
Bibliography

References

[1] 511. http://www.511.org/.

[2] Amazon ec2 documentation. http://aws.amazon.com/ec2.

[3] Amazon's mechanical turk. https://www.mturk.com/mturk/welcome.

[4] Cabspotting. http://www.cabspotting.org.

[5] California Department of Transportation. http://www.dot.ca.gov/.

[6] CCTV information. http://www.cctv-information.co.uk/i/An_Introduction_to_ANPR.

[7] Citris cluster documentation. https://sites.google.com/a/lbl.gov/ucb-citris-cluster-user-guide.

[8] Data breaches. http://www.privacyrights.org/ar/ChronDataBreaches.htm.

[9] Fcc government website. http://www.fcc.gov/Bureaus/Wireless/.

[10] Google maps. http://maps.google.com/.

[11] Inrix. http://www.inrix.com/.

[12] Mesos project. http://www.mesosproject.org.

[13] Mesos schema. https://github.com/mesos/mesos/wiki/Mesos-Architecture.

[14] Mobile millenium. http://traffic.berkeley.edu/.

[15] Mpi run documentation. http://linux.die.net/man/1/mpirun.

[16] NAVTEQ Inc. http://www.navteq.com.

[17] Nersc cluster documentation. http://www.nersc.gov/nusers/systems/carver.

[18] Next generation simulation. http://ngsim-community.org/.

[19] Ngsim. http://www.ngsim.fhwa.dot.gov/.

[20] Nokia Inc. http://www.nokia.com.

[21] Paramics. http://www.paramics-online.com.

[22] PeMS. http://pems.eecs.berkeley.edu/.

[23] PostGIS extensions for PostgreSQL. http://postgis.refractions.net/.

[24] PostgreSQL database. http://www.postgresql.org/.

[25] Scala programming language. http://scala-lang.org.

[26] Sensys Networks. http://www.sensysnetworks.com.

[27] Spark documentation. https://github.com/mesos/spark/wiki.

[28] Traffic.com. http://www.traffic.com/.

[29] United States Department of Transportation. http://www.dot.gov.

[30] *Highway Capacity Manual*. TRB, National Research Council, Washington, D.C., 2000.

[31] The darpa network challenge, 2009. https://networkchallenge.darpa.mil/.

[32] The netflix prize, 2009. http://www.netflixprize.com/.

[33] Facebook, 2011. www.facebook.com.

[34] Flickr-photo sharing, 2011. www.flickr.com.

[35] Foursquare, 2011. www.foursquare.com.

[36] Google privacy practices, 2011. http://www.google.com/mobile/privacy.html.

[37] Linkedin - world's largest professional network, 2011. http://www.linkedin.com/.

[38] Loopt, 2011. http://www.loopt.com.

[39] Merriam–webster, 2011. http://www.merriam-webster.com/.

[40] The nokia sportstracker program, 2011. http://www.sports-tracker.com/.

[41] Oakland crime spotting, 2011. http://oakland.crimespotting.org/.

[42] Techcrunch, 2011. http://techcrunch.com/2011/05/20/ infographic-a-look-at-the- size-and- shape-of-the-geosocial-universe-in-2011/.

[43] Wikipedia, the free encyclopedia that anyone can edit, 2011. www.wikipedia.org.

[44] Yahoo! answers, 2011. http://answers.yahoo.com/.

[45] Zillow real estate, homes for sale, home prices and values, 2011. http://www.zillow.com.

[46] O. M. Aamo, A. Smyshlyaev, P. Rouchon, and P. Martin. Boundary control of a nonlinear Stefan problem. In *Proceedings of the IEEE Conference on Decision and Control*, pages 1309–1314, Maui, HI, Dec. 2003.

[47] P. Abbeel and A. Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1. ACM, 2004.

[48] I. Abraham, A. Fiat, A.V. Goldberg, and R.F. Werneck. Highway dimension, shortest paths, and provably efficient algorithms. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA10), Society for Industrial and Applied Mathematics*, 2010.

[49] L. Alvarez-Icaza, L. Munoz, X. Sun, and R. Horowitz. Adaptive observer for traffic density estimation. In *Proceedings of the American Control Conference*, pages 2705–2710, Boston, MA, June 2004.

[50] S. Amin, A. Cardenas, and S. Sastry. Safe and Secure Networked Control Systems under Denial-of-Service Attacks. Number 5469 in Lecture Notes in Computer Science, pages 31–45. Springer, San Francisco, CA, 2009.

[51] B. Anderson and J. Moore. *Optimal filtering*. Prentice-Hall, inc, Englewood Cliffs, N.J., 1979.

[52] R. Ansorge. What does the entropy condition mean in traffic flow theory? *Transportation Research*, 24B(2):133–143, 1990.

[53] O. A.Oleinik. On discontinuous solutions of nonlinear differential equations. *Uspekhi Mat. Nauk.*, 12:3–73, 1957. English translation: *American Mathematical Society*, Ser. 2 No. 26 pp. 95–172, 1963.

[54] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on signal processing*, 50(2), 2002.

[55] D. Assaf and B. Levikson. Closure of phase type distributions under operations arising in reliability theory. *The Annals of Probability*, 10(1):265–269, 1982.

[56] V. Astarita. A continuous time link model for dynamic network loading based on travel time function. In *th International Symposium on Transportation and Traffic Theory*, pages 79–102, Lyon, France, 1996.

[57] J.-P. Aubin. *Viability Theory*. Systems and Control: Foundations and Applications. Birkhäuser, Boston, MA, 1991.

[58] J.-P. Aubin. Viability kernels and capture basins of sets under differential inclusions. *SIAM J. Control Optim.*, 40:853–881, 2001.

[59] J.-P. Aubin, A. M. Bayen, and P. Saint-Pierre. Dirichlet problems for some Hamilton-Jacobi equations with inequality constraints. *In press: SIAM Journal on Control and Optimization*, 47(5):2348–2380, 2008.

[60] J.-P. Aubin and A. Cellina. *Differential inclusions*. Springer-Verlag, New York, NY, 1984.

[61] J.-P. Aubin and H. Frankowska. *Set Valued Analysis*. Birkhäuser, Boston, MA, 1990.

[62] A. Aw and M. Rascle. Resurrection of ’second order’ models of traffic flow. *SIAM Journal on Applied Mathematics*, 60(3):916–938, 2000. www.scopus.com.

[63] Q. Jacobson M. Gruteser A. Bayen J-.C. Herrera R. Herring D. Work M. Annavaram J. Ban B. Hoh, T.Iwuchukwu. Enhancing privacy and accuracy in probe vehicle based traffic monitoring via virtual trip lines. 2011.

IEEE Transactions on Mobile Computing.

[64] X. Ban, R. Herring, P. Hao, and A. Bayen. Delay pattern estimation for signalized intersections using sampled travel times. In *Proceedings of the 88th Annual Meeting of the Transportation Research Board*, Washington, D.C., January 2009.

[65] X. Ban, R. Herring, J. Margulici, and A. Bayen. Optimal sensor placement for freeway travel time estimation. *Proceedings of the 18th International Symposium on Transportation and Traffic Theory*, July 2009.

[66] X. Ban, Y. Li, A. Skabardonis, and J. Margulici. Performance evaluation of travel time methods for real time traffic applications. In *Proceedings of the 11th World Congress on Transport Research (CD-ROM)*, 2007.

[67] Y. Bar-Shalom, X.R. Li, and T. Kirubarajan. *Estimation with applications to tracking and navigation*. Wiley New York, 2001.

[68] M. Bardi and I. Capuzzo-Dolcetta. *Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman equations*. Birkhäuser, Boston, MA, 1997.

[69] C. Bardos, A. Y. Leroux, and J. C. Nedelec. First order quasilinear equations with boundary conditions. *Communications in partial differential equations*, 4(9):1017–1034, 1979.

[70] E. N. Barron and R. Jensen. Semicontinuous viscosity solutions for Hamilton-Jacobi equations with convex Hamiltonians. *Comm. Partial Differential Equations*, 15:1713–1742, 1990.

[71] A. M. Bayen, R. L. Raffard, and C. Tomlin. Network congestion alleviation using adjoint hybrid control: Application to highways. Number 1790 in Lecture Notes in Computer Science, pages 95–110. Springer Verlag, 2004.

[72] R.E. Bellman and R.E. Kalaba. *Numerical Inversion of the Laplace Transform*. American Elsevier Publishing Company, 1966.

[73] D.P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 2005.

[74] D.P. Bertsekas and J.N. Tsitsiklis. *Neuro-dynamic Programming*. Athena Scientific, 1996.

[75] S. Blandin, G. Bretti, A. Cutolo, and B. Piccoli. Numerical simulations of traffic data via fluid dynamic approach. *Applied Mathematics and Computation*, 210(2):441–454, 2009.

[76] S. Blandin, D. Work, P. Goatin, B. Piccoli, and A. Bayen. A general phase transition model for vehicular traffic. *Preprint*, 2009.

[77] H.W. Block and T.H. Savits. The ifra closure problem. *The Annals of Probability*, 4(6):1030–1032, 1976.

[78] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[79] M. Brand. Coupled hidden Markov models for modeling interacting processes. Technical report, The Media Lab, Massachusetts Institute of Technology, Boston, MA, 1997.

[80] A. Bressan. *Hyperbolic Systems of Conservation Laws: The One-dimensional Cauchy Problem*. Oxford University Press, Oxford, UK, 2000.

[81] A. Bressan, G. Crasta, and B. Piccoli. *Well-posedness of the Cauchy problem for nxn systems of conservation laws*. American Mathematical Society, 2000.

[82] Yingyi Bu, Bill Howe, Magdalena Balazinska, and Michael D. Ernst. Haloop: Efficient iterative data processing on large clusters. In *VLDB*, 2010.

[83] G. Burgers, P. Jan van Leeuwen, and G. Evensen. Analysis scheme in the ensemble Kalman filter. *Monthly Weather Review*, 126(6):1719–1724, 1998.

[84] W. Burghout. Hybrid microscopic-mesoscopic traffic simulation. *Royal Institute of Technology Doctoral Dissertation*, 2004.

[85] B. Starr H. Schuster C. Perry, J. McIntyre and R. Habib. Cell phone tracking helped find al-zarqawi u.s. military: Terrorist alive briefly after airstrike. *CNN*, 2006. CNN Article.

[86] P. Cardaliaguet, M. Quincampoix, and P. Saint-Pierre. Optimal times for constrained nonlinear control problems without local controllability. *Applied Mathematics and Optimization*, 36:21–42, 1997.

[87] P. Cardaliaguet, M. Quincampoix, and P. Saint-Pierre. Set-valued numerical analysis for optimal control and differential games. In M. Bardi, T.E.S. Raghavan, and T. Parthasarathy, editors, *Stochastic and Differential Games: Theory and Numerical Methods*, Annals of the International Society of Dynamic Games. Birkhäuser, Boston, MA, 1999.

[88] M. Cassidy and J Windover. Methodology for assessing dynamics of freeway traffic flow. *Transportation Research Record*, 1484:73–79, 1995.

[89] J. Del Castillo and F. Benitez. On the functional form of the speed-density relationship i: General theory. *Transportation Research Part B*, 29(5):373–389, 1995.

[90] J. Del Castillo, P. Pintado, and F. Benitez. The reaction time of drivers and the stability of traffic flow. *Transportation research. Part B*, 28(1):35–60, 1994.

[91] C. Chalons and P. Goatin. Godunov scheme and sampling technique for computing phase transitions in traffic flow modeling. *Interfaces and Free Boundaries*, 10(2):195–219, 2008.

[92] E. Charniak. *Statistical Language Learning*. MIT Press, Cambridge, Massachusetts, 1993.

[93] C. Chen, K. Petty, A. Skabardonis, and P. Varaiya. Freeway performance measurement system: mining loop detector data. In *th Annual Meeting of the Transportation Research Board*, Washington, D.C., January 2001.

[94] P. Cheng. Arstechnica. In *Arterial, crowdsourced traffic info comes to Google Maps*, 2009.

[95] S. Y. Cheung, S. C. Ergen, and P. Varaiya. Traffic surveillance with wireless magnetic sensors. In *Proceedings of the 12th ITS World Congress*, 2005.

[96] T. Choe, A. Skabardonis, and P. Varaiya. Freeway performance measurement system: an operational analysis tool. *Transportation Research Record*, 1811(-1):67–75, 2002.

[97] A. J. Chorin. Numerical solution of Boltzmann's equation. *Communications on Pure and Applied Mathematics*, 25(2):171–186, 1972.

[98] A. J. Chorin and X. Tu. Implicit sampling for particle filters. *Proceedings of the National Academy of Sciences*, 106(41):17249, 2009.

[99] Mosharaf Chowdhury, Matei Zaharia, Justin Ma, Michael I. Jordan, and Ion Stoica. Managing data transfers in computer clusters with Orchestra. In *SIGCOMM*, 2011.

[100] P. D. Christofides. *Nonlinear and robust control of PDE systems: methods and applications to transport-reaction processes*. Birkhauser, 2001.

[101] Cheng-Tao Chu, Sang Kyun Kim, Yi-An Lin, YuanYuan Yu, Gary Bradski, Andrew Y. Ng, and Kunle Olukotun. Map-reduce for machine learning on multicore. In *NIPS*, 2007.

[102] L. Chu, H. Liu, and W. Recker. Using microscopic simulation to evaluate potential intelligent transportation system strategies under nonrecurrent congestion. *Transportation Research Record*, 1886(-1):76–84, 2004.

[103] P. H. Clarke, Y. S. Ledyaev, R. J. Stern, and P. R. Wolenski. Qualitative properties of trajectories of control systems: a survey. *Journal of dynamical and control systems*, 1(1):1–48, 1995.

[104] C. Claudel and A. Bayen. Lax-Hopf based incorporation of internal boundary conditions into Hamilton-Jacobi equation. Part I: theory. *IEEE Transactions on Automatic Control*, 55(5):1142–1157, 2010. doi:10.1109/TAC.2010.2041976.

[105] C. Claudel and A. Bayen. Lax-Hopf based incorporation of internal boundary conditions into Hamilton-Jacobi equation. Part II: Computational methods. *IEEE Transactions on Automatic Control*, 55(5):1158–1174, 2010. doi:10.1109/TAC.2010.2045439.

[106] C. Claudel, M. Nahoum, and A. Bayen. Minimal error certificates for detection of faulty sensors using convex optimization. In *Proceedings of the 47th Annual Allerton Conference on Communication, Control, and Computing*, Allerton, IL, Sep. 2009.

[107] C. G. Claudel and A. M. Bayen. Solutions to switched Hamilton-Jacobi equations and conservation laws using hybrid components. In M. Egerstedt and B. Mishra, editors, *Hybrid Systems: Computation and Control*, number 4981 in Lecture Notes in Computer Science, pages 101–115. Springer Verlag, Saint Louis, MO, April 2008.

[108] C. G. Claudel and A. M. Bayen. Convex formulations of data assimilation problems for a class of Hamilton-Jacobi equations. *Submitted to SIAM Journal on Control and Optimization*, 2009.

[109] G. M. Coclite, M. Garavello, and B. Piccoli. Traffic flow on a road network. *SIAM Journal on Mathematical Analysis*, 36(6):1862–1886, 2005.

[110] B. Coifman. Estimating travel times and vehicle trajectories on freeways using dual loop detectors. *Transportation Research Part A*, 36(4):351–364, 2002.

[111] R. Colombo. On a 2×2 hyperbolic traffic flow model. *Math. Comput. Modelling*, 35(5-6):683–688, 2002.

[112] R. Colombo. Hyperbolic phase transitions in traffic flow. *SIAM J. Appl. Math.*, 63(2):708–721, 2003.

[113] R. Colombo, P. Goatin, and F. Priuli. Global well-posedness of traffic flow models with phase transitions. *Nonlinear Analysis*, 66(11):2413–2426, 2007.

[114] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. The MIT Press, 2001.

[115] J.-M. Coron, B. d'Andrea Novel, and G. Bastin. A strict Lyapunov function for boundary control of hyperbolic systems of conservation laws. In *Proceedings of the American Control Conference*, pages 3319–3323, Paradise Island, Bahamas, Dec. 2004.

[116] M. G. Crandall, L. C. Evans, and P.-L. Lions. Some properties of viscosity solutions of Hamilton-Jacobi equations. *Transactions of the American Mathematical Society*, 282(2):487–502, 1984.

[117] M. G. Crandall and P.-L. Lions. Viscosity solutions of Hamilton-Jacobi equations. *Transactions of the American Mathematical Society*, 277(1):1–42, 1983.

[118] M. Cremer and M. Papageorgiou. Parameter identification for a traffic flow model. *Automatica*, 17(6):837–843, 1981.

[119] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, Cambridge, UK, 2000.

[120] C. M. Crowe. Data reconciliation-progress and challenges. *Journal of Process Control*, 6(2):89–98, 1996.

[121] R. C.Smith and M. A. Demetriou. *Research Directions in Distributed Parameter Systems*. SIAM, Philadelphia, PA, 2000.

[122] C. M. Dafermos. Polygonal approximations of solutions of the initial value problem for a conservation law. *Journal of Mathematical Analysis and Applications*, 38(1):33–41, 1972.

[123] C. Daganzo. Requiem for second-order fluid approximations of traffic flow. *Transportation Research Part B*, 29(4):277–286, 1995.

[124] C. Daganzo, M. Cassidy, and R. Bertini. Possible explanations of phase transitions in highway traffic. *Transportation Research Part A*, 33(5):365–379, 1999.

[125] C. Daganzo, W. Lin, and J. Castillo. A simple physical principle for the simulation of freeways with special lanes and priority vehicles. *Transportation Research Part B*, 31(2):103 – 125, 1997.

[126] C. F. Daganzo. The cell transmission model: a dynamic representation of highway traffic consistent with the hydrodynamic theory. *Transportation Research*, 28B(4):269–287, 1994.

[127] C. F. Daganzo. The cell transmission model, part II: network traffic. *Transportation Research Part B*, 29(2):79–93, 1995.

[128] C. F. Daganzo. A variational formulation of kinematic waves: basic theory and complex boundary conditions. *Transporation Research B*, 39B(2):187–196, 2005.

[129] C. F. Daganzo. On the variational theory of traffic flow: well-posedness, duality and applications. *Networks and heterogeneous media*, 1:601–619, 2006.

[130] C. de Fabritiis, R. Ragona, and G. Valenti. Traffic estimation and prediction based on real time floating car data. In *Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on*, pages 197–203, 2008.

[131] B.C. Dean. Algorithms for minimum-cost paths in time-dependent networks with waiting policies. *Networks*, 44:41–46, 2004.

[132] E.W. Dijkstra. A note on two problems on connection with graphs. *Numerishe Mathematik*, 1:269–271, 1959.

[133] S. Dreyfus. An appraisal of some shortest-path algorithms. *Operations Research*, 17:395–412, 1969.

[134] John Duchi, Alekh Agarwal, and Martin Wainwright. Distributed dual averaging in networks. In *NIPS*. 2010.

[135] B. Efron and G. Gong. A leisurely look at the bootstrap, the jackknife, and cross-validation. *American Statistician*, pages 36–48, 1983.

[136] Jaliya Ekanayake, Hui Li, Bingjing Zhang, Thilina Gunarathne, Seung-Hee Bae, Judy Qiu, and Geoffrey Fox. Twister: a runtime for iterative mapreduce. In *HPDC '10*, 2010.

[137] H. Engl, K. Kunisch, and A. Neubauer. Convergence rates for tikhonov regularization of nonlinear ill-posed problems. *Inverse Problems*, 5(4):523–540, 1989.

[138] D. Estrin. Participatory sensing: Applications and architecture. 2010. Proceedings of the 8th International Conference on Mobile systems, applications and services (MobiSys'10).

[139] L. C. Evans. *Partial Differential Equations*. American Mathematical Society, Providence, RI, 1998.

[140] L. C. Evans and P. E. Souganidis. Differential games and representation formulas for solutions of Hamilton-Jacobi-Isaacs equations. *Indiana University Mathematics Journal*, 33(5):773–797, 1984.

[141] G. Evensen. Using the extended Kalman filter with a multilayer quasi-geostrophic ocean model. *Journal of Geophysical Research*, 97(C11):17905–17924, 1992.

[142] G. Evensen. The ensemble Kalman filter: theoretical formulation and practical implementation. *Ocean Dynamics*, 53(4):343–367, 2003.

[143] G. Evensen. *Data Assimilation: The Ensemble Kalman Filter*. Springer-Verlag, Berlin Heidelberg, 2007.

[144] M. Falcone and R. Ferretti. Semi-Lagrangian schemes for Hamilton-Jacobi equations, discrete representation formulae and Godunov methods. *Journal of computational physics*, 175(2):559–575, 2002.

[145] Y. Fan and Y. Nie. Optimal routing for maximizing the travel time reliability. *Networks and Spatial Economics*, 6(3-4):333–344, September 2006.

[146] Y.Y. Fan, R.E. Kalaba, and J.E. Moore. Arriving on time. *Journal of Optimization Theory and Applications*, 127(3):497–513, 2005.

[147] M. Fliess, J. Levine, P. Martin, and P. Rouchon. Flatness and defect of non-linear systems: introductory theory and examples. *International Journal of Control*, 61(6):1327–1361, 1995.

[148] M. Fliess, P. Martin, N. Petit, and P. Rouchon. Active signal restoration for the telegraph equation. In *IEEE Conference on Decision and Control*, volume 2, pages 1107–1111, 1999.

[149] P. Le Floch. Explicit formula for scalar non-linear conservation laws with boudary condition. *Math. Meth. Appl. Sci.*, 10:265–287, 1988.

[150] H. Frank. Shortest paths in probabilistic graphs. *Operations Research*, 17:583–599, 1969.

[151] H. Frankowska. Solutions to initial-boundary value problem for scalar conservation laws. In preparation.

[152] H. Frankowska. Lower semicontinuous solutions of Hamilton-Jacobi-Bellman equations. *SIAM Journal of Control and Optimization*, 31(1):257–272, 1993.

[153] L. Fu and L. Rilett. Expected shortest paths in dynamic and stochastic traffic networks. *Transportation Research Part B*, 32(7):499–516, 1998.

[154] C. Furtlehner, J. Lasgouttes, and A. de la Fortelle. A belief propagation approach to traffic prediction using probe vehicles. In *Proceedings of the IEEE 10th International Conference on Intelligent Transportation Systems*, pages 1022–1027, 2007.

[155] M. Garavello and B. Piccoli. *Traffic Flow on Networks*. American Institute of Mathematical Sciences on Applied Math. Springfield, MO, 2006.

[156] M. Garavello and B. Piccoli. On fluido-dynamic models for urban traffic. *Netw. Heterog. Media*, 4(1):107–126, 2009.

[157] H. Gault and I. Taylor. The use of output from vehicle detectors to access delay in computer-controlled area traffic control systems. Technical Report Research Report No. 31, Transportation Operation Research Group, University of Newcastle upon Tyne, United Kingdom, 1977.

[158] D. Gazis and C. Liu. Kalman filtering estimation of traffic counts for two network links in tandem. *Transportation Research Part B: Methodological*, 37(8):737 – 745, 2003.

[159] R. Geisberger, P. Sanders, D. Schultes, and D. Delling. Contraction hierarchies: Faster and simpler hierarchical routing in road networks. In *Proceedings of the 7th Workshop on Experimental Algorithms (WEA'08)*, pages 319–333. Springer, 2008.

[160] N. Geroliminis and C.F. Daganzo. Macroscopic modeling of traffic in cities. In *Proceedings of the 86th Annual Meeting of the Transportation Research Board*, Washington, D.C., January 2007.

[161] N. Geroliminis and A. Skabardonis. Prediction of arrival profiles and queue lengths along signalized arterials by using a Markov decision process. *Transportation Research Record*, 1934(1):116–124, May 2006.

[162] P. Gilbert, L. P. Cox, J. Jung, and D. Wetherall. *Toward trustworthy mobile sensing*. 2010.

[163] J. F. Gimpel. A theory of discrete patterns and their implementation in snobol4. *Commun. ACM*, 16:91–100, February 1973.

[164] J. Glimm. Solutions in the large for nonlinear hyperbolic systems of equations. *Comm. Pure Appl. Math.*, 18:697–715, 1965.

[165] P. Goatin. The Aw-Rascle vehicular traffic flow model with phase transitions. *Math. Comput. Modelling*, 44(3-4):287–303, 2006.

[166] S. Godunov. A difference method for the numerical calculation of discontinuous solutions of hydrodynamic equations. *Mathematics Sbornik*, 47(3):271–306, 1959.

[167] A. V. Goldberg, H. Kaplan, and R.F. Werneck. Better landmarks within reach. In *Workshop on Experimental Algorithms (WEA), Rome, Italy*, 2007.

[168] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi. Understanding individual human mobility patterns. *Nature*, 453:779–782, 2008.

[169] N.J. Gordon, D.J. Salmond, and A.F.M. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEE Proceedings F Radar and Signal Processing*, 140(2):107–113, 1993.

[170] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 1.21.

http://cvxr.com/cvx, July 2010.

[171] H. Greenberg. An analysis of traffic flow. *Oper. Res.*, 7(1):79–85, 1959.

[172] L. Greenemeier, E. Malykhina, P. McGougall, A. Ricadela, and M. K. McGee. The high cost of data loss. Mar 2006.

[173] B. Greenshields. A study of traffic capacity. *Proceedings of the Highway Research Board*, 14(1):448–477, 1935.

[174] B.D. Greenshields. A study of traffic capacity. *Highway Research Board*, 14:448–477, 1935.

[175] M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. 2003. ACM MobiSys.

[176] M. Gruteser and B. Hoh. On the anonymity of periodic location samples. 2005. Proceedings of the Second International Conference on Security in Pervasive Computing.

[177] R.W. Hall. The fastest path through a network with random time-dependent travel times. *Transportation Science*, 20(3):182, 1986.

[178] T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning*. Springer, corrected edition, July 2003.

[179] B. Hellinga, P. Izadpanah, H. Takada, and L. Fu. Decomposing travel times measured by probe-based traffic monitoring systems to individual road segments. *Transportation Research Part C: Emerging Technologies*, 16(6):768 – 782, 2008.

[180] J.-C. Herrera and A. Bayen. Traffic flow reconstruction using mobile sensors and loop detector data. In *TRB Annual Meeting* 87th, Washington D.C., Jan. 12-17 2008. Transportation Research Board.

[181] J.-C. Herrera, D. Work, R. Herring, J. Ban, Q. Jacobson, and A. Bayen. Evaluation of traffic data obtained via GPS-enabled mobile phones: the Mobile Century experiment. To appear, *Transportation Research Part C*, 2009, doi:10.1016/j.trc.2009.10.006.

[182] J.C. Herrera and A.M. Bayen. Incorporation of Lagrangian measurements in freeway traffic state estimation. *Transportation Research Part B: Methodological*, 44(4):460–481, May 2010.

[183] R. Herring, A. Hofleitner, P. Abbeel, and A. Bayen. Estimating arterial traffic conditions using sparse probe data. In *Proceedings of the 13th International IEEE Conference on Intelligent Transportation Systems*, Madeira, Portugal, September 2010.

[184] R. Herring, A. Hofleitner, S. Amin, T. Abou Nasr, A. Abdel Khalek, P. Abbeel, and A. Bayen. Using mobile phones to forecast arterial traffic through statistical learning. In *Proceedings of the 89th Annual Meeting of the Transportation Research Board*, Washington D.C., 2010.

[185] M. Herty and A. Klar. Modeling, simulation, and optimization of traffic flow networks. *SIAM Journal on Scientific Computing*, 25(3):1066–1087, 2003.

[186] Benjamin Hindman, Andy Konwinski, Matei Zaharia, Ali Ghodsi, Anthony D. Joseph, Randy Katz, Scott Shenker, and Ion Stoica. Mesos: A platform for fine-grained resource sharing in the data center. Technical report, UC Berkeley, Technical Report UCB/EECS-2010-87, EECS Dept, September 2010.

[187] A. Hofleitner, R. Herring, and A. Bayen. Arterial travel time forecast with streaming data: a hybrid flow model - machine learning approach. *In preparation*, 2010.

[188] A. Hofleitner, R. Herring, and A. Bayen. A hydrodynamic theory based statistical model of arterial traffic. *Technical Report UC Berkeley*, August 2010.

[189] A. Hofleitner, R. Herring, T. Hunter, P. Abbeel, and A. Bayen. A learning and estimation approach towards arterial traffic monitoring. 2010.

[190] B. Hoh, M. Gruteser, R. Herring, J. Ban, D. Work, J.-C. Herrera, A. Bayen, M. Annavaram, and Q. Jacobson. Virtual trip lines for distributed privacy-preserving traffic monitoring. In *th International Conference on Mobile Systems, Applications, and Services*, pages 15–28, Breckenridge, CO, June 17-18 2008.

[191] B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady. *Enhancing security and privacy in traffic-monitoring systems*. 2006. IEEE Pervasive Computing.

[192] B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady. Preserving privacy in gps traces via uncertainty-aware path cloaking. October 2007.

[193] H. Holden and N. Risebro. *Front tracking for hyperbolic conservation laws*. Springer-Verlag, 2002.

[194] H. Holden and N.H. Risebro. A mathematical model of traffic flow on a network of unidirectional roads. *SIAM Journal on mathematical analysis*, 26(4):999 – 1017, July 1995.

[195] T. Hosaka. Facebook asks users to translate for free crowdsourcing aids company's aggressive worldwide expansion. 2008. MSNBC.

[196] P. L. Houtekamer and H. L. Mitchell. A sequential ensemble Kalman filter for atmospheric data assimilation. *Monthly Weather Review*, 129:123–137, 2001.

[197] J. Howe. The rise of crowdsourcing. June 2006.

Wired Magazine.

[198] T. Hunter, R. Herring, P. Abbeel, and A.M. Bayen. Path and travel time inference from GPS probe vehicle data. In *Proceedings of the Neural Information Processing Systems foundation (NIPS)*, Vancouver, Canada, 2009.

[199] T. Hunter, R. Herring, A. Hofleitner, A. Bayen, and P. Abbeel. Trajectory reconstruction of noisy GPS probe vehicles in arterial traffic. *In progress*, 2010.

[200] D. Jacquet, C. Canudas de Wit, and D. Koenig. Traffic control and monitoring with a macroscopic model in the presence of strong congestion waves. In *Proc. of the 44th IEEE Conference on Decision and Control, and European Control Conference*, pages 2164–2169, Sevilla, Spain, 2005.

[201] Denis Jacquet, Miroslav Krstic, and Carlos Canudas de Wit. Optimal control of scalar one-dimensional conservation laws. In *Proc. of the 25th American Control Conference*, pages 5213–5218, Minneapolis, MN, 2006.

[202] Z. Jia, C. Chen, B. Coifman, and P. Varaiya. The PeMS algorithms for accurate, real-time estimates of g-factors and speeds from single-loop detectors. In *Intelligent Transportation Systems, 2001. Proceedings. 2001 IEEE*, pages 536–541, 2001.

[203] Z. Jia, C. Chen, B. Coifman, and P. Varaiya. The PeMS algorithms for accurate, real time estimates of *g*-factors and speeds from single loop detectors. In *IEEE Intelligent Transportation Systems Conference Proceedings*, pages 536–541, Oakland, CA, Aug. 2001.

[204] S. Julier and J. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, 2004.

[205] P. Kachroo, K. Ozbay, and A. Hobeika. Real-time travel time estimation using macroscopic traffic flowmodels. *Proceedings of IEEE Intelligent Transportation Systems*, pages 132–137, 2001.

[206] J. Kaipio and E. Somersalo. *Statistical and Computational Inverse Problems*. Springer, New York, NY, 2005.

[207] J. Kaipio and E. Somersalo. Statistical inverse problems: Discretization, model reduction and inverse crimes. *Journal of Computational and Applied Mathematics*, 198(2):493–504, 2007.

[208] R.E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME Journal of Basic Engineering*, 82:35–45, 1960.

[209] Y. Kamarianakis and P. Prastacos. Space-time modeling of traffic flow. *ERSA Conference Papers*, June 2006.

[210] C. Y. Kao, S. Osher, and J. Qian. Lax-Friedrichs sweeping scheme for static Hamilton-Jacobi equations. *Journal of Computational Physics*, 196(1):367–391, 2004.

[211] B. Kerner. Experimental features of self-organization in traffic flow. *Phys. Rev. Lett.*, 81(17):3797–3800, 1998.

[212] B. Kerner. Phase transitions in traffic flow. *Traffic and granular flow*, pages 253–283, 2000.

[213] M. Krstic and A. Smyshlyaev. Backstepping boundary control for first-order hyperbolic PDEs and application to systems with actuator and sensor delays. *Systems & Control Letters*, 57(9):750–758, 2008.

[214] J. Krumm. Inference attacks on location tracks. In *Proceedings of the Fifth International Conference on Pervasive Computing*, Toronto, Ontario, Canada, May 2007.

[215] S. Kruzhkov. First order quasilinear equations in several space variables. *Sb. Math.*, 10(2):217–243, 1970.

[216] J. Kwon, C. Chen, and P. Varaiya. Statistical methods for detecting spatial configuration errors in traffic surveillance sensors. *Transportation Research Record: Journal of the Transportation Research Board*, 1870(-1):124–132, 2004.

[217] K. Kwong, R. Kavaler, R. Rajagopal, and P. Varaiya. Arterial travel time estimation based on vehicle re-identification using wireless magnetic sensors. *Transportation Research Part C: Emerging Technologies*, 17(6):586–606, December 2009.

[218] G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M. Jordan. Learning the kernel matrix with semidefinite programming. *J. Mach. Learn. Res.*, 5:27–72, 2004.

[219] J.P. Lebacque. Intersection modeling, application to macroscopic network traffic flow models and traffic management. In *Traffic and Granular Flow 2003*, pages 261–278. Springer Berlin Heidelberg, 2005.

[220] J.P. Lebacque. The Godunov scheme and what it means for first order macroscopic traffic flow models. *Proceedings of the 13th ISTTT, Ed. J.B. Lesort,*, pages 647–677, Lyon, 1996.

[221] L. Leclercq. Bounded acceleration close to fixed and moving bottlenecks. *Transportation research. Part B: methodological*, 41(3):309–319, 2007.

[222] P. L'Ecuyer. Stochastic simulation in java, http://www.iro.umontreal.ca/ simardr/ssj/indexe.html, 2008.

[223] R. Leveque. *Finite volume methods for hyperbolic problems*. Cambridge University Press, 1992.

[224] R.J. LeVeque. *Numerical Methods for Conservation Laws*. Birkhäuser Verlag, Basel, Switzerland, 1992.

[225] J. M. Lewis, S. Lakshmivarahan, and S. Dhall. *Dynamic Data Assimilation: A Least Squares Approach*. Cambridge University Press, Cambridge, UK, 2006.

[226] M. Lighthill and G. Whitham. On kinematic waves. II. A theory of traffic flow on long crowded roads. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 229(1178):317–345, 1955.

[227] W. Lin and D. Ahanotu. Validating the basic cell transmission model on a single freeway link. Technical report, Institute of Transportation Studies, University of California, Berkeley, 1995.

[228] J. Van Lint and H. Van Zuylen. Monitoring and predicting freeway travel time reliability: Using width and skew of day-to-day travel time distribution. *Transportation Research Record*, 1917:54–62, 2005.

[229] J. Lite. Obama's cell phone hacked, privacy issues murky. http://www.scientificamerican.com/blog/post.cfm?id=obamas-cell-phone-hacked-privacy-is-2008-11-21.

[230] X. Litrico. Robust flow control of single input multiple outputs regulated rivers. *Journal of Irrigation and Drainage Engineering*, 127(5):281–286, 2001.

[231] X. Litrico and V. Fromion. Boundary control of linearized saint-venant equations oscillating modes. In *Proceedings of the American Control Conference*, pages 2131–2136, Paradise Island, Bahamas, Dec. 2004.

[232] H. Liu and W. Ma. A virtual probe approach for time-dependent arterial travel time estimation. *Presented at the 87th Annual Conference on Transportation Research Board, and Submitted for publication*, 2008.

[233] R.P. Loui. Optimal paths in graphs with stochastic or multidimensional weights. *Communications of the ACM.*, 26(9):670–676, 1983.

[234] Yucheng Low, Joseph Gonzalez, Aapo Kyrola, Danny Bickson, Carlos Guestrin, and Joseph M. Hellerstein. Graphlab: A new parallel framework for machine learning. In *UAI*, 2010.

[235] Grzegorz Malewicz, Matthew H. Austern, Aart J.C Bik, James C. Dehnert, Ilan Horn, Naty Leiser, and Grzegorz Czajkowski. Pregel: A system for large-scale graph processing. In *SIGMOD*, 2010.

[236] Highway Capacity Manual. Special report 209. *Transportation Research Board, Washington, DC*, 1985.

[237] Ryan T. McDonald, Keith Hall, and Gideon Mann. Distributed training strategies for the structured perceptron. In *Conference of the North American Chapter of the Association of Computation Linguistics*, pages 456–464, 2010.

[238] Samy Merzgui. Parallelization of sensor data processing and learning algorithms in the cloud. Master's thesis, Swiss Federal Institute of Technology in Lausanne (EPFL), School of Computer & Communication Sciences, Switzerland, March 2011. In collaboration with the California Center for Innovative Transportation (CCIT) at UC Berkeley.

[239] L. Mihaylova and R. Boel. A particle filter for freeway traffic estimation. In *Proc. of the 43rd IEEE Conference on Decision and Control*, volume 2, pages 2106–2111, 2004.

[240] L. Mihaylova, R. Boel, and A. Hegyi. Freeway traffic estimation within recursive bayesian framework. *Automatica*, 43(2):290–300, 2007.

[241] E.D. Miller-Hooks and H.S. Mahmassani. Least expected time paths in stochastic, time-varying transportation networks. *Transportation Science*, 34(2):198–215, 2000.

[242] L. Mimbela, L. Klein, P. Kent, J. Hamrick, K. Luces, and S. Herrera. *Summary of Vehicle Detection and Surveillance Technologies used in Intelligent Transportation Systems*. Federal Highway Administration's (FHWA) Intelligent Transportation Systems Program Office, August 2007.

[243] X. Min, J. Hu, Q. Chen, T. Zhang, and Y. Zhang. Short-term traffic flow forecasting of urban network based on dynamic STARIMA model. In *Intelligent Transportation Systems, 2009. ITSC '09. 12th International IEEE Conference on*, pages 1–6, 2009.

[244] P. Misra and P. Enge. *Global Positioning System: Signals, Measurements and Performance*. Lincoln, MA: Ganga-Jamuna Press, 2 edition, 2006.

[245] H.L. Mitchell, P.L. Houtekamer, and G. Pellerin. Ensemble size, balance, and model–error representation in an ensemble Kalman filter. *Montly Weather Review*, 130:2791–2808, 2002.

[246] I. Mitchell. A toolbox of level set methods. http://www.cs.ubc.ca/ mitchell, 2005.

[247] I. Mitchell. Games of two identical vehicles. Technical Report SUDAAR 740, Department of Aeronautics and Astronautics, Stanford University, Stanford, CA, July 2001.

[248] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin. Computing reachable sets for continuous dynamic games using level set methods. *IEEE Transactions on Automatic Control*, 50(7):947–957, 2005.

[249] Nick Mitchell and Gary Sevitsky. Building memory-efficient Java applications: Practices and challenges. PLDI 2009 Tutorial.

[250] M.Krstic and A. Smyshlyaev. Adaptive boundary control for unstable parabolic PDEs. Part I: Lyapunov design. *IEEE Transactions on Automatic Control*, 53(7):1575, 2008.

[251] J. E. Moore, S. Cho, A. Basu, and D. B. Mezger. Use of Los Angeles freeway service patrol vehicles as probe vehicles. California Partners for Advanced Transit and Highways (PATH). Research Report: UCB-ITS-PRR-2001-05. Technical report, Feb. 2005.

[252] K. Moskowitz. Discussion of 'freeway level of service as influenced by volume and capacity characteristics' by D.R. Drew and C. J. Keese. *Highway Research Record*, 99:43–44, 1965.

[253] M. Mun, S. Reddy, K. Shilton, N. Yau, P. Boda, J. Burke, D. Estrin, M. Hansen, E. Howard, and R. West. Peir, the personal environmental impact report, as a platform for participatory sensing systems research. In *th Annual International Conference on Mobile Systems, Applications and Services*, 2009.

[254] L. Munoz, X. Sun, R. Horowitz, and L. Alvarez. Traffic density estimation with the cell transmission model. In *American Control Conference, 2003. Proceedings of the 2003*, volume 5, 2003.

[255] G. Newell. A simplified car-following theory: a lower order model. *Transportation Research Part B*, 36(3):195–205, 2002.

[256] G. F. Newell. A simplified theory of kinematic waves in highway traffic, Part (I), (II) and (III). *Transporation Research B*, 27B(4):281–313, 1993.

[257] A. Ng. Lecture notes. CS 229: Machine learning. *Stanford University*, 2003.

[258] Y. Nie and Y. Fan. Arriving-on-time problem. *Transportation Research Record*, pages 193–200, 2006.

[259] D. Nikovski, N. Nishiuma, Y. Goto, and H. Kumazawa. Univariate short-term prediction of road travel times. *IEEE Intelligent Transportation Systems, 2005. Proceedings*, pages 1074–1079, 2005.

[260] N.Petit. *Delay Systems. Flatness in Process Control and Control of some Wave Equations*. PhD thesis, Ecole des Mines de Paris, Paris, France, 2000.

[261] C. Oh. *Anonymous Vehicle Tracking for Real-Time Traffic Performance Measures*. PhD thesis, University of California, Irvine, Irvine, CA, 2003.

[262] C. Oh and S. Ritchie. Anonymous vehicle tracking for real-time traffic surveillance and performance on signalized arterials. In *Proceedings of the 82nd Annual Meeting of the Transportation Research Board (CD-ROM)*, 2003.

[263] O. Oleinik. Discontinuous solutions of non-linear differential equations. *Uspekhi Mat. Nauk*, 12(3):3–73, 1957.

[264] A. Orda and R. Rom. Shortest-path and minimum-delay algorithms in networks with time-dependent edge-length. *Journal of the ACM*, 37(3):607–625, July 1990.

[265] M. Papageorgiou. *Applications of Automatic Control Concepts to Traffic Flow Modeling and Control*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1983.

[266] M. Papageorgiou. Some remarks on macroscopic traffic flow modelling. *Transportation Research Part A*, 32(5):323–329, 1998.

[267] T. Park and S. Lee. A Bayesian approach for estimating link travel time on urban arterial road network. In *Computational Science and Its Applications - ICCSA 2004*, pages 1017–1025. Perugia, Italy, May 2004.

[268] T. Park and S. Lee. A Bayesian approach for estimating link travel time on urban arterial road network. *Lecture notes in computer science*, pages 1017–1025, 2004.

[269] Anthony D. Patire, Alexandre M. Bayen, Daniel B. Work, Juan C. Herrera, Ryan Herring, Xuexang (Jeff) Ban, Quinn Jacobson, Olli-Pekka Tossavainen, Sebastien Blandin, Christian Claudel, Ali Mortazavi, Steve Andrews, Baik Hoh, Marco Gruteser, Murali Annavaram, Toch Iwuchukwu, and Kenneth Tracton. Mobile Century Final Report for TO 1021 and TO 1029: A traffic sensing field experiment using GPS mobile phones. Technical Report UCB-ITS-CWP-2010-4, ISSN 1557-2269, UC Berkeley, Institute of Transportation Studies (ITS), 2010.

[270] H. Payne. *Models of freeway traffic and control*. Simulation Councils, Inc., 1971.

[271] K. Petty, P. Bickel, M. Ostland, J. Rice, F. Schoenberg, J. Jiang, and Y. Ritov. Accurate estimation of travel times from single-loop detectors. *Transportation Research Part A*, 32(1):1–17, 1998.

[272] P.E. Pfeifer and S.J. Deutsch. A three-stage iterative procedure for space-time modeling. *Technometrics*, 22(1):35–47, February 1980.

[273] L.A. Pipes. Car following models and the fundamental diagram of road traffic. *Transportation Research*, 1(1):21–29, 1967.

[274] Russel Power and Jinyang Li. Piccolo: Building fast, distributed programs with partitioned tables. In *OSDI*, 2010.

[275] A. Prekopa. Logarithmic concave measures with application to stochastic programming. *Acta Scientiarum Mathematicarum*, 32:301–315, 1971.

[276] A. Prekopa. On logarithmic concave measures and functions. *Acta Scientiarum Mathematicarum*, 34:335–343, 1973.

[277] C. Prieur and J. de Halleux. Stabilization of a 1-D tank containing a fluid modeled by the shallow water equations. *Systems & Control Letters*, 52(3-4):167–178, 2004.

[278] T.S. Rabbani, F. Di Meglio, X. Litrico, and A.M. Bayen. Feed-forward control of open channel flow using differential flatness. *IEEE Transactions on Control Systems Technology*, 18(1):213–221, 2009.

[279] L. Rabiner. A tutorial on hidden markov models and selected applications inspeech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[280] S. Reddy, D. Estrin, and M. Srivastava. Recruitment framework for participatory sensing data collections. In *Eighth International Conference on Pervasive Computing*, 2010.

[281] S. Reddy, K. Shilton, J. Burke, D. Estrin, M. Hansen, and M. Srivastava. Evaluating participation and performance in participatory sensing. In *International Workshop on Urban, Community, and Social Applications of Networked Sensing Systems - UrbanSense08*, 2008.

[282] S. Reddy, K. Shilton, J. Burke, D. Estrin, M. Hansen, and M. Srivastava. Using context annotated mobility profiles to recruit data collectors in participatory sensing. In *th International Symposium on Location and Context Awareness (LOCA)*, 2009.

[283] D. Reid. *An algorithm for tracking multiple targets*. Dec 1979. IEEE Transactions on Automatic Control.

[284] J. Rice and E. Van Zwet. A simple and effective method for predicting travel times on freeways. *IEEE Transactions on Intelligent Transportation Systems*, 5(3):200–207, 2004.

[285] P. I. Richards. Shock waves on the highway. *Operations Research*, 4(1):42–51, 1956.

[286] S. Ritchie, S. Park, S. Jeng, and A. Tok. Anonymous vehicle tracking for real-time freeway and arterial street performance measurement. Technical Report Research Report, UCB-ITS-PRR-2005-9, California PATH, 2005.

[287] C. Robert. *The Bayesian choice: a decision-theoretic motivation*. Springer-Verlag, 1994.

[288] R.T. Rockafellar. *Convex Analysis*. Princeton University Press, 1970.

[289] S. Russell and P. Norvig. *Artificial Intelligence - A Modern Approach*. Prentice-Hall, Inc, Englewood Cliffs, NJ, 1995.

[290] P. Saint-Pierre. Approximation of the viability kernel. *Applied Mathematics and Optimization*, 29:187–209, 1994.

[291] S. Saroiu and A. Wolman. *I am a sensor, and i approve this message*. 2010. ACM HotMobile.

[292] J. Sau, N.E. El Faouzi, A. Ben Assa, and O. De Mouzon. Particle filter-based real-time estimation and prediction of traffic conditions. *Applied Stochastic Models and Data Analysis*, 12, 2007.

[293] B. Scholkopf and A. Smola. *Learning with kernels*. MIT press, 2002.

[294] R. Sedgewick. *Algorithms in C*. Addison Wesley Publishing Company, 1990.

[295] D. Serre. *Systems of conservation laws*. Diderot, 1996.

[296] J. A. Sethian. *Level Set Methods and Fast Marching Methods*. Cambridge University Press, New York, NY, 1999.

[297] K. Shilton, J. Burke, D. Estrin, R. Govindan, and J. Kang. Designing the personal data stream: Enabling participatory privacy in mobile personal sensing. In *th Research Conference on Communication, Information and Internet Policy (TPRC)*, 2009.

[298] K. Shilton, J. Burke, D. Estrin, M. Hansen, and M. Srivastava. Participatory privacy in urban sensing. In *International Workshop on Mobile Device and Urban Sensing (MODUS 2008)*, 2008.

[299] K. Shilton, N. Ramanathan, V. Samanta, J. Burke, D. Estrin, M. Hansen, and M. Srivastava. Participatory design of urban sensing networks: Strengths and challenges. In *Participatory Design Conference*, 2008.

[300] D. Simon. *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Wiley-Interscience, 2006.

[301] V. Sisiopiku and N. Rouphail. Travel time estimation from loop detector data for advanced traveler information system applications. Technical report, Illinois University Transportation Research Consortium, 1994.

[302] A. Skabardonis and R. Dowling. Improved speed-flow relationship for planning applications. *Transportation Research Record: Journal of the Transportation Research Board*, 1572:18–23, 1997.

[303] A. Skabardonis and N. Geroliminis. Real-time estimation of travel times on signalized arterials. In *Proceedings of the 16th International Symposium on Transportation and Traffic Theory*, 2005.

[304] A. Skabardonis and N. Geroliminis. Real-Time monitoring and control on signalized arterials. *Journal of Intelligent Transportation Systems*, 12(2):64–74, March 2008.

[305] S. Smulders. Control of freeway traffic flow by variable speed signs. *Transportation Research Part B: Methodological*, 24(2):111 – 132, 1990.

[306] C. Snyder, T. Bengtsson, P. Bickel, and J. Anderson. Obstacles to high-dimensional particle filtering. *Monthly Weather Review*, 136:4629–4640, 2008.

[307] D. Sobel. *Longitude: the true story of a lone genius who solved the greatest scientific problem of his time*. Walker, 1995.

[308] K. Srinivasan and P. Jovanis. Determination of number of probe vehicles required for reliable travel time measurement in urban network. *Transportation Research Record*, 1537(-1):15–22, 1996.

[309] I. S. Strub and A. M. Bayen. Weak formulation of boundary conditions for scalar conservation laws. *International Journal of Robust and Nonlinear Control*, 16:733–748, 2006.

[310] D. Sun and A. M. Bayen. Multicommodity Eulerian-Lagrangian large-capacity cell transmission model for en route traffic. *Journal of Guidance Control and Dynamics*, 31(3):616, 2008.

[311] X. Sun, L. Munoz, and R. Horowitz. Methodological calibration of the cell transmission model. In *American Control Conference*, Boston, MA, June 2004.

[312] X. Sun, L. Munoz, and R. Horowitz. Mixture Kalman filter based highway congestion mode and vehicle density estimator and its application. In *Proc. of the American Control Conference*, volume 3, pages 2098–2103, Boston, MA, 2004.

[313] L. Sweeney. *Achieving k-Anonymity Privacy Protection Using Generalization and Suppression*. 2002. International Journal on Uncertainty, Fuzziness and Knowledge-based Systems.

[314] M. Szeto and D. Gazis. Application of kalman filtering to the surveillance and control of traffic systems. *Transportation Science*, 6(4):419–439, 1972.

[315] C. Tampere and L. Immers. An extended Kalman filter application for traffic state estimation using CTM with implicit mode switching and dynamic parameters. In *IEEE Intelligent Transportation Systems Conference*, pages 209 –216, Seattle, WA, Sept. 30–Oct. 3 2007.

[316] B. Temple. Systems of conversation laws with invariant submanifolds. *Trans. Amer. Math. Soc.*, 280(2):781–795, 1983.

[317] A. Thiagarajan, L. Sivalingam, K. LaCurts, S. Toledo, J. Eriksson, S. Madden, and H. Balakrishnan. VTrack: Accurate, Energy-Aware Traffic Delay Estimation Using Mobile Phones. In *th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, Berkeley, CA, November 2009.

[318] Kurt Thomas, Chris Grier, Justin Ma, Vern Paxson, and Dawn Song. Design and evaluation of a real-time url spam filtering service. In *IEEE Symposium on Security and Privacy*, May 2011.

[319] A. Tikhonov. Solution of incorrectly formulated problems and the regularization method. In *Soviet Math. Dokl*, volume 4, pages 1035–1038, 1963.

[320] L. Tong. Nonlinear dynamics of traffic jams. *Phys. D*, 207(1-2):41–51, 2005.

[321] E. Toro. *Riemann solvers and numerical methods for fluid dynamics*. Springer-Verlag, 1997.

[322] O. P. Tossavainen, J. Percelay, A. Tinka, Q. Wu, and A.M. Bayen. Ensemble kalman filter based state estimation in 2d shallow water equations using lagrangian sensing and state augmentation. In *th IEEE Conference on Decision and Control, 2008*, pages 1783–1790, 2008.

[323] P. Turney. A theory of cross-validation error. *Journal of Experimental and Theoretical Artificial Intelligence*, 6:361–361, 1994.

[324] R. Underwood. Speed, volume, and density relationships: Quality and theory of traffic flow. *Yale Bureau of Highway Traffic*, pages 141–188, 1961.

[325] USGS. Did you feel it?, 2011. http://http://earthquake.usgs.gov/earthquakes/dyfi/.

[326] C. P. IJ. van Hinsbergen, T. Schreiter, F. S. Zuurbier, J. W. C. van Lint, and H. J. van Zuylen. Fast traffic state estimation with the localized extended kalman filter. In *th International IEEE Annual Conference on Intelligent Transportation Systems*, pages 917 – 922, Madeira Island, Portugal, September 19–22 2010.

[327] P. Varaiya. Reducing highway congestion: an empirical approach. *Eur. J. Control*, 11(4-5):301–309, 2005.

[328] P. Varaiya. Congestion, ramp metering and tolls. *Philos. Trans. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci.*, 366(1872):1921–1930, 2008.

[329] M. Wainwright and M. Jordan. Graphical models, exponential families, and variational inference. Technical report, Dept. of Statistics, September 2003. Published: Technical Report 649.

[330] S.T. Waller and A.K. Ziliaskopoulos. On the online shortest path problem with limited arc cost dependencies. *Networks*, 40(4):216–227, 2002.

[331] Y. Wang and M. Papageorgiou. Real-time freeway traffic state estimation based on extended Kalman filter: a general approach. *Transportation Research Part B*, 39(2):141–167, 2005.

[332] Y. Wang, M. Papageorgiou, A. Messmer, P. Coppola, A. Tzimitsi, and A. Nuzzolo. An adaptive freeway traffic state estimator. *Automatica*, 45(1):10–24, 2009.

[333] P. D. Wasserman. *Neural computing: theory and practice*. Van Nostrand Reinhold Co., New York, NY, USA, 1989.

[334] J. Wasson, J. Sturdevant, and D. Bullock. Real-time travel time estimates using mac address matching. *Institute of Transportation Engineers Journal*, 78(6):20–23, 2008.

[335] P. Wendykier. Jtransforms, http://sites.google.com/site/piotrwendykier/software/jtransforms , 2009.

[336] G. Whitham. *Linear and Nonlinear Waves*. Pure Appl. Math., 1974.

[337] S. Winchester. The professor and the madman. 1999. HarperPernnial, New York, 1999.

[338] Jason Wolfe, Aria Haghighi, and Dan Klein. In *ICML*, 2008.

[339] D. Work, S. Blandin, O. Tossavainen, B. Piccoli, and A. Bayen. A traffic model for velocity data assimilation. *Applied Research Mathematics eXpress (ARMX)*, 1:1–35, April 2010.

[340] D. Work, O-P. Tossavainen, S. Blandin, A. Bayen, T. Iwuchukwu, and K. Tracton. An ensemble Kalman filtering approach to highway traffic estimation using GPS enabled mobile devices. In *47th IEEE Conference on Decision and Controls, 2008 Cancun, Mexico*, pages 5062–5068, 2008.

[341] D. Work, O.P. Tossavainen, Q. Jacobson, and A. Bayen. Lagrangian Sensing: Distributed traffic estimation with mobile devices. Saint Louis, MO, 2009. To appear in the American Control Conference.

[342] D. B. Work, S. Blandin, O.-P. Tossavainen, B. Piccoli, and A. M. Bayen. A traffic model for velocity data assimilation. pages 2010(1):1–35, 2010. Applied Mathematics Research eXpress.

[343] Q. Wu, X. Litrico, and A. M. Bayen. Data reconciliation of an open channel flow network using modal decomposition. *Advances in Water Resources*, 32(2):193–204, 2009.

[344] X. Xie, R. Cheu, and D. Lee. Calibration-free arterial link speed estimation model using loop data. *Journal of Transportation Engineering*, 127(6):507–514, 2001.

[345] H. Xiong and G. Davis. Travel time estimation on arterials. In *Proceedings of the 87th Annual Meetings of Transportation Research Board (CD-ROM)*, 2008.

[346] J. Yeon, L. Elefteriadou, and S. Lawphongpanich. Travel time estimation on a freeway using discrete time markov chains. *Transportation Research Part B*, 42(4):325–338, 2008.

[347] Yuan Yu, Michael Isard, Dennis Fetterly, Mihai Budiu, Úlfar Erlingsson, Pradeep Kumar Gunda, and Jon Currey. DryadLINQ: A system for general-purpose distributed data-parallel computing using a high-level language. In *OSDI*, 2008.

[348] Matei Zaharia, Mosharaf Chowdhury, Michael J. Franklin, Scott Shenker, and Ion Stoica. Spark: Cluster Computing with Working Sets. In *HotCloud*, 2010.

[349] Matei Zaharia, Mosharaf Chowdhury, Michael J. Franklin, Scott Shenker, and Ion Stoica. Spark: Cluster computing with working sets. Technical report, UC Berkeley, Technical Report UCB/EECS-2010-53, EECS Dept, May 2010.

[350] H. Zhang. A link journey speed model for arterial traffic. *Transportation Research Record*, 1676:109–115, 1998.

[351] H. Zhang. A theory of nonequilibrium traffic flow. *Transportation Research Part B*, 32(7):485–498, 1998.

[352] H. Zhang. A non-equilibrium traffic model devoid of gas-like behavior. *Transportation Research Part B*, 36(3):275–290, 2002.

Part IX

Appendices

A. Technical Appendix

A.1. Derivation of the probability distribution of total delay between arbitrary locations in the congested regime

We derive the probability distribution of travel times for vehicles traveling from a location x_1 to a location x_2 on the link. As in the previous notations, x represents the distance to the intersection.

We call n_s the maximum number of stops in the remaining queue experienced by the vehicles between the locations x_1 and x_2, and omit the indices x_1 and x_2 for notational simplicity. In the duration of a light cycle, the distance traveled by vehicles stopped in the queue is $l_{\max}$. Thus, the maximum number of stops in the remaining queue, between x_1 and x_2,

$$n_s = \left\lceil \frac{min\,(x_1, l_r) - min\,(x_2, l_r)}{l_{max}} \right\rceil.$$

The delay experienced when reaching the triangular queue is readily derived from the expression of the delay in the undersaturated regime. The delay experienced when reaching the remaining queue is the duration of the red time R. The expression of the delay at location x is then

$$\delta^c(x) = \begin{cases} R & \text{if} x \leq l_r \\ R\,\dfrac{l_r + l_{max} - x}{l_{max}} & \text{if} x \in [l_r, l_r + l_{max}] \\ 0 & \text{if} x \geq l_r + l_{max} \end{cases}$$

Case 1: x_1 is upstream of the total queue and x_2 is in the remaining queue (Figure A.1)

<u>Condition 1:</u> $x_1 \geq l_r + l_{max} \quad x_2 \leq l_r$

Since x_1 is upstream of the total queue and x_2 is in the remaining queue, all the vehicles stop once in the triangular queue between x_1 and x_2. We define the critical location x_c as the location in the triangular queue such that:

- Vehicles reaching the triangular queue upstream of x_c stop n_s times in the remaining queue. They represent a fraction $\frac{l_r+l_{max}-x_c}{l_{max}}$ of the vehicles entering the link in a cycle.
- Vehicles reaching the triangular queue downstream of x_c stop $n_s - 1$ times in the remaining queue. They represent a fraction $\frac{x_c-l_r}{l_{max}} = 1 - \frac{l_r+l_{max}-x_c}{l_{max}}$ of the vehicles entering the link in a cycle. The location x_c is given by $x_c = x_2 + n_s l_{max}$.

The values of the minimum and maximum delays are given by $\delta_{\min} = (n_s - 1)R + \delta^c(x_c)$ and $\delta_{max} = n_s R + \delta^c(x_c)$. The delay experienced by the vehicles is uniformly distributed on $[\delta_{max}, \delta_{max}]$.

We note that $n_s \geq 1$ since $x_2 \leq l_r$.

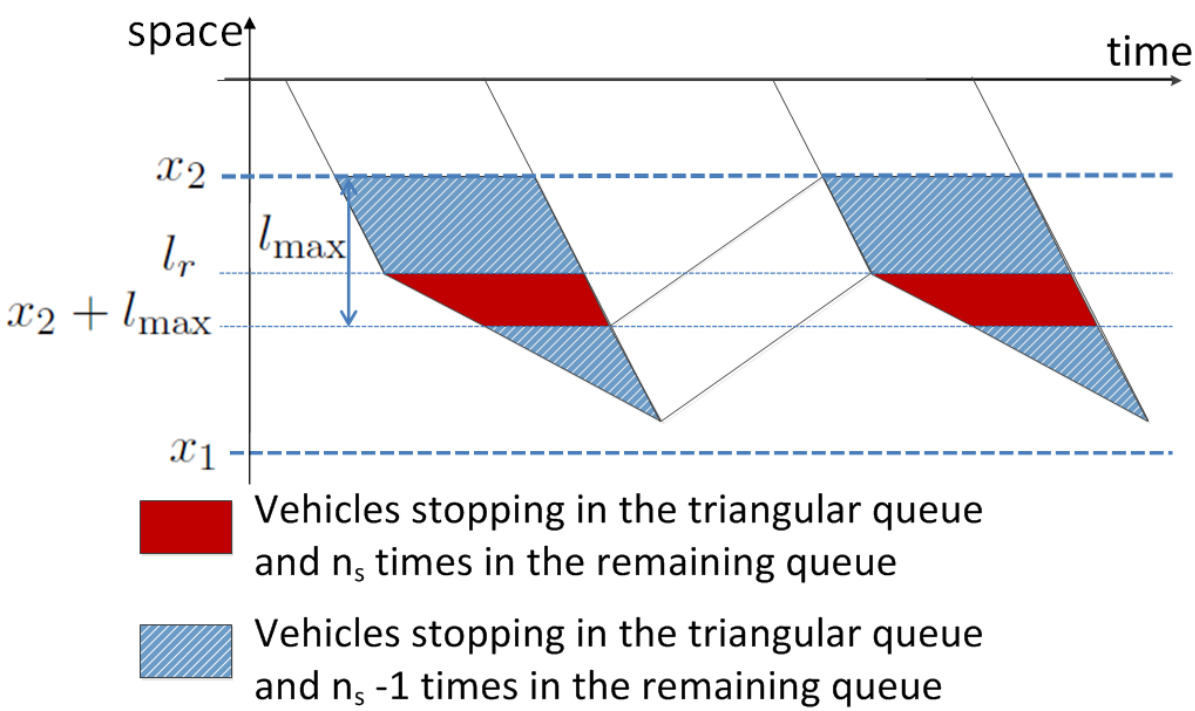


Figure A.1. Case 1: All the vehicles stop in the triangular queue. A fraction stops n_s times in the remaining queue, the other ones stop $n_s - 1$ times.

Case 2: x_1 and x_2 are upstream of the remaining queue (Figure A.2)

<u>Condition 2:</u> $x_1 \geq l_r \quad x_2 \geq l_r$

Given that x_2 is upstream of the remaining queue, this case is similar to the undersaturated regime. A fraction of the vehicles does not experience delay between x_1 and x_2. The vehicles reaching the queue between x_1 and x_2 experience a delay in the triangular queue. This delay is a random variable, uniformly distributed on $[\delta^c(x_1), \delta^c(x_2)]$.

The fraction of vehicles experiencing delay is $\eta^c_{x_1,x_2} = \frac{min\,(l_{max}+l_r,x_1)-min\,(l_{max}+l_r,x_2)}{l_{max}}$

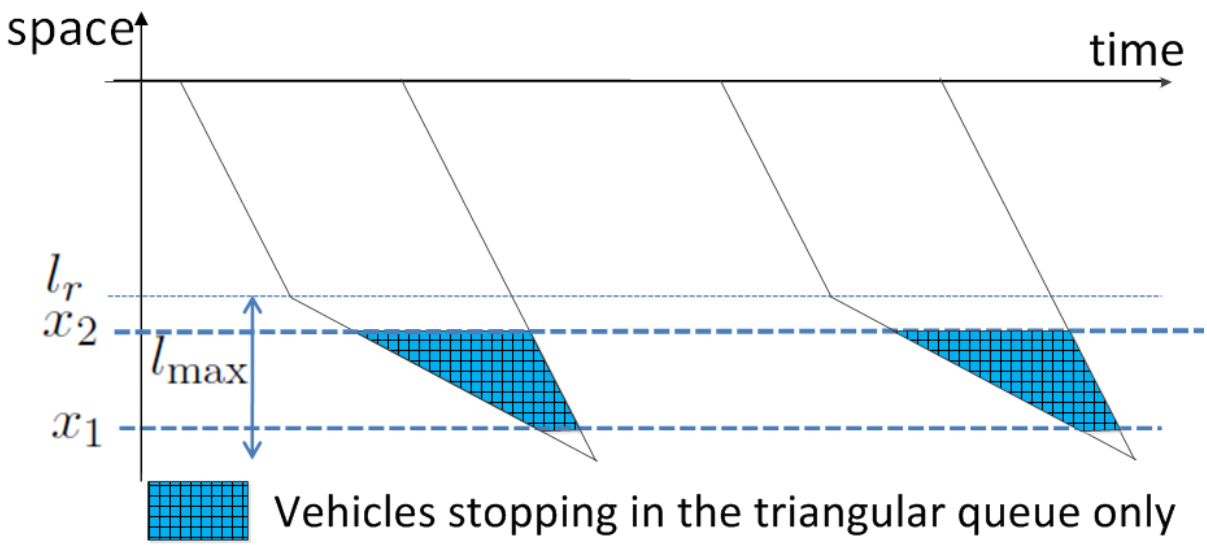


Figure A.2. Case 2: Some vehicles stop in the triangular queue. The others do not experience delay.

Case 3: x_1 is in the remaining queue, and thus so is x_2 (Figure A.3)

<u>Condition 3:</u> $x_1 \leq l_r$ (which implies $x_2 \leq l_r$)

The path starts downstream of the triangular queue. Some vehicles stop n_s times and experience a delay $n_s R$ and the other vehicles stop $n_s - 1$ times and experience a delay $(n_s - 1)R$.

We define the critical location x_c as the location in the remaining queue such that

- Vehicles joining the queue between x_1 and x_c stop n_s times between x_1 and x_2. Their stopping time is $n_s R$ and they represent a fraction $(x_1 - x_c)/l_{\max}$ of the vehicles entering the link in a cycle.
- Vehicles joining the queue between x_c and $x_c - l_{max}$ stop $n_s - 1$ times between x_1 and x_2. Their stopping time is $(n_s - 1)R$ and they represent a fraction $1 - (x_1 - x_c)/l_{max}$ of the vehicles entering the link in a cycle.

The critical location x_c is given by $x_c = x_2 + (n_s - 1)l_{max}$.

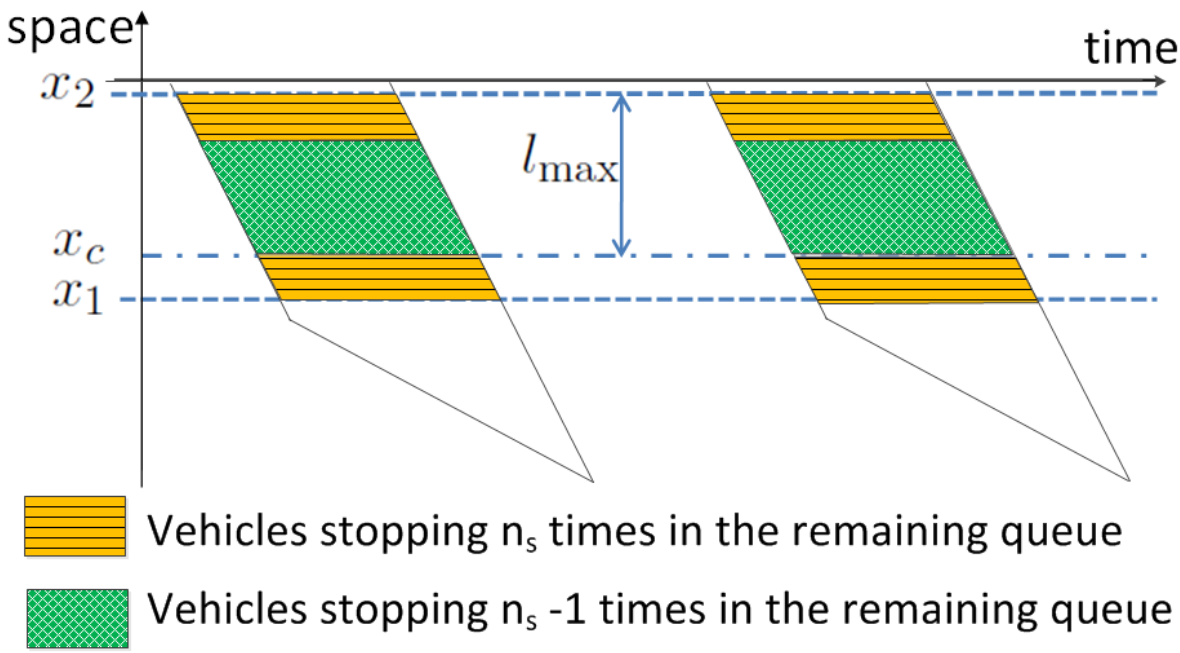


Figure A.3. Case 3: A fraction of the vehicles stop n_s times in the remaining queue. The rest stop $n_s - 1$ times in the remaining queue.

Case 4: x_1 is in the triangular queue, x_2 is in the remaining queue

We distinguish two different cases to derive the probability distribution of travel times. We define the critical location x_c as $x_c = x_2 + n_s l_{max}$ and derive probability distributions of travel times for the two subcases 4a ($x_c \leq x_1$, Figure A.4 (top)) and 4b ($x_c \geq x_1$, Figure A.4 (bottom)).

<u>Case 4a.</u> $x_c \leq x_1$. The delay patterns are the following:

- One stop in the triangular queue and n_s stops in the remaining queue. The queue is first reached between x_1 and x_c. The delay is a random variable with uniform distribution with support $[\delta^c(x_1) + n_s R, \delta^c(x_c) + n_s R]$. The vehicles following this pattern represent a fraction $\frac{x_1 - x_c}{l_{max}}$ of the vehicles entering the link in a cycle.
- One stop in the triangular queue and $n_s - 1$ stops in the remaining queue. The queue is first reached between x_c and l_r. The delay is a random variable with uniform distribution with support $[\delta^c(x_c) + (n_s - 1)R, \delta^c(l_r) + (n_s - 1)R]$. Noticing that $\delta^c(l_r) = R$, we derive that the support of the delay distribution is $[\delta^c(x_c) + (n_s - 1)R, n_s R]$. The vehicles following this pattern represent a fraction $\frac{x_c - l_r}{l_{max}}$ of the vehicles entering the linkin a cycle.
- No stop in the triangular queue and n_s stops in the remaining queue. The queue is first

reached between l_r and $x_1 - l_{max}$. The delay is $n_s R$. The vehicles following this pattern represent a fraction $\frac{l_r-(x_1-l_{max})}{l_{max}}$ of the vehicles entering the link in a cycle.

We can check that the weights of the different components sum to 1:

$$\frac{x_1 - x_c}{l_{max}} + \frac{x_c - l_r}{l_{max}} + \frac{l_r - (x_1 - l_{max})}{l_{max}} = 1$$

We remark that, $x_2 \leq l_r$ implies that $n_s \geq 1$. Then using the definition of x_c, $x_c = x_2 + n_s l_{max}$ and the fact that $x_1 \geq x_c$, we prove that $x_1 - l_{max} \geq x_2$ and all vehicles reach the queue between x_1 and $x_1 - l_{max}$.

<u>Case 4b.</u> $x_c \geq x_1$. The delay patterns are the following:

- One stop in the triangular queue and $n_s - 1$ stops in the remaining queue. The queue is first reached between x_1 and l_r. The delay is a random variable with uniform distribution on $[\delta^c(x_1) + (n_s - 1)R, \delta^c(l_r) + (n_s - 1)R]$, *i.e.* uniform distribution on $[\delta^c(x_1) + (n_s - 1)R, n_s R]$. The vehicles following this pattern represent a fraction $\frac{x_1-l_r}{l_{max}}$ of the vehicles entering the link in a cycle.
- No stop in the triangular queue and n_s stops in the remaining queue. The queue is first joined between l_r and $x_c - l_{max}$. The delay is $n_s R$. The vehicles following this pattern represent a fraction $\frac{l_r-(x_c-l_{max})}{l_{ax}}$ of the vehicles entering the link in a cycle.
- No stop in the triangular queue and $n_s - 1$ stops in the remaining queue. The queue is first joined between $x_c - l_r$ and $x_1 - l_{max}$. The delay is $(n_s - 1)R$. The vehicles following this pattern represent a fraction $\frac{x_c-x_1}{l_{max}}$ of the vehicles entering the link in a cycle.

We can check that the weights of the different components sum to 1:

$$\frac{l_r - (x_c - l_{max})}{l_{max}} + \frac{x_1 - l_r}{l_{max}} + \frac{x_c - x_1}{l_{max}} = 1.$$

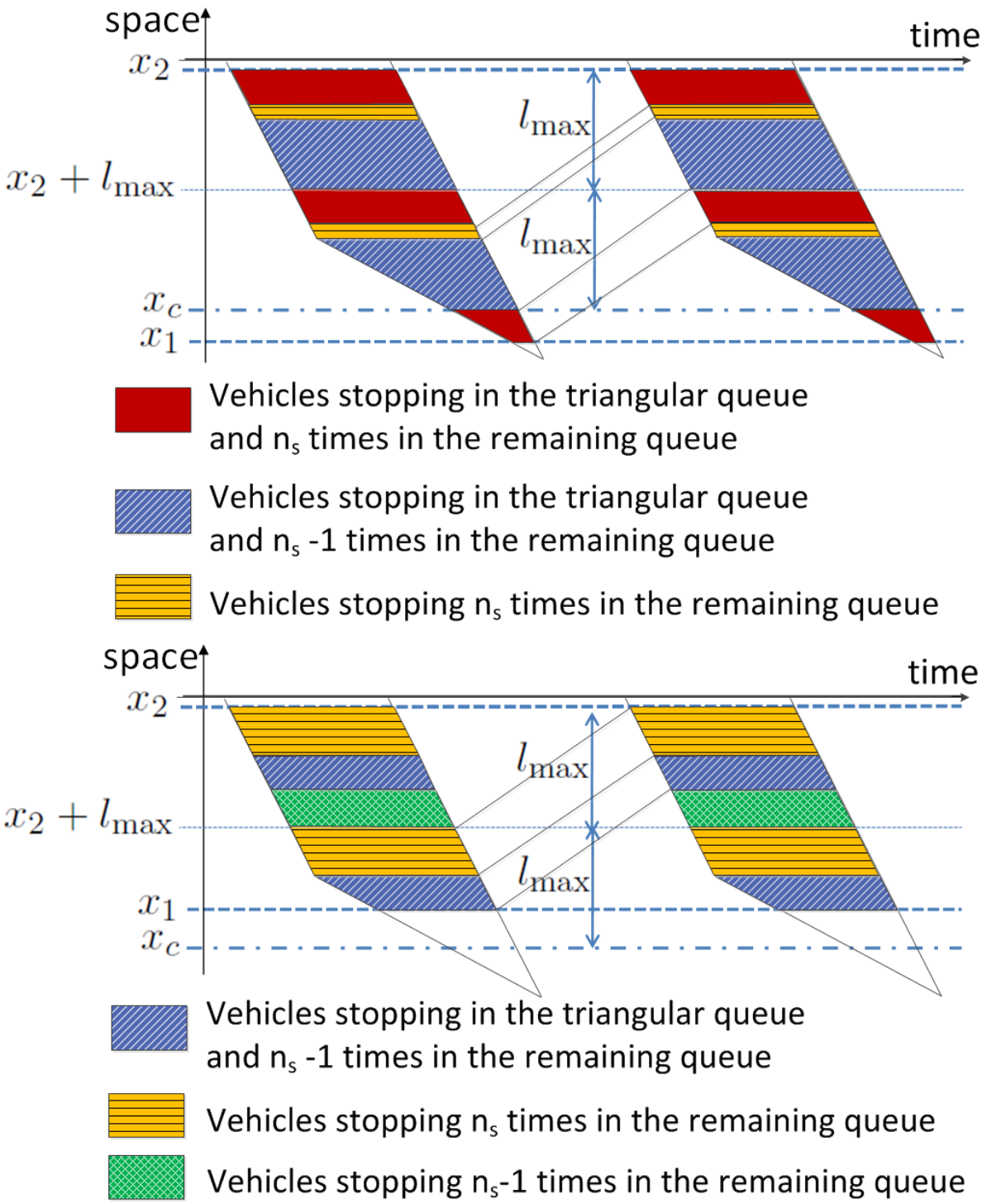


Figure A.4. Case 4: (Top) Case 4a: a fraction of the vehicles stop in the triangular queue and n_s times in the remaining queue, a fraction of the vehicles stop in the triangular queue and n_s times in the remaining queue, the rest stop n_s times in the remaining queue. (Bottom) Case 4b: a fraction of the vehicles stop in the triangular queue and $n_s - 1$ times in the remaining queue, a fraction of the vehicles stop n_s times in the remaining queue, the rest stop $n_s - 1$ times in theremaining queue.