ONLINE AD HOC DISTRIBUTED TRAFFIC SIMULATION WITH OPTIMISTIC EXECUTION

A Dissertation
Presented to
The Academic Faculty

By

Wonho Suh

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in Civil Engineering

Georgia Institute of Technology

August 2012

ONLINE AD HOC DISTRIBUTED TRAFFIC SIMULATION WITH OPTISTIC EXECUTION

Approved by:

Dr. Michael P. Hunter, Advisor
School of Civil and Environmental Engineering
Georgia Institute of Technology

Dr. John D. Leonard
College of Engineering
Georgia Institute of Technology

Dr. Michael O. Rodgers
School of Civil and Environmental Engineering
Georgia Institute of Technology

Dr. Jorge A. Laval
School of Civil and Environmental Engineering
Georgia Institute of Technology

Dr. Richard M. Fujimoto
College of Computing
Georgia Institute of Technology

Date Approved: June 26, 2012

To my wife, Hyunjin Son and two children, Minju & Jaimin for their support and love.

ACKNOWLEDGEMENTS

My journey through the PhD program at Georgia Tech allowed me to develop the courage to overcome failure, heartache, and frustration that accompany all PhD candidates. Now as this journey comes to an end, I hope to live up to all the responsibilities that accompany the title of PhD.

For one's life long journey, there exists a team of support that assists in one’s physical, mental, and emotional development. I too have had a team of supporters guiding me through the toughest times as well as celebrating all of my accomplishments. I could not have successfully completed this journey without the encouragement of my mentor and advisor, Dr. Michael Hunter. Dr. Hunter’s guidance and wisdom has shown me to strive for perfection. His mentoring has influenced not only my academics but also various aspects of my life. I am truly indebted to you, thank you.

To my other supporters, members of my dissertation committee, Dr. John Leonard, Dr. Michael Rodgers, Dr. Jorge Laval, and Dr. Richard Fujimoto, your professionalism, words of wisdom, and patience, allowed me to process through my research in meaningful manner. Thank you for your endless support and guidance. My gratitude also goes to Dr. Randall Guensler, Dr. Michael Meyer and Dr. Angshuman Guin for their invaluable guidance.

To the rest of my friends, Dr. Saroch Boonsiripant, Dr. Danjue Chen, Dr. Dwayne Henclewood, Dr. Hoe Kyoung Kim, Dr. Mshadoni Smith, Dr. Seung Kook Wu, Lance Ballard, Christina Barry, Lacy Bell, Stephanie Box, Bhargava (Rama) Chilukuri, Kate D’Ambrosio, Vetri Venthan Elango, Aaron Greenwood, Ya-Lin Huang, Ying Li, Chenhao Liu, Narasimham (Lakshmi) Peesapati, Luke Taylor, and Tom Wall, your friendship, support, and knowledge challenged me question my ideas and beliefs. Knowing that you all would have my back allowed me to take chances and forced me to get back up when I had fallen.

Finally, I would like to thank my wonderful parents and parents-in-law for their unconditional endless love and continued support throughout my journey.

Thank you!

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

As roadside and in-vehicle sensors are deployed under the Connected Vehicle Research program (formerly known as Vehicle Infrastructure Integration initiative and Intellidrive^{SM}), an increasing variety of traffic data is becoming available in real time. This real time traffic data is shared among vehicles and between vehicles and traffic management centers through wireless communication. This course of events creates an opportunity for mobile computing and online traffic simulations.

However, online traffic simulations require faster than real time running speed with high simulation resolution, since the purpose of the simulations is to provide immediate future traffic forecast based on real time traffic data. However, simulating at high resolution is often too computationally intensive to process a large scale network on a single processor in real time. To mitigate this limitation an online ad hoc distributed simulation with optimistic execution is proposed in this study.

The objective of this study is to develop an online traffic simulation system based on an ad hoc distributed simulation with optimistic execution. In this system, data collection, processing, and simulations are performed in a distributed fashion. Each individual simulator models the current traffic conditions of its local vicinity focusing only on its area of interest, without modeling other less relevant areas. Collectively, a central server coordinates the overall simulations with an optimistic execution technique and provides a predictive model of traffic conditions in large areas by combining simulations geographically spread over large areas. This distributed approach increases computing capacity of the entire system and speed of execution. The proposed model

manages the distributed network, synchronizes the predictions among simulators, and resolves simulation output conflicts. Proper feedback allows each simulator to have accurate input data and eventually produce predictions close to reality. Such a system could provide both more up-to-date and robust predictions than that offered by centralized simulations within a single transportation management center. As these systems evolve, the online traffic predictions can be used in surface transportation management and travelers will benefit from more accurate and reliable traffic forecast.

CHAPTER 1 INTRODUCTION

1.1 Background

While demands on transportation system continue to grow, resources to address these demands are becoming increasing scarce. According to statistics from the U.S. Census Bureau, FWHA, and the Texas Transportation Institute, the number of vehicles in the United States has increased more than 50% and the vehicle miles traveled have almost doubled from 1982 to 2010 [1, 2]. While there are more vehicles in the system and many more miles being driven, the total highway lane miles during this same time period have increased only 7.5% (Figure 1). This prolonged failure of highway construction to match increasing travel demands has resulted in increasing traffic congestion. The delay per each traveler has increased more than 160 percent over the past 25 years and the congestion cost has reached $713 per each traveler in 2010 from $301 in 1982 [3]. To help address these issues increasing emphasis is being placed on real time system efficiency. However, to actively manage transportation operations, capacity, etc., it is necessary to know the current and likely near term state of the system. Unfortunately, a significant challenge faced today is a lack of detailed knowledge of the current real time state of the roadway network, particularly off the freeway system. An online ad hoc distributed simulation approach is proposed to address this lack of current and near term knowledge. Through this distributed and adaptive approach, transportation infrastructure may be provided the information necessary to automatically reconfigure itself to maximize efficiency, minimize the effects of unexpected events such as localized incidents, and provide near term system performance predications.

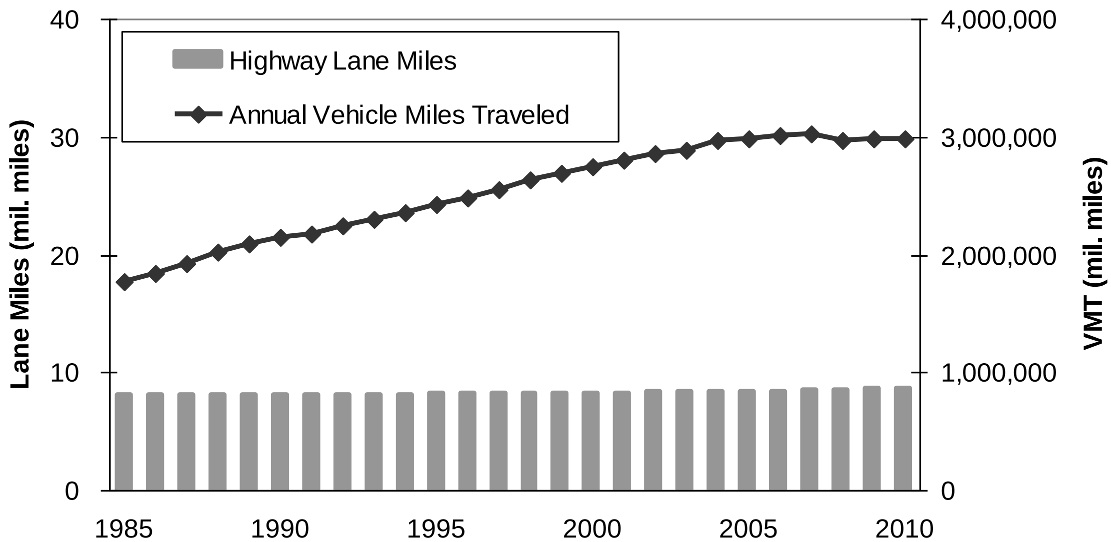

Figure 1 VMT vs. Highway Lane Miles [1, 2]

Recent advancements in sensor, mobile computing, and wireless communication technologies offer new opportunities to address the needs for real time information required to improve system efficiencies. These technologies have contributed to the integration of vehicles and infrastructure in the surface transportation system. New applications from this integration have been rapidly growing with support from public and private sectors. In 2002, ITS America in cooperation with the US DOT included the use of dedicated short-range communications for ITS safety applications in the "National Intelligent Transportation Systems Program Plan: A Ten-Year Vision". In 2003, the Federal Communications Commission allocated 75 MHz at 5.9 GHz for dedicated short-range infrastructure-to-vehicle and vehicle-to-vehicle communications. Later during the

2003 ITS World Congress of Madrid, Spain, the US DOT launched the Vehicle Infrastructure Integration (VII) initiative [4-9]. The VII Initiative (later renamed Intellidrive[SM] and Connected Vehicle Program) focuses on deploying a communication infrastructure for Dedicated Short Range Communications (DSRC) to support safety, operational, data collection, design and other applications. A public-private VII Coalition including AASHTO, state/local agencies, and automotive manufacturers has been formed and actively participated in the design, testing, and evaluation of a deployable VII system for the US.

Under the Vehicle Infrastructure Integration initiative, roadside units (RSU) and in-vehicle processing units collect and process traffic data. While in-vehicle processing units reside inside vehicles, roadside units (RSU) are stationary and deployed through the transportation system. Both RSU and in-vehicle processing units are equipped with DSRC wireless technology and disseminate traffic data to other units, which in turn forward information to other nearby units. This wireless data transmission creates an opportunity for online simulation applications to enhance traffic safety and operations.

To date the primary field deployed VII example has been in-vehicle collision avoidance systems [10, 11] that monitor and model traffic conditions within close proximity of the vehicle, enabling the detection and avoidance of hazardous conditions. Such systems tend to only consider very immediate future traffic conditions, seconds from current time, allowing for highly accurate predictions. Other applications commonly considered include traffic prediction [12, 13], route planning [14], traffic management [15, 16], and signal operation [17-19].

However, it is possible to consider a broader application of the integration of VII and onboard processing capabilities and intelligence. For example, one may envision in-vehicle simulation applications that model traffic conditions over a broader, but still localized area (e.g., the downtown section of a city), focusing on the vehicle's area of interest. Detailed real time traffic data could be utilized as an input to the in-vehicle simulations with the simulation providing localized traffic estimates. Combining the traffic estimates generated from multiple vehicles throughout the local area and the wider region provides the potential for more accurate and quick responsive traffic models. Such a system could provide both more up-to-date and more robust estimates than that offered by centralized simulations within a single transportation management center. Collectively, the aggregation of in-vehicle simulations may be able to provide a predictive model of the transportation infrastructure and have the ability to automatically revise forecasts as unexpected events occur.

1.2 Problem Definition

To actively manage arterial transportation operations, it is necessary to know the current and likely near term state of the system. Unfortunately, there is a lack of detailed knowledge of the current and near term state of the roadway network, particularly off the freeway system. To address the lack of sufficient real time network state and near term future traffic state of arterials, an online traffic simulation is proposed.

In the envisioned online traffic simulation, data collection, processing, simulations, and estimates are performed in a distributed fashion by roadside units and onboard vehicles. A central server coordinates the overall simulation with an optimistic execution technique. Such a distributed approach can decrease communication bandwidth requirements and increase computing capacity. Communication middleware would act to manage the distributed network, synchronize the estimates among in-vehicle simulators, and resolve simulation output conflicts. Proper feedback would allow each vehicle to have accurate input data and eventually produce estimates close to reality. As these systems evolve, the online traffic estimates can be used in surface transportation management, and travelers will benefit from a more accurate and reliable traffic forecast.

Two significant challenges exist to satisfactorily implement the envisioned system. Online traffic simulations are required to have 1) a resolution sufficient to enable the detailed estimates of traffic conditions on a local street network and 2) fast running speed (faster than real time) in order to provide sufficiently fast and detailed information.

Simulations in the system are envisioned to be microscopic, that is they model individual vehicles, allowing the simulations to realistically represent individual traffic characteristics and capture dynamically changing traffic conditions, such as localized traffic incidents in the network. Microscopic traffic simulation offers the high level of accuracy necessary for online traffic estimates.

With the precision of microscopic simulation come limitations in terms of computing loads, which increases with network size and number of vehicles simulated. Simulating at high resolution is often too computationally intensive to process a large scale network as a single monolithic model faster than real time. Simulation performance degrades significantly as the network size increases and number of vehicles in the network increases. Therefore, it is unrealistic to simulate a large traffic network, such as the Metro Atlanta, faster than real time on the resources generally available to most departments of transportation and other public agencies.

This potential processing constraint is a significant issue as simulations must run faster than real time, since the purpose is to provide drivers with short-term traffic forecasts based on real time traffic estimates. Execution speed becomes increasingly critical if the applications are to be used for emergency response scenarios [20-23]. Numerous researchers have attempted to address this scalability problem of microscopic simulation. Parallel and distributed simulation has been considered as one of the promising solutions to achieve reasonably fast processing of large network microscopic simulations. In these schemes, a traffic simulation program is partitioned into multiple processors and communication middleware is used to coordinate between multiple single-processor machines. The most established idea is that a large network microscopic simulation can be achieved faster when the network is divided into a set of sub-networks, each of which is assigned to a different processor [20, 21, 24].

Although parallel and distributed simulation increases performance and saves resources in a large-scale computation, it requires simulation time managing processes to synchronize all logical processes, which often significantly reduces efficiency. Since neither speed of each processor nor the computational loads for each processor are the same, speed of the entire simulation is dependant on the slowest processor [25-28]. Faster simulators always have to wait for the slowest processor while all processors need to be synchronized with respect to simulation time. This synchronization overhead can take abundant simulation resources and degrade overall simulation performance.

Despite of these issues, it is believed that the lack of detailed knowledge of the current and likely near term state of the traffic system can be addressed by distributed in-vehicle simulations which provide real time traffic data processing and traffic estimates with increased computing capacity and less communication bandwidth requirements. A distributed approach allows the system to operate in close proximity to real time data, offering the potential to use more accurate data with shorter response time than centralized simulations within a single transportation management center. Further, the redundancy inherent in ad hoc distributed simulations provides more robustness of the system and the simulations would offer more reliable information regarding traffic states and future estimates of the roadway network.

1.3 Research Objectives

The goal of this study is to develop an online ad hoc distributed traffic simulation system based on optimistic execution. Objectives of this study are as follows;

- *Develop a distributed traffic simulation framework:* Each in-vehicle simulation models a small portion of the overall network and provides detailed traffic state information. Traffic simulation and data processing are performed in a distributed fashion by multiple vehicles. Each in-vehicle simulation is designed to run in real time and update its estimates when it is necessary.
- *Integrate communication middleware and traffic simulation*: Middleware is necessary for the distributed simulation to perform on multiple platforms. TRTI, a

communication middleware developed based on object-oriented client/server technology as a parallel effort of other researchers is integrated with traffic simulation. This integration manages the distributed network to synchronize the predictions among logical processes.

- *Implement Space-Time Memory management into a transportation simulation approach:* A local central server receives the traffic states from multiple in-vehicle simulations. Traffic estimates are not guaranteed to be received in time-stamp order, since in-vehicles simulations run concurrently. Also, a traffic state can be projected by multiple in-vehicle simulations. A mechanism is needed to coordinate the transmitted data, combine values into a composite value, and save in Space-Time Memory.
- *Create an optimistic (rollback-based) synchronization protocol*: Optimistic execution inspired by Time Warp can mitigate the synchronization problem allowing each logical process to execute asynchronously. This approach provides increased computing capacity with a time-synchronized approach.

The implementation of these four objectives will be referred to as an online ad hoc distributed traffic simulation.

1.4 Research Contributions

Transportation impacts every aspect of daily life. For many decades efforts to improve transportation have been made to ensure quality of life and higher standards of living.

However, utilization of real time traffic data into our surface transportation system has not been fully accomplished. Recent advancements in sensor, mobile computing, and wireless communication technologies is creating new opportunities to effectively exploit real time traffic data. Onboard vehicles collect, process, simulate traffic states in a distributed fashion and a local transportation management center coordinates the overall simulation with an optimistic execution technique. Such a distributed approach can provide more up-to-date and robust estimates with decreased communication bandwidth requirements and increased computing capacity.

This research effort is expected to provide the following contributions:

- *Develop a distributed traffic simulation framework:* Traffic simulation and data processing are performed in a distributed fashion by multiple in-vehicle simulations which model small portions of the overall network.
- *Integration of TRTI (communication middleware) and traffic simulation*: This integration manages the distributed network to synchronize the predictions among logical processes.
- *Implementation of Space-Time Memory management into a transportation simulation approach:* The estimates across the multiple logical processes are aggregated, transferred into composite values and saved in Space-Time Memory.
- *Create an optimistic (rollback-based) synchronization protocol*: Optimistic execution inspired by Time Warp can mitigate the synchronization problem allowing each

logical process to execute asynchronously. Invalidated estimates are updated quickly by this mechanism to ensure more robust and reliable estimates.

- *Demonstration of the feasibility of the ad hoc distributed model:* The performance of the ad hoc distributed simulation model provides the feasibility of the model under various steady and non-steady traffic conditions.
- *Investigation of the sensitivity of the ad hoc distributed model with different geographical distributions of LPs and rollback thresholds*: The sensitivity analysis provides insights into the parameters of the ad hoc approach and guidance for future research and field implementations.
- *Examination of the performance of the ad hoc distributed simulation under congested traffic conditions:* The congested traffic experiment examines the robustness of the system and the likelihood that a large-scale implementation of the model in real-world settings could be successful.
- *Development of a methodology to incorporate real time field sensor data:* The ad hoc distributed traffic simulation works with the data feed from the real time field sensor data and incorporate them in its model.

Finally, this research is anticipated to provide a framework for an online ad hoc distributed simulation which features dynamic collections of logical processes interacting with each other and with real time data. The ad hoc distributed simulation with optimistic execution will be able to capture, process, and incorporate data into simulation models, and transfer useful information with reasonably fast response time.

1.5 Dissertation Outline

Following the research introduction in Chapter 1, this research effort is structured as follows. Chapter 2 summarizes the previous vehicular ad hoc network studies and reviews the parallel and distributed simulation technologies, optimistic execution methodologies and their related researches. Chapter 3 discusses the running environment and main process for the development of the ad hoc distributed traffic simulation, including functions in global / logical process. Chapter 4 evaluates the ad hoc distributed simulation with graphical and analytical methods. Chapter 5 explores the ad hoc distributed simulation with different traffic conditions, including steady traffic state, volume increase, and incident scenarios. Chapter 6 investigates the sensitivity of the ad hoc distributed simulation with different geographical logical process distributions and different level of rollback thresholds. Chapter 7 examines the ad hoc distributed simulation model under congested traffic conditions and provides discussions about the limitation of the proposed approach. Chapter 8 evaluates the ad hoc distributed simulation model when real time field sensor data is available allowing for real time state estimates of the roadway network. Lastly, the summary of findings, research contributions and future research is described in Chapter 9. The remainder of this dissertation includes Appendix A – Server script and Appendix B – Logical process script.

CHAPTER 2 LITERATURE REVIEW

In this study, an online ad hoc distributed traffic simulation is proposed which incorporates VANET (vehicular ad hoc network), network communication, and optimistic execution. This chapter describes the previous works on parallel and distributed simulation, parallel traffic simulation, and optimistic execution.

2.1 Chapter Organization

This chapter begins with an overview of vehicular ad hoc network in Section 2.2. This is followed in Section 2.3 by a description of parallel and distributed simulation. Section 2.4 provides the previous application of parallel and distributed simulation in transportation area. Section 2.5 addresses optimistic execution and its application in traffic simulation.

2.2 VANET

VANET (Vehicular Ad Hoc Network) refers to a network created by vehicles equipped with short range wireless communication technology. Data communication occurs between vehicles inside their radio range so that real time traffic data from onboard and roadside sensors can be transmitted to and shared among vehicles and between vehicles and traffic management centers. By utilizing this real time data transmission various online simulation applications have been studied including collision avoidance, traffic prediction, route planning, traffic management, and signal timing [10, 12-19, 29-31]. Research on VANET has been actively conducted worldwide including Europe, Japan and the United State [32-36].

In Europe several national and European projects have been carried out. "FleetNet - Internet on the Road" project started in Germany on September 2000 and ended in 2003. It was founded by a consortium of six companies and three universities. Its main objective was to develop a wireless ad hoc network for inter-vehicle communications and it successfully studied and demonstrated the feasibility of ad hoc networking and vehicular communication based on IEEE 802.11 [37].

The NOW (Network on Wheels) is the successor of the FleetNet project. It was founded by several automobile manufacturers in combination with other communication technology companies in 2004 and supported by Federal Ministry of Education and Research in Germany. The main objective is to provide technology on the communication protocols and data security for car-to-car communications, in addition to supporting active safety applications as well as infotainment (information-based media content) applications with infrastructure and between vehicles [38] (Figure 2).

The Car2Car Communication Consortium is a non-profit organization initiated by European vehicle manufacturers. Its first meeting was held in 2004 and its goal is to create a European industrial standard for car-to-car communication to increase road traffic safety and efficiency by means of inter-vehicle communications (Figure 3). NOW is working closely with Car2Car Communication Consortium and the results of NOW project are implemented in standardization activities of the Car2Car Communication Consortium [39].

Figure 2 NOW (Network on Wheels) Applications
(source: http://www.network-on-wheels.de/objectives.html)

GST (Global System for Telematics) is an EU-funded integrated project to create a standardized end-to-end architecture for automotive telematics services. GST consists of seven sub-projects; four technology-oriented sub-projects (Open systems, Certification, Service payment, and Security) and three service-oriented sub-projects (Rescue, Enhanced floating car data, and Safety Channel). Its vision is to provide drivers and occupants on-board integrated telematics system to access a dynamic online safety, efficiency- and comfort-enhancing services wherever they drive in Europe [40].

CVIS (Cooperative Vehicle-Infrastructure Systems) is a European research and development project to design, develop, and test vehicle to vehicle (V2V) and vehicle to nearby roadside infrastructure (V2I) communications. The consortium consists of 60 partners including top vehicle manufacturers, suppliers, universities, research institutes, national road administrations, and representative organizations from the European member states [41].

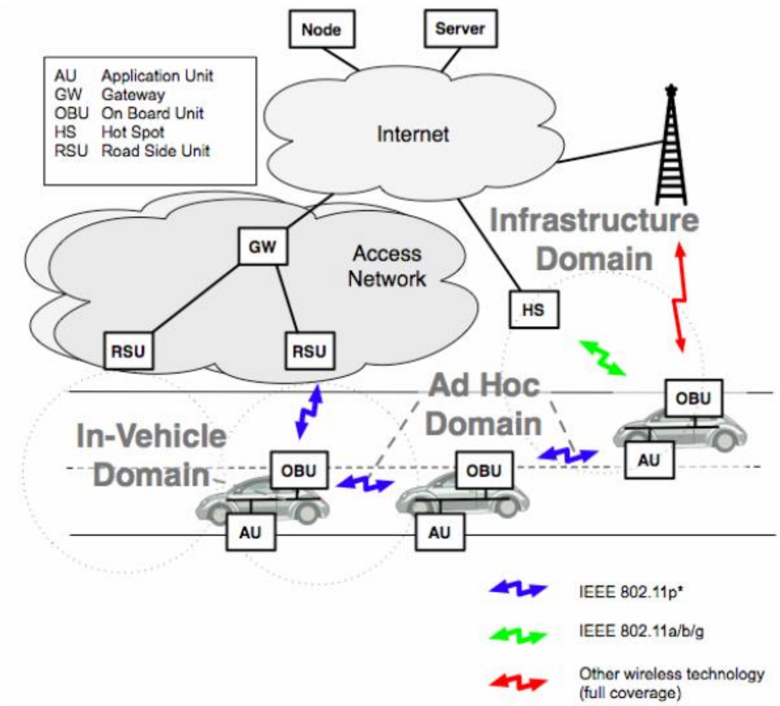

Figure 3 Car2Car System Architecture
(source: http://www.car-2-car.org/index.php?id=11)

In Japan, ASV (Advanced Safety Vehicle) Promotion Project has been in place since 1991. Through collaboration between industry, educational institutions, and the administration, an Advanced Safety Vehicle (ASV) is designed to collect traffic information with various onboard sensors and telecommunications systems and provide safety information based on the information collected. During phase 3 (2001-2005) applications of "infrastructure to car communication" were developed. "Car to car communication" test is in plans for phase 4. Two DSRC standards have been adopted

(ARIB STD-T75 in 2001 and ARIB STD-T88 in 2004) and Ad-hoc Network Platform Consortium has been established including 14 universities and 14 industry members [42-44].

Connected vehicle program in the United States is known as Intellidrive^{SM} and VII (Vehicle Infrastructure Integration) (Figure 4). Its research is focused on technologies and applications that use wireless communications to deliver safety, mobility, and environmental improvements in surface transportation via an open communications platform. It supports data transmission among vehicles (V2V) and between vehicles and roadway infrastructure (V2I) or hand held devices (V2D) to enable numerous safety and mobility applications. Coalition partners including the U.S. Department of Transportation, state and local transportation agencies, and nine major automobile manufacturers have participated in advancing the initiative [7].

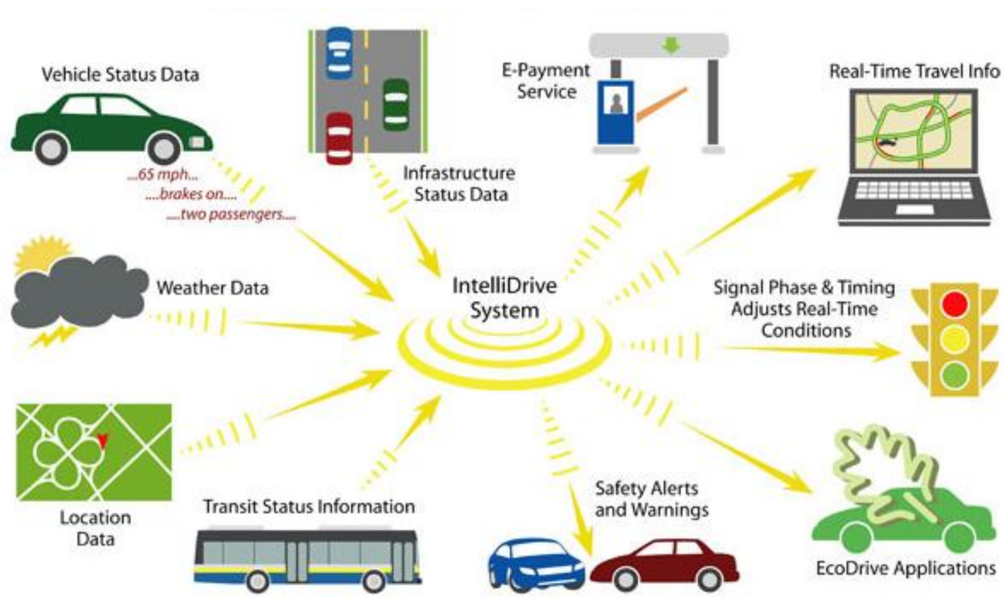

Figure 4 Intellidrive^{SM} Future Vision (source: http://www.its.dot.gov/)

2.3 Parallel and Distributed Simulation

Parallel and distributed simulation refers to technologies that enable a simulation model to execute on multiple processors [45]. Its benefit includes reduced execution time, larger model scale, and integration with other simulators. Parallel and distributed simulations can be distinguished by the geographical distribution, the composition of the processors used, and the network to interconnect the processors. While the processors in a parallel simulation are homogeneous machines and located in close physical proximity, the processors in distributed simulation are often composed of heterogeneous machines that may be geographically distributed (Table 1). For communication between processors, parallel simulation uses customized interconnection switches and distributed simulation utilizes widely accepted telecommunication standards including LAN (Local Area Network) and WAN (Wide Area Network) [45].

Table 1 Parallel and Distributed Computing [45]

	Parallel	Distributed
Physical Extent	Machine room	Single building to global
Processors	Homogeneous	Often heterogeneous
Communication Network	Customized switch	Commercial LAN or WAN

When a simulation program is distributed over multiple processors in parallel and distributed simulation, a number of LPs (logical processes) execute simulations concurrently. In such simulations time stamp ordered processing is not guaranteed, as in a sequential execution on a single machine. Errors resulting from out-of-order processing are referred as causality errors. Out-of-order execution must be prevented to ensure the parallel and distributed simulation produces the same results as a sequential execution. To avoid causality errors, synchronization algorithms are required which refer to the coordination of simulation processes in a time stamp order to complete a task. Under the synchronization algorithms, LPs execute simulations while obeying a rule known as Local Causality Constraint (LCC). Two different synchronization approaches have been proposed to satisfy the local causality constraint, conservative execution and optimistic execution [46-50]. LPs in conservative synchronization protocols strictly avoid violating LCC. Each LP only advances when it is safe to proceed after satisfying LCC. However, optimistic algorithms assume "optimistically" that there are no causality errors and allow LPs to process asynchronously. LCC violation can occur, since optimistic execution does not determine when it is safe to proceed for each LP. Instead, when a causality error is detected, a mechanism to recover is provided in the optimistic approach. Once a causality error is detected simulation states prior to the causal violation are recalled and the simulation is executed forward from that state, with the LCC violation corrected.

The operation of recovering a previous state is known as a rollback and this recovering process requires state saving and anti-message. State saving stores state variables values prior to an event computation. Two widely used techniques for state saving are Copy state saving and Incremental state saving. Copy state saving creates an entire copy of the modifiable state variables, whereas Incremental state saving records 1) the address of the state variables that was modified and 2) the value of the state variable prior to the modification. If a small number of state variables are changed, incremental state saving is more efficient, reducing the time and memory overheads. However, incremental state saving does not perform well when most of the state variables are modified by each event. Infrequent state saving is an alternative to reduce the overheads by decreasing frequency of LP state-saving [51]. When a rollback event happens, the simulation state being rolled back may have sent messages which are not consistent with the rolled back state. Those messages have to be annihilated or cancelled in the anti-messaging process [45].

2.4 Parallel and Distributed Simulation in Traffic Simulation

Among many possible ways of dividing a large scale simulation over different processors, two approaches are popular; 1) task parallelization and 2) domain decomposition [21, 27]. MITSIM, DynaMIT, and DYNASMART are utilizing task parallelization for faster processing [52-54] and different modules of a traffic simulation package (vehicle generation, signal operation, routing, etc.) are assigned to different computers in those models. This approach is conceptually straightforward and fairly insensitive to network bottlenecks. On the other hand, domain decomposition is splitting a simulation with respect to time or space. For time decomposition, the domain is partitioned into a number of time intervals and each processor is responsible for running simulation of an assigned time interval. Space decomposition is more popular for traffic simulation. In this scheme, a simulation network is divided into multiple sub-networks and each sub-network is assigned to a different machine.

Several traffic simulation models have implemented this domain decomposition approach to split computational loads over different computers in order to achieve fast running speed. The models include Transportation Analysis and Simulation System (TRANSIMS) [55-57], Advanced Interactive Microscopic Simulator for Urban and Non-Urban Networks (AIMSUN) [58, 59], and Parallel Microscopic Simulation (Paramics) [60, 61].

Table 2 Parallel and Distributed Computing in Traffic Simulation

	Parallelization Type	Parallelization Detail
MITSIM	Task Parallelization	MITSIM and traffic management simulator Master controller is used to synchronize the execution of all modules.
DynaMIT	Task Parallelization	Demand simulator estimates O-D flows. Supply simulator represents mesoscopic traffic network model.
DYNASMART	Task Parallelization	Different modules are deployed on a distributed computational platform using the CORBA architecture.
TRANSIMS	Domain Decomposition	Each CPU is responsible for a different geographical area of the simulated region.
AIMSUN2	Domain Decomposition	Network is partitioned into blocks and layers.
PARAMICS	Domain Decomposition	Network is divided into several regions and run simultaneously with synchronization algorithm.

TRANSIMS is an agent-based transportation forecast model developed by Los Alamos National Laboratory. It is a micro-simulation based model utilizing cellular automata (CA) approach to simulate second-by-second movements of every vehicle in a large metropolitan area. In TRANSIMS the network can be partitioned into tiles of similar size and boundary information is exchanged between processors for global synchronization [55-57]. AIMSUN2 is a microscopic simulation program originally developed as a sequential version, but later parallel computing architectures AIMSUN2/MT (the multi-thread parallelized AIMSUN2) were added. For distributed simulation implementation, a network is divided into layers, blocks, and entities. AIMSUN simulates each vehicle based on lane changing and car following model at the level of the entity (section and junction entity) which are updated at every time step. Entities updated together are grouped into blocks that may be allocated to a single thread. Blocks which need to be updated simultaneously by threads are grouped into a layer. Threads can be executed in parallel by the multiple machines. It was reported that the parallel AIMSUN2 operating on a SUN SPARC station with four processors completed a network simulation consisting of 561 sections and 428 junctions 3.5 times faster than its sequential version [58, 59].

PARAMICS (PARAllel MICroscopic Simulation) is a suite of microscopic traffic simulation tools. Cameron et al. [60, 61] implemented data parallel programming in Paramics. In their study, multiple simple processors connected in a tightly coupled network executed the same code while having their own input data (Single Instruction Multiple Data). Researchers at the National University of Singapore [20] divided the whole network and each sub network was dedicated to a different processor. To maintain the spatial connectivity between regions simulated separately vehicles were transferred to the next processor when they cross the network boundary. The method was implemented on a hypothetical grid-type network with over 150 sq. km, 500 nodes, 1000 links and 72 signalized intersections. Their results showed speed increase from 1.50 to 2.25 times when using two processors and from 1.75 to 3.75 times when using three processors, compared with the speed of simulation without parallel execution. Liu et al. [21] have developed a distributed modeling framework with low-cost networked PCs. Windows Sockets were used as the communication middleware to transfer vehicle information and synchronize the simulation time between the client controller and server simulators. Researchers at the University of California, Irvine [23] developed ParamGrid as a scalable and synchronized framework. They distributed the simulation across low-cost personal computers (PCs) connected by local area network (LAN). A large traffic network was divided into a grid of smaller, rectangular sub-networks. Each sub-network was called a tile and ran on a single-process simulator on a single PC. They developed methodologies to transfer vehicles across tiles and synchronize the simulation time globally using CORBA middleware. They found the simulation performance increased approximately linearly with the number of added low cost processors.

Bononi et al. [25, 26, 62, 63] proposed Mobile Wireless Vehicular Environment Simulation (MoVES) as a scalable and efficient framework for the parallel and distributed simulation of vehicular ad hoc networks. MoVES was implemented on the ARTIS (Advanced RTI System) simulation middleware which partially adopted the High Level Architecture (HLA) standard IEEE 1516 and supported conservative time management based on time-stepped approach. They developed solutions for communication overhead reduction and computational/communicational load balancing. Their vehicular model followed a microscopic approach including car following model. However, lane changing policies were not implemented. Their performance analysis demonstrated that MoVES had better performance in scalability, efficiency, and accuracy.

2.5 Optimistic Execution in Traffic Simulation

In the field of computer science, both conservative execution and optimistic execution have been well-researched. However, only conservative execution is employed for the most distributed traffic simulation works, since simulation state saving is not available and additional overhead computation is not supported for most of commercial traffic simulation packages. In the literature reviewed it appears that researchers at Oak Ridge National Laboratory made the first attempt at applying optimistic simulation techniques to parallel vehicular network simulation [64, 65]. They developed a parallel vehicular traffic simulation model called SCATTER-OPT, standing for an optimistic-parallel version of the SCATTER simulation system, to reduce execution time for simulating emergency vehicular traffic scenarios. A simplified traffic model was used in their work. For example, the road network was modeled as a graph representing road segments and intersections. Each road segment was modeled with a few basic attributes (number of lanes, length of road segment, speed limit, and traffic lights). They considered a constant time of 1 second as the time required for a vehicle to cross any intersection. They compared the simulation runtime of OREMS (Oak Ridge Evacuation Modeling System) and SCATTER-OPT (with one and two processors) on the same 16×16 road network to demonstrate the absolute speedup of the SCATTER-OPT. Also, both optimistic and conservative synchronization techniques were tested with different numbers of processors, three different vehicular networks (64×64, 128×128, and 256×256) and different simulation parameters (lookahead values). It was observed that optimistic synchronization preformed well with increasing network size and decreased amount of lookahead. For the largest network size (256×256) a speedup of nearly 20 was recorded with 32 processors. They concluded that in modeling vehicular traffic network, where the lookahead is not fixed, optimistic synchronization (reverse-computing) provides a better promise for timely simulation results.

2.6 Summary

This chapter reviewed the previous research regarding vehicular ad hoc network and parallel and distributed simulation technologies associated with vehicular ad hoc network. Research on VANET has been actively conducted worldwide for various online simulation applications collision avoidance, traffic prediction, route planning, traffic management, and signal timing. In order to run a large simulation with fast speed, parallel and distributed simulation has been utilized in transportation area. Generally, parallel and distributed simulations are differentiated by the geographical distribution, the composition of the processors used, and the network to interconnect the processors. Also, two popular approaches for the synchronization were discussed; 1) conservative time synchronization and 2) optimistic time synchronization.

In transportation area, previous research efforts to divide a large scale traffic simulation over different processors can be classified into two approaches; 1) task parallelization and 2) domain decomposition. Most of the researches in traffic simulation followed the conservative time synchronization. However, in this conservative time synchronization approach, speed of the entire simulation is dependant on the slowest processor, since all processors need to be synchronized with respect to simulation time. To evaluate the potential of concurrent simulation run by geographically distributed heterogeneous processors, this dissertation proposes an ad hoc distributed approach based on optimistic time synchronization. In this optimistic time synchronization approach, geographically-distributed heterogeneous processors are allowed to run concurrently while obeying LCC.

CHAPTER 3 AD HOC DISTRIBUTED SIMULATION MODEL

This study attempts to integrate distributed traffic simulations with wireless technology and build a data dissemination framework in VANET environment. This distributed simulation environment is referred to as an online ad hoc simulation. The following sections discuss the proposed online ad hoc distributed traffic simulation model. In Chapter 8 it will be seen that this approach may be extended to introduce a real time field data driven simulation client allowing for real time state estimate of the roadway network. In a field implementation this real time field data driven simulation client would be replaced with the streaming detector data

First, the overall system is represented. Second, the physical operating platform for the model is described. In this description, detailed information about operating system, communicational middleware, and traffic simulation model is included. The communication process and its message structure are demonstrated. Two major components of the initial algorithmic approach to the ad hoc distributed simulation; global process and logical process, are proposed in Section 3.4 and 3.5, respectively. Three main functions of the global process; data aggregation, rollback detection, and anti-messaging are illustrated. Then, details about the proposed logical process operation are explained in four subsections; traffic simulation, estimate, state saving and traffic update when rollbacks occur.

3.1 Model Overview

An ad hoc distributed simulation is a set of interacting online simulations that collectively predict future states of a physical system. Each LP receives information concerning the current state of the system from one or more sensors as well as estimated future system states from other LPs, and generates estimated future states of some portion of the physical system. For example, as shown in Figure 5, one LP might model some set of road segments and intersections, receive vehicle flow rates on links carrying vehicles into the region modeled by the LP, and predict vehicle flow rates on links carrying vehicles out of that region. The LPs collectively model the larger transportation system covered by all the participants.

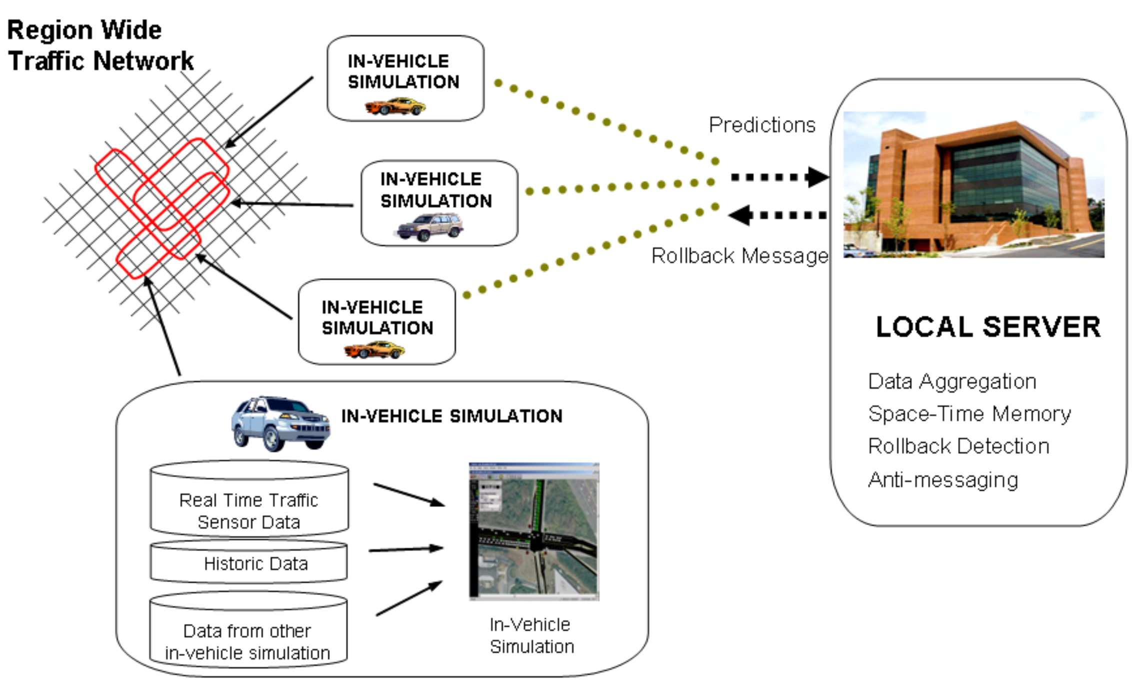

Figure 5 Ad Hoc Distributed Traffic Simulation

The region modeled by each LP is determined by the LP itself. In this sense, the overall distributed simulation consists of an "ad hoc" collection of LPs. In general, a specific road segment will be modeled by multiple LPs. The state estimates produced by the different LPs must be aggregated, and the aggregated value transmitted to other LPs that utilize this state information as input. In the ad hoc distributed simulation approach the LPs operate in an asynchronous fashion, that is, LP is not required to operate in time synchronous lock step with other participating LPs, allowing for largely autonomous operation.

The proposed ad hoc distributed simulation model provides transportation network monitoring and near term predication of the system where embedded, LPs are combined with information servers and simulations running within the roadside infrastructure. In the proposed implementation each LP represents a participating simulator that models the roadway network in the immediate vicinity of the LP. Each LP publishes projections of near term future system states, and utilizes projected state information from other LPs, real time embedded traffic sensor data, and historical traffic behavior patterns. This state information is saved and managed in Space-Time Memory inside the server. Based on an approach inspired by the Time Warp algorithm [47] the server aggregates projected state information from LPs, detects rollbacks, and processes anti-messages, while traffic simulation, estimate, state saving and traffic update occur in the logical process level as illustrated in Figure 6.

As seen in Figure 5, the LPs within the transportation network may cover overlapping areas. This is a distinct difference from conventional distributed simulation where simulated areas are commonly partitioned into non-overlapping sections, and an

LP is mapped to each one. An additional characteristic unique to an ad hoc distributed simulation with mobile simulator platforms (e.g. in-vehicle simulators) is that the network area modeled by an LP can vary over time, for example, as the vehicle traverses the network the area that it models may change. Finally, the set of participating LPs may be dynamic as new LPs can join and existing LPs leave during the analysis period.

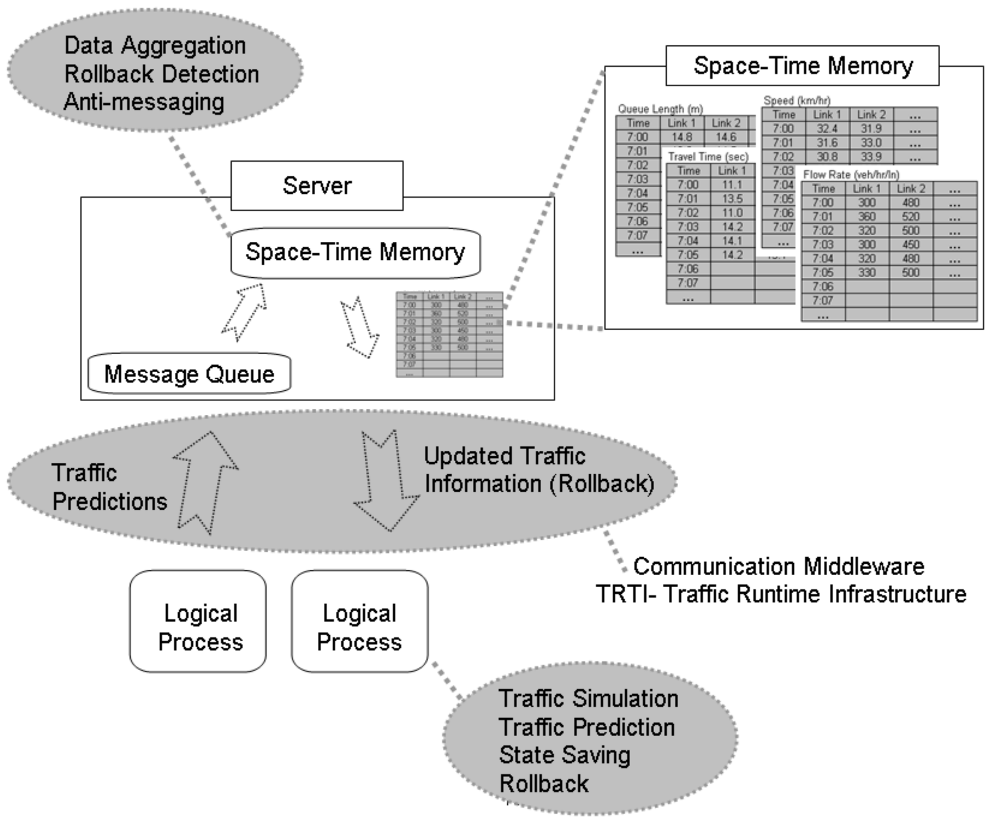

Figure 6 Server and Logical Process Data Process Map

An advantage of the ad hoc distributed simulation approach is that an embedded distributed simulation operates in close proximity to the real time data, allowing near term estimates to be based on detailed, up-to-date data collected from nearby sensors. In addition, as mentioned earlier it is anticipated that multiple vehicles will be simulating overlapping areas, resulting in significant redundancy, offering the potential for greater robustness and resilience to failures.

3.2 Running Environment

The following section discusses the physical operating platform for the proposed development, the communication middleware, and the individual simulation instance platform (VISSIM®).

3.2.1 Experimental Platform

The model is comprised of one server and multiple LPs (logic processes). Each LP represents an in-vehicle simulator. To provide for realistic testing each LP uses a separate laptop computer. All computers are equipped with a middleware communication program (TRTI: Traffic Runtime Infrastructure) and a simulation script coded in Microsoft Visual .NET language. The script controls the traffic simulation (VISSIM®) execution (e.g. advancement of time steps, rollback implementation, etc.) and aggregation of simulation output while the middleware facilitates communication between the server and other LPs. The area modeled by each LP covers only a small portion of the overall network. The simulation results, after some aggregation to be discussed in a subsequent section, are sent to the server. A detailed architecture of the model is depicted in Figure 7.

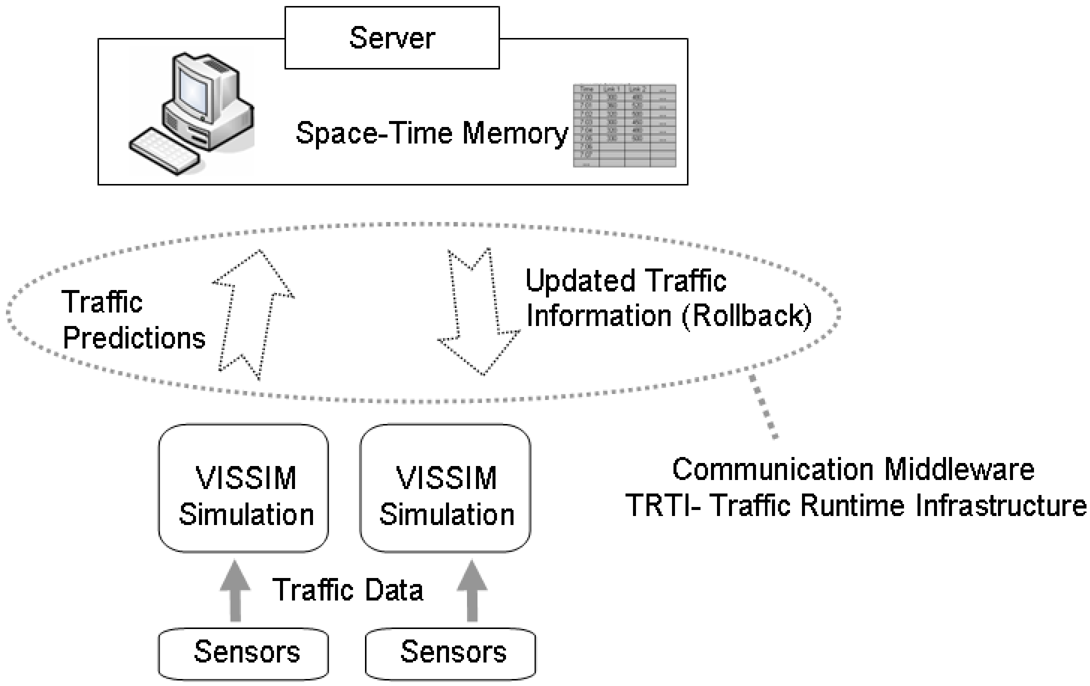

Figure 7 System Architecture

3.2.2 Communication and Communication Middleware

Distributed simulation provides better usability, flexibility, and capability in a large scale microscopic traffic simulation than centralized simulation. However, it requires an object-oriented system with client/server technology to handle the complexity of its application. This problem can be managed by communication middleware, which refers to a layer of software above the operation system API (Application Programming Interface) between platforms and applications. Middleware runs on multiple platforms and supports standard interfaces and protocols. It provides a higher level building block than API to manage the complexity and heterogeneity inherent in distributed systems [49, 63, 66-68]. Several middleware technologies are available for VANET simulation, including Windows Socket (Winsock), Remote Procedure Call (RPC), and Object Request Broker (ORB) including Distributed Component Object Model (DCOM), Common Object Request Broker Architecture (CORBA) [68-72].

High Level Architecture (HLA), a standard (IEEE 1516) is a distributed simulation architecture developed by the U.S. Department of Defense. It supports simulation reuse and ensures interoperability between heterogeneous distributed simulation system platforms [46, 73, 74]. Communications are available with other computers regardless of the computing platforms and all communications between the units of software reuse, called Federates, are accomplished via a distributed middleware called RTI (Run-Time Infrastructure). RTI is a communication module designed to provide a clean API to application developers while adhering to the rules to HLA. Each federate uses its own local copy of RTI software library for communication and manages global state of communicating federates by RTI [63].

TRTI is utilized as a HLA inspired middleware. TRTI has been developed by Georgia Insitute of Technology research team and it employs TCP (Transmission Control Protocol) and UDP (User Datagram Protocol) as the protocol for communication between computers. LP initializes TRTI with the network information of the communication destination (server IP address) and registers the local federate with TRTI through a message handler function. Once a connection between the server and the LP is established, any messages can be transmitted using the TRTI API over the existing connection. In this study, LP to LP communication is not considered, only communication between server and LP is investigated. For each LP incoming messages are queued by TRTI in the order in which they are received. Microsoft Visual .NET application in each LP can process the messages by calling the TRTI function to get the messages from the queue. Apart from communications, TRTI also keeps track of the list of registered groups by a Federate at the server and can send rollback messages to designated LPs registered to particular groups. For the communication, the Georgia Tech Local Area Wireless Network (LAWN) is utilized for wireless communication. LAWN is a campus-wide local-area network.

3.2.3 Traffic Simulation

For the optimistic distributed approach a traffic simulation model should be capable of producing interim simulation data and simulation state saves during runtime. Very few commercial microscopic traffic simulation models offer these features. VISSIM® is a commercial simulation package capable of producing simulation results and runtime state saves. VISSIM® is a discrete, stochastic, time step based microscopic simulation model. This behavior-based multi-purpose traffic simulation program has been developed to model a wide range of traffic conditions including freeway, arterial, and public transit operations. In this model all vehicles are modeled individually, based on a psycho-physical driver behavior model developed by Wiedemann [75]. The basic assumption of this model is that a driver can be in one of four driving modes: free driving, approaching, following, or braking. Access to VISSIM® simulation data and simulation states is available through the COM (Component Object Model) interface, which allows developers to import the objects and properties during runtime. The VISSIM® COM interface can be operated through computer languages including Visual Basic, Visual

C++, and Java. The snapshot function in VISSIM® saves and restores current simulation state. Detailed traffic information is saved in a snapshot file including location, speed, and acceleration of each vehicle on the network and the state of all traffic control devices. Through the VISSIM® COM interface it is possible to run simulations while saving the simulation state periodically, stop the simulation, and restore one of the saved past states to resume with different input parameters. VISSIM® 5.1 is used in this study.

3.3 Data Communication

As stated the TRTI is used for the data exchange including traffic estimates and rollback messages. The exchange is accomplished based on data packets which are transmitted in form of radio broadcasts. Details about the data are described in the following section. Also, the following assumptions are made on the platform.

1. Messages are not lost during communication.
2. Messages are received in the order sent.
3. Server and LPs have sufficient buffers to handle the message queues.

3.3.1 Data from Server to Logical Process

The server sends a message to LPs on three different occasions; 1) send rollback information (described below), 2) send a message to end the current simulation run, and 3) send a message to start a new simulation run. Message “7777” is used to inform LPs to close the current simulation run and “9999” to start a new simulation run. While contents of the messages to end or start a simulation run are very simple, each rollback message contains 22 characters with the following traffic information.

AAAABBBBBBCCCCCCDDDEEE – 22 character message structure

where: AAAA is the rollback logical process ID: 4 characters (starting from 0001)
BBBBBB is the rollback link number: 6 characters
CCCCCC is the rollback simulation time: 5 characters (starting from 00001)
DDD is the average speed: 3 character (0.1km/hr) (starting from 000)
EEE is the average flow rate per hour per lane: 3 characters (starting from 000)

Each message is sent using the below TRTI function call with declaration of the group, destination (IP address of logical process), and rollback information.

TRTI_sendMsgToGroupAt(group, LP, message(AAAA BBBBBB CCCCCC DDD EEE))

3.3.2 Data from Logical Process to Server

On a periodic basis (every 1 simulation minute in this study) each LP collects its traffic estimates and sends them to the server. Two different delivery methods are considered, 1) sending a separate message for each link and 2) sending one message including all link data. While the first method is very simple and straightforward, it requires numerous TRTI function calls. On the other hand, the message size in the second method becomes larger and extra computational loads are necessary to break down the message on both server and LP. However, less frequent TRTI calls significantly reduce the communication load resulting in simulation speed increase. Each message starts with a logical process ID, run number, and simulation time. Link characteristic, link ID, and traffic estimates for each link are followed. Traffic estimates include speed, flow rate, travel time, delay, and queue length. The structure of each message is as follows;

All link data in one message

AAAA BBBBBB CCCCC (D EEEEEE FFF GGG HHH IIII JJJJ)……

where: AAAA is the logical process ID: 4 characters (starting from 0001)

BBBBBB is the run number: 6 characters (starting from 000001)

CCCCC is the simulation time: 5 characters (starting from 00001)

For every link in the network

D is the link characteristic: 1 character (1-inbound link, 2-outbound link, 3-internal link)

EEEEEE is the link number: 6 characters

FFF is the 4 minute average speed: 3 character (0.1km/hr) (starting from 000)

GGG is the 4 minute average flow rate per hour per lane: 3 characters (starting from 000)

HHHH is the 2 minute average travel time: 4 characters (starting from 0000)

IIII is the 2 minute average delay: 4 characters (starting from 0000)

JJJJ is the 2 minute average queue length: 4 characters (starting from 0000)

Each LP delivers a message to the server calling the following TRTI function with group, destination (IP address of server), and message information.

TRTI_sendMsgToGroupAt(group, server, message(AAAA BBBBBB CCCCC D EEEEEE…)

3.4 Global Process

As seen in Figure 5, the LPs within the transportation network may cover overlapping areas. Also, the network area modeled by an LP can vary over time, for example, as the vehicle traverses the network the area that it models may change. Additionally, the set of participating LPs may be dynamic as new LPs can join and existing LPs leave during the analysis period. Therefore, it is necessary to create a database to store global predications in the Space-Time Memory, which is accomplished through the data aggregation algorithm, demonstrated in this section. Also, as described in Section 2.3, rollback and anti-messaging process is required to manage the Space-Time Memory. Details about the three Global Process (Server) functions, data aggregation, rollback detection, and anti-messaging are described in 3.4.2 thru 3.4.4.

3.4.1 Simulation Time and Wall-clock Time

Before describing the Global Processes, it is necessary to review fundamental terminologies used in simulations to refer different notions of time. The following provides definitions of “Simulation Time” and Wall-clock Time”, which are used in the remainder of the dissertation.

- *Simulation Time* is “an abstraction used by the simulation to model physical time” [45].
- *Wall-clock Time* refers to “time during the execution of the simulation program” and “A Simulation program can usually obtain the current value of wall-clock time by reading a hardware clock maintained by the operating system” [45].

To better illustrate the differences, suppose a traffic simulation of Metro Atlanta at traffic management center. At 7:00AM wall-clock time, the center is predicting traffic states and its simulation model can run at the speed of 30 minute simulation time at 1 minute wall-clock time. Therefore, the center's estimates are available until 7:30AM simulation time at 7:01AM wall-clock time. One minute later, at 7:02AM wall-clock time, the estimates reach 8:00AM. "Real-time Factor" / "Time Scale Factor", which is defined as the ratio of the simulation time to the time of the real process is 30 in this example [20, 45].

Additionally, "simulation executions where advances in simulation time are paced by wall-clock time" are referred to as "real-time execution" and simulators running in this rule are called "real-time simulators" [45]. In this case, "Real-time Factor" / "Time Scale Factor" is 1.

3.4.2 Data Aggregation

Ad hoc distributed simulation is a collection of logical processes, LP_1, LP_2, LP_3, … , LP_n, which share a global state G that contains object instances G_1, G_2, G_3, … , G_m. The global object instances are saved in Space-Time Memory (STM) inside the server (Figure 7), and synchronized in an optimistic fashion. All the notations described in this study are summarized in Table 3.

Once LP_i publishes $LP_{i,j,k}^{m,p}$ which denotes local state estimates of LP_i on link j at simulation time k to the server (m; data type and p; link type), the estimates are transferred from LP_i to the message queue located inside the server. The message queue contains traffic data of different links at different simulation times from multiple LPs in the order of time when the message is received. The server processes messages from the message queue in FIFO (first in first out) order by local-to-global transition function $\{f\ (\cdot)\}$. This function converts the local state estimates $LP_{i,j,k}^{m,p}$ to global variables. In other words,

$$g_{i,j,k}^{\mathrm{m,p}} = f(LP_{i,j,k}^{m,p})$$

where, $g_{i,j,k}^{m,p}$ represents global variables on link j at simulation time k generated by LP_i with m as data type and p as link type.

When the server receives any data from LP_i, the composition function $C\ (\cdot)$ aggregates the values of $g_{i,j,k}^{m,p}$ into one global instance $G_{j,k}^{m}$. Specifically,

$$G_{j,k}^{\mathrm{m}} = C\ (g_{i,j,k}^{m,p})$$

For example, global state instances; $G_{j,k}^{FlowRate}$, $G_{j,k}^{Speed}$, $G_{j,k}^{TravelTime}$, $G_{j,k}^{Delay}$, and $G_{j,k}^{QueueLength}$ can be calculated based on the set of $g_{i,j,k}^{m,p}$.

$$G_{j,k}^{FlowRate} = \frac{\sum_i g_{i,j,k}^{FlowRate,Internal}}{n}$$

where, n represents all available number of estimates $g_{i,j,k}^{FlowRate,Internal}$ of link j as an internal link at simulation time k.

Table 3 Local and Global Process Notation Summary

Symbol	Description
LP_i	i th Logical Process
$LP_{i,j,k}^{m,p}$	Local estimate of LP_i on link j at simulation time k with data type m (flow, speed, travel time, delay, or queue length) and link type p (inbound, internal, or outbound)
$g_{i,j,k}^{m,p}$	Global variable generated by LP_i on link j at simulation time k with data type m (flow, speed, travel time, delay, or queue length) and link type p (inbound, internal, or outbound)
$G_{j,k}^{m}$	Global state G on link j at simulation time k with data type m (flow, speed, travel time, delay, or queue length)
$G_{j,k,l}^{m}$	Global state G (based on estimates from LP) on link j at simulation time k at wall-clock time l with data type m (flow, speed, travel time, delay, or queue length)
$RollbackThreshold$	Rollback threshold (Flow rate)
$SpeedThreshold$	Speed threshold

It is noted that in the calculation of global state instances that only data from LP internal simulation links is utilized. Inbound and outbound link data is excluded from the aggregation process as they may poorly represent the actual traffic conditions. For example, inbound link traffic performance (travel time, delay, and queue length) may not be accurately modeled as the vehicle arrival headway distribution at the entry point of inbound link may differ from the real traffic pattern on the link. For example, entry link data will not reflect platoon characteristics of arriving vehicles due to upstream intersections not reflected in the model. When considering outbound links it is noted that vehicles may exit the outbound link regardless of the actual traffic conditions of the link.

For instance, when traffic constraints outside the boundaries of the LP simulation result in a spillback of congestion into the region being modeled this spillback will not be reflected in the model. As they have no knowledge of downstream traffic condition, vehicles on the LP simulation may exit the link at free flow speed, providing inaccurate traffic estimates. As discussed later, to address this situation, outbound link speed is controlled to meter the outflow rate from the LP simulation model. Thus, upstream internal link behavior will reflect the spillback due to a bottleneck outside the modeled area however the outbound link itself is being artificially manipulated to capture this impact, resulting in its data not being suitable for the global aggregation. More details are discussed in later section on how to represent these intermitted capacity bottlenecks on outbound links and the link speed selection process to meter vehicles.

The server keeps track of all available estimates. $G_{j,k,l}^{m}$ represents global state on link j at simulation time k at wall-clock time l with data type m. Two attributes of the ad hoc distributed system estimate are considered 1) length of prediction horizon, that is, how far in advance of the current wall-clock time the system provides estimates and 2) how accurate the estimates are at specific prediction horizon, i.e. how accurate is the estimate. For example, the following analysis would be available. Suppose at 7:00AM wall-clock time the system was able to predict until 7:30AM simulation time and its estimates regarding 10 minute period between 7:20AM simulation time and 7:30AM simulation time over-estimated by 15%. However, at 7:10AM wall-clock time with more updated information the system was able to provide the same time period estimates (7:20AM-7:30AM simulation time) with better accuracy (5% difference). Further comparisons between $G^{m}_{j,k,l}$ and the actual traffic state will be conducted to quantify the system's estimate capability in Chapter 8.

Lastly, global-to-local transition function $\{f^{-1}(\cdot)\}$ is called when any LPs need to rollback in order to revise its estimates with updated information. This function $\{f^{-1}(\cdot)\}$ converts the global object instances to local state, i.e.

$$LP^{m,p}_{i,j,k} = f^{-1}(G^{m}_{j,k})$$

Whenever this function is called, $G^{m}_{j,k,l}$, aggregated value from the estimates of the LPs on link j at simulation time k, is converted to the local instance for LP_i . Then, $LP^{m,p}_{i,j,k}$ is utilized as new input to revise its estimates.

3.4.3 Rollback Detection

When the server receives estimates from logical process LP_i it determines whether a rollback should be triggered for any LPs based on rollback detection function $Rollback(\cdot)$. The rollback detection function compares the flow rate estimates of each LP with the corresponding global instances in the Space-Time Memory and decides which LP needs to renew its estimates. Since LPs model their own network portions of interest and their model networks overlap, each link can be simulated by multiple LPs. Also, the link can be an inbound, outbound, or internal link depending on the network configuration of each LP.

Consider link j, a link which some LPs have as an inbound link, some as an outbound link, and some as an internal link of their own network simulations. Whenever there is an update on the global instance, $G_{j,k}$ in the Space-Time Memory, the server checks the difference between $G_{j,k}^{FlowRate}$ and estimates on link j as boundary links (inbound link or outbound link) for the individual LPs, that is $g_{i,j,k}^{FlowRate,Inbound}$ or $g_{i,j,k}^{FlowRate,Outbound}$. If the difference is greater than a given threshold $RollbackThreshold$, then the estimates of the corresponding LP are considered invalid and a rollback is issued from the server.

Consider LPs which have link j as an outbound link. They only model the upstream area of link j, not including downstream area of link j in their network. Since link j is the end link of the network, vehicles on link j exit the network at free flow speed unless there is an outflow constraint. In this case, they may not have a good estimate on link j when the downstream traffic condition outside the boundaries results in a spillback of congestion into the network being modeled. Thus, the traffic condition on link j needs to be adjusted to reflect the spillback traffic condition. On the other hand, LPs which have link j as an inbound link and generate vehicles at a pre-determined flow rate may not represent traffic condition well when a sudden change in incoming traffic is predicted outside of the network boundaries. In this case, input rate on link j is adjusted based on $G_{j,k}^{FlowRate}$ which is included in rollback messages sent from the server.

When the simulation time k is far ahead from the current wall-clock time, $G_{j,k}^{FlowRate}$ is calculated from the LP estimates which are available in the Space-Time Memory at the time period when the server checks $G_{j,k}^{FlowRate}$. Therefore, it is expected that rollback statistics would vary depending on how many LPs are contributing to the aggregated global values. The number of LPs contributing to the aggregated global values is determined by geographical distributions of LP locations. The impact of the geographical distributions of LP locations will be investigated in Chapter 6.

3.4.4 Anti-Messaging

Optimistic synchronization algorithm in an online ad hoc distributed simulation can distribute data through all LPs and allow independent running of LPs. Anti-messaging for invaliding estimates and synchronizing valid estimates are essential for reliable data management and efficient simulation speed. To ensure the accuracy of the global estimates, global instances should be only aggregated using currently valid estimates and invalid estimates should be removed from the Space-Time Memory and its message queue in the server.

If the server detects rollback on LP_i at simulation time k, it means the estimates of LP_i regarding simulation time k and thereafter (for example, $LP_{i,j,k}$, $LP_{i,j,k+1}$, $LP_{i,j,k+2}$, …) are not valid and should be eliminated. The server removes all estimates of LP_i from the simulation time k and thereafter from its Space-Time Memory (where already processed data is saved) and the message queue (where received but not-processed data is located). After removing estimates of LP_i, the server delivers a rollback message to LP_i. The message contains new state value $G_{j,k}$, simulation time, link number and identity of the logical process (see section 3.3.1).

3.5 Logical Process

Optimistic synchronization algorithm in an ad hoc distributed simulation allows each LP to run independently without time synchronization with other LPs. As illustrated in Figure 6, each LP simulates its own network of interest, publishes its traffic estimates, saves its simulation states periodically, and updates its simulation when new information is available. Details about the logical process are described in the following section.

3.5.1 Traffic Simulation

An LP starts its simulation with initial input and updates its input whenever it obtains updated information from available sources. The sources can be 1) projected state information from other LPs through the server in the current approach, 2) real time embedded traffic sensor data, or 3) historical traffic behavior patterns. The input data includes traffic flow rate and average vehicle speed of each entering link. Vehicle generation time on entering links is calculated using the input flow rate and time headway is uniformly distributed in this proposed model. At each time step, each LP checks the next vehicle generation time to decide whether a vehicle needs to be released.

3.5.2 Traffic Estimate

As stated, each LP in an ad hoc distributed simulation runs independently while sending estimates to the server at every given time interval. During LP execution, simulation results (flow rate and average speed of vehicles on each link) are recorded at a pre-determined time interval (1 minute in this study). Then, the LP aggregates the results into an average over a longer time period and saves this to its own Space-Time Memory. Aggregation into a longer time internals prevents rollbacks invoked due to short flow rate fluctuations that result from expected variability in a traffic stream, such as flow fluctuations resulting from an upstream signal.

Regarding the aggregation interval selection, there is no definite regulation, with this being one aspect requiring further study. Smaller time intervals can provide more accurate simulation, since the response time to new traffic information would be reduced. However, the number of rollbacks would also increase, raising the communication load and potentially reducing the simulation speed and shortening the prediction horizon. The solution to this dilemma depends on the objective of the simulation and required accuracy. However, it should be noted that as the time interval becomes smaller than the cycle length of nearby intersections, variation in traffic flow become much more pronounced. Therefore, a shorter time interval may result in continuous back and forth rollbacks between two traffic states (for example, from state A when upstream light is green to state B when upstream light is red and then back from state B to state A).

In this study four minutes is chosen as an aggregation time interval. A four minute aggregation period is considered sufficiently long not to be affected by local signal cycles while capturing flow rate changes within a reasonably small response time. Also, travel time, delay, and queue length are collected every two minutes for each link inside the network. All estimates are aggregated into a single message and sent to the server every minute (see 3.3.2.). The basic operation inside logical process is shown in

Figure 8. Vehicle generation in the network (left) and the work flow of logical process (right) are illustrated.

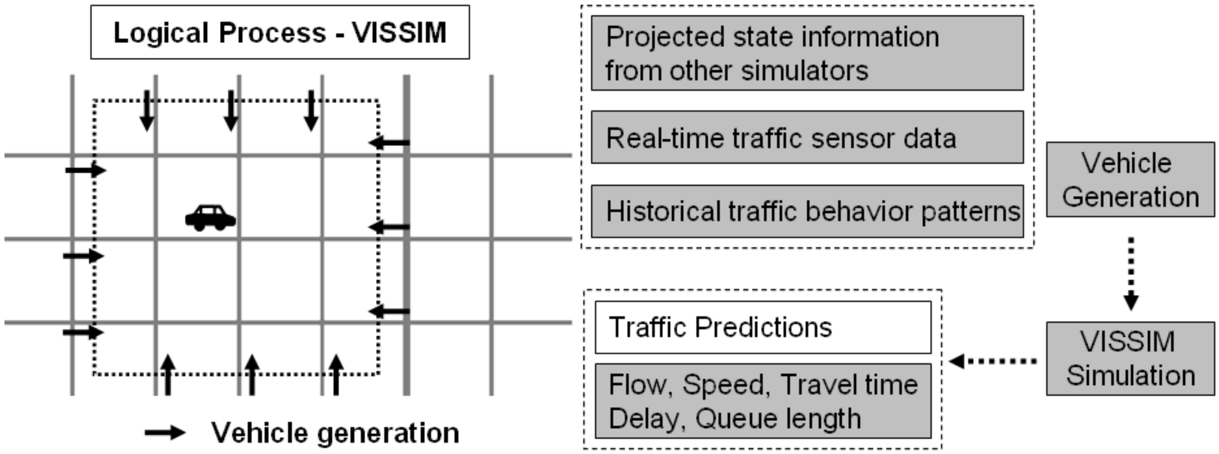

Figure 8 Logical Process

3.5.3 Traffic State Saving

While running its simulation each LP saves its simulation state in its local storage area (a local hard disk of each laptop computer in this study). This allows LPs to roll back to any past simulation time which has been completed and restore the traffic state to resume its simulation with different traffic input parameters. The simulation state saved in its local storage area contains information about all vehicles in the network including speed, acceleration/deceleration, and coordinate. In VISSIM® the traffic state is saved via snapshot file (*.SNP). In this study snapshot files are created at every simulation minute and each file contains simulation time information in its file name. Therefore, traffic state at specific simulation time can be loaded easily when it is necessary. State saving script is as follows;

```
If  SimulationTime Mod 60 = 0 Then
        Simulation.SaveSnapshot(Directory Name\ SimulationTime ".snp")
End If
```

3.5.4 Traffic Update - Overview

In the proposed ad hoc distributed simulation rollback process enables the simulation to adapt to new traffic states and update its own estimates if necessary. The traffic update has two processes, as discussed more in the next section; 1) updating downstream states based on upstream traffic information and 2) updating upstream traffic conditions according to downstream states.

Consider two LPs which are simulating the network regions as shown in Figure 9. LP 1 models the left side of the network (Grey area) and LP 2 simulates the right part of the network (Black dotted box). Each LP starts its simulation using historical average flow rate as an initial input. Suppose that while the two LPs are projecting future traffic states of their own network, the server receives new information from the other LPs. Further suppose that the difference between the flow from the server and input rate of LP 1 on Link A exceeds a given threshold at some wall-clock time (either present or future estimate). The server will detect a rollback and deliver a rollback message which contains rollback logical process ID information, rollback link number, rollback simulation time, new average link speed, new average flow rate, as described in section 3.3.1. Once LP 1 receives this rollback message, LP 1 will update its simulation with this new flow rate by undertaking a rollback and publishes new traffic estimates on the links inside its network to the server. These updated estimates from LP 1 will be transmitted to the server and the Space-Time Memory in the server will be updated accordingly.

To provide additional detail the following is a specific potential example of the preceding general discussion. Assume at wall-clock time 7:00AM, LP 1 starts its simulation based on 300 veh/hr/ln input flow rate on Link A. At wall-clock time 7:10AM, its prediction horizon extends to 8:00AM simulation time. However, new information arrives to the Space-Time Memory in the server at wall-clock 7:10AM forecasting that a 600 veh/hr/ln input rate on Link A is expected at 7:25AM simulation time (15 minute future from the current wall-clock time 7:10AM). The server compares this new 600 veh/hr/ln input rate with the initial 300 veh/hr/ln input rate which LP 1 has reported to the server and was saved in the Space-Time Memory. Since the 300 veh/hr/ln difference exceeds the assumed current threshold and this new 600 veh/hr/ln input rate is regarded as a valid data, the server issues a rollback to LP 1 and sends the new traffic information. Immediately after receiving the rollback message, LP 1 restores its 7:25AM simulation state and continues to renew its estimates of 7:25AM simulation time and thereafter with the updated input data at the current wall-clock time 7:10AM. Two minutes (wall-clock time) later, at wall-clock time 7:12AM, LP 1's prediction horizon reaches 7:35AM simulation time and its updated estimates are sent to the server. After updating its Space-Time Memory with the new estimates, the server checks if there are any threshold violations. At the current wall-clock time 7:12AM, the server realizes that increased traffic volume is expected to reach Link B at 7:35AM simulation time and the flow rate difference between the estimates of LP 1 and input rate of LP 2 surpasses the threshold, causing a rollback on LP 2. With the same method, the server sends a rollback message to LP 2 regarding new traffic information at 7:35AM simulation time (23 minute future from the current wall-clock time 7:12AM). After updating its input data of 7:35AM simulation time at the 7:12AM wall-clock time, LP 2 will continue its simulation and send new estimates of 7:35AM simulation time and thereafter. The server will update its Space-Time Memory and check the rollback violations every time it receives estimates from any LP.

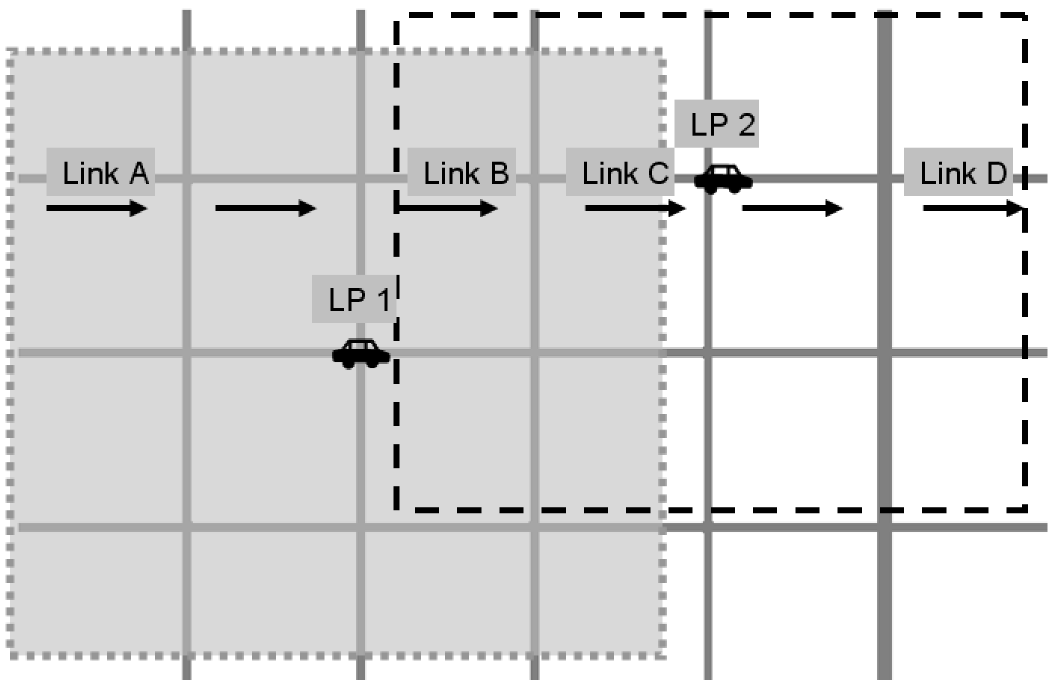

Figure 9 Two Logical Process Example

In this case, at 7:10AM wall-clock time LP 1 is able to update its 7:25AM simulation before the actual volume increase "actually" reaches its modeling area at 7:25AM wall-clock time. Similarly, LP 2 renews its simulation based on new information when it receives the updated traffic state of 7:35AM simulation time at 7:12AM wall-clock time (23 minutes before the new traffic condition "actually" arrives in the area where LP 2 models). This chain of rollbacks between LPs allows other LPs to obtain information about future traffic state changes before they "actually" occur. From a system perspective LPs in the entire network share the most reliable and up-to-date information, even though they are spatially separated from each other. Their estimates are constantly updated through rollbacks to reflect any traffic changes which warrant a threshold violation. Details of updating traffic information are described in the following sections.

A) Traffic update selection

As described in Section 3.4, there are two different types of traffic update when there is a rollback. First case is when changes in traffic conditions outside the boundaries of the LP simulation result in a significant increase or decrease in the entering flow rate. In this case, upstream traffic information needs to be transmitted to downstream LPs to update their traffic input rates. Secondly, there is a case where traffic constraints outside the boundaries of the LP simulation result in a spillback of congestion into the region being modeled. To address this situation, outbound link speed is controlled to meter the outflow rate from the LP simulation model. Thus, upstream internal link behavior will reflect the spillback due to the bottleneck outside the modeled area. The next two sections present the two different traffic update process needed to implement these cases.

B) Traffic update – upstream to downstream

Suppose the traffic condition of the network in Figure 9 is uncongested. Assume that while the two logical processes, LP 1 and LP 2, are simulating the future traffic states of their local network, LP 1 receives new traffic information from the server regarding a sudden influx of eastbound traffic on Link A. LP 1 corrects its simulation with the new information resulting in higher outflow rate on Link B. For example, the new flow rate on Link B from the LP 1 simulation is 600 veh/hr/ln and the input flow rate on the link of LP 2 is 300 veh/hr/ln with 100 veh/hr/ln as the rollback threshold. Thus, there is a threshold violation on Link B. In this case, upstream LPs' estimates are considered valid, since they may have captured a sudden change of upstream flow rate. Whenever there is a threshold violation, the server instructs LP 2 to correct its simulation with the data given by LP 1. After receiving new data from the server regarding the traffic state at simulation time T, LP 2 recalls the past traffic state, resets the traffic flow on the Link B by updating the input vehicle headway (600 veh/hr/ln, one vehicle at every 6 seconds), and sends the traffic estimates from simulation time T to the server accordingly.

C) Traffic update – downstream to upstream

Assume a traffic incident occurs on Link D resulting in arrivals to Link C exceeding possible departures. This results in congestion (i.e. queued vehicles unable to be served) spreading outward from Link D. LP 2 would receive new traffic information from the server indicating the congestion – i.e., significantly reduced traffic volume with very low speed (below SpeedThreshold in Table 3). LP 2 corrects its simulation to better represent the new traffic conditions. For example, assume the incident occurred at 8:00AM wall-clock time. LP 2 would update its simulation shortly after 8:00 AM wall-clock time depending on detection technology available. This updated LP 2 simulation would predict that flow rate on Link C at 8:20AM simulation time would be reduced to 100 veh/hr/ln from 300 veh/hr/ln due to the congestion. Without any information regarding the downstream incident, LP 1 predicts the outflow flow rate on Link C to be 300 veh/hr/ln in its model. Since the flow rate difference exceeds the given threshold, LP 1 needs to match its outflow rate to the downstream estimates (100 veh/hr/ln level for its 8:20AM simulation time traffic estimates).

Updating the LP 1 simulation to reflect this congestion is a non-trivial problem. In uncongested conditions, the upstream flow rate can be easily reproduced by changing vehicle headway on the entering link upon which vehicles are released into downstream LPs. However, changing headway is not an option in congested conditions, as the constraint occurs on an exit link. Unfortunately the currently simulation model (VISSIM®) does not have a way to directly reduce potential flows (i.e. reduce capacity) on an unrestricted link. In this study, the outflow rate is controlled by changing the speed of vehicles on the exiting link. If the server recognizes a difference in flow rate (300 veh/hr/ln for LP 1 and 100 veh/hr/ln for LP 2 on Link C) is over the threshold, a rollback message will be sent to LP 1 to lower the outflow rate to 100 veh/hr/ln. LP 1 applies a sufficiently low speed on vehicles on Link C to produce the same flow rate with LP 2. This leaves the question concerning what speed is required to create the appropriate flow constraint on the exit link of LP 1. For this study, the necessary speeds for various desired flow rates have been estimated based on an empirical analysis of VISSIM® model performance. A graphical analysis regarding speed selection will be presented in Chapter 4.

D) Traffic update – Summary

The purpose of having two different traffic updating methods is to allow for maintaining the same flow rate between upstream LPs and downstream LPs and the transmission of accurate traffic conditions to other LPs beyond the network boundaries of the LPs. These updates keep the flow rate difference between LPs within prescribed threshold. Eventually all LPs will be able to capture dynamically changing traffic conditions and provide reliable system-wide traffic estimates by aggregating estimates generated by LPs.

3.6 Summary

This chapter described the proposed online ad hoc distributed simulation. The physical operating platform for the model including operating system, communicational middleware, and traffic simulation model were demonstrated. Also, two major components of the initial algorithmic approach; global process and logical process, were proposed along with data communication mechanism. Finally, main functions of the global process and logical process were illustrated. The proposed methodology is aimed to provide asynchronous execution of LPs, integrate distributed traffic simulations with communication middleware and coordinate the estimates generated by multiple processes with an aggregation technique. Also, the rollback process allows for maintaining similar traffic conditions between LPs and transmitting accurate traffic conditions to other LPs beyond the network boundaries of the LPs. With proper feedback the proposed simulation will be able to capture dynamically changing traffic conditions and provide more up-to-date and more robust estimates.

CHAPTER 4 GRAPHYCAL ANALYSIS

This chapter investigates the analytical background of the proposed ad hoc distributed simulation model and its extension with a real time field data driven simulation client which represents real time field sensor data. In a field implementation this client would be replaced with the streaming detector data. In Section 4.1 the rollback process between two LPs is described in two different diagrams; flow rate diagram and cumulative arrival diagram. The examination is extended in Section 4.2 adding a real time field data driven simulation client into the graphical analysis. In this section, two measures for the system's predictability are graphically presented. Also, graphical analysis for speed selection of outflow control is presented as well as an empirical solution.

4.1 Graphical Presentation of Rollback Process

Suppose that LP 1 and LP 2 are running the ad hoc simulation (Figure 9) with a rollback threshold (μ = 200 veh/hr/ln). LP 1 has estimated the average flow rate on Link B until 7:20AM simulation time would be 120 veh/hr/ln and a sudden flow increase would occur at 7:20AM simulation time to 600 veh/hr/ln (Figure 10). Further suppose LP 2 utilizes 120 veh/hr/ln as an initial input flow rate on Link B until it receives updated information regarding the flow change. Even though LP 1 sends a higher flow rate of 7:21AM simulation time traffic state to the server on Link B, the 4 minute flow rate average (240 veh/hr/ln) does not warrant a rollback immediately in the server, since the difference between 240 veh/hr/ln (the 4 minute flow rate average by LP 1) and 120 veh/hr/ln (the current input rate of LP 2) is smaller than the given threshold (μ : 200 veh/hr/ln). However, LP 1's simulation advances one more simulation minute and LP 1 sends a much higher average flow rate 360 veh/hr/ln at 7:22AM simulation time. The server compares the difference between 360 veh/hr/ln and 120 veh/hr/ln (the current input rate of LP 2) and sends a rollback message to LP 2, since the difference is greater than the given threshold. Similarly, the difference of 7:23AM simulation time traffic states (difference between 480 veh/hr/ln by LP 1 and 360 veh/hr/ln, the new input rate for LP 2) is not large enough to force a rollback. One more simulation minute later, LP 2 needs to alter its input flow rate again when the 4 minute flow rate average at 7:24AM simulation time traffic state is greater than 360 veh/hr/ln (the new input rate for LP 2) by more than the given threshold (μ : 200 veh/hr/ln).

As shown in Figure 10, a rollback is processed whenever the difference between estimates is greater than the given threshold. This implies the system is dependent on the size of threshold. For example, if the size of threshold becomes smaller, then the system would have more rollbacks, which implies more computational overheads although generally higher agreements between LP estimates across the network. On the other hand, a larger threshold is expected to reduce the computational overheads although may result in higher discrepancies between the LPs. The sensitivity of rollback threshold will be examined later in Chapter 6.

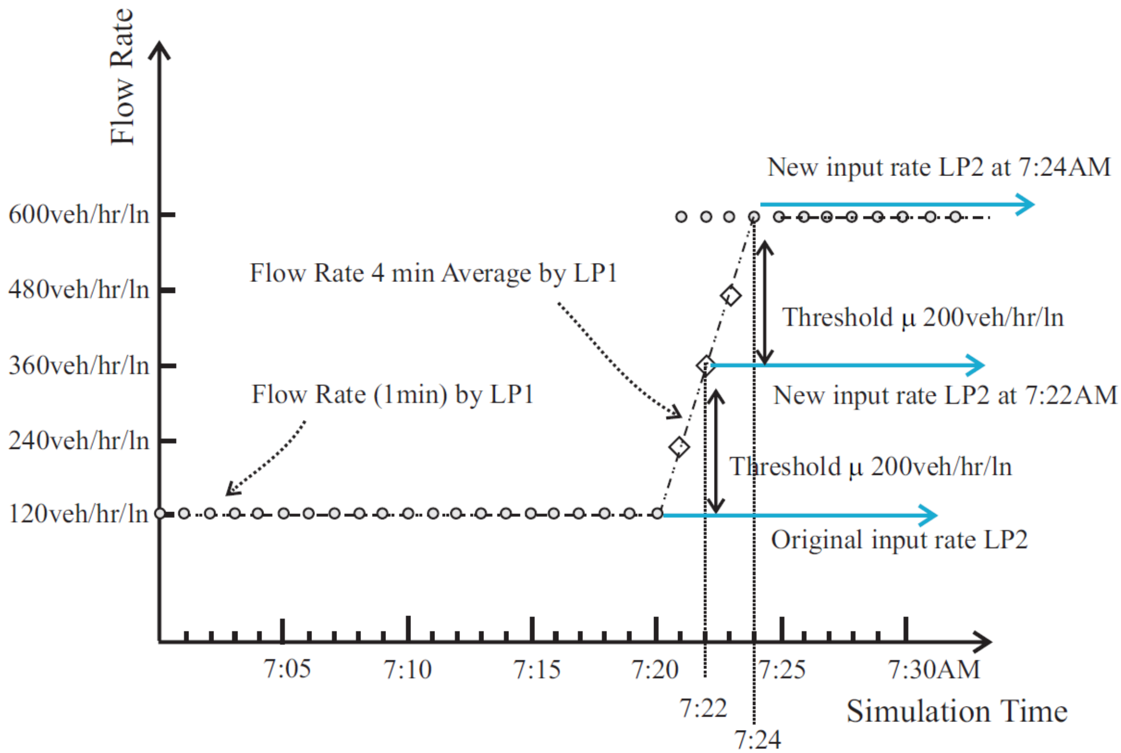

Figure 10 Flow Rate Diagram for Ad Hoc Distributed Simulation

Figure 11 demonstrates cumulative number of vehicles served on Link B. A(t) represents the cumulative arrivals on Link B in LP 1 and D(t) corresponds to Link B cumulative departure in LP 2 (i.e., cumulative number of entering vehicles on LP 2).

As seen in Figure 10 and Figure 11, (1) LP 1 sends its traffic estimates to the server and the arrival flow rate from upstream LP 1 is constant as a(t) from simulation time 7:00AM to 7:20AM, (2) downstream LP 2 continues its simulation with the departure flow rate equal to d(t), (3) the estimated arrival flow rate from upstream LP 1 begins to increase after 7:20AM simulation time and the information is sent to the server and saved in the Space-Time Memory, (4) when the arrival rate of 7:22AM simulation time is sent to the server, the difference between a(t) and d(t) is greater than the given threshold, which prompts the first rollback by the server, (5) the server invalidates traffic states of simulation time 7:22AM and after (grey dotted line) provided by LP 2 in its Space-Time Memory, (6) the server also sends the new arrival rate information to LP 2, and (7) after receiving the rollback message with the updated flow rate, LP 2 rolls back to simulation state of 7:22AM simulation time and renews its simulation. Similarly, LP 2 processes another rollback at 7:24AM.

A drawback of the current threshold method may also be seen in this analysis. A rollback occurs where the slope difference of two curves, i.e. the difference between a(t) and d(t), is greater than the given threshold, since the rollback comparison is based on the point flow rate difference (i.e., absolute flow difference at a time instance, not cumulative difference) in the proposed model. For example, 100 veh/hr/ln arrival rate and 150 veh/hr/ln departure rate with 100 veh/hr/ln threshold does not warrant a rollback in the proposed model, even though 50% more vehicles (50 vehicles) would be generated over an hour. While difference in cumulative vehicle counts would be a good potential measure to detect changes in traffic conditions, it would require additional system measurements, such as counting the number of vehicles entering and exiting the network. Furthermore, the proposed system is associated with numerous rollbacks across the network during the simulation time period. Therefore, the impact of system overhead would need to be considered in collecting cumulative vehicle counts.

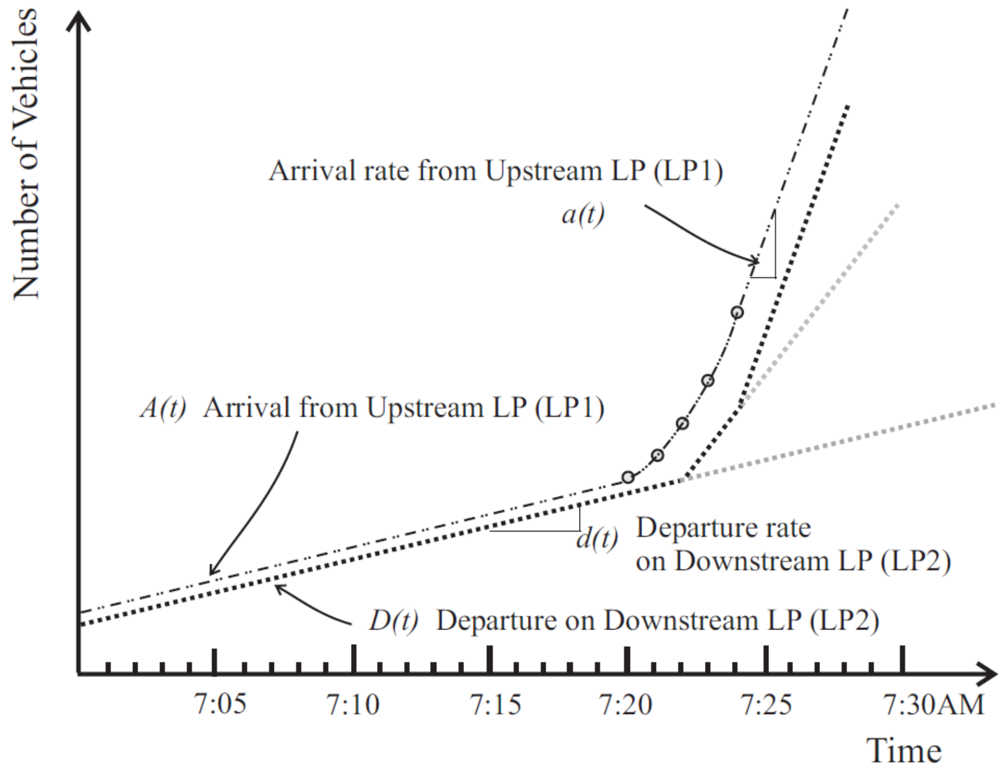

Figure 11 Cumulative Number of Vehicle Diagram for Ad Hoc Distributed Simulation

4.2 Graphical Analysis of Ad Hoc Distributed Simulation with Real Time Field Data Driven Simulation Client

In section 4.1, the rollback process in the ad hoc distributed simulation was graphically presented and it was seen that the threshold size may have a significant impact on computational overheads and estimate accuracy. In this section, as discussed in Section 3.4, a real time field data driven simulation client (LP) is included, allowing for an LP which represents real time sensor data from the field. In a field implementation this client would be replaced with the streaming detector data. In Space-Time Memory at the server, available estimates would be different at varying wall-clock times. Two potential measure of system performance are; 1) length of prediction horizon, that is, how far in advance of the current wall-clock time the system provides estimates and 2) how accurate the estimates are at specific prediction horizon, i.e. how accurate is the estimate (compared with the actual traffic conditions). Figure 12 illustrates available estimates over varying wall-clock times. Suppose the simulation starts at 7:00AM wall-clock time as in Figure 10 and Figure 11 and the simulation proceeds at the speed of 3 minute simulation time / 1 minute wall-clock time (speed-up factor 3). Since there is no rollback between 7:00AM wall-clock time and 7:22AM wall-clock time, it is seen that estimates until 8:00AM simulation time are available at 7:20AM wall-clock time. Suppose the first rollback occurs at 7:22AM wall-clock time based on data from the real time field sensor data. Then, the server invalidates all the available estimates from LP 2 at 7:22AM wall-clock time and LP 2 starts to send updated estimates after the rollback. Similarly the second threshold violation at 7:24AM wall-clock time initiates the second rollback. Without any additional rollbacks, the system produces estimates over an increasing long time horizon as the wall-clock time progresses. It is shown that available estimates from a single LP at varying wall-clock times can be determined as a function of the simulation speed and the time duration after the most recent rollback.

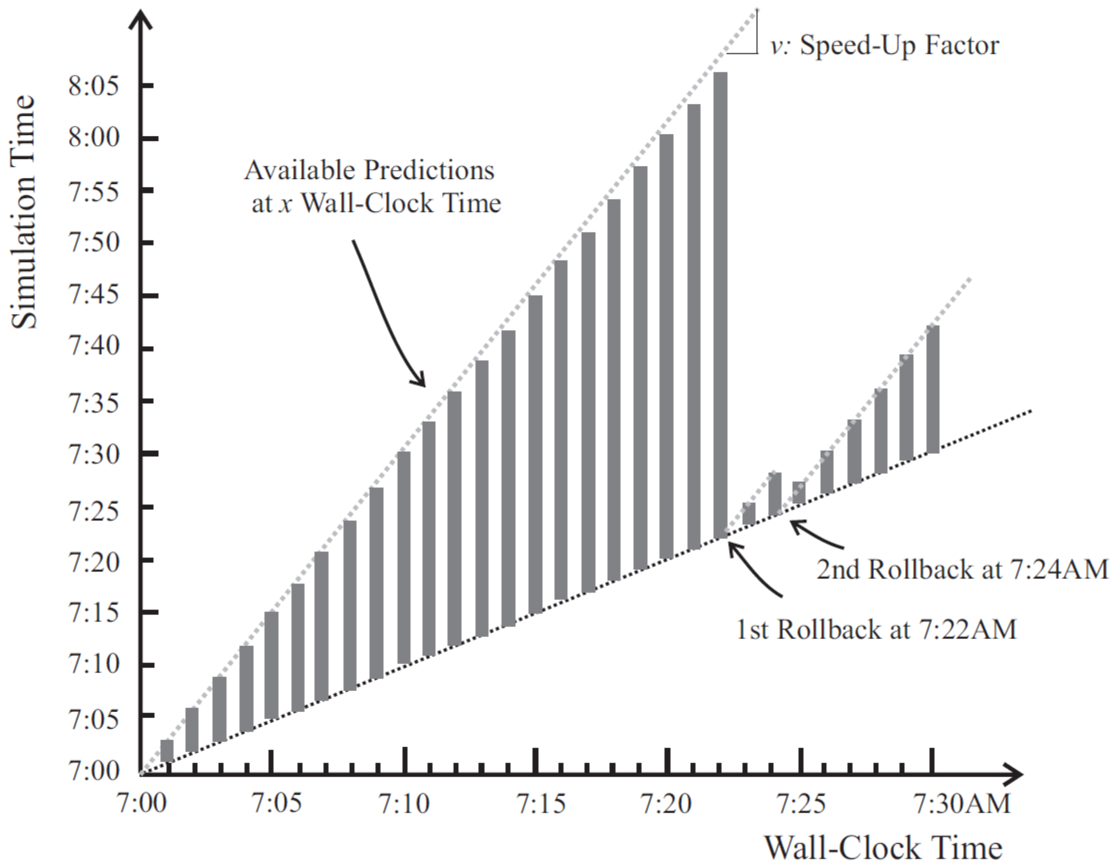

Figure 12 Predicted Simulation Time Period with Wall-clock Time in Ad Hoc Distributed Simulation

However, the preceding is only concerned with the length of the prediction horizon at varying wall-clock time. The second measure is focused on the accuracy of the available estimates. Figure 13 presents simplified available flow rate estimates over the simulation time period. From 7:00AM wall-clock time to 7:22AM wall-clock time, the system predicts Flow A as a future estimated flow rate and its estimates are available up until 8:00AM simulation time at 7:20AM wall-clock time. However, traffic conditions are measured to change in the field at 7:20AM wall-clock time and the first rollback occurs at 7:22AM wall-clock time. Looking over the estimates at 7:22AM wall-clock time regarding the predicted traffic states of 7:20AM simulation time and after (the estimates made between 7:07AM wall-clock time and 7:20AM wall-clock time), they are significantly different from the traffic conditions which occur. Right after the first rollback, limited future traffic estimates (as discussed above) are available as Flow D. However, accuracy is improved after the rollback, since the estimates (Flow D) are more accurate than the previous estimates based on flow rates that did not account for the updated real time detections (Flow A). Similarly, Flow B is predicted after the second rollback at 7:24AM wall-clock time and its estimates are available until 7:42AM simulation time at 7:30AM wall-clock time.

Thus, by way of example, imagine that an incident occurs at 7:20AM wall-clock time. Detectors would not begin to recognize flow changes due to the incident until the incident occurs. Therefore, the ad hoc system could not reflect the incident until receiving these new detections. All estimates made prior to the incident that stretched beyond the incident time would be invalid, and new estimates would be required that account for the incident.

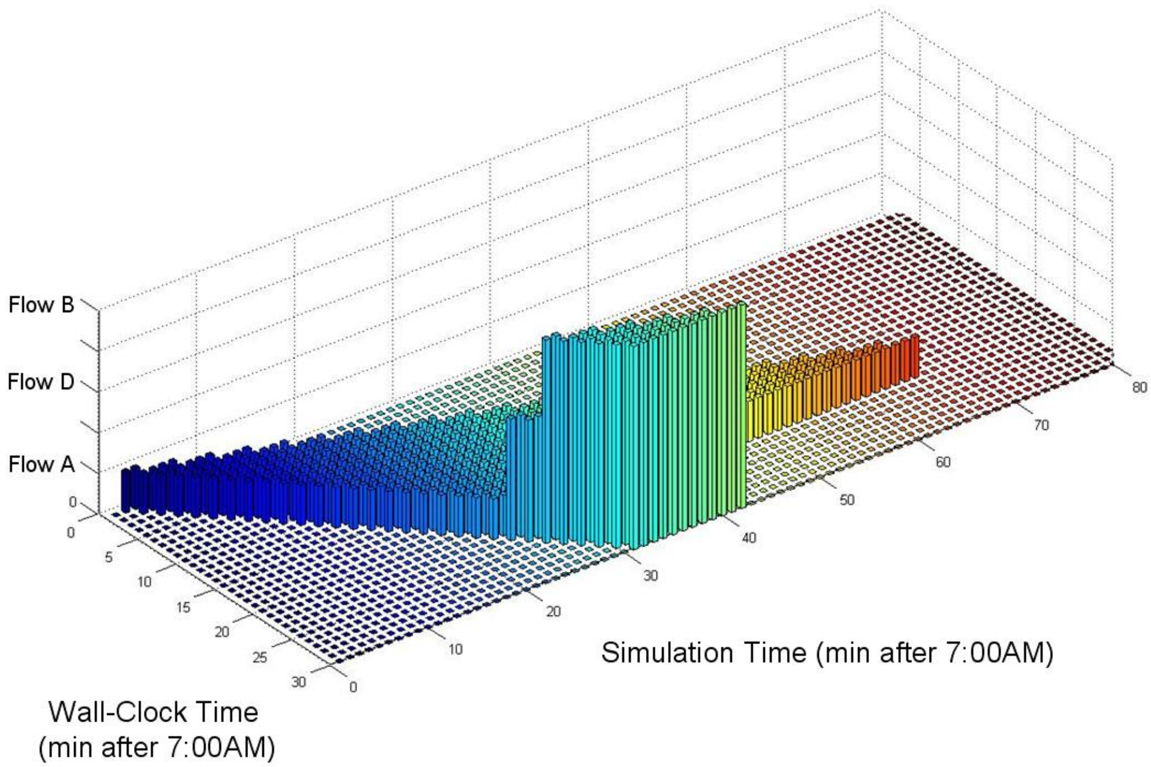

Figure 13 Flow Rate Estimates in Ad Hoc Distributed Simulation

4.3 Speed Selection

As discussed in Section 3.4.4, downstream traffic information is transmitted to upstream LPs in congested traffic conditions. To accomplish this transmission, the outflow rate on the exiting link of the upstream LPs is controlled by changing speed of vehicles on the link. This is required as the simulation model (VISSIM®) has no direct means to throttle the flow rate on an unconstrained link. A question concerning selection of the speed to apply in order to meter the same number of vehicles with downstream LPs is addressed in this section.

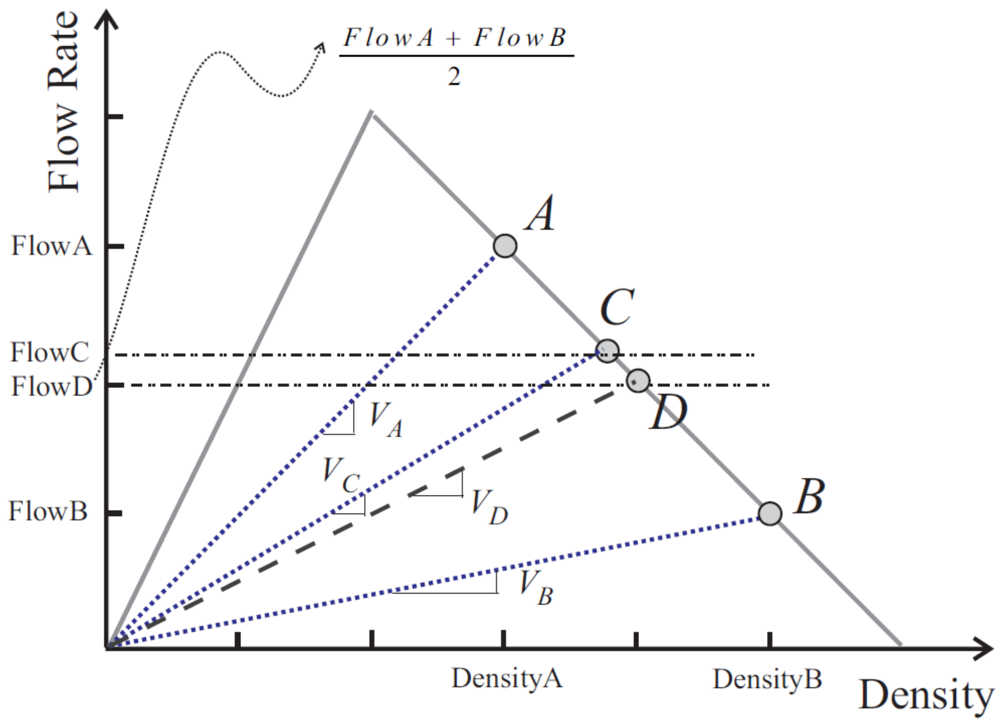

Figure 14 Applying Speed Estimation

Suppose the current 4 minute average flow rate which is stored in the Space-Time Memory is calculated as an average of Flow A and Flow B in Figure 14. To process the average flow rate – Flow D (which is $\frac{(FlowA + FlowB)}{2}$, i.e. the average of Flow A and Flow B) in upstream LP, new speed needs to be applied on Link C of the upstream LP in Figure 9. If the slope of $\overline{OC}$ (average of slope between $\overline{OA}$ and $\overline{OB}$; that is average speed of V_A and V_B) is applied as a new speed, the actual flow rate which the upstream LP processes would be the flow rate of C (Flow C), which is higher than the flow rate of D. This can explain why constant speed can produce more traffic volume. For example, more traffic can be processed during the constant 50 km/hr for two hour period than 100km/hr for one hour and 0 km/hr for one hour. Based on the flow and density relationship, the speed to produce $\frac{(FlowA + FlowB)}{2}$ on Link C in LP 1 would be V_D (a slope of $\overline{OD}$), instead of V_C (a slope of $\overline{OC}$). Therefore, the average speed of downstream LP should not be applied directly onto the link in upstream LPs to reproduce the same traffic flow rate. Instead, lower speed than the average speed of downstream LP needs to be exercised.

After recognizing the average speed may not meter the intended number of vehicles, applying speed selection is investigated based on empirical analysis from VISSIM® model. Figure 15 depicts speed flow diagram obtained from VISSIM® test runs. In each test run the desired speed of each vehicle is altered from 48km/hr to a lower speed at simulation time 600 seconds with at total simulation time of 1200 seconds. The lower speeds simulated where 0.5km/hr to 15km/hr in 0.5km/hr increment and 16km/hr to 30km/hr in 1km/hr increment. Five runs were completed for each speed for a total 245 runs. Each dot represents the average of flow rate and speed during a one minute interval between 660 seconds and 1200 seconds excluding transition period data between 600 seconds and 660 seconds. Figure 15 reveals that the data display a close agreement with flow and speed relationship. In the proposed model, the first 120 second estimates are excluded in aggregation right after a new speed is applied to exclude the transition period data. Based on this VISSIM® data, Table 4 is used in this study to apply on to vehicles to meter the intended traffic flow.

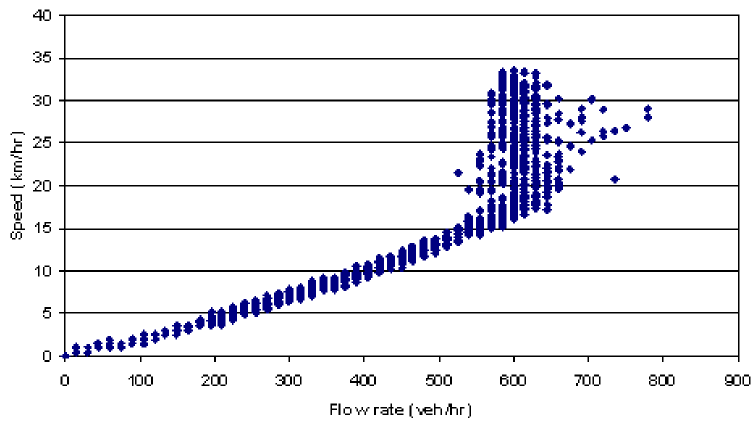

Figure 15 VISSIM® Speed Flow Diagrams

Table 4 Applied Speed Table

Applied Speed	Flow Rate (veh/hr)	Applied Speed	Flow Rate (veh/hr)
1 km/hr	≤ 60	11 km/hr	410 < ≤ 440
2 km/hr	60 < ≤ 120	12 km/hr	440 < ≤ 470
3 km/hr	120 < ≤ 160	13 km/hr	470 < ≤ 500
4 km/hr	160 < ≤ 200	14 km/hr	500 < ≤ 530
5 km/hr	200 < ≤ 240	15 km/hr	530 < ≤ 560
6 km/hr	240 < ≤ 280		
7 km/hr	280 < ≤ 320		
8 km/hr	320 < ≤ 350		
9 km/hr	350 < ≤ 380		
10 km/hr	380 < ≤ 410		

4.4 Summary

This chapter presents the fundamental analytical background on the ad hoc distributed simulation model. The rollback process in the ad hoc distributed simulation is graphically described in Section 4.1. The flow rate diagram and cumulative number of vehicle diagram shows that the overall system simulation speed and estimate accuracy may differ significantly as a function of the selected threshold.

Two main criteria to measure the system's predictability are presented: 1) length of prediction horizon, that is, how far in advance of the current wall-clock time the system provides estimates and 2) how accurate the estimates are at specific prediction horizon, i.e. how accurate is the estimate (compared with the actual traffic conditions). These two criteria and the relation between LP simulations and the real time field data driven simulation client data are graphically demonstrated in Section 4.2. Also, speed selection of outflow control is presented with graphical and empirical methods. The graphical method proves that applying average speed may not process the correct traffic flow. Speed selection is then proposed based on an empirical analysis.

After introducing graphical backgrounds regarding the ad hoc distributed simulation, Chapter 5 explores the ad hoc distributed simulation with three different traffic conditions, including steady traffic state, volume increase, and incident scenarios. In order to examine robustness of the ad hoc distributed simulation, Chapter 6 delves more deeply into the ad hoc distributed simulation and investigates the impact of the geographical LP distributions and different level of rollback thresholds under uncongested traffic conditions. Chapter 7 examines the ad hoc distributed simulation model under congested traffic conditions and provides discussions about the limitation of the proposed approach. Chapter 8 evaluates the ad hoc distributed simulation model when real time field sensor data is available – which is represented by the real time field data driven simulation client. Lastly, Chapter 9 describes the findings from this research effort and suggests the future research.

CHAPTER 5 EVALUATION OF AD HOC DISTRIBUTED SIMULATION

5.1 Introduction

This chapter explores the performance of the ad hoc distributed simulation, developed in Chapter 3. In order to investigate the model in various traffic environments, three different traffic conditions are examined including steady traffic state, volume increase, and incident traffic condition. Section 5.2 describes the experimental design and the results are discussed in Section 5.3.

5.2 Experimental Design

In this section the traffic simulation network utilized in these ad hoc simulation experiments and details of the traffic conditions tested are described. The experiments in this study are performed over a set of heterogeneous personal computers connected through a wireless network. In total, 11 machines (1 server and 10 LPs) are utilized for each experiment. LPs run the experiments on a separate laptop computer with 3 GHz Intel PC with 2 GB RAM. Current experimental design is intended for initial evaluation of the proposed ad hoc approach. Key design parameters in the ad hoc experiments include rollback threshold (flow rate) and rollback threshold (speed). For these experiments design parameters are selected based on engineering judgment. Chapter 6 will provide a sensitivity analysis for several key parameters and future efforts will endeavor to further investigate the parameter selection for the proposed ad hoc approach.

5.2.1 VISSIM® Network

To implement the ad hoc distributed approach the utilized traffic simulation model is required to have the following capabilities: 1) ability to modify simulation objects at runtime, 2) generate interim simulation data at runtime, 3) produce runtime simulation states, and 4) recall the simulation states. VISSIM®, a widely used off-the-shelf traffic simulation program, is a commercial simulation package meeting all requirements mentioned above. VISSIM® is a discrete, stochastic, time step based microscopic simulation model. This behavior-based multi-purpose traffic simulation program has been developed to model a wide range of traffic conditions including freeway, arterial, and public transit operations [75]. In this model all vehicles are modeled individually, based on a psycho-physical driver behavior model developed by Wiedemann [75]. The basic assumption of this model is that a driver can be in one of four driving modes: free driving, approaching, following, or braking.

Access to simulation objects is available through the COM (Component Object Model) interface, which allows developers to import objects and properties during runtime [76]. The VISSIM® COM interface can be operated through computer languages including Visual Basic, Visual C++, and Java. VISSIM® objects are accessed during each simulation time step from scripting languages. For example, vehicle(s) can be generated on any link at any time step by calling 'Vehicle' object, allowing for direct control and runtime adjustment to flow rates. In addition, through the VISSIM® COM interface, interim simulation results can be captured during runtime.

In VISSIM® the current state of the simulation model can be saved at any time during a simulation run using the snapshot function. This function enables each LP to save simulation states in its local storage area at a pre-specified interval. This allows an LP to stop its current simulation, roll back to any stored simulation time step, restore the traffic state to that time period, and resume its simulation with the updates that triggered the roll back, e.g. a change in flow rate at the earlier time step. The simulation state contains information about all vehicles in the network including speed, acceleration/deceleration, and coordination. In this study the traffic states are created via snapshot files (*.SNP) at every simulation minute.

Figure 16 illustrates the VISSIM® network utilized for the experiments in this study. This Manhattan-style 3-by-6 grid network consists of a two-way, 8-lane road (Fifth Street) with all other roads being 4-lane, two-way facilities. Each of the eighteen signalized intersections operates using a pre-timed, 120 second four-phase cycle (10 second protected-only leading lefts and a 50 second through/right movement on all approaches), and a 0 second offset. Table 5 lists the signal timing for the intersections. For this network each roadway link is 400m in length with a 180m single lane left-turn bay, the vehicle fleet is assumed to be 100% autos, and the desired speed is 48km/hr. At each intersection approach 95% of vehicles are assumed to pass straight through, 3% turn right, and 2% turn left, respectively. Each LP models a 3-by-3 grid network, centered on the LP location.

In the experiments it is assumed that the LPs are pre-configured to model the designated scenario area at the start of a run. Each LP sends estimated flow rate, speed, travel time, and queue length data on all simulated links to the server every 60 seconds of simulated time. The 60-second predictions are the aggregate flows over the previous 240 seconds. At the initialization of each LP a 240 second fill period is completed before rollbacks are allowed. LPs do not send updates to the server during the fill period. The duration of each experiment is 90 simulated minutes, including a 30 minute warm-up to allow the system to reach steady state. Results presented below do not include the warm-up period.

As described in Chapter 3, an LP may experience a rollback when the boundary link flow rate for a time interval, utilized as the entering flow rate for that LP, differs by a preset threshold (a rollback threshold of 150 veh/hr is utilized) from the composite value used previously. Finally, ten replicate simulations of the entire large network, representing 'ground truth', were also generated to compare with the ad hoc distributed simulation. The three metrics considered for the experiments are the traffic flow rate, travel time, and queue length.

Table 5 Signal Timing

Phase	Movement Served	Phase Length (secconds)
Φ1 (East/West)	East West Left Turn	7 (Green) / 3 (Yellow)
Φ2 (East/West)	East West Thru	40 (Green) / 3 (Yellow) / 2 (All-red)
Φ3 (North/South)	North South Left Turn	7 (Green) / 3 (Yellow)
Φ4 (North/South)	North South Thru	40 (Green) / 3 (Yellow) / 2 (All-red)

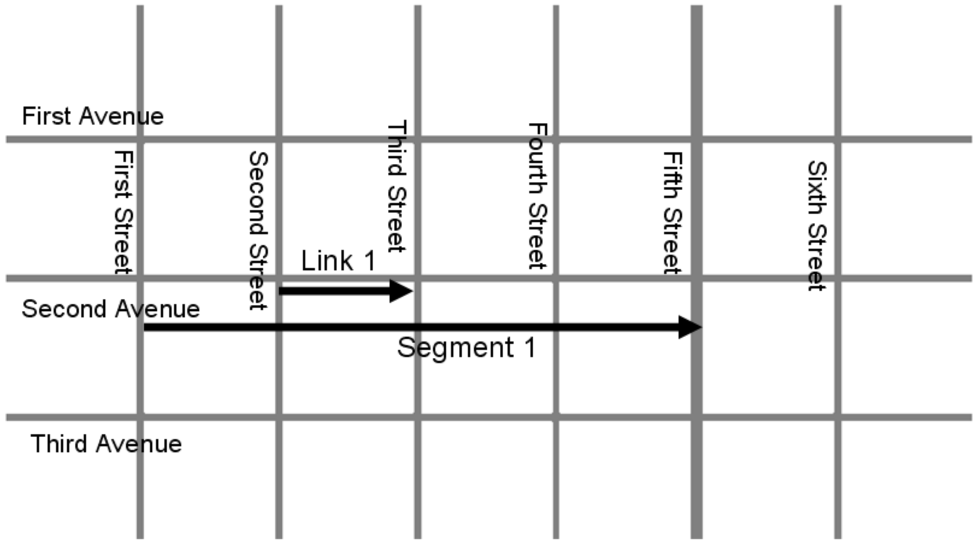

Figure 16 VISSIM® Network

5.2.2 Experimental Scenarios

In this chapter, five different traffic demand scenarios are investigated (Table 6). Scenarios 5.1, 5.2, and 5.3 utilize constant arrival rates (100, 300, and 500 vehicles/hr, respectively), exploring the impact of differing demand levels. Scenarios 5.4 and 5.5 explore dynamically changing traffic conditions, allowing for an evaluation of the responsiveness of the system to a change in traffic conditions. Scenario 5.4 starts with an initial arrival rate of 100 vehicles/hr on all external links, with the arrival rate of eastbound traffic increasing to 500 vehicle/hr 20 minutes after the system warm-up period ends. This experiment is intended to model a sudden influx of traffic at the western end of the network. In Scenario 5.5 a traffic incident occurs on an eastbound exit point on Second Avenue, reducing the average speed of vehicles from 48km/hr to 1 km/hr for 600 seconds and resulting in significant upstream queueing. This experiment examines how well the ad hoc distributed simulation model responds to downstream traffic congestion.

Ten LPs are participating in the simulation; five LPs simulate the 3-by-3 grid network covering the western half of the network (white box in Figure 17), and the remaining five LPs (LP 6 thru LP 10) model the eastern half of the network (grey box in Figure 17). Ten replications are completed for each scenario.

Table 6 Experimental Scenarios

Scenario NO	Traffic State	Input Volume (veh/hr/ln)	Threshold & Traffic Incident
5.1	Steady	100	Threshold 150 veh/hr/ln
5.2	Steady	300	Threshold 150 veh/hr/ln
5.3	Steady	500	Threshold 150 veh/hr/ln
5.4	Non-steady (Volume Increase)	100-500	Threshold 150 veh/hr/ln Volume Increase at 20 minutes
5.5	Non-steady (Incident)	500	Threshold 150 veh/hr/ln Incident between 10 and 20 minutes

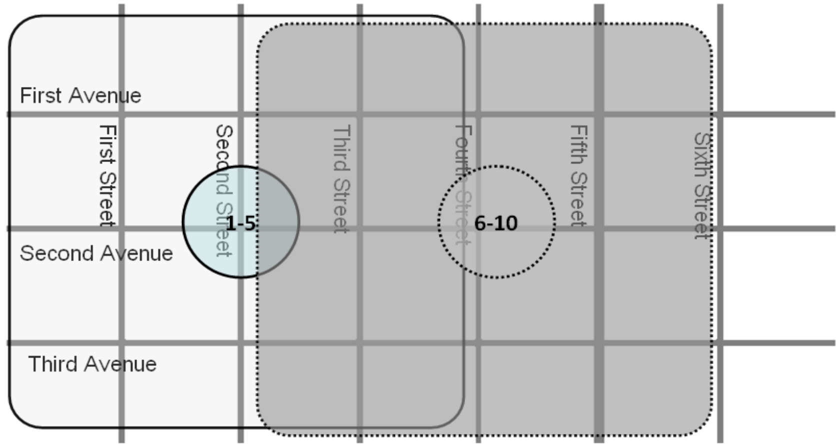

Figure 17 Logical Process (LP) Distribution

5.3 Results and Analysis

The following presents the performance results of the ad hoc approach under both steady state and dynamic traffic demands. To measure the performance quantitatively, flow rate, travel time, and queue length are recorded at pre-determined time interval, utilizing data collection points placed on each link in the network for the ad hoc and ground truth simulations. Section 5.3.1 demonstrates how flow rate, travel time, and queue length are estimated. Quantitative analysis is provided for each scenario in the later section.

5.3.1 Performance Measure Calculation

Flow rate is the number of vehicles crossing a point measured 150m downstream from the entering point of each link, and queue length is measured from downstream intersection stop bar) of each link. In order to assess the performance over a larger area a segment comprised of multiple links is utilized (Figure 16). Segment queue length is the sum of the given performance measure over all links in the segment. Segment flow rate is the numerical average of flow rates of the links in the segment.

$$QueueLength_t^{Segment} = \sum_i QueueLength_t^i$$

$$FlowRate_t^{Segment} = \frac{\sum_i FlowRate_t^i}{n}$$

where n = total number of links in segment

i = link

t = time

Link travel time is the time to traverse a given link (including the downstream intersection). This travel time is recorded when a vehicle completes its travel between the predefined two points (start point and end point), which is "responsive". To provide predicted travel time estimates when a vehicle enters the link, the travel time estimates need to be converted as "predictive" travel time.

$$PreditiveLinkTravelTime_{t-a}^i = ResponsiveLinkTravelTime_t^i$$

where a = $ResponsiveLinkTravelTime_t^i$

i = link

t = time

Segment travel time is calculated based on the estimated "predicted" link travel times. Since modeling areas of LPs are varying, a segment may be modeled partially by overlapping LPs. Therefore, predictive travel time of $Segment^s$ is estimated based on the sum of link travel times on $Segment^s$. Suppose $Segment^s$ comprises of Link 1, Link 2, and Link 3. Predictive travel time of $Segment^s$ is calculated as follows;

$$PreditiveSegmentTravelTime_t^s = PredictiveLinkTravelTime_t^1 + PredictiveLinkTravelTime_{t+b}^2 + PredictiveLinkTravelTime_{t+b+c}^3$$

where $b = PredictiveLinkTravelTime_t^1$

$c = PredictiveLinkTravelTime_{t+b}^2$

t = time

Based on the calculation methods described here, quantitative analysis is provided for each scenario in the next section.

5.3.2 Steady State

Under the tested steady state no rollbacks were reported (Table 7). Even though the threshold was relatively tight (150 veh/hr/ln - 10 vehicles per 4 minute aggregation interval), this indicates relatively stable performance of the ad hoc system under steady state conditions. Rollbacks are designed to correct errors, e.g. when unexpected changes occur in traffic conditions. Since the network is operating in steady state, no rollbacks are required, which indicates that the individual LPs provide reasonable estimates without receiving additional data from other participating LPs. As a result, the traffic estimates by the distributed simulation are similar to the estimates produced by the single "ground truth" model.

Figure 18 shows the "ground truth" traffic flow (average of ten replicated runs of the full network) and modeled ad hoc traffic flow for Scenarios 5.1 through 5.3. The flow rate projections from the ad hoc approach correlate well with the replicated average over the course of the simulation. Also, travel time and queue length on the distributed simulators typically provide reasonable agreement with the ground truth, less than 13.1% mean absolute percentage error for travel time and queue length, for each scenario (Figure 19 and Figure 20, respectively).

Table 7 Rollback Statistics

Scenario NO	Traffic State	Replicate Runs with Rollback(s)
5.1	Steady (100 veh/hr/ln)	0
5.2	Steady (300 veh/hr/ln)	0
5.3	Steady (500 veh/hr/ln)	0
5.4	Non-steady (Volume Increase)	10
5.5	Non-steady (Incident)	10

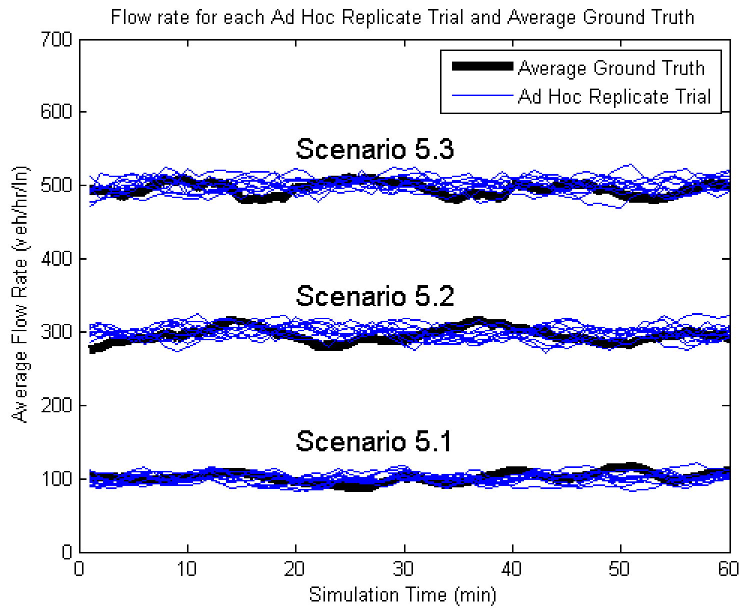

Figure 18 Flow Rate for Each Ad Hoc Replicate Trial and Average Ground Truth (Scenarios 5.1 through 5.3–Segment1)

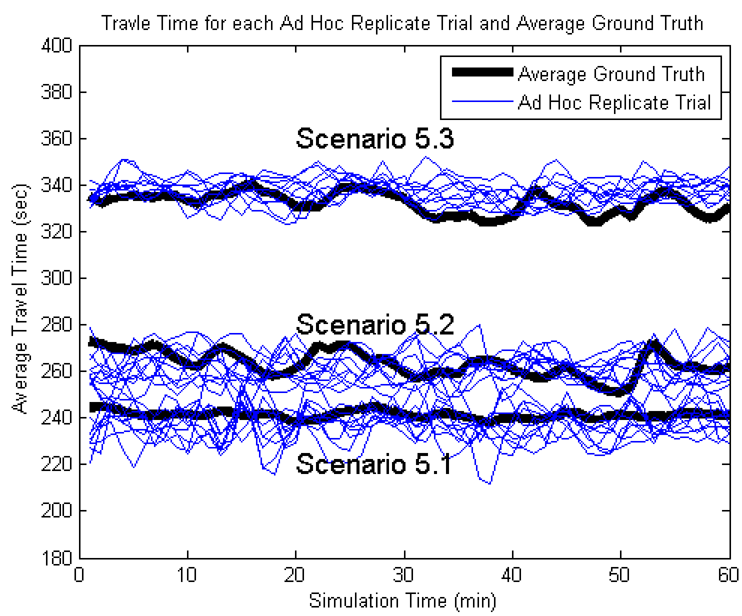

Figure 19 Travel Time for Each Ad Hoc Replicate Trial and Average Ground Truth (Scenarios 5.1 through 5.3–Segment1)

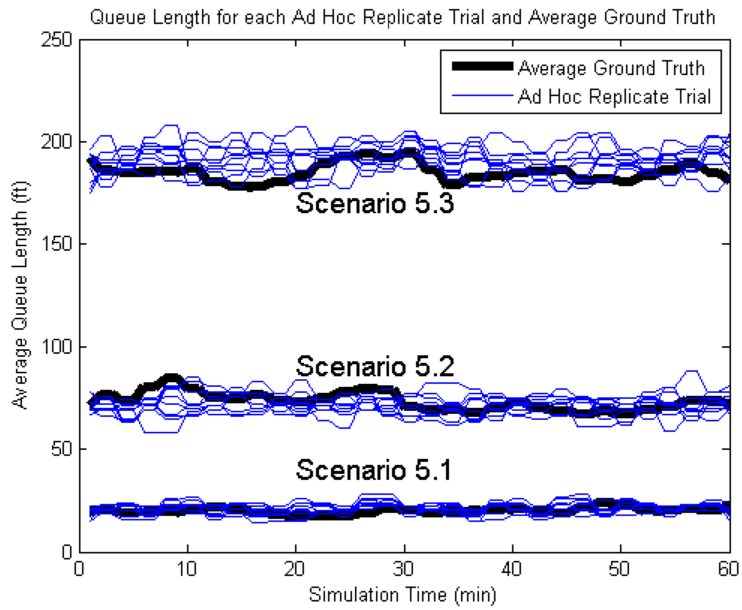

Figure 20 Queue Length for Each Ad Hoc Replicate Trial and Average Ground Truth (Scenarios 5.1 through 5.3–Segment1)

5.3.3 Non-Steady State - Volume Increase

After the steady state traffic experiments Scenario 5.4 explores input arrival rate changes in under capacity conditions. This experiment explores the ability of the distributed simulation approach to propagate the impact of a change in arrival rates on boundary links throughout the network. In this experiment, the network input flow rate on the boundary links was initially set to 100 veh/hr/ln. At simulation time T equal to 20 minutes (after the 30 minute warm-up period), the arrival rate of all border input roads on the west side of the traffic network were increased to 500 veh/hr/ln. Unlike the steady state simulations, rollbacks were required to successfully simulate the increase in traffic through the network (Table 7).

Figure 21 shows the estimated flow rate for Link1 by LP 6 thru LP 10. This estimated flow rate of LP 6 thru LP 10 is adjusted estimate after the downstream LPs receive rollbacks based on the estimates from the upstream LPs. Also, Figure 22 depicts the estimated flow rate for Segment1 of Scenario 5.4. Solid black lines represent the average flow rate of the ten ground truth network simulations and thin lines show the flow rate for each ad hoc distributed simulation replicated trial. As seen in the ground truth data (Figure 21) the increase in the eastbound flow rate on Second Avenue at 20 minutes reaches Link 1 at approximately 23 minutes. The traffic flow increase is propagated downstream according to the travel speed of the vehicles and interaction with the signals. In the ad hoc simulations, the increased arrival demand also reaches Link 1 at approximately 23 minutes in simulation time, 3 minutes after the increase enters the network. As expected, the ten simulators in the ad hoc simulations produce similar results as the "ground truth" runs, successfully propagating the increased flow rate across LP boundaries. Furthermore, it is shown in Figure 23 and Figure 24 that other traffic measurements (travel time and queue length) are also successfully modeled in the ad hoc environment. Quantitative analysis will be presented in Table 8 in Chapter 5.4 where it will be shown that the ad hoc simulations provide reasonably high agreement with the average of "ground truth" simulations. However, more difference between the ad hoc simulation estimates and the ground truth runs is observed after the volume increase (30 minutes and later in simulation time) than before the volume increase. Chapter 6 will provide more detailed analysis of the accuracy of the ac hoc simulation based on the time period (before the increase, during the increase, and the after the increase).

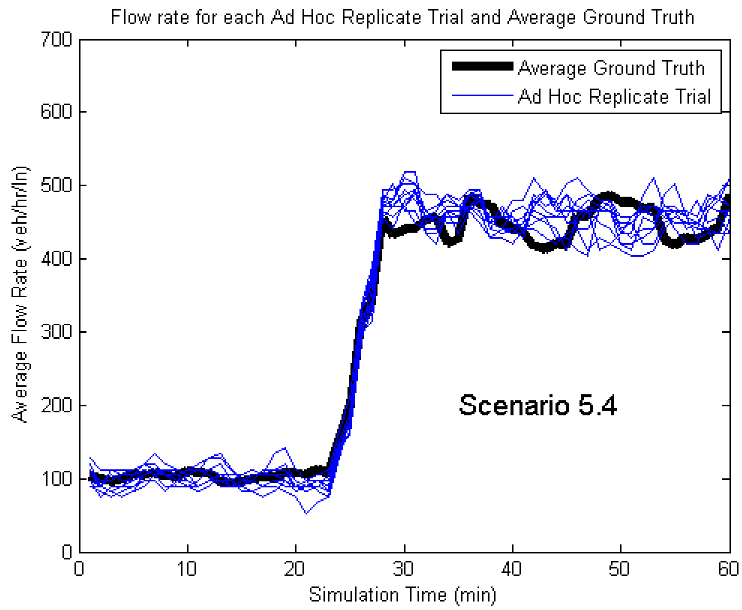

Figure 21 Flow rate for each Ad Hoc Replicate Trial and Average Ground Truth (Scenario 5.4 – Link 1)

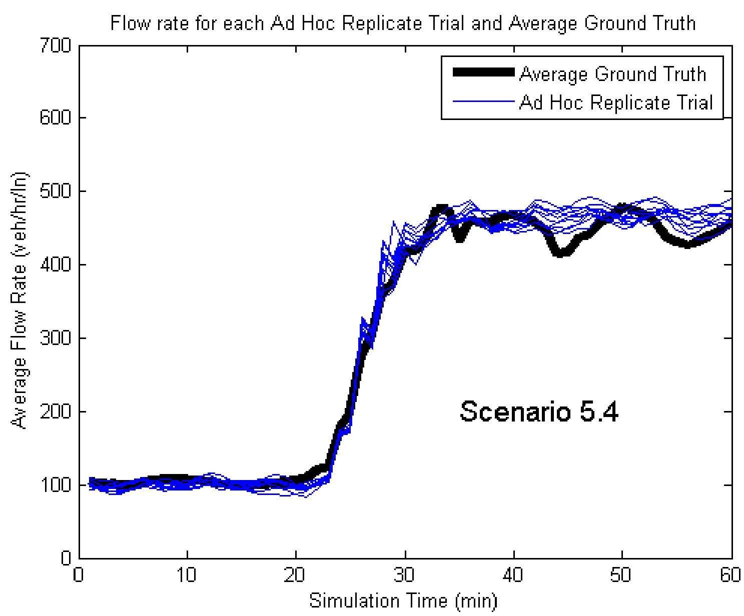

Figure 22 Flow rate for each Ad Hoc Replicate Trial and Average Ground Truth (Scenario 5.4 – Segment 1)

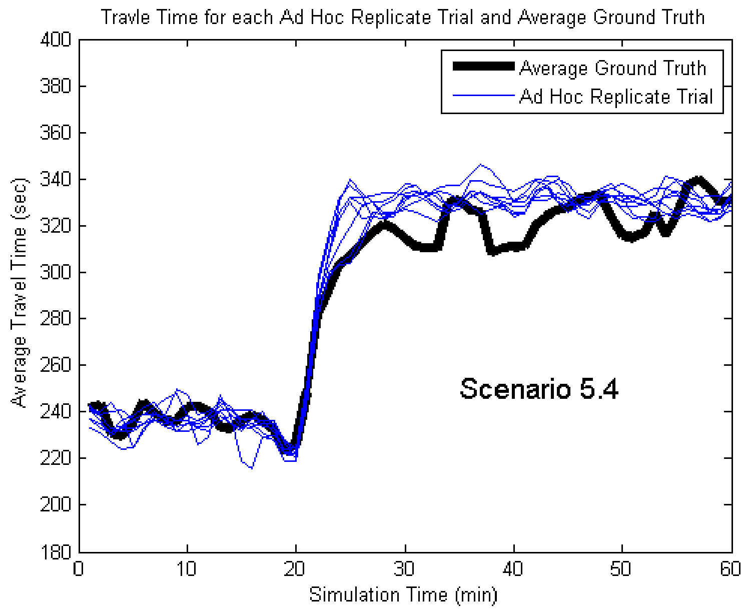

Figure 23 Travel time for each Ad Hoc Replicate Trial and Average Ground Truth (Scenario 5.4 – Segment 1)

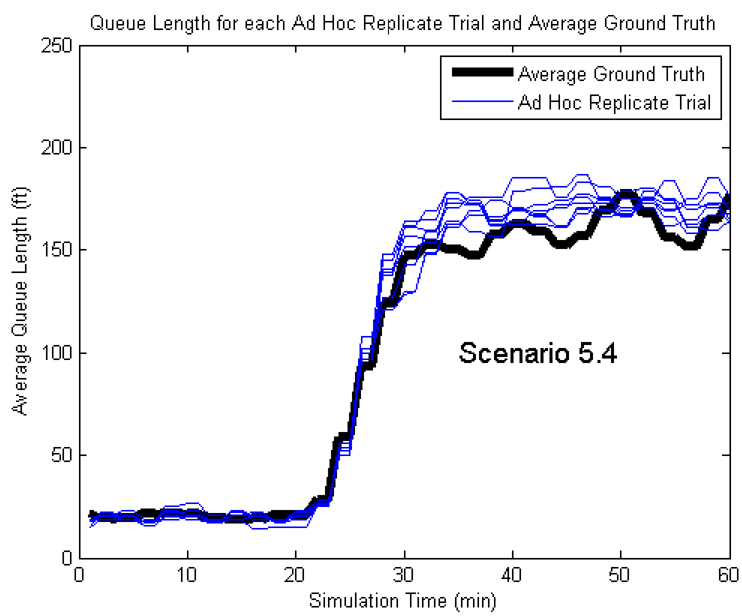

Figure 24 Queue length for each Ad Hoc Replicate Trial and Average Ground Truth (Scenario 5.4 – Segment 1)

5.3.4 Non-Steady State - Incident

This scenario is intended to investigate responsiveness of the ad hoc simulations when traffic incident occurs. A traffic incident is assumed to occur on Second Avenue eastbound exit point, reducing the average speed of vehicles from 48 km/hr to 1 km/hr for 600 seconds and creating queues toward upstream eastbound links. The network input flow rate on the boundary links was constant during the simulation as 600 veh/hr/ln. At simulation time T equal to 10 minutes (after the 30 minute warm-up period), the speed of the vehicles on the eastbound exit link of Second Avenue (Figure 16) was decreased to 1 km/hr, essentially representing a 10 minute blockage of the roadway. Unlike the volume increase scenario, upstream LPs were required to make rollbacks to update the incident traffic conditions from downstream LPs.

Figure 25 shows the estimated flow rate for Segment1 of Scenario 5.5. Solid black lines represent the average flow rate of the ten ground truth network simulations and thin lines show the flow rate for each ad hoc distributed simulation replicated trial. Figure 25 shows the ad hoc distributed simulation were able to regenerate not only the congestion, but also the periods before, during, and after the congestion. While it is shown in Figure 26 and Figure 27 that other traffic measurements (travel time and queue length) are over-estimated in the ad hoc trial runs, the general traffic patterns in congested traffic conditions were reasonably modeled in the ad hoc environment. It is believed that higher flow rates on the upstream LPs estimated after the congestion (between 30 and 55 minutes) in Figure 25 are the primary reason for the over estimation of queues and travel times. The upstream LPs fail to capture the buildup of unserved demand accurately with lower traffic flow during the congestion (between 10 and 30 minutes) and higher traffic after the congestion. Therefore, the estimates of the ad hoc simulation diverge from the actual traffic measures. The impact from the higher flow rates during this time period (a difference of less than 10% between the ad hoc flow and ground truth) resulted in a significant impact on travel time and queue length during the same time period (Figure 26 and Figure 27). More discussion about the ad hoc simulation in congested traffic conditions will be provided in Chapter 7.

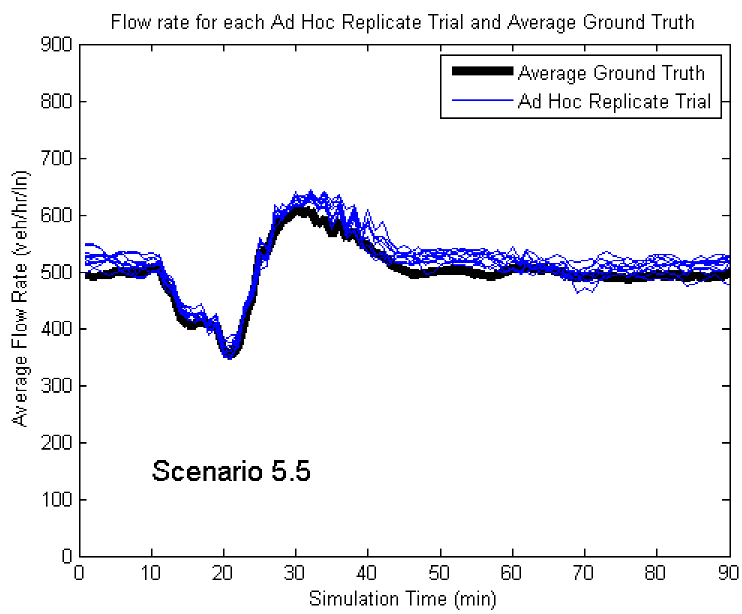

Figure 25 Flow rate for each Ad Hoc Replicate Trial and Average Ground Truth (Scenario 5.5 – Segment 1)

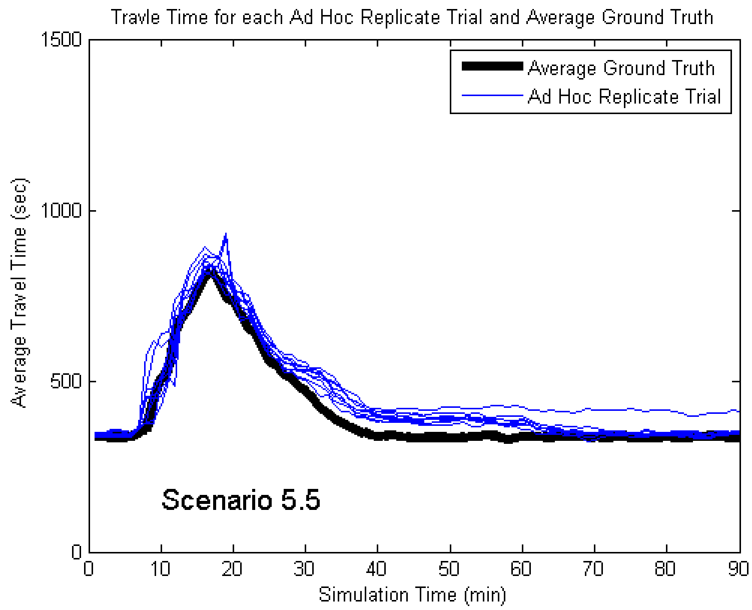

Figure 26 Travel time for each Ad Hoc Replicate Trial and Average Ground Truth (Scenario 5.5 – Segment 1)

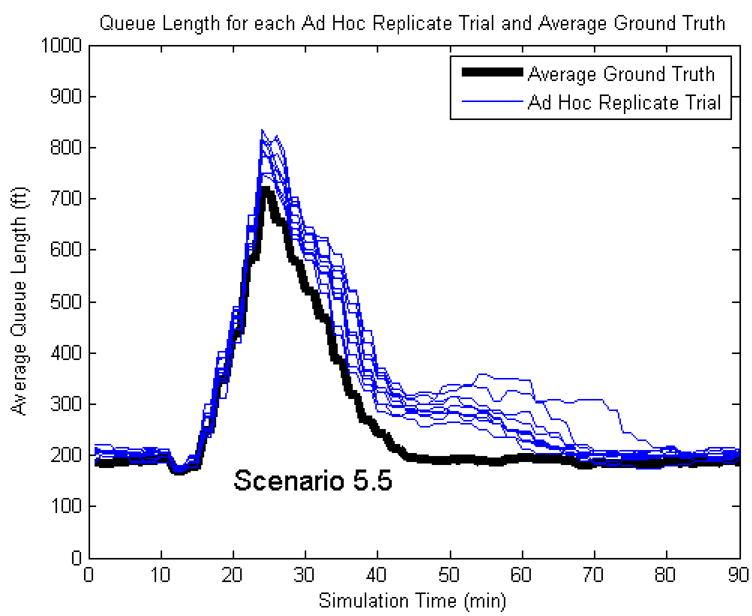

Figure 27 Queue length for each Ad Hoc Replicate Trial and Average Ground Truth (Scenario 5.5 – Segment 1)

5.3.5 Analysis

To provide a quantitative analysis of the performance of the ad hoc distributed simulation approach the mean absolute percentage error (MAPE) values were computed for the given performance metrics.

$$MAPE_{j,k} = \frac{\sum_{i=1}^{n} \frac{\left| Ad\ Hoc\ Replicate\,Trial_{i,j,k} - Ground\,Truth\ Average_{j,k} \right|}{Ground\,Truth\ Average_{j,k}}}{n} \times 100$$

$$MAPE_{j} = \frac{\sum_{k=1}^{m} (MAPE_{j,k})}{m}$$

Where:

$MAPE_{j,k}$ Mean Absolute Percentage Error of the ad hoc simulation estimates (estimate type j) for simulation time interval k.

$MAPE_{j}$ Mean Absolute Percentage Error of the ad hoc simulation estimates (estimate type j) over the total simulation period.

$Ad\ Hoc\ Replicate\,Trial_{i,j,k}$ Average of estimates (estimate type j) for simulation time interval k, produced by LPs for replicate trail run number i.

$GroundTruthAverage_{j,k}$ Average of estimates (estimate type j) for simulation time interval k, produced by 10 ground truth replicate simulation runs:

m Number of simulation intervals

n Number of ad hoc replicate trial runs

j Estimate type (flow rate, travel time, and queue length).

Table 8 presents the mean absolute percentage error of the traffic measures. It is shown that the ad hoc simulations provide reasonably high agreement with the average of "ground truth" simulations, with 2.0% to 13.1% MAPE in steady scenarios. It is noted that there is a trend of decreasing MAPE values as the input rate increases. For example, the MAPE of flow rate estimates for the 100 veh/hr/ln input rate under Scenario 5.1 (7.2%) are significantly higher than those under Scenario 5.2 (300 veh/hr/ln input rate, 4.2%) and Scenario 5.3 (500 veh/hr/ln input rate, 2.3%). In addition, the MAPE values of travel time and queue length estimates show the decreasing trend as the input rate increases. These results imply that the vehicle arrival pattern will impact the performance of ad hoc distributed simulations. It is believed that in higher input rates (300 veh/hr/ln and 500 veh/hr/ln in the experiments), the impact of vehicle platooning with signal coordination is reduced (e.g. the overall variation in traffic estimates decrease) and the ad hoc simulations more closely reflect the "ground truth" simulation. This is likely an indication that the reliability of the ad hoc solution may decease at lower volumes. Also, this is likely at least in part a result of the threshold value becoming close to flow rate. More detailed investigation regarding thresholds is provided in Chapter 6.

Also, it is shown that the ad hoc simulations reasonably model two non-steady traffic scenarios. In congested traffic conditions of Scenario 5.5, the flow rate MAPE is lower than Scenarios 5.1 and 5.2 uncongested traffic conditions. However, in later chapters it will be shown that these results are specific to the ad hoc simulation parameters and traffic scenario tested. For instance, in Section 6.3.2, it will be demonstrated that the performance of the ad hoc simulations are dependent on the rollback threshold in volume increase scenarios. Chapter 7 will further present scenarios showing poor performance in congested traffic conditions. Based on these examples overall discussion will be provided regarding the performance of the ad hoc simulation with different traffic conditions and simulation parameters.

Table 8 MAPE by traffic conditions (%, Segment 1)

Scenario NO	Traffic State	Flow Rate	Travel Time	Queue Length
5.1	Steady (100 veh/hr/ln)	7.2	2.6	13.1
5.2	Steady (300 veh/hr/ln)	4.2	2.7	7.0
5.3	Steady (500 veh/hr/ln)	2.3	2.0	4.5
5.4	Non-steady (Volume Increase)	6.1	3.7	9.5
5.5	Non-steady (Incident)	3.4	7.7	18.8

5.4 Summary

This chapter investigated the performance of the ad hoc distributed simulation under five traffic scenarios. Firstly, the detail of the experimental environments and VISSIM® simulation parameters were described. Secondly, five different traffic scenarios tested were explained. After the simulation, the ad hoc replicate trials were graphical compared with the ground truth runs and the rollback statistics were analyzed. In order to quantify the performance, flow rate, travel time, and queue length were evaluated based on the mean absolute percentage error.

In steady state, the ad hoc distributed simulation provided very similar projections without any rollbacks. However, rollbacks were triggered in every replicate run in non-steady traffic condition scenarios. The results demonstrated that the ad hoc approach provided reasonable agreement with the ground truth. Based on the quantitative analysis, it was seen that the ad hoc distributed simulation provides comparable results with the ground truth under various steady and non-steady traffic conditions. Chapter 6 will investigate more general applicability of the ad hoc distributed simulation with different geographical LP distributions and conduct a sensitivity analysis of rollback thresholds with different level of traffic inputs under uncongested traffic conditions. Chapter 7 then examines the ad hoc distributed simulation model under congested traffic conditions and provides discussions about the limitation of the proposed approach.

CHAPTER 6 SENSITIVITY ANALYSIS OF AD HOC DISTRIBUTED SIMULATION (UNCONGESTED)

6.1 Introduction

The performance of the ad hoc distributed simulation is investigated under five different traffic scenarios with a pre-determined parameter setup in Chapter 5. Chapter 6 delves more deeply into the influence of two primary factors of the ad hoc distributed simulation: 1) the geographical LP distribution over the network and 2) the size of the rollback threshold. Firstly, geographical distribution of LP's is explored as the LP locations in a field implementation of an ad hoc distributed simulation would not be fixed and would move according to each LP's traveling direction (i.e. the movement of the vehicle on which the LP resides). Thus, the areas modeled by different LPs overlap unpredictably. Therefore, it is necessary to understand the impact of the geographical distribution of LP locations. Secondly, in Chapter 4 the size of the rollback threshold is identified as a significant factor that may influence the simulation speed and accuracy of the system. Thus, the sensitivity of the rollback threshold to the overall communication overhead and accuracy is investigated. Sensitivity analysis for these two parameters allows for more insights into the ad hoc approach and an improved understanding of potential issues regarding a field implementation.

6.2 Experimental Design

The traffic network used in this chapter is the same as the network previously used in Chapter 5. Details about the parameter selections are as follows.

6.2.1 Logical Process Location Distribution

In Chapter 5, five LPs simulate the 3-by-3 grid network covering the western half of the network (white box in Figure 17), and the remaining five LPs (LP 6 through LP 10) model the eastern half of the network (grey box in Figure 17). In addition to this initial setup, a randomly selected geographical distribution of ten LPs is evaluated in this chapter. In this setup, the LPs are randomly assigned to network locations, assuring all network links are covered by at least one LP. Figure 28 shows the center point of the ten LPs, each location the center of the 3-by-3 LP grid network. Where the 3-by-3 network area centered on the LP location extends beyond the boundary of the large network the LP simulates only those portions in the large network. To compare the impact of the geographical distribution of LP in the same conditions, the five traffic scenarios in Chapter 5 are tested given the randomly selected locations of LPs. Ten replications are completed for each scenario.

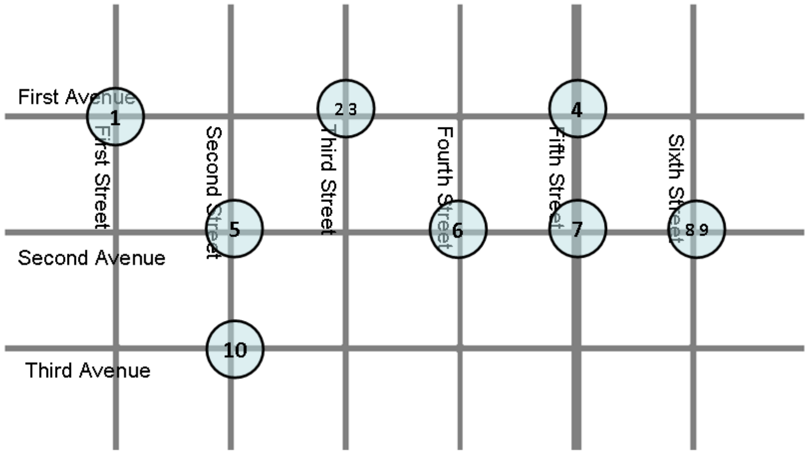

Figure 28 Logical Process Distribution

6.2.2 Rollback Threshold

As the size of the rollback threshold is identified as a major factor which may influence the communication overhead and accuracy of the system in Chapter 4, the impact of the rollback threshold is investigated. Since no rollbacks occurred in three steady states with 150 veh/hr/ln threshold in Chapter 5, these scenarios are excluded in the rollback sensitivity analysis. In this exploration a volume increasing scenario (i.e. input volume increases on the west boundary input links at a preselected time) is tested for various rollback thresholds and combinations of initial and final input rates (Table 9) under the same network setup with Chapter 5. It is expected that smaller rollback thresholds will increase the ad hoc simulation accuracy while requiring more rollbacks, which will increase the communication overhead. Similarly with MAPE in Chapter 5, the travel time estimates from the ad hoc simulation are compared with the ground truth estimates. However, MAE (Mean Absolute Error) is calculated for the flow rate estimate comparison in order to better present the absolute difference between the estimates.

$$MAE_{j,k} = \frac{\sum_{i=1}^{n} \left| Ad\ Hoc\ Replicate\,Trial_{i,j,k} - Ground\,Truth\ Average_{j,k} \right|}{n}$$

$$MAE_{j} = \frac{\sum_{k=1}^{m} (MAE_{j,k})}{m}$$

Where:

$MAE_{j,k}$ Mean Absolute Error of the ad hoc simulation estimates (estimate type j) for simulation time interval k

MAE_{j} Mean Absolute Percentage Error of the ad hoc simulation estimates (estimate type j) over the total simulation period

$Ad\ Hoc\ Replicate\,Trial_{i,j,k}$ Average of estimates (estimate type j) for simulation time interval *k*, produced by LPs for replicate trail run number *i*

$GroundTruthAverage_{j,k}$ Average of estimates (estimate type j) for simulation time interval k, produced by 10 ground truth replicate simulation runs

m Number of simulation intervals

n Number of ad hoc replicate trial runs

j Estimate type (flow rate, travel time, & queue length)

6.3 Results and Analysis

Sensitivity analysis is conducted using the same network in Chapter 5. Also, the same traffic measures (flow rate, travel time, and queue length) are collected. In 6.3.1, presenting the results for the LP location distribution exploration segment travel time and segment queue length (the sum of the given performance measure over all links in the segment) and segment flow rate (the numerical average of flow rates of the links in the segment) are compared with the results in Chapter 5. In addition, segment travel time and flow rate of Link 2 (Figure 29) are compared to further investigate the downstream traffic condition propagation in 6.3.2.

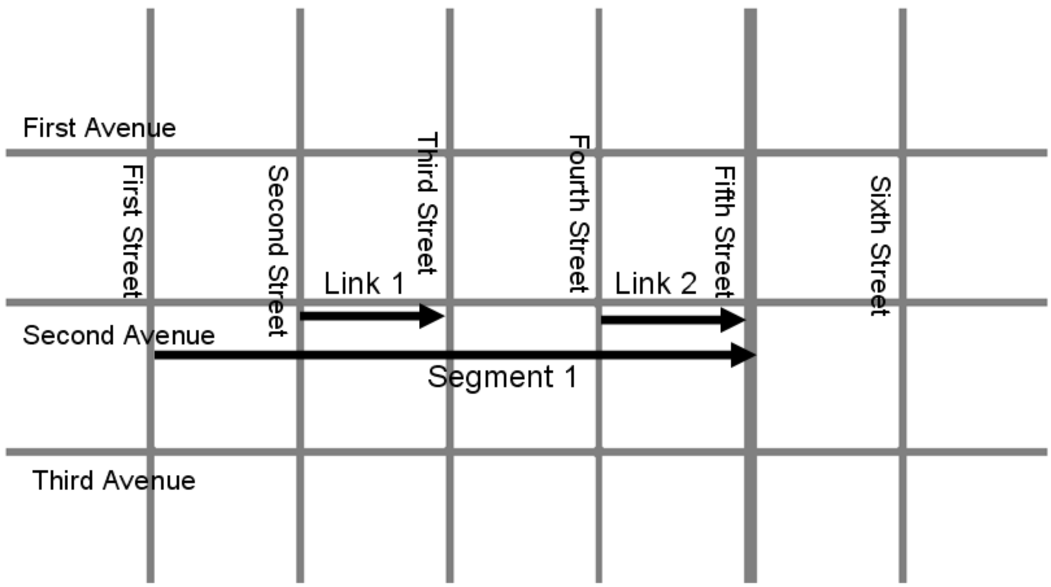

Figure 29 Traffic Measuring Locations

6.3.1 Logical Process Location Distribution

As discussed above, each traffic condition scenario is simulated with two different LP distributions. In Chapter 5, LP 1 thru LP 5 model the left portion of the network and the other five LPs model the right portion (Figure 17). In this setup, rollbacks may only be triggered by operations on eastbound links between Second Street and Third Street and westbound links between Third Street and Fourth Street. For example, LP 1 thru LP 5 each provide the server with flow rate estimates for eastbound link on Second Avenue between Second Street and Third Street, and LP 6 thru LP 10 each receive arrival flow data for the link from the server.

Table 9 Rollback Statistics

Scenario NO	Traffic State	LP Distribution	Replicate Runs with Rollback(s)	Total Number of Rollbacks
5.1	Steady (100 veh/hr/ln)	Initial*	0	0
6.1		Random	0	0
5.2	Steady (300 veh/hr/ln)	Initial	0	0
6.2		Random	0	0
5.3	Steady (500 veh/hr/ln)	Initial	0	0
6.3		Random	0	0
5.4	Non-steady (Volume Increase)	Initial	10	100
6.4		Random	10	127
5.5	Non-steady (Incident)	Initial	10	162
6.5		Random	10	184

* Initial refers to LP distribution found in Chapter 5

In this section, the Scenarios 5.1 through 5.5 in Chapter 5 are tested with the LP distribution in Figure 28 to explore the impact of a random geographical distribution of the LPs on the ability of the ad hoc approach to capture traffic conditions. Each scenario is replicated ten times using the random LP distribution. In these LP distributions, rollbacks may be triggered by thresholds violations on additional links (recall rollbacks are only triggered on three eastbound links and three westbound links in the experiments in Chapter 5). Furthermore, fewer LPs may contribute to the composite value, creating more variation in aggregated flow rate. For example, in Chapter 5, five LPs provide estimates for the three eastbound and three westbound links where rollbacks are triggered. However, in the random LP distribution, links where rollbacks may be triggered are spread across the network, depending on the distribution of LP locations. As a result it is anticipated that the number of rollbacks may increase, as seen in Table 9 where 27% and 14% more rollbacks occurred given the random distribution of LP locations, for the Scenario 6.4 and 6.5 traffic conditions (100 rollbacks to 127 rollbacks in Scenario 6.4 and 162 rollbacks to 184 rollbacks in Scenario 6.5). However, again, no rollback occurred in three steady state scenarios (Table 9). However, the mean absolute percentage error (MAPE) analysis for Segment 1 (Table 10) reveals that the different geographical distribution of LP locations did not significantly impact on the overall results of the ad hoc approach. While this result is for a single random distribution of LPs and may not be generalized to any possible distribution it does demonstrate that it is possible to introduce randomness into the LP distribution and maintain reasonable results. Future efforts will explore conditions identifying potential characteristic of reliable LP location distributions and potential characteristics of unreliable distributions.

6.3.2 Rollback Threshold

While smaller rollback thresholds are expected to increase higher accuracy between LPs over the network, they are also expected to increase the total number of rollbacks, resulting in additional communication overhead. The sensitivity analysis results for several rollback thresholds and traffic input combination are presented in Table 12 and

Table 13. It is readily seen that as the threshold decreases, the overall accuracy of the ad hoc approach improves and more rollbacks occur as the rollback threshold becomes stricter.

Table 10 Mean Absolute Percentage Error (%, Segment 1)

Scenario NO	Traffic State	LP Distribution	Flow Rate	Travel Time	Queue Length
5.1	Steady (100 veh/hr/ln)	Initial*	7.2	2.6	13.1
6.1		Random	6.7	3.4	10.7
5.2	Steady (300 veh/hr/ln)	Initial	4.2	2.7	7.0
6.2		Random	4.8	3.1	7.5
5.3	Steady (500 veh/hr/ln)	Initial	2.3	2.0	4.5
6.3		Random	3.6	3.3	5.2
5.4	Non-steady (Volume Increase)	Initial	6.1	3.7	9.5
6.4		Random	5.8	4.1	7.6
5.5	Non-steady (Incident)	Initial	3.4	7.7	18.8
6.5		Random	5.1	6.7	8.1

* Initial refers to LP distribution found in Chapter 5

However, it is noted that the performance is specific to the rollback threshold and traffic scenario tested, even though the overall relationship is observed. In several scenarios, the ad hoc simulation with a larger threshold performed better than the simulation with a smaller threshold. For example, the 330 veh/hr/ln threshold in Scenario 6.9 produces higher accuracy than smaller thresholds (150 veh/hr/ln and 240 veh/hr/ln in Scenario 6.7 and Scenario 6.8, respectively) in the MAE of the flow rate on Link 2 between 31 and 50 minutes (Figure 30 and Table 12). Similarly, higher accuracy is witnessed with the larger threshold (330 veh/hr/ln in Scenario 6.9) compared to the smaller thresholds (150 veh/hr/ln or 240 veh/hr/ln in Scenario 6.7 and Scenario 6.8) in the

MAPE of the travel time on Segment 1 (Table 13). However, the 330 veh/hr/ln threshold does not provide high accuracy in Scenario 6.14 and Scenario 6.23 (Figure 31 and Figure 32). Instead, the 240 veh/hr/ln threshold offers higher estimate accuracy than 150 veh/hr/ln threshold in these scenarios (Figure 31 and Figure 32).

These inconsistencies are likely a result of the ratio between the selected thresholds and the absolute change in the input volume. For example, the flow difference between the initial flow rate and the increased flow rate is slightly greater than the selected threshold in Scenario 6.9 (361 veh/hr/ln flow rate increase, 400 veh/hr/ln increase* 95%*95% thru movement rate, versus 330 veh/hr/ln threshold). Therefore each LP in Scenario 6.9 was able to publish its flow rate and travel time estimates with better accuracy than Scenario 6.7 and Scenario 6.8. In Scenario 6.7 a total increase of 300 veh/hr/ln would be reflected (i.e. two thresholds) and in Scenario 6.8 a total increase of 240 veh/hr/ln would be reflected (i.e. one threshold). In both Scenario 6.7 and Scenario 6.8 additional rollbacks incrementing the volumes higher would not occur as the volume increase was not sufficient. On the other hand, the 330 veh/hr/ln was not able to accurately reflect the traffic flow changes from 100 veh/hr/ln to 400 veh/hr/ln (Figure 30) and 200 veh/hr/ln to 500 veh/hr/ln (Figure 32) in Scenario 6.14 and Scenario 6.23, respectively. Since the flow rate increment (300 veh/hr/ln) was smaller than the selected rollback threshold (330 veh/hr/ln), rollbacks were not triggered to update the changes in traffic conditions. However, the smaller thresholds of 150 veh/hr/ln and 240 veh/hr/ln provided flow rate and travel time estimates with better accuracy than 330 veh/hr/ln in these scenarios (Scenarios 6.8, 6.9, 6.21, and 6.22). It should be noted that the estimate errors in travel time of Segment 1 appear smaller than the errors in flow rate estimate on

Link 1, since Segment 1 travel time includes travel times of Link 1 and Link 2 (estimated by LPs 1-5 which were pre-configured to model the designated scenario) and travel times of Link 3 and Link 4 (produced by LPs 6-10 which received the updated traffic conditions from the LPs 1-5). Therefore, travel time estimates on Link 1 and Link 2 were close to the ground truth estimates regardless of the rollback size, since they did not require updated traffic information from rollbacks. Thus, while no rollback was instanced with higher threshold (i.e. Scenarios 6.10, 6.14, and 6.23) overall segment travel time estimates were not significantly different from ground truth simulations.

As expected, the 60 veh/hr/ln threshold was found to provide the most consistent accuracy over the traffic conditions tested. While this performance is specific to the network and traffic scenarios tested, it may be reasonably concluded that the anticipated overall relationship between the estimate accuracy and the threshold level will hold. That is, decreasing thresholds improve ad hoc performance with some potential for scenario specific aberrations.

As mentioned, the threshold selection is very crucial to system performance from the perspective of system accuracy and efficiency. As seen in Table 12 and Table 13, there is a trade-off between the accuracy and communication overhead. Even though a communication overhead constraint was not considered in this analysis, it could be a significant factor in a full implementation of the ad hoc simulation. In addition, while 60 veh/hr/ln (2 vehicles in one cycle) provided the highest accuracy in the experiments, it is possible that sufficiently small thresholds would be highly inefficient. The potential exists that nearly continuous rollbacks may occur particularly where rollbacks are triggered by normal variations in flow conditions, such as fluctuations in flow rate due to traffic control. Recall the objective of threshold is to catch changes in traffic conditions, not expected, small variations in traffic conditions. Future research should consider determining a reasonable rollback threshold concerning the system efficiency.

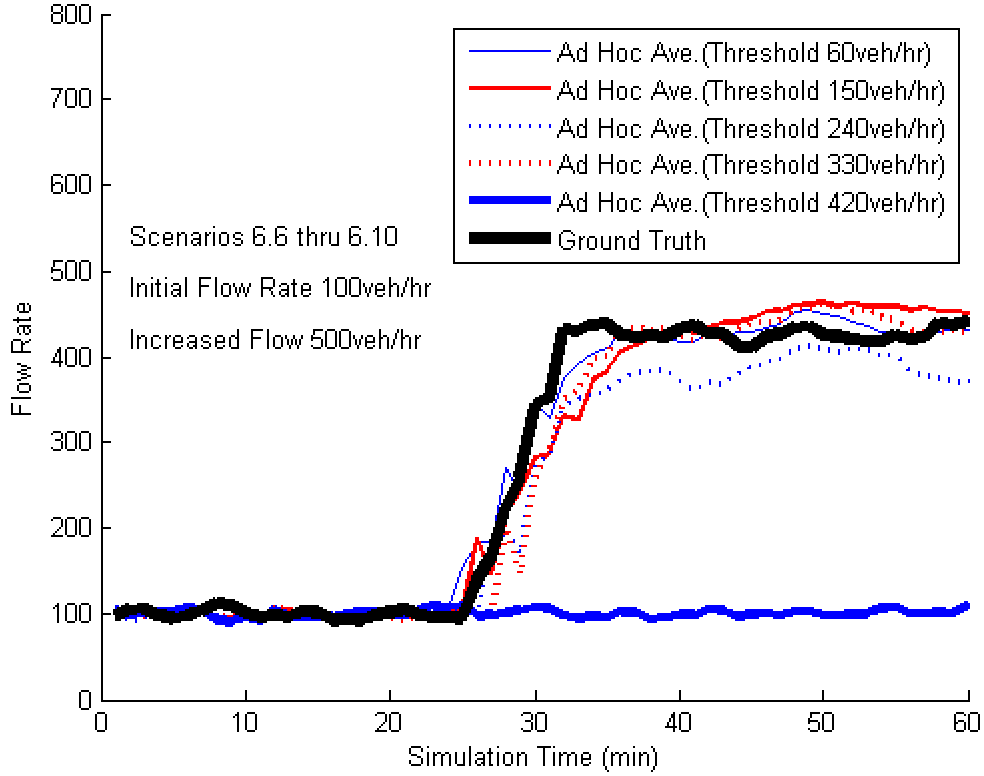

Figure 30 Flow Rate Comparisons (Scenario 6.6 through 6.10–Link 2)

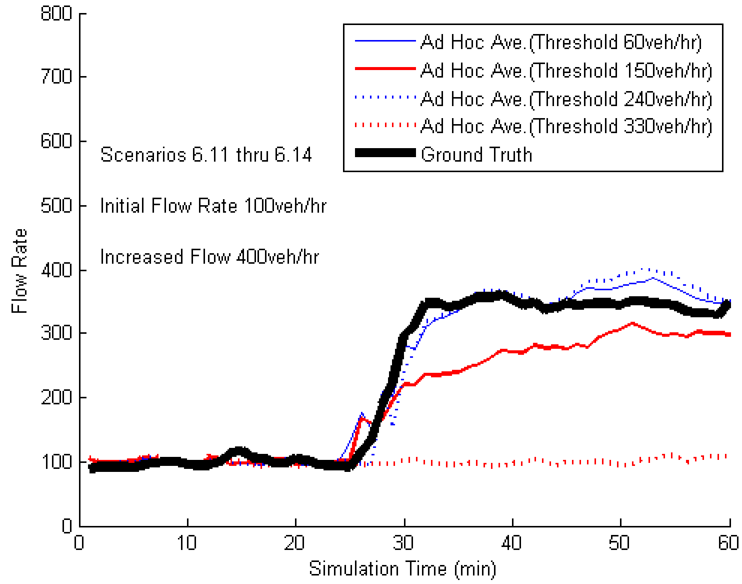

Figure 31 Flow Rate Comparisons (Scenario 6.11 through 6.14–Link 2)

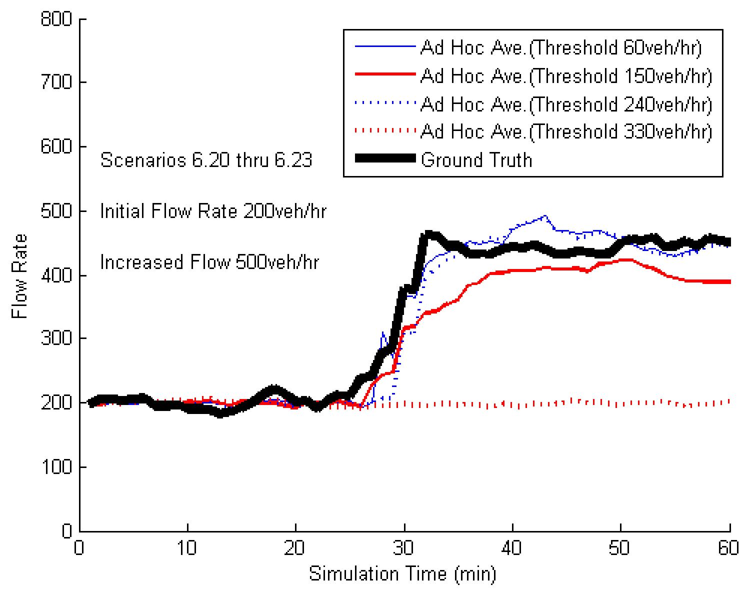

Figure 32 Flow Rate Comparisons (Scenario 6.20 through 6.23–Link 2)

Table 11 Rollback Thresholds Tested

Scenario No	Initial Traffic Flow Rate	Increased Traffic Flow Rate	Increase Increment	Rollback Threshold
6.6	100	500	400	60
6.7	100	500	400	150
6.8	100	500	400	240
6.9	100	500	400	330
6.10	100	500	400	420
6.11	100	400	300	60
6.12	100	400	300	150
6.13	100	400	300	240
6.14	100	400	300	330
6.15	100	300	200	60
6.16	100	300	200	150
6.17	100	300	200	240
6.18	100	200	100	60
6.19	100	200	100	150
6.20	200	500	300	60
6.21	200	500	300	150
6.22	200	500	300	240
6.23	200	500	300	330
6.24	200	400	200	60
6.25	200	400	200	150
6.26	200	400	200	240
6.27	200	300	100	60
6.28	200	300	100	150
6.29	300	500	200	60
6.30	300	500	200	150
6.31	300	500	200	240
6.32	300	400	100	60
6.33	300	400	100	150
6.34	400	500	100	60
6.35	400	500	100	150

Table 12 Flow Rate Mean Absolute Error (veh/hr/ln, Link 2)

Scenario NO	Initial flow rate	Volume at 20 minutes	Threshold	Mean Absolute Error (veh/hr/ln)			Number of Rollback
				Before increase 1-20 minutes	During increase 21-30 minutes	After Increase 31-50 minutes	
6.6	100	500	60vph (2veh/cycle)	7.1	22.6	26.0	203
6.7			150vph (5veh/cycle)	7.5	24.2	41.4	100
6.8			240vph (8veh/cycle)	9.2	35.2	61.0	50
6.9			330vph (11veh/cycle)	7.2	38.2	29.4	50
6.10			420vph (14veh/cycle)	11.0	67.6	323.3	0
6.11	100	400	60vph (2veh/cycle)	9.2	20.0	19.9	151
6.12			150vph (5veh/cycle)	9.0	25.1	84.3	65
6.13			240vph (8veh/cycle)	8.6	24.6	20.2	50
6.14			330vph (11veh/cycle)	11.6	52.4	247.6	0
6.15	100	300	60vph (2veh/cycle)	6.6	22.8	20.0	109
6.16			150vph (5veh/cycle)	6.8	26.1	19.9	50
6.17			240vph (8veh/cycle)	5.2	48.7	163.2	0
6.18	100	200	60vph (2veh/cycle)	7.3	18.9	22.0	54
6.19			150vph (5veh/cycle)	7.0	28.6	83.7	0
6.20	200	500	60vph (2veh/cycle)	10.1	20.1	31.6	135
6.21			150vph (5veh/cycle)	10.3	26.7	60.3	65
6.22			240vph (8veh/cycle)	9.8	35.4	33.4	50
6.23			330vph (11veh/cycle)	11.2	50.7	239.7	5
6.24	200	400	60vph (2veh/cycle)	8.8	17.5	24.4	120
6.25			150vph (5veh/cycle)	7.1	24.0	25.1	50
6.26			240vph (8veh/cycle)	9.5	30.7	164.8	5
6.27	200	300	60vph (2veh/cycle)	11.3	26.5	25.4	90
6.28			150vph (5veh/cycle)	14.9	34.3	83.3	0
6.29	300	500	60vph (2veh/cycle)	12.6	19.6	25.7	215
6.30			150vph (5veh/cycle)	13.4	17.8	28.8	50
6.31			240vph (8veh/cycle)	9.7	21.6	147.6	10
6.32	300	400	60vph (2veh/cycle)	9.1	19.1	27.7	75
6.33			150vph (5veh/cycle)	19.5	27.6	82.1	5
6.34	400	500	60vph (2veh/cycle)	20.5	21.6	23.5	161
6.35			150vph (5veh/cycle)	17.3	12.6	86.7	5

Table 13 Travel Time Mean Absolute Percentage Error (%, Segment 1)

Scenario NO	Initial flow rate	Volume at 20 minutes	Threshold	Mean Absolute Error (veh/hr/ln)		
				Before increase 1-20 minutes	During increase 21-30 minutes	After Increase 31-50 minutes
6.6	100	500	60vph (2veh/cycle)	2.5	6.9	1.6
6.7			150vph (5veh/cycle)	2.7	9.4	1.9
6.8			240vph (8veh/cycle)	2.3	2.3	3.4
6.9			330vph (11veh/cycle)	2.6	8.6	1.7
6.10			420vph (14veh/cycle)	3.2	12.3	13.4
6.11	100	400	60vph (2veh/cycle)	2.4	6.1	2.3
6.12			150vph (5veh/cycle)	2.7	6.9	4.7
6.13			240vph (8veh/cycle)	2.5	6.5	2.5
6.14			330vph (11veh/cycle)	3.3	8.8	8.3
6.15	100	300	60vph (2veh/cycle)	2.6	3.9	1.9
6.16			150vph (5veh/cycle)	2.5	4.2	1.9
6.17			240vph (8veh/cycle)	2.0	3.0	3.7
6.18	100	200	60vph (2veh/cycle)	2.9	2.6	1.8
6.19			150vph (5veh/cycle)	2.5	2.9	1.9
6.20	200	500	60vph (2veh/cycle)	1.7	4.7	1.7
6.21			150vph (5veh/cycle)	1.7	7.3	3.6
6.22			240vph (8veh/cycle)	1.6	6.6	1.5
6.23			330vph (11veh/cycle)	1.8	10.6	12.5
6.24	200	400	60vph (2veh/cycle)	1.5	1.8	1.9
6.25			150vph (5veh/cycle)	1.7	5.1	2.1
6.26			240vph (8veh/cycle)	1.5	7.4	9.0
6.27	200	300	60vph (2veh/cycle)	1.8	2.8	2.4
6.28			150vph (5veh/cycle)	2.1	3.1	2.4
6.29	300	500	60vph (2veh/cycle)	2.2	2.8	1.7
6.30			150vph (5veh/cycle)	1.7	3.1	1.4
6.31			240vph (8veh/cycle)	1.9	7.6	9.5
6.32	300	400	60vph (2veh/cycle)	1.9	2.5	1.9
6.33			150vph (5veh/cycle)	2.6	4.5	5.6
6.34	400	500	60vph (2veh/cycle)	2.3	1.9	1.5
6.35			150vph (5veh/cycle)	1.5	2.5	3.5

6.4 Summary

This chapter discussed the sensitivity of the system performance to 1) the geographical LP distribution over the network and 2) the size of the rollback threshold under varying combinations of traffic inputs for uncongested traffic conditions. The findings from the sensitivity analysis are summarized below:

More rollbacks were observed to occur when the LPs were widely spread over the network. This is likely a result of an increase in the number of links where rollbacks may be triggered and fewer LPs may contribute to the composite value of each link, creating more variation in aggregated flow rate. However, it is seen in the mean absolute percentage error (MAPE) analysis that there is no significant impact on the overall accuracy by the different geographical distribution of LP locations. While these results are based on a single random distribution of LPs, and may not be generalized to any possible distribution, it does demonstrate that it is possible to introduce randomness into the LP distribution and maintain reasonable results.

It was also seen that the system performance differs significantly with the size of the rollback threshold. Generally, as the threshold increases, the number of rollbacks decreases as expected. Additionally, a general trend is discovered that the estimate accuracy increases with smaller thresholds, although the accuracy is also found to be specific to the traffic input conditions.

CHAPTER 7 AD HOC DISTRIBUTED SIMULATION IN CONGESTED TRAFFIC CONDITIONS AND DISCUSSION

7.1 Introduction

Chapters 5 demonstrated that the ad hoc distributed simulation performs reasonably well under various traffic conditions. In Chapter 6, it was shown that the system performs well with different LP locations and various uncongested traffic input. This chapter investigates the performance of the ad hoc distributed simulations under congested traffic conditions. Also, a detailed discussion of the ad hoc simulation in congested traffic conditions will be provided.

7.2 Experimental Design

To compare the performance of the ad hoc distributed simulations under various congested traffic conditions, the same traffic network is used as in Chapter 5. Input flow rate and incident duration are tested to demonstrate the sensitivity of the proposed approach under congested situations where larger variations exist in flow control. This experiment setup is intended to show future research needs to better handle congested traffic conditions, such as introducing additional rollback criteria and different methods to represent congested conditions.

As mentioned above, the same traffic network is utilized as in Chapter 5. Also, the locations of LP remains the same, five LPs simulating the 3-by-3 grid network covering the western half of the network (white box in Figure 17), and the remaining five LPs (LP 6 thru LP 10) modeling the eastern half of the network (grey box in Figure 17).

The experiment in this section is intended to demonstrate how the ad hoc simulation performs under congested traffic conditions and its sensitivity in various traffic conditions. The performance of the model is examined not only during the incident, but also before and after the incident. Input flow rate is carefully selected in order to demonstrate all transitional traffic conditions in the limited simulation time period (90 minutes). For example, relatively high traffic input is necessary to create congested traffic conditions in the network. Also, incident duration should be sufficiently long to create congested conditions. However, if too high an input rate is selected with relatively long incident duration, it would take more time to return to the initial traffic condition after the incident is removed. Therefore, the incident duration selection should be short enough to show uncongested after traffic conditions in the given 90 minute simulation time period. The traffic inputs and incident duration were selected based on iterative experiments using the network wide model, ensuring the impacts of the incident could be modeled within the selected simulation time window. Based on this testing two traffic inputs (500 veh/hr/ln and 600 veh/hr/ln) and two different incident durations are selected for each traffic input (15 minutes and 20 minutes for 500 veh/hr/ln and 5 minutes and 10 minutes for 600 veh/hr/ln) were selected to demonstrate before, during, and after incident conditions

Table 14 Experimental Scenarios

Scenario NO	Input Volume (veh/hr/ln)	Incident	Note
7.1	500	5-20 minutes	
7.2	500	10-20 minutes	Scenario 5.5
7.3	600	10-20 minutes	
7.4	600	15-20 minutes	

7.3 Results and Discussion

As described in 3.4.4, two different types of traffic update are available, upstream to downstream in uncongested conditions, and downstream to upstream in congested conditions. Experiments in this section are intended to investigate the traffic data propagation from downstream to upstream in congested traffic conditions. Sensitivity analysis is conducted using the same network in Chapter 5 with two different traffic inputs and two different incident durations for each input. The same traffic measures (flow rate, travel time, and queue length) as in earlier experiments are collected for each scenario. Each scenario is replicated ten times and compared with the average of the ten ground truth replicate runs. Segment queue length represents the sum of the given performance measure over all links in the segment and segment flow rate is computed based on the numerical average of flow rates of the links in the segment, as presented in Chapter 5.

It is seen in (Figure 33 thru Figure 35) that the ad hoc approach is capable of reasonably modeling congested traffic conditions for the scenarios tested. For example, Scenario 7.1 provides the best match in queue length estimates while Scenario 7.2 and Scenario 7.3 over-predict queue length although provide better travel time estimates. Also, it is noted that the ad hoc simulation performance can be significantly different between replicate trials for the same scenario. For instance, significant variation is seen in the queue length estimates of Scenario 7.2 and Scenario 7.3. Recall the small flow over-estimation of Scenario 5.5 likely resulted in significant over estimations of queue length and travel time. Errors in traffic flow rate over time are significantly more critical in congested traffic conditions, as the rate of long term queue formation is directly related to the number of inbound vehicles over link processing capacity. Whereas in uncongested traffic conditions previous inbound traffic flow (as long as it is under capacity) leaves a link each cycle with no long term queue buildup. Thus, travel time and queue length estimates are significantly affected by the flow rate difference over time under congested conditions.

Recall that downstream congested traffic conditions are propagated toward upstream by metering vehicles on an exit link with speed control as described in 3.4.4. B. Unlike traffic updates from upstream to downstream, which are processed by changing the input rate of an entering link on downstream LP, flow rate is indirectly controlled by applying reduced speed onto vehicles at the exit link of a model to approximately match the flow dictated by the downstream constraint not directly modeled. Therefore, the intended traffic flow might not be achieved and more variation in flow rate can be involved in this process, resulting in reduced estimate accuracy.

As described above, the performance of the proposed ad hoc simulations in congested conditions is significantly impacted by minor deviations from the true exit flow values over an extended time period. For example, if the real-world traffic flow is near capacity and the ad hoc simulation traffic flow is slightly under capacity, the threshold value may not be violated with both operating at uncongested conditions. Over a short time period at uncongested conditions, the ad hoc simulation would provide reasonable estimates. However, if the real-world traffic flow falls into congested conditions and the ad hoc simulation fails to capture the buildup of unserved demand, the estimates of the ad hoc simulation will diverge from the true traffic measures on the field. Future research may consider 1) finding a better mechanism to meter the traffic flow without additional computational overhead and 2) tracking cumulative flow rate difference over time and implementing this as a secondary rollback criterion.

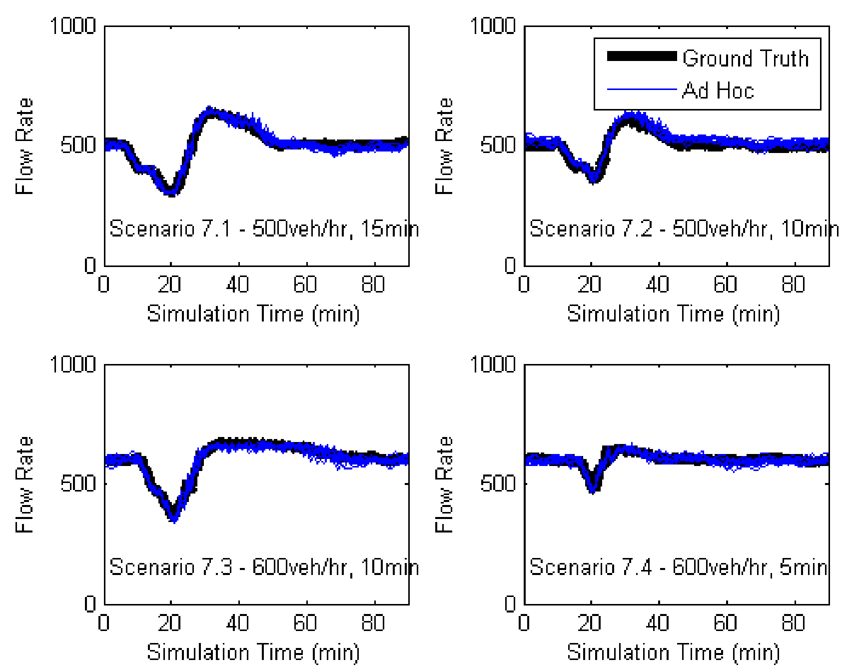

Figure 33 Flow Rate for Each Ad Hoc Replicate Trial and Average Ground Truth (Scenario 7.1 through 7.4–Segment1)

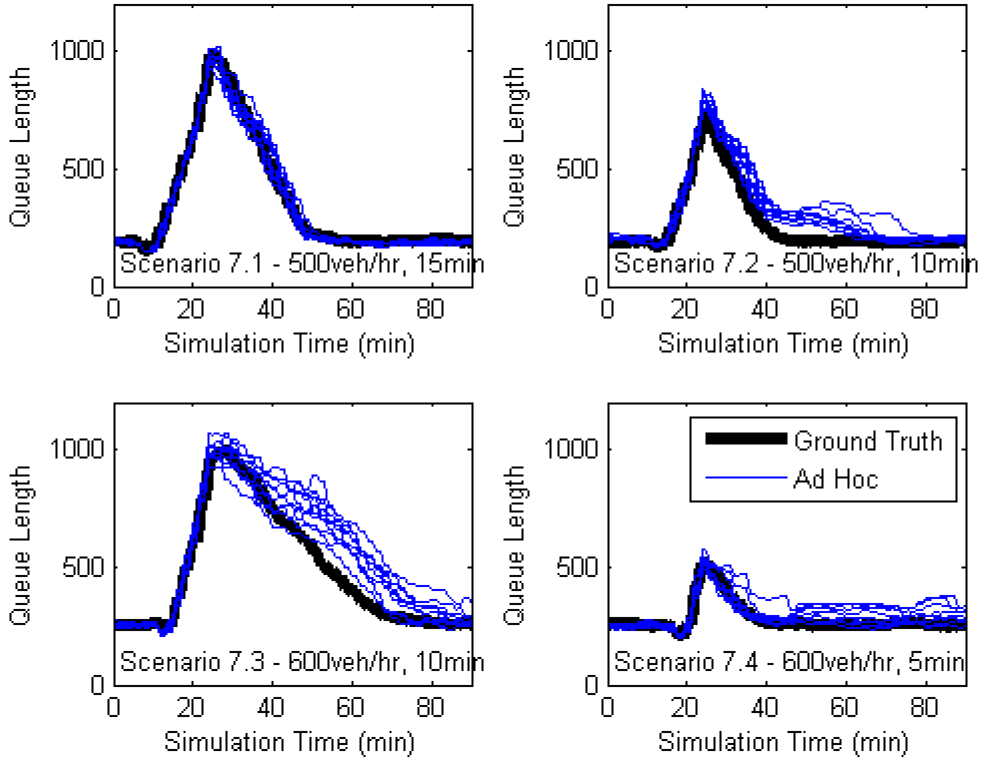

Figure 34 Queue Length for Each Ad Hoc Replicate Trial and Average Ground Truth (Scenario 7.1 through 7.4–Segment1)

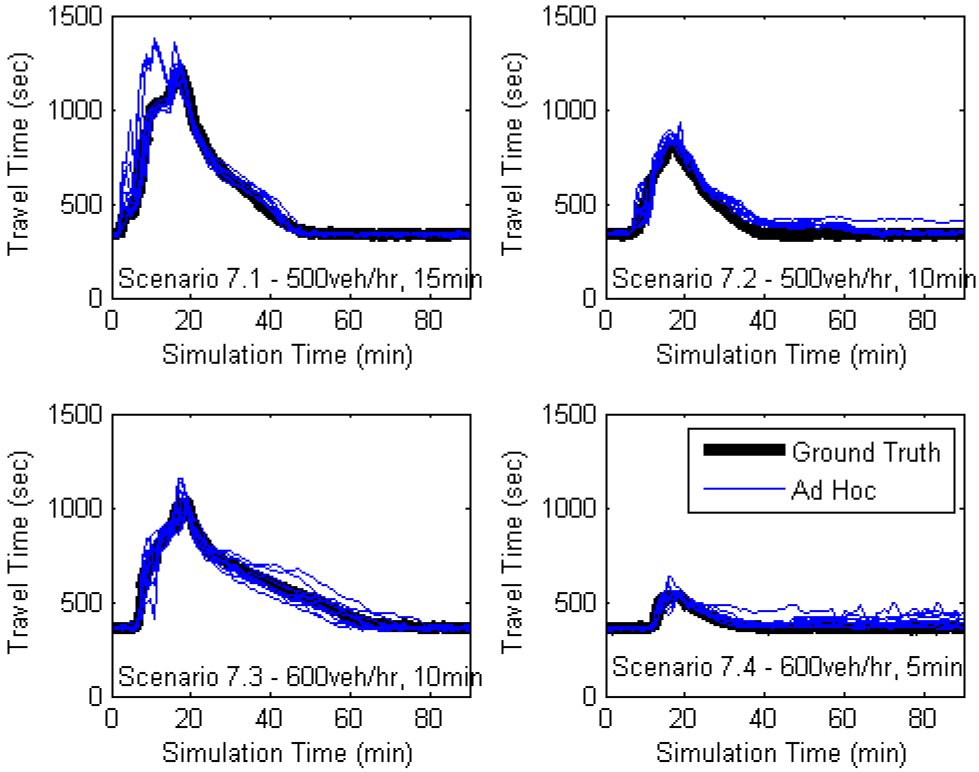

Figure 35 Travel Time for Each Ad Hoc Replicate Trial and Average Ground Truth (Scenario 7.1 through 7.4–Segment1)

7.4 Summary

This chapter described the ad hoc distributed simulation in congested traffic conditions. With two different flow rates and incident durations, it was demonstrated that the ad hoc approach is capable of reasonably modeling congested traffic conditions. However, higher variations were observed than under uncongested traffic conditions. The reasons for these deviations likely result from 1) traffic flow rate is controlled indirectly and 2) slight flow rate differences over time can significantly impact traffic performance measures under congested conditions. In order to capture the exact number of unserved vehicles in the system, several techniques are suggested, 1) better outflow control and 2) utilizing cumulative flow rate difference as a potential rollback trigger.

In conclusion, it is seen that the ad hoc simulation performs relatively well in the scenarios tested. It is also demonstrated that the cumulative flow difference over time is crucial to the system performance in congested traffic conditions.

CHAPTER 8 EVALUATION OF AD HOC DISTRIBUTED SIMULATION WITH REAL TIME FIELD DATA DRIVEN SIMULATION CLIENT

8.1 Introduction

As stated, the online ad hoc distributed traffic simulation can utilize projected state information from other LPs, real time embedded traffic sensor data, and historical traffic behavior patterns as input. In previous analysis in Chapter 5, 6, and 7, LPs exploit initial traffic input (given at the beginning of the simulation) and estimates from other LPs (received during the simulation). In this chapter, a real time field data driven simulation client is introduced representing the real time state estimate of the roadway network. This simulation is running at wall-clock speed, i.e. the real time simulation clock is synchronized with the wall-clock. In a field implementation this client would be replaced with the streaming detector data. This chapter evaluates the performance of the online ad hoc distributed traffic simulation model given real time information, exploring the predictive capability of this approach and feasibility of the large scale field implementation. In this evaluation, a volume increase scenario, similar to Scenario 5.4 in uncongested traffic conditions and an incident scenario, similar to Scenario 5.5, are analyzed.

8.2 Data Process with Real Time Field Data Driven Simulation Client

Data from the real time field data driven simulation client represents real time field sensor data, i.e. streaming detector data. For example, $RealTimeFieldDataDrivenSimulationLP_{i,j,k}^{m}$ corresponds to real time sensor data on link j, at Wall-clock time k, with data type m (flow, speed, travel time, delay, or queue length). Unlike $LP_{i,j,k}$, described in Section 3.4.2, traffic data from the real time field data driven simulation, $RealTimeFieldDataDrivenSimulationLP_{i,j,k}^{m}$ does not require an aggregation process as it corresponds to the traffic state from the field, not the estimated state. Therefore, it is converted to global variable $G_{j,k}^{m}$ (global state on link j at simulation time k with data type m - flow, speed, travel time, delay, or queue length) without any aggregation.

As demonstrated in Chapter 4, the Space-Time Memory saves all available predictions throughout the simulation duration. Thus, predictions at the current wall-clock time represent real time predictions of the system performance. The predictions are dynamic, updating as wall-clock time and the ad hoc simulation advance. The system's predictability can be measured based on 1) length of prediction horizon - how far in advance the system predicts at a specific wall-clock time and 2) how accurate the predictions are at specific point in the prediction horizon. For example, suppose at 7:00AM wall-clock time the system was able to provide traffic predictions until 7:30AM (thus. a 30 minute prediction horizon) and its predictions regarding the 10 minute period between 7:20AM and 7:30AM are found to be over predicted by 15% (the error at this point in the prediction horizon). However, at 7:10AM wall-clock time, with updated information regarding actual traffic conditions that occurred between 7:00AM and 7:10AM, the system was able to provide traffic predictions until 7:35 (a 25 minute prediction horizon) with a 7:20AM-7:30AM traffic prediction with a higher accuracy (5% difference). In this chapter, the accuracy of the available estimates with various near term horizon lengths (1-5 minute future predictions, 6-10 minute future predictions, and 11-15 minute future predictions) will be explored for two scenarios. Experimental design and results are presented in the next sections.

8.3 Experimental Design

This set of experiments utilizes the same traffic network as in Chapter 5 through 7. The primary experimental design difference is the addition of a field data driven simulation client which provides a real time state estimate of the roadway network. To fully investigate the ability of the ad hoc system to utilized real time data no LPs are initialized with accurate demand conditions. In previous experiments it was assumed that LPs were pre-configured to model the designated scenario. For example, upstream LPs had information about the volume increase in Chapter 6 and downstream LPs ran their simulation based on the given incident information in Chapter 7. However, in this chapter, initial rollbacks are expected to be instanced based on the field sensor data from the real time field data driven simulation client. The field sensor data is shared and propagated between LPs based on the ad hoc algorithms.

For these experiments LPs are uniformly distributed over the network. Locations of the eight LPs used for these experiments are shown in Figure 36. Each LP models a 3-by-3 grid network, centered on the vehicle location. For example, LP 8 models a network covering Fourth Street, Fifth Street, and Sixth Street with First Avenue, Second Avenue, and Third Avenue. The real time field data driven simulation client covers the entire 3-by-6 grid network representing real time traffic data. In a field implementation this client would be replaced with the streaming detector data.

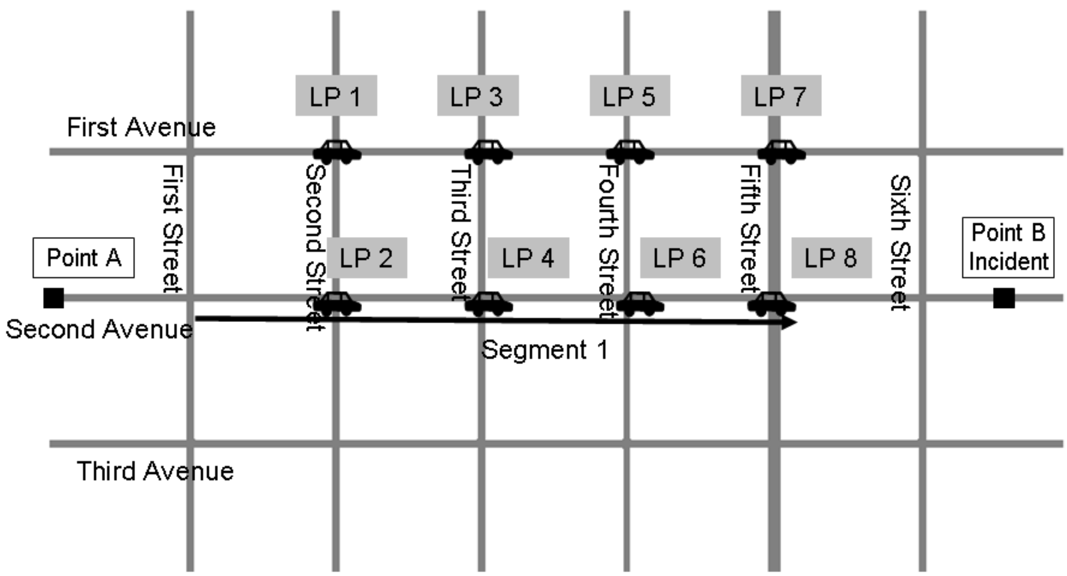

Figure 36 Logical Process Location

Two different traffic conditions are examined: a peak traffic scenario in uncongested traffic conditions and an incident scenario. The first scenario, Scenario 8.1, assumes that a sudden increase of eastbound traffic on Second Avenue is detected at Point A. This scenario explores how traffic flow change is transferred to downstream LPs. In the second scenario, Scenario 8.2, a traffic incident is assumed to occur eastbound on Second Avenue at Point B, reducing the average speed of vehicles from 48km/hr to 1 km/hr for a 900 seconds. It reduces the roadway capacity below demand, resulting in significant upstream queueing. This scenario models congested conditions and examines the responsiveness of the system to a downstream bottleneck. Average speed and flow rate are measured every minute for each link. Details about the scenarios are presented in Table 15. Each scenario with one real time field data driven simulation client and eight LPs is replicated 10 times with different VISSIM® random seed numbers.

Table 15 Experimental Scenarios

Scenario NO	Initial Flow Rate (veh/hr/ln)	Note
8.1	100	Volume Increase (500 veh/hr/ln) 20-40 minutes
8.2	500	Incident 10-25 minutes

8.4 Results and Analysis

As stated previously, the objective of the experiments in this chapter is to investigate performance of the proposed ad hoc simulation when real time field sensor data is available. Scenario 8.1 examines how the system adequately captures changes in traffic conditions when the traffic volume experiences a short duration peaking, in uncongested traffic conditions. Scenario 8.2 explores how well the ad hoc simulation operates under incident conditions. As the real time field data driven simulation client represents real time field data, the system's performance can be measured by the accuracy of the predictions at the current wall-clock time for future wall-clock times. Predictions for future traffic states at future wall-clock time can be found from global instance $G_{j,k,l}^{m}$, which corresponds to aggregated predictions on link j at simulation time k at wall-clock time l with data type m (flow, speed, travel time, delay, or queue length). For example, $G_{j,7:30AM,7:20AM}^{FlowRate}$ refers to aggregated flow rate predictions on link j of 7:30AM simulation time when current wall-clock time is 7:20AM.

To quantify the accuracy of the predictions, MAPE is calculated. Unlike the MAPE measures in Chapter 5 thru 7, which were calculated after the end of the simulation runs, MAPE measures in the analysis of predictive abilities are calculated for each wall-clock minute, since available predictions at each wall-clock time may differ. For example, $MAPE_{Wall-colck\,T\,\mathrm{min}}^{T+i\,\mathrm{min}}$, is the mean absolute percentage error for the prediction for T+i minutes calculated at wall-clock time T minutes. It is computed based on the T+i minute simulation time prediction at wall-clock time T minutes and the real time field data driven simulation client at T+i minute wall-clock time. MAPE of 1-5 minute, 6-10 minute and 11-15 minute future predictions can be computed to check the system's prediction performance with various near term horizon lengths.

8.3.1 Peak Traffic Scenario

Scenario 8.1 examines how the system adequately captures changes in traffic conditions when traffic volume is suddenly increased or decreased under uncongested traffic conditions. This is achieved by modeling under-capacity 100 veh/hr/ln traffic demand for 20 minutes (after initialization) followed by a sudden flow increase to 500 veh/hr/ln for 20 minutes on Second Avenue (Point A on Figure 36), with traffic then returning to the original 100 veh/hr/ln rate (Table 15).

In order to model the traffic volume changes over the network, new traffic volume information needs to be transferred from upstream LPs to downstream LPs. In this scenario, the real time field data driven simulation client is expected to reflect the increased traffic volume at 20 minute wall-clock time, under the assumption that the increased volume has been detected by field detectors. This traffic increase triggers the server to send a rollback message to the upstream LPs (LP 1 and LP 2 in Figure 36). LP 1 and LP 2 update their predictions with this new information and send their future traffic predictions regarding the links they are modeling. Based on the new predictions by LP 1 and LP 2, global variables, $G^{m}_{j,k,l}$ are updated in the Space-Time Memory and rollbacks are triggered on downstream LPs if necessary. This process is continued allowing the downstream LPs to received predictive data regarding the flow increase prior to the increase reaching the LPs' simulation area.

The system's performance will be measured by two attributes; 1) length of prediction horizon - how far in advance the system provides predictions at specific wall-clock time and 2) how accurate the predictions are at specific wall-clock time. By focusing on these two attributes, a comprehensive quantitative comparison is conducted to explore the quality of available predictions of the ad hoc distributed simulation approach. The accuracy of the available predictions is calculated for various near term horizon lengths (1-5 minute future predictions, 6-10 minute future predictions, and 11-15 minute future predictions).

Mean absolute error (flow rate) and Mean absolute percentage error (travel time) analysis are conducted. Detail of the calculation is as follows;

$$MAE\_FlowRate_{k,l}^{a-b\,\min} = \frac{\sum_{i=a}^{b} \left| Ad\ Hoc\ FlowRate_{k,l+i,l} - RealTimeField\ FlowRate_{k,l+i} \right|}{b-a+1}$$

$$MAE\_FlowRate^{a-b\,\min} = \frac{\sum_{l=1}^{n} \sum_{k=1}^{m} (MAE\_FlowRate_{k,l}^{a-b\,\min})}{m*n}$$

$$MAPE\_TT_{k,l}^{a-b\,\min} = \frac{\sum_{i=a}^{b} \frac{\left| Ad\ Hoc\ TT_{k,l+i,l} - RealTimeField\ TT_{k,l+i} \right|}{RealTimeField\ TT_{k,l+i}}}{b-a+1} \times 100$$

$$MAPE\_TT^{a-b\,\min} = \frac{\sum_{l=1}^{n} \sum_{k=1}^{m} (MAPE\_TT_{k,l}^{a-b\,\min})}{m*n}$$

Where:

$MAE\_FlowRate_{k,l}^{a-b\,\min}$: Mean absolute error of the ad hoc simulation (run number *k)* flow rate predictions for the next a-b minute simulation time at wall-clock time *l*.

$MAPE\_TT_{k,l}^{a-b\,\min}$: Mean absolute percentage error of the ad hoc simulation (run number *k)* travel time predictions for the next a-b minute simulation time at wall-clock time *l*.

$Ad\ Hoc\ TT_{k,l+1,l}$ Average of travel time predictions (run number *k*) produced by LPs for the next *i* minute simulation time at wall-clock time *l*

$RealTimeField\ FlowRate_{k,l+i}$ Flow rate from real time filed sensor data (run number *k*) at wall-clock time *l+i*.

$RealTimeField\ TT_{k,l+i}$ Travel time from real time filed sensor data (run number *k*) at wall-clock time *l+i*.

m Total number of replicate trial runs

k Number of replicate trial runs

n Number of intervals

For example, $MAE\_FlowRate_{k,l}^{1-5\min}$ represents mean absolute error of the ad hoc simulation (run number k) flow rate predictions for the next 1-5 minute simulation time at wall-clock time *l*.

Figure 37, Figure 38, and Table 16 show that the ad hoc distributed simulations are able to present a high degree of agreement with the field sensor data for the immediate near term future (1-5 minute future predictions). As expected, it is readily seen that the prediction accuracy on flow rate and travel time decrease with traffic states change at 20 minutes (when the increase begins) and 40 minutes (when the decrease begins). However, the ad hoc distributed simulations quickly adapt to the new traffic state and the overall accuracy of the ad hoc approach improves. In the replicated trials the increased arrival demand on average reaches Point B approximately 8 minutes after the initial increase at Point A. Since the upstream LPs (LP 1 and LP 2 in Figure 36) are able to reflect the new traffic state immediately after the increase, they are expected to demonstrate similar results as the field sensor data. However, the new traffic information is not available to the downstream LPs (LP 7 and LP 8 containing Point B) from the field sensors until 8 minutes after crossing Point A. Although, the downstream LPs in the ad hoc distributed simulations, coupled with the upstream LPs, were able to roll back and predict the increased/decreased traffic flow before the new traffic reached the field detectors. Exchanging predicted flow rate information between LPs in an ad hoc distributed simulation allows the downstream LPs to reflect the oncoming traffic changes.

Figure 37 and Figure 38 also demonstrate that the agreement between the ad hoc distributed simulations and the real time field sensor data is significantly reduced as the prediction horizon increases to 6-10 minutes and 11-15 minutes. It is not possible to have any updated prediction until the event is occurs on some area modeled by any LPs (Point A in this scenario) and is reflected on any LPs. Additionally, the propagation time is approximately 5 minutes from Second Street and Fifth Street. Therefore, the ad hoc simulation is not able to make accurate further future predictions over 6 minutes. Thus all of the prediction over the 6 minutes can be erroneous. The larger horizons will thus have more errors and it is believed that the length of prediction horizon is correlated with the propagation time which is function of the network size, vehicle propagation speed, and LP simulation speed. For example, the ad hoc simulation could make accurate further 30 minute future predictions, if the traffic propagation time is 30 minutes or more. This will be revisited later in this chapter.

Table 16 Mean Absolute Error and Mean Absolute Percentage Error (Segment 1)

Prediction		MAE / MAPE
Flow Rate	1-5 minute prediction	31.6 veh/hr/ln
	6-10 minute prediction	82.5 veh/hr/ln
	11-15 minute prediction	138.7 veh/hr/ln
Travel Time	1-5 minute prediction	6.6 %
	6-10 minute prediction	11.5 %
	11-15 minute prediction	16.3 %

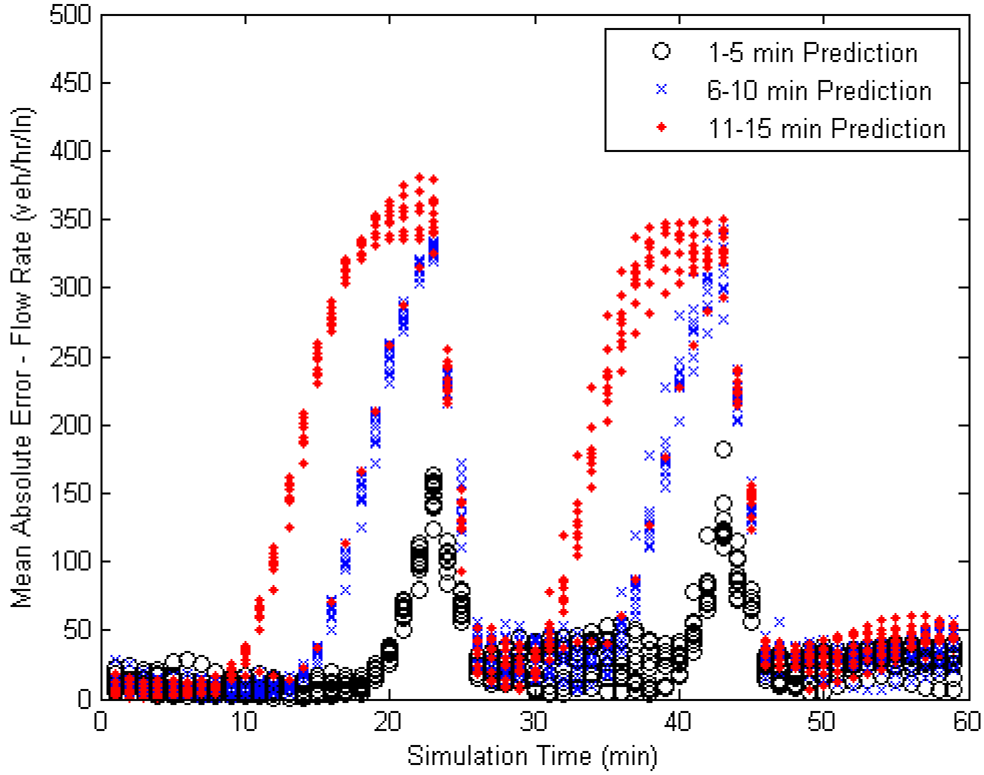

Figure 37 Ad Hoc Simulation with Real Time Field Data Driven Simulation Client (Segment 1 Flow Rate)

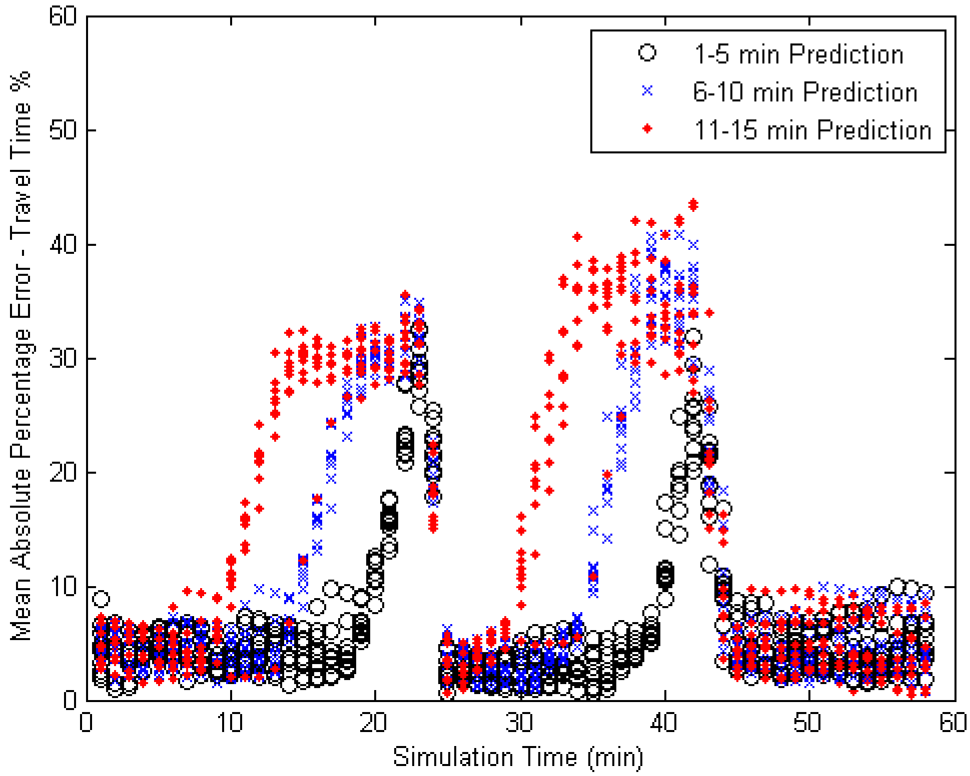

Figure 38 Ad Hoc Simulation with Real time Field Data Driven Simulation Client (Segment 1 Travel Time)

8.3.2 Incident Scenario

This scenario is intended to investigate responsiveness of the online ad hoc distributed simulations when under traffic incident conditions. As discussed in Chapter 3, traffic information transfer in this scenario is not as straightforward as in the uncongested traffic conditions. While upstream traffic information (flow rates) is propagated from upstream LPs toward downstream LPs in the volume increase scenario, downstream traffic information (speed reduction) is transmitted to upstream LPs from downstream LPs, as congestion builds from downstream to upstream. To investigate how the ad hoc simulations perform during before-incident, during-incident, and after-incident periods, Scenario 8.2 is constructed. A traffic incident is set to create congested conditions by reducing vehicle speed from 48 km/hr to 1 km/hr at Point B for 15 minutes (Figure 36). The incident starts at 10 minutes after the 20 minute warm up period. After an additional 10 minutes the vehicle queue extends to Second Street. The queue does not begin to clear from this link until after the incident is removed from Point B. This experiment allows for an investigation of how the system represents not only the congestion, but also the periods before, during, and after the congestion. Similar to Scenario 8.1, ten replicated runs with one real time data driven simulation client and eight LPs are conducted. After the runs, a comprehensive quantitative comparison is performed to examine how accurate the predictions are at specific prediction horizons.

First, the progress of the incident traffic conditions in the real time field data driven simulation client is described and how ad hoc distributed simulations successfully model the incident is explained later. Due to the incident at Point B capacity on Second Avenue is reduced significantly, reducing the average speed of vehicles from 48km/hr to 1 km/hr for 600 seconds and resulting in significant upstream queueing towards Point A. Average speed drops as the impact of the incident reaches the upstream links. It requires approximately about 15 minutes for the impact to reach Second Street. At the same time only limited traffic flow (far less than 500 veh/hr/ln input flow rate – approximately 50-100 veh/hr/ln) can be served. After the incident is cleared at 25 minutes, vehicles are able to pass Point B at free flow speed. However, more than 15 minutes is required to pass all the unserved vehicles in the queue and for the traffic to return to pre-incident state. Flow rate and travel time on the real time field data driven simulation client are depicted in Figure 39. Also, a three dimensional plot of travel time of the real time field data driven simulation client is presented in Figure 40 by 5 minute time interval. It is shown that the Segment 1 travel time reaches approximately 700 seconds during the congestions, while approximately 350 seconds is the uncongested travel time.

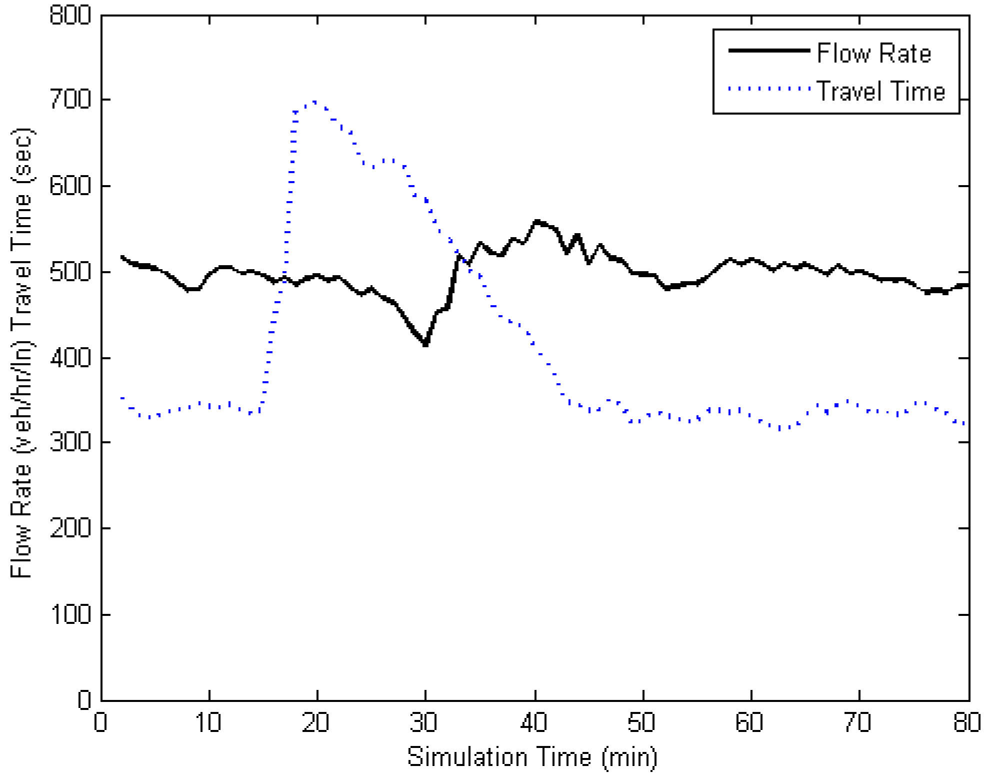

Figure 39 Real Time Field Data Driven Simulation Client (Segment 1 Flow Rate and Travel Time)

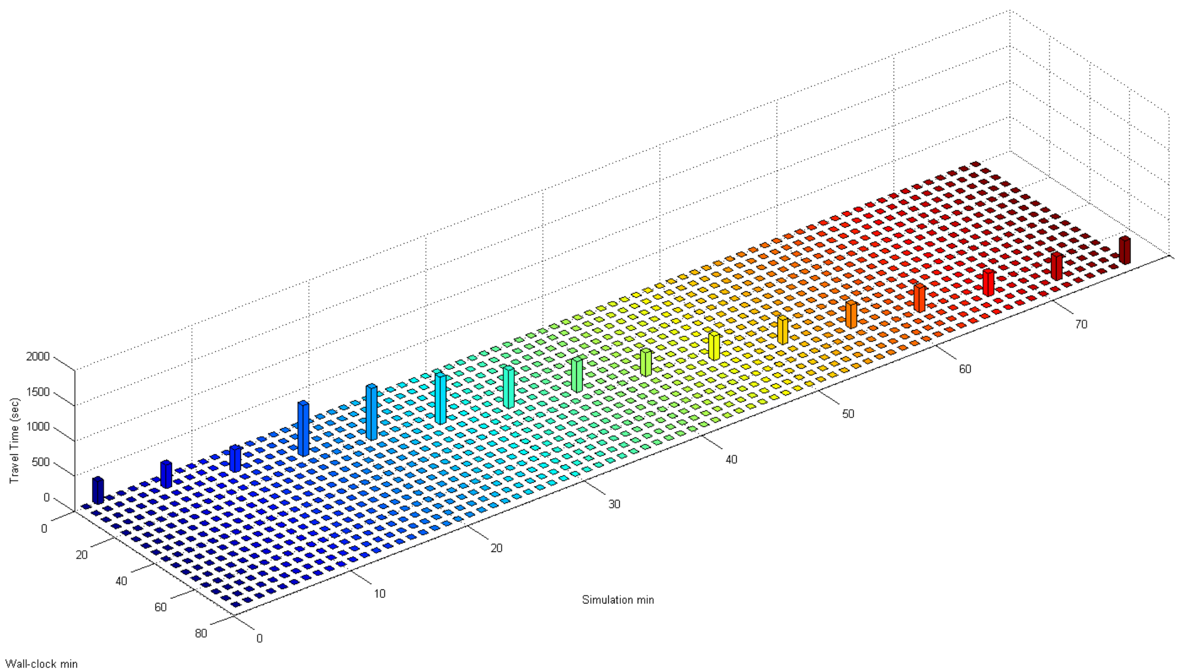

Figure 40 Real Time Field Data Driven Simulation Client 3-D Plot (Segment 1 Travel Time)

This traffic condition is reproduced in the ad hoc distributed simulations, as follows. Each LP is running its simulation based on the initial flow rate. No LP has information about the incident until the server receives the low speed and low flow rate from the real time field data driven simulation client and sends rollback messages to corresponding LPs. The incident starts at Point B reducing the average speed of vehicles from 48km/hr to 1 km/hr for 900 seconds. It results in significant upstream queueing on Second Avenue such that measured traffic flow and speed of the real time field data driven simulation client are significantly reduced. Right after the incident start at 10 minute wall-clock time, the real time field sensor data starts to show lower speed and lower flow rate on the link at Point B. The server receives the low speed and low flow rate from the real time field data driven simulation client and find a rollback threshold violation between the data from the real time field data driven simulation client and the already received estimates from LP 7 and LP 8. The server then issues a rollback to LP 7 and LP 8 in Figure 36. They begin to update their future traffic predictions assuming newly received traffic conditions continue. Reproducing congested conditions on LP 7 and LP 8 is accomplished by controlling outflow rate by altering the 'desired speed' of each vehicle on the link. When LP 7 and LP 8 update their predictions and the difference between their predictions and the flow rates already predicted by LP 5 and LP 6 (which did not have the incident information) violates the rollback threshold and the average speed predicted by LP 7 and LP 8 is below the speed threshold in Table 3, rollbacks are triggered on the upstream LPs (LP 5 and LP 6). In a similar fashion, LP 3 and LP 4 (and LP 1 and LP 2 later) make a rollback as the queueing continues to build up towards Point A. This allows congested traffic information to be passed to the upstream LPs, even before the impact of the incident actually reaches the area which the upstream LPs are modeling. Once there is another threshold violation (i.e. incident is removed), updated information is again transmitted from the real time field data driven simulation client to LP 7 and LP 8 and from LP 7 and LP 8 to other LPs in the same way.

Figure 41 illustrates a three dimensional plot of travel time predictions and the real time field sensor data with wall-clock time on the y axis. Initially (at 0, 5, and 10 minute wall-clock time) predictions are available until 80 minutes of simulation time (Area A in Figure 41). These predictions were made during the 20 minute warm up time period. Travel time is predicted to be approximately 350 seconds, as these predictions are constructed without knowledge of the incident (as the incident has not yet occurred). Once a rollback is triggered by the incident, existing predictions on the rolled back clients are removed from the Space-Time Memory and they are updated with new predictions based on updated rollback information. Until the ad hoc simulation receives new traffic information, it is seen that travel time is predicted to continue increasing (Area B) since the current traffic condition is assumed to continue. Therefore, it is anticipated that the ad hoc simulations make predictions with high accuracy if estimated incident clear-up time information is provided. Empty cells in Area C show that the predictions beyond 50 minute simulation time are not available at 25, 30, and 35 minute wall-clock time, as the earlier prediction have been removed and sufficient computational time has not yet passed to allow updated predictions at this point in the time horizon. Finally it is seen that when the impact from the incident disappears around 40 minute wall-clock time the ad hoc simulation is able to adjust predictions to reflect his new data (Area D).

Using the same method with Scenario 8.1, MAE and MAPE are calculated (Table 17). Figure 42 and Figure 43 show considerably higher MAE/MAPE, compared to Scenario 8.1. This implies that the ability of the ad hoc distributed simulations to reflect congested traffic condition due to incidents is reduced, as discussed in Chapter 6. This is an expected outcome. The simulation performance worsens in the incident scenario as the outflow constraint by speed does not provide highly accurate flow control. In addition more randomness is involved in modeling congested networks. However, it is revealed that the ad hoc simulations offer reasonable replicates of the real time field data driven simulation client for the immediate future travel time predictions (1-5 minutes) and are capable of providing reasonable predictions for the longer horizons although delay exists in updating predictions.

Table 17 Mean Absolute Error and Mean Absolute Percentage Error

Prediction		MAE / MAPE
	1-5 minute prediction	20.0 veh/hr/ln
Flow Rate	6-10 minute prediction	27.3 veh/hr/ln
	11-15 minute prediction	40.1 veh/hr/ln
	1-5 minute prediction	25.5 %
Travel Time	6-10 minute prediction	45.5 %
	11-15 minute prediction	63.1 %

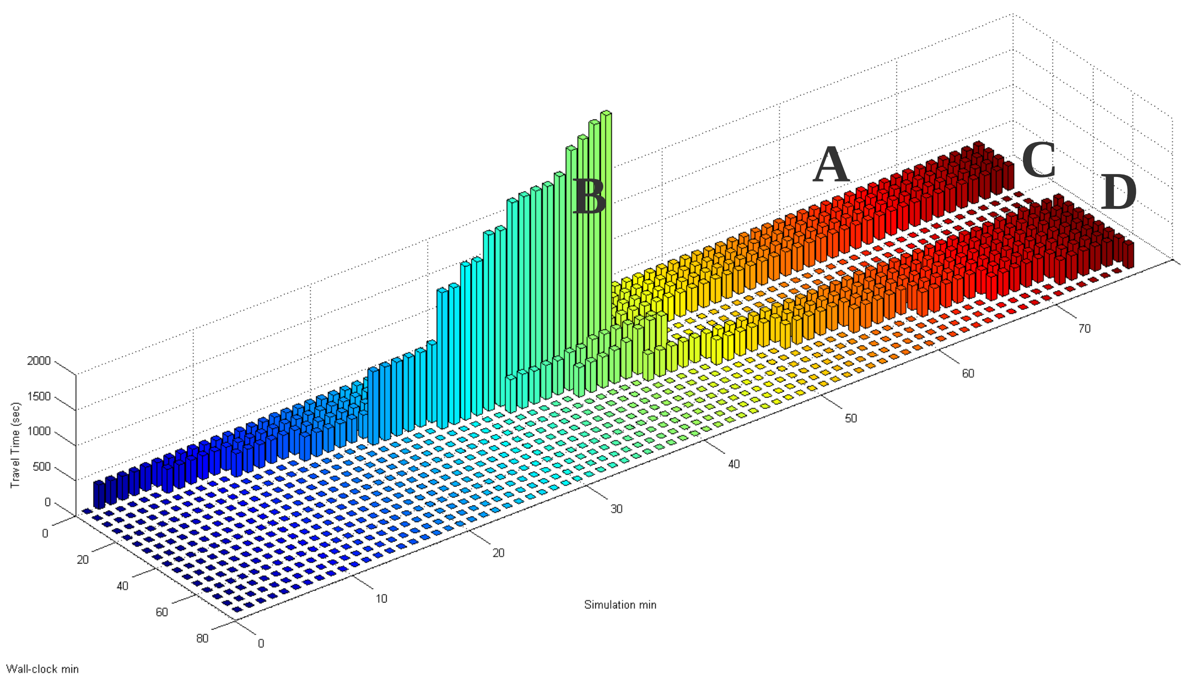

Figure 41 Ad Hoc Simulation Segment 1 Travel Time Prediction

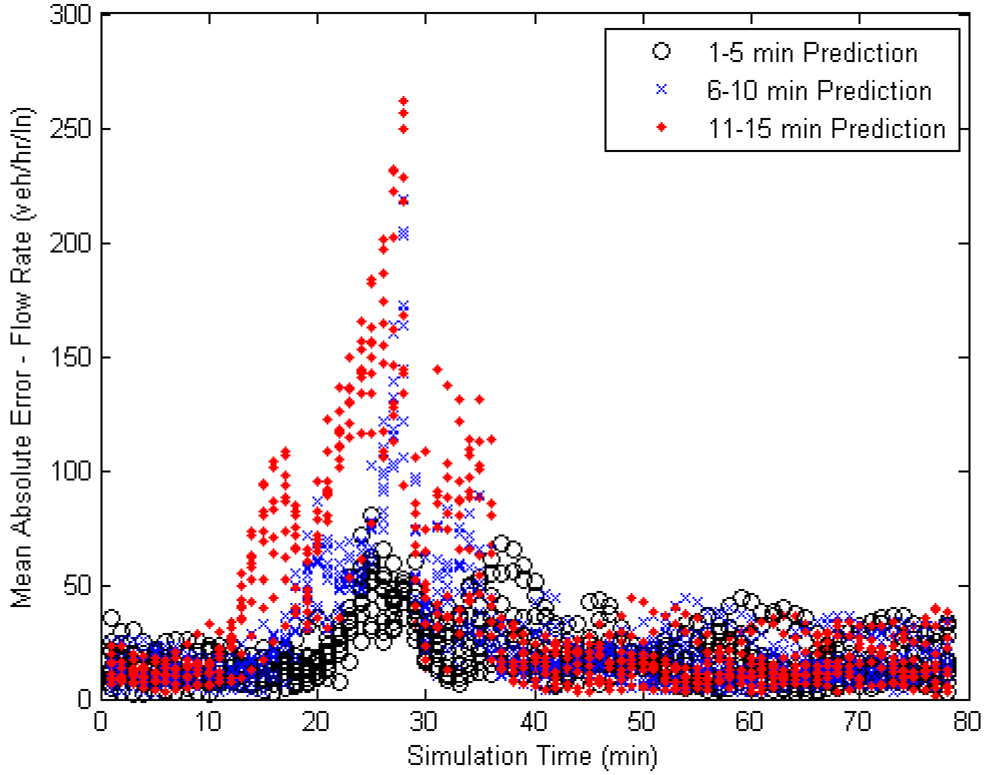

Figure 42 Ad Hoc Simulation with Real Time Field Data Driven Simulation Client (Segment 1 Flow Rate)

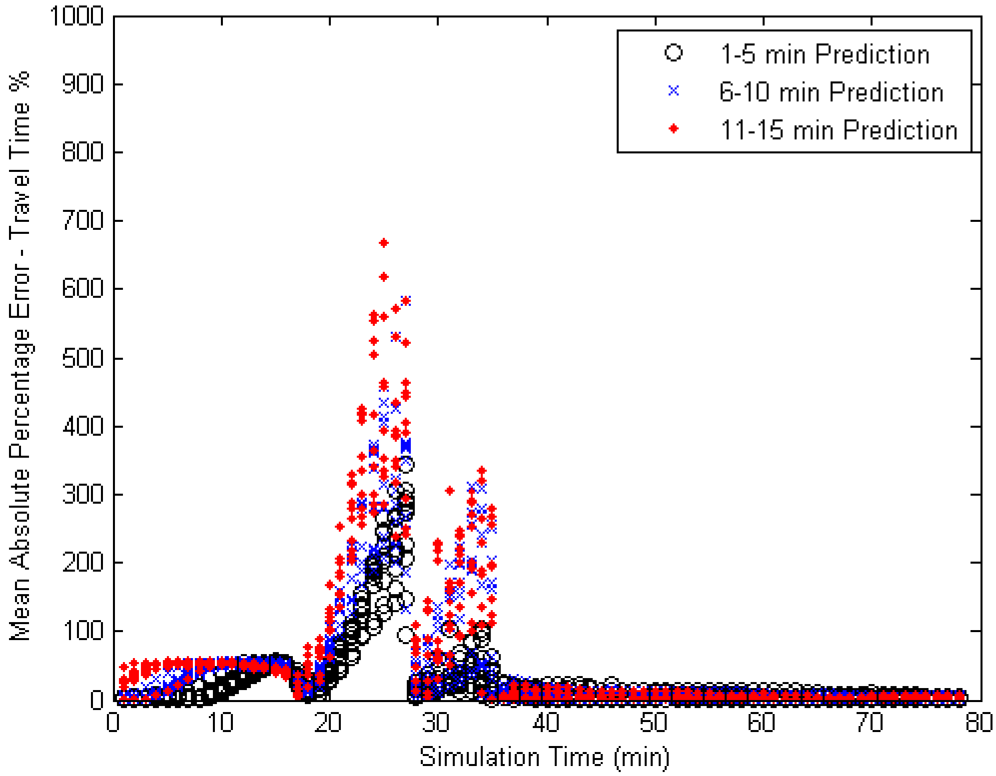

Figure 43 Ad Hoc Simulation with Real Time Field Data Driven Simulation Client (Segment 1 Travel Time)

8.4 Summary

This chapter presented the performance of the ad hoc approach when real time field sensor data is available. The real time field data is represented by the real time field data driven simulation client and this client would be replaced with the streaming detector data in a field implementation. Scenario 8.1 examined how the system adequately captures changes in traffic conditions when traffic volume is suddenly increased and decreased in uncongested traffic conditions. Scenario 8.2 investigated how well the ad hoc simulation operates when a traffic incident occurs.

It was found that the proposed ad hoc distributed simulation is capable of capturing dynamically changing traffic conditions on both the peak traffic scenario and incident scenario. In both scenarios the prediction accuracy drops when the traffic state changes. Additional performance degradation is seen in the incident scenario, since the predictions are produced based on the assumption that current traffic conditions continue, i.e. potential incident clearing is not assumed. However, for the immediate future predictions, the proposed simulation presents relatively good prediction capability.

CHAPTER 9 CONCLUSIONS

This chapter summarizes the findings from this research in Section 9.1. Contributions are presented in Section 9.2. Future works are suggested in Section 9.3.

9.1 Summary of Findings

This research developed an online ad hoc distributed traffic simulation using optimistic execution. The proposed model was described in Chapter 3 and graphically demonstrated in Chapter 4. The proposed model was examined under different traffic conditions, including steady traffic state, volume increase, and incident scenarios (Chapter 5). Also, the performance of the model was further investigated by 1) a sensitivity analysis of the model under uncongested traffic conditions with regard to geographical distribution of logical process and varying rollback thresholds (Chapter 6) and 2) a sensitivity analysis of the model under congested traffic conditions for several demand levels and incident scenarios (Chapter 7). Finally, the model was evaluated given real time field sensor data allowing for real time state predictions of future roadway network performance (Chapter 8). The findings of this research are as follows;

- Integration of communication middleware and traffic simulation: Communication middleware allows the distributed simulation to perform on multiple platforms.

- Dissemination of traffic information across the multiple LPs: Object-oriented client/server technology helps to efficiently disseminate traffic estimates and predictions and incorporate this data across the multiple LPs.
- Space-Time Memory management: A local central server is able to coordinate the traffic states from multiple logical process simulations. The traffic states from multiple logical process simulations can be projected by multiple logical processes and are not required to be received in time-stamp order.
- Optimistic (rollback-based) synchronization protocol: Optimistic execution inspired by Time Warp mitigates the synchronization problem allowing each logical process to execute asynchronously.
- Ad hoc distributed simulation in steady traffic state, volume increase, and incident scenarios: The ad hoc approach provides very comparable results with the large scale model under various steady and non-steady traffic conditions.
- Ad hoc distributed simulation under uncongested traffic conditions with different distributions of geographical logical process: As the number of links where rollbacks may be triggered increases and fewer logical processes contribute to the composite value of a link (creating more variation in aggregated flow rate) an increased number of rollbacks is seen. No significant impact is found on the overall accuracy by the different geographical distributions of logical processes tested.
- Ad hoc distributed simulation under uncongested traffic conditions with different rollback thresholds: A general trend is discovered that the prediction accuracy

increases with smaller thresholds, although in certain conditions the accuracy is also found to be specific to the traffic input conditions.

- Ad hoc distributed simulation under congested traffic conditions for several traffic demands and incident scenarios: While increased deviation in the estimations is present than in the uncongested volume increase scenarios, the ad hoc approach is capable of reasonably modeling congested traffic conditions.
- Ad hoc distributed simulation when real time field sensor data is available allowing for real time state predictions of the roadway network performance: In both peak traffic scenario and incident scenario the prediction accuracy drops when the traffic state changes. However, the ad hoc approach appears generally capable of capturing dynamically changing traffic conditions when the real time field sensor data is available.

9.2 Contributions

Transportation impacts every aspect of daily life. For many decades efforts to improve transportation have been made to ensure quality of life and higher standards of living. However, the potential benefits of the utilization of real time traffic data have not yet been fully achieved. Recent advancements in sensor, mobile computing, and wireless communication technologies is creating new opportunities to effectively exploit real time traffic data. In the presented research LPs collect, process, and simulate traffic states in a distributed fashion and a central server coordinates the overall simulation with an optimistic execution technique. Such a distributed approach can provide more up-to-date and robust predictions with decreased communication bandwidth requirements and increased computing capacity.

This research effort provided the following contributions:

- Development of a new approach to distributed traffic simulations: Traffic simulation and data processing are performed in an online ad hoc distributed fashion by multiple logical process simulations, which model small portions of the overall network.
- Integration of TRTI (communication middleware) and traffic simulation: TRTI, developed as a parallel effort of other researcher is integrated with traffic simulation. This integration manages the distributed network to synchronize the predictions among logical processes.
- Implementation of Space-Time Memory management into a transportation simulation approach: The predictions across the multiple logical processes are aggregated, transferred into composite values, and saved in Space-Time Memory.
- Created an optimistic (rollback-based) synchronization protocol for an asynchronous distributed transportation simulation: Optimistic execution inspired by Time Warp addresses the synchronization problem across LPs, allowing each LP to execute asynchronously. Invalidated predictions are updated quickly by this mechanism to ensure more robust and reliable estimates and predictions.
- Demonstrated the feasibility of the ad hoc distributed traffic simulation under steady state, volume increase, and incident scenarios: The ad hoc distributed

simulation provides very comparable results with the large scale simulation under various steady and non-steady traffic conditions.

- Investigated the sensitivity of the ad hoc distributed traffic simulation under uncongested traffic conditions with different geographical distributions of LPs and several rollback thresholds: The sensitivity analysis provided insights into the parameters of the ad hoc approach and guidance for future research and field implementations.
- Explored the performance of the ad hoc distributed simulation under congested traffic conditions for several traffic demands and incident scenarios: This congested traffic experiment supported the robustness of the system and the likelihood that a large-scale implementation of the model in real-world settings could be successful.
- Developed a methodology to incorporate real time field sensor data into the ad hoc distributed traffic simulation allowing for predictions of near term traffic conditions: The ad hoc distributed traffic simulation receives the data feed from the real time field sensor data and incorporate them in its model.

Finally, this research is provides a framework for an ad hoc distributed simulation which features dynamic collections of logical processes interacting with each other and with real time field data. The ad hoc distributed simulation with optimistic execution is able to capture, process, and incorporate data into simulation models, and transfer useful information with reasonably fast response time.

9.3 Future Research

This research used simplified communication and traffic simulation attributes such as perfect communication environments without errors, unchanging threshold, and constant turning ratios. To facilitate successful field implementation, there are additional research tasks needed to improve the model's accuracy and robustness. Figure 44 shows necessary steps toward successful large scale field implementation of the proposed ad hoc distributed traffic simulation model. As seen in Figure 44, the previous chapters in this study developed an ad hoc distributed traffic simulation model (Chapter 3) and demonstrated the feasibility of the model under various traffic scenarios (Chapter 5), the impact of various rollback thresholds and different LP locations (Chapter 6), the performance of the model under congested traffic conditions (Chapter 7), and the feasibility of the model with real time sensor data (Chapter 8). Based on the finding of the study, future research should evaluate the model with more realistic assumptions and develop new implementation methodologies to make the model more robust. The following sections will detail future research topics.

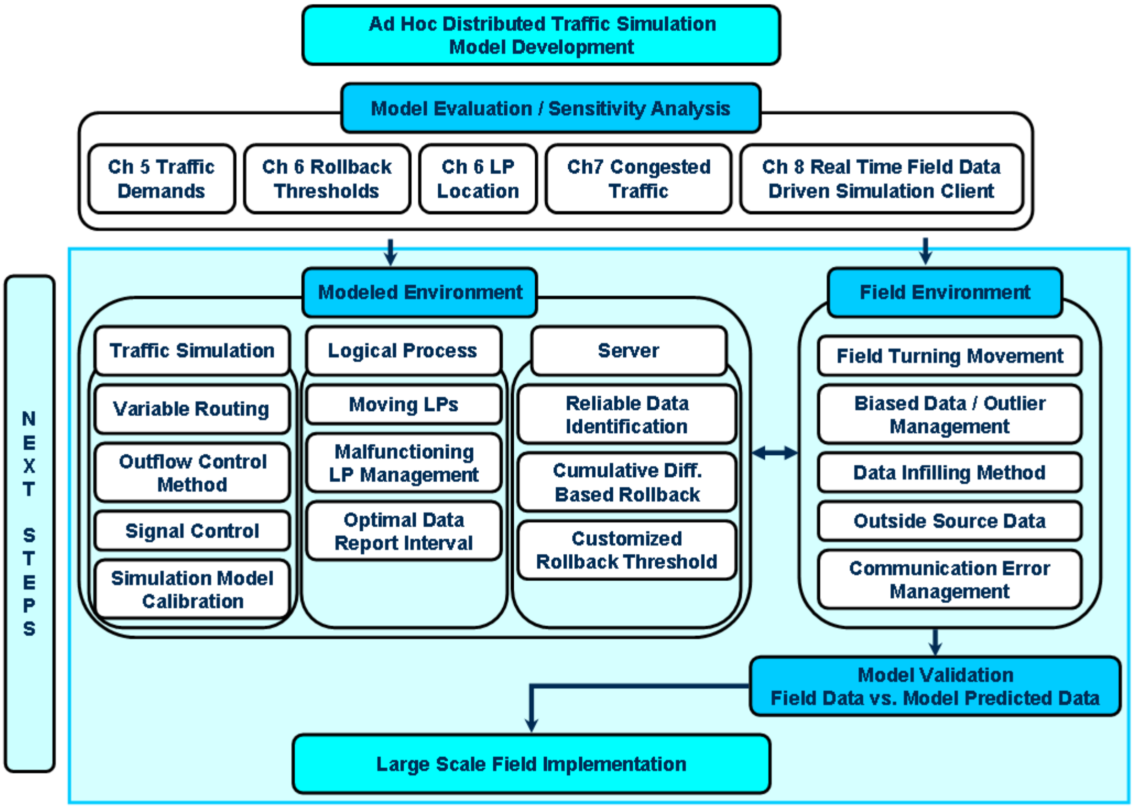

Figure 44 Future Research Map

9.3.1 Future Research Task - Modeled Environment (Traffic Simulation)

- Variable routing: In this study, turning movement ratios at intersections are assumed to be constant over time, which is not realistic, especially in congested traffic conditions when drivers change their routing decisions to avoid delays. Additional experiments will be conducted with varying turning ratios and errors in turning ratios.
- Outflow control method: In Chapter 7, it was demonstrated that the proposed outflow control method by applying corresponding speed on exiting vehicles is

not highly accurate. Alternative methods can be investigated to better reflect congested traffic conditions.

- Signal control: Fixed signal timing plan was utilized in this study. Future research should investigate how to incorporate various historic and online signal control data in the model, including actuated signal control and pedestrian push bottom crossing.
- Simulation model calibration: Traffic simulation should be calibrated to reflect the local traffic condition / driver behavior more accurately. A methodology should be developed to implement a calibration process into the ad hoc distributed traffic simulation model. Further, real time calibration should be considered and it will enable the proposed model to become mode robust.

9.3.2 Future Research Task - Modeled Environment (Logical Process)

- Moving LPs: In this study, LPs are assumed to be stationary and model their neighboring areas. In the large scale field implementation, it is envisioned that LPs may be mobile. Therefore, the modeling area can change over time and a proper method to incorporate the changes in the modeling area during simulation should be investigated.
- Malfunctioning LP management: The experiments in this study were performed based on the perfect simulation environment assumption. However, in the large scale field implementation LPs might perform improperly providing inaccurate

data. This can affect the accuracy of the entire system. A methodology should be developed to manage LPs if necessary.

- Optimal data report interval: LPs collect traffic measures at every minute and aggregate into 4 minute average in this study. However, the data collection frequency and aggregation interval should be further examined to ensure the data communication efficiency and the model performance.

9.3.3 Future Research Task - Modeled Environment (Server)

- Reliable data identification: As discussed before, the experiments in this study were performed in a perfect simulation environment. Field data is expected to have higher variations with biased data points. It is necessary to have a filtering process to identify more likely reliable data in Space-Time Memory. Both erroneous data and the potential for internally false data stream should be considered.
- Cumulative difference based rollback: In this model, rollback criteria compare only cross-sectional flow difference and can not identify cumulative demand differences over time. However, it was found in Chapter 7 that small differences over time can lead to significant different in traffic measures in congested traffic conditions. The cumulative difference can complement the model as secondary rollback criteria to increase the robustness of the model.
- Customized rollback threshold: Rollback thresholds can be designed based on 1) absolute difference in flow rate, 2) relative difference in flow rate, or 3) other

criteria. For example, a tight threshold can be selected for a highway link where traffic flow is constant most of the time. Also, thresholds can be customized to meet the objective of the system management. Strict thresholds can be applied when the traffic management wants to detect any traffic changes in a certain time period. This variable rollback threshold selection should be investigated the robustness and flexibility of the model.

9.3.4 Future Research Task – Field Environment

- Field turning movement: Accurate turning movement is necessary to reflect the accurate traffic states. Current technologies can detect turning movements through videos or loop detectors if a lane is designated for one movement. Estimating turning movements for a lane which multiple movements share is challenging. New methodologies should be developed to estimate real time field turning movement and implement the field turning movement into the simulation model in real time.
- Biased data / outlier management: Traffic predictions are provided by multiple LPs. It is envisioned that LPs use their own simulation packages. Therefore, the predictions can be biased and inconsistent with predictions from other LPs. An algorithm should be designed to distinguish biased data and handle outlier data properly in the field implementation.
- Data infilling method: Since LPs are envisioned moving over the network, it is possible that some area might not be modeled by any LPs for a certain period of

time. Realistic assumptions can be made, for example using neighboring link data and lastly reported data. Methodologies should be developed to infill these missing data.

- Outside source data: In the algorithmic approach, predictions are made based on the assumption that current traffic conditions will continue. For example, in Scenario 8.2 when the congestion from the incident builds up, the predicted delay will continue to grow the entire prediction horizon length, regardless potential future clearing of the incident. Incorporating outside source information, such as expected incident clear-up time and planned event information may improve the prediction accuracy.
- Communication error management: The proposed model is developed based on the perfect communication environment assumption. Communication error including communication message loss, messages in reverse order, and messages over buffer limit should be examined for a successful field implementation of the model.

9.3.4 Future Research Task – Model Validation

- Model validation: All steps described above are expected to have a positive contribution to the robustness of the model. However, the output of the model needs to be validated with the field data. Based on the validation results, the model can be calibrated more to increase the performance. A variety of statistical tests should be employed to compare the field data and the predictions from the model.

APPENDIX A: SERVER SCRIPT

```
Imports System.Threading
Imports System.Net
Imports System.Net.Sockets
Imports System.Runtime.InteropServices
Imports Microsoft.Office.Interop
Imports System.Math
Imports System
Imports system.net.mail
Public Class Form2
    Public Declare Auto Sub TRTI_Initialize Lib "trti.dll" (ByVal port As Integer, ByVal a As VB_Reflect)
    Public Declare Auto Sub TRTI_ProcessAllMessagesInQueue Lib "trti.dll" ()
    Public Declare Auto Sub TRTI_getOneMessage Lib "trti.dll" ()

    Delegate Sub VB_Reflect(ByVal a As Integer, ByVal b As String, ByVal c As Integer)
    Public Declare Auto Function TRTI_addToGroupAt Lib "trti.dll" (ByVal destination As IntPtr, ByVal group
As IntPtr) As Integer
    Public Declare Auto Function TRTI_removeFromGroupAt Lib "trti.dll" (ByVal destination As IntPtr, ByVal
group As IntPtr) As Integer
    Public Declare Auto Function TRTI_createNewGroupAt Lib "trti.dll" (ByVal destination As IntPtr, ByVal
group As IntPtr) As Integer
    Public Declare Auto Function TRTI_sendMsgToGroupAt Lib "trti.dll" (ByVal destination As IntPtr, ByVal
group As IntPtr, ByVal message As IntPtr) As Integer

    Public Declare Auto Function TRTI_reliable_addToGroupAt Lib "trti.dll" (ByVal destination As IntPtr,
ByVal group As IntPtr) As Integer
    Public Declare Auto Function TRTI_reliable_removeFromGroupAt Lib "trti.dll" (ByVal destination As
IntPtr, ByVal group As IntPtr) As Integer
    Public Declare Auto Function TRTI_reliable_createNewGroupAt Lib "trti.dll" (ByVal destination As
IntPtr, ByVal group As IntPtr) As Integer
    Public Declare Auto Function TRTI_reliable_sendMsgToGroupAt Lib "trti.dll" (ByVal destination As
IntPtr, ByVal group As IntPtr, ByVal message As IntPtr) As Integer
```

```
    Public Declare Auto Function TRTI_reliable_sendSingleMsgToGroupAt Lib "trti.dll" (ByVal destination As
IntPtr, ByVal group As IntPtr, ByVal message As IntPtr) As Integer

    Public reflectptr As VB_Reflect = AddressOf TRTI_OnLineReceived
    Dim Port As Integer
    Dim ServerIP As String

    Private mobjThread As Thread
    Private mobjListener As TcpListener
    Private mcolClients As New Hashtable

    Dim Actual_Speed(0 To 200, 0 To 500) As Integer
    Dim Actual_Flow(0 To 200, 0 To 500) As Integer
    Dim Actual_TTime(0 To 200, 0 To 500) As Integer
    Dim Actual_Delay(0 To 200, 0 To 500) As Integer
    Dim Actual_QL(0 To 200, 0 To 500) As Integer
    Dim WallClock As Integer

    Dim Speed(0 To 20, 0 To 200, 0 To 500) As Integer
    Dim Flow(0 To 20, 0 To 200, 0 To 500) As Integer
    Dim TTime(0 To 20, 0 To 200, 0 To 500) As Integer
    Dim Delay(0 To 20, 0 To 200, 0 To 500) As Integer
    Dim QL(0 To 20, 0 To 200, 0 To 500) As Integer

    Dim AveTTime(0 To 300, 0 To 200, 0 To 500) As Integer
    Dim AveDelay(0 To 300, 0 To 200, 0 To 500) As Integer
    Dim AveQL(0 To 300, 0 To 200, 0 To 500) As Integer

    Dim AveInboundSpeed(0 To 300, 0 To 200, 0 To 500) As Integer
    Dim AveInboundFlow(0 To 300, 0 To 200, 0 To 500) As Integer
    Dim AveOutboundSpeed(0 To 0, 0 To 200, 0 To 500) As Integer
    Dim AveOutboundFlow(0 To 0, 0 To 200, 0 To 500) As Integer

    Dim RBInfo(0 To 20, 0 To 10)
    Dim RBcount As Integer
    Dim Message(0 To 5000000, 0 To 10)
    Dim WallclockIntNO As Integer
    Dim To_send_string As String
```

```
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        TRTI_Initialize(Port, reflectptr)
        TRTI_reliable_createNewGroupAt(Marshal.StringToHGlobalAnsi(ServerIP),
Marshal.StringToHGlobalAnsi("Server"))

        While 1 > 0
            TRTI_ProcessAllMessagesInQueue()
            ArrayInitialize()

            If MsgReceivedNO > MsgProcessedNO And ServerIdle = 0 Then
                ServerIdle = 1
                MsgProcessedNO = MsgProcessedNO + 1

                ClientNum = Message(MsgProcessedNO, 0)
                RunNum = Message(MsgProcessedNO, 1)
                LinkChar = Message(MsgProcessedNO, 2)
                LinkNum = Message(MsgProcessedNO, 3)
                IntNO = Message(MsgProcessedNO, 4)

                If RunNum >= ClientRunNO(ClientNum) And Message(MsgProcessedNO, 5) > 0 Then

                    Speed(ClientNum, LinkNum, IntNO) = Message(MsgProcessedNO, 5)
                    Speed(ClientNum, LinkNum, 0) = LinkChar
                    Flow(ClientNum, LinkNum, IntNO) = Message(MsgProcessedNO, 6)
                    Flow(ClientNum, LinkNum, 0) = LinkChar
                    TTime(ClientNum, LinkNum, IntNO) = Message(MsgProcessedNO, 7)
                    Delay(ClientNum, LinkNum, IntNO) = Message(MsgProcessedNO, 8)
                    QL(ClientNum, LinkNum, IntNO) = Message(MsgProcessedNO, 9)

                    AverageInOut()

                        If LinkChar = 1 Then
                                CheckFromInbound()
                        ElseIf LinkChar = 2 Then
                                CheckFromOutbound()
                        ElseIf LinkChar = 3 Then
                            CheckFromInternal()
```

```
                End If
            End If
            ServerIdle = 0
        End If
    End While
End Sub
Private Sub CheckFromInbound()
    If IntNO * 60 > WallClock Then
        OutboundFlow = AveOutboundFlow(0, LinkNum, IntNO)
        OutboundSpeed = AveOutboundSpeed(0, LinkNum, IntNO)
        InboundFlow = AveInboundFlow(0, LinkNum, IntNO)
        InboundSpeed = AveInboundSpeed(0, LinkNum, IntNO)

        For Client0 = ClientNum To ClientNum
            FindInboundRB()
        Next

        If RB = 0 Then
            For Client0 = 1 To 10
                FindOutboundRB()
            Next
        End If
    End If
    RB = 0
End Sub
Private Sub CheckFromOutbound()
    If IntNO * 60 > WallClock Then
        OutboundFlow = AveOutboundFlow(0, LinkNum, IntNO)
        OutboundSpeed = AveOutboundSpeed(0, LinkNum, IntNO)
        InboundFlow = AveInboundFlow(0, LinkNum, IntNO)
        InboundSpeed = AveInboundSpeed(0, LinkNum, IntNO)

        For Client0 = ClientNum To ClientNum
            FindOutboundRB()
        Next

        If RB = 0 Then
            For Client0 = 1 To 10
```

```
                    FindInboundRB()
                Next
            End If
        End If
        RB = 0
    End Sub
    Private Sub CheckFromInternal()
        If IntNO * 60 > WallClock Then
            OutboundFlow = AveOutboundFlow(0, LinkNum, IntNO)
            OutboundSpeed = AveOutboundSpeed(0, LinkNum, IntNO)
            InboundFlow = AveInboundFlow(0, LinkNum, IntNO)
            InboundSpeed = AveInboundSpeed(0, LinkNum, IntNO)

            For Client0 = 1 To 10
                FindInboundRB()
            Next
            For Client0 = 1 To 10
                FindOutboundRB()
            Next
            RB = 0
        End If
    End Sub
    Private Sub FFRB(ByVal clt As Integer)
        ClearDB(clt)
        SendFFRB(clt)
        Anti(clt)
    End Sub

    Private Sub SendFFRB(ByVal clt As Integer)

        To_send_string = String1 & String2 & String3 & String4
        Try
            TRTI_reliable_sendMsgToGroupAt(Marshal.StringToHGlobalAnsi(ServerIP),
Marshal.StringToHGlobalAnsi(clt.ToString), Marshal.StringToHGlobalAnsi(To_send_string))
            TRTI_ProcessAllMessagesInQueue()
        Catch
        End Try
    End Sub
```

```
Private Sub Anti(ByVal clt As Integer)
    TRTI_ProcessAllMessagesInQueue()
    Dim Antimessage As Integer
    For Antimessage = MsgProcessedNO + 1 To 500000
        If Message(Antimessage, 0) = clt Then
            Message(Antimessage, 4) = 999
            Message(Antimessage, 5) = 999
        End If
    Next
End Sub
Private Sub ClearDB(ByVal clt As Integer)
    For Link0 = 100 To 200
        For Int0 = IntNO To 200
            Speed(clt, Link0, Int0) = -1
            Flow(clt, Link0, Int0) = -1
            TTime(clt, Link0, Int0) = -1
            Delay(clt, Link0, Int0) = -1
            QL(clt, Link0, Int0) = -1
        Next
    Next
End Sub
```

APPENDIX B: LOGICAL PROCESS SCRIPT

```
Imports System.Text
Imports System.Net.Sockets
Imports VISSIM_COMSERVERLib
Imports System.Runtime.InteropServices
Imports Microsoft.Office.Interop
Imports System.Convert
Imports System.Math
Imports System
Imports System.IO

Public Class VehicleRemoval
    Public Declare Auto Sub TRTI_Initialize Lib "trti.dll" (ByVal port As Integer, ByVal a As VB_Reflect)
    Public Declare Auto Sub TRTI_ProcessAllMessagesInQueue Lib "trti.dll" ()
    Public Declare Auto Sub TRTI_getOneMessage Lib "trti.dll" ()
    Delegate Sub VB_Reflect(ByVal a As Integer, ByVal b As String, ByVal c As Integer)

    Public Declare Auto Function TRTI_addToGroupAt Lib "trti.dll" (ByVal destination As IntPtr, ByVal group
As IntPtr) As Integer
    Public Declare Auto Function TRTI_removeFromGroupAt Lib "trti.dll" (ByVal destination As IntPtr, ByVal
group As IntPtr) As Integer
    Public Declare Auto Function TRTI_createNewGroupAt Lib "trti.dll" (ByVal destination As IntPtr, ByVal
group As IntPtr) As Integer
    Public Declare Auto Function TRTI_sendMsgToGroupAt Lib "trti.dll" (ByVal destination As IntPtr, ByVal
group As IntPtr, ByVal message As IntPtr) As Integer

    Public Declare Auto Function TRTI_reliable_addToGroupAt Lib "trti.dll" (ByVal destination As IntPtr,
ByVal group As IntPtr) As Integer
    Public Declare Auto Function TRTI_reliable_removeFromGroupAt Lib "trti.dll" (ByVal destination As
IntPtr, ByVal group As IntPtr) As Integer
```

```
    Public Declare Auto Function TRTI_reliable_createNewGroupAt Lib "trti.dll" (ByVal destination As
IntPtr, ByVal group As IntPtr) As Integer
    Public Declare Auto Function TRTI_reliable_sendMsgToGroupAt Lib "trti.dll" (ByVal destination As
IntPtr, ByVal group As IntPtr, ByVal message As IntPtr) As Integer

    Public reflectptr As VB_Reflect = AddressOf TRTI_OnLineReceived

    Dim Port As Integer
    Dim Retval As Integer

    Dim Str As String = ""
    Dim ServerIP As String
    Dim ClientNum As Integer
    Dim InpNum As Integer
    Dim TotalSimulationTime As Integer
    Dim Resolution As Integer
    Dim Inputline() As String
    Dim WallClockSimulationTime As Integer
    Dim EnteringLink(0 To 60)
    Dim ExitingLink(0 To 60)
    Dim RemoveLink(0 To 60)
    Dim VolIncrease(0 To 30)
    Dim ReplicationNO As Integer = 10

    Dim Vissim As Vissim
    Dim Simulation As Simulation
    Dim Net As Net
    Dim Vehicles As Vehicles
    Dim Vehicle As Vehicle
    Dim Links As Links
    Dim Link As Link
    Dim Eval As Evaluation
    Dim LinkEval As LinkEvaluation
    Dim Speed As Integer
    Dim RBMessageNO As Integer
    Dim MessageNO As Integer
    Dim ProcessNO As Integer
    Dim MessageNOCompleted As Integer
```

```
Dim TTimes As TravelTimes
Dim TTime(0 To 168) As TravelTime
Dim Delays As Delays
Dim Delay(0 To 168) As Delay
Dim QCounters As QueueCounters
Dim Queue(0 To 168) As QueueCounter
Dim Out(0 To 200, 0 To 7, 0 To 2000)
Dim IntNO As Integer = 1

Dim InputEntering(0 To 50, 0 To 1)
Dim InputExiting(0 To 50, 0 To 1)

Dim Detectors As DataCollections
Dim Detec(0 To 168) As DataCollection
Dim Detector As DataCollectionEvaluation
Dim SimTime As Long
Dim Rollback(0 To 10000, 0 To 100)
Dim Message(0 To 10000, 0)
Dim SameRB As Integer

Private mobjClient As TcpClient
Private marData(1024) As Byte
Private mobjText As New StringBuilder
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    TRTI_Initialize(Port, reflectptr)

    While 1 > 0
        If MessageNO = MessageNOCompleted Then
            TRTI_getOneMessage()
            If MessageNOCompleted > ProcessNO Then
                If Message(ProcessNO + 1, 0) = 9999 Then
                    ProcessNO = ProcessNO + 1
                    NextCaseNO = CaseNO + 1
                End If
            End If
        End If

        If CaseNO = 0 Or NextCaseNO = CaseNO + 1 Then
```

```
            For Each tmpfile As String In IO.Directory.GetFiles("C:\tmp", "*.snp")
                My.Computer.FileSystem.DeleteFile(tmpfile)
            Next

            CaseNO = CaseNO + 1
            NewInputText()
            KeepRunning = 0
            ReadInput()

            Vissim = CreateObject("vissim.vissim")
            TRTI_reliable_createNewGroupAt(Marshal.StringToHGlobalAnsi(ServerIP),
Marshal.StringToHGlobalAnsi(ClientNum.ToString))
            Simulation = Vissim.Simulation
            Vissim.LoadNet
            Net = Vissim.Net
            Vehicles = Vissim.Net.Vehicles
            Links = Net.Links
            Simulation.Period = TotalSimulationTime
            Simulation.Resolution = Resolution
            Simulation.RandomSeed = RandomNum

            Dim Link0 As Integer
            Dim Link1 As Integer
            Dim LinkNO As Integer
            Dim LinkProperty(0 To Links.Count, 0 To 5)

            Eval = Vissim.Evaluation
            Eval.AttValue("datacollection") = True
            Eval.AttValue("vehiclerecord") = True
            Eval.AttValue("traveltime") = True
            Eval.AttValue("delay") = True
            Eval.AttValue("queuecounter") = True
            QCounters = Vissim.Net.QueueCounters
            TTimes = Vissim.Net.TravelTimes
            Delays = Vissim.Net.Delays
            Detector = Vissim.Evaluation.DataCollectionEvaluation
            Detector.LoadConfiguration
            Detectors = Vissim.Net.DataCollections
```

```
For x = 1 To 168
    Detec(x) = Detectors(x)
Next

For x = 1 To TTimes.Count
    TTime(x) = TTimes(x)
Next

For x = 1 To Delays.Count
    Delay(x) = Delays(x)
Next

For x = 1 To QCounters.Count
    Queue(x) = QCounters(x)
Next
IntNO = 1

For Link0 = 1 To Links.Count
    If Links(Link0).ID > 100000 And Int(Links(Link0).ID / 100) Mod 10 = 0 Then
        LinkNO = LinkNO + 1

        LinkProperty(LinkNO, 0) = Links.GetLinkByNumber(Links(Link0).ID)
        LinkProperty(LinkNO, 1) = LinkProperty(LinkNO, 0).getvehicles

        For x = 1 To 30
            If InputEntering(x, 0) = LinkProperty(LinkNO, 0).attvalue("id") Then
                If LinkProperty(LinkNO, 0).attvalue("numlanes") = 4 Then
                    LinkProperty(LinkNO, 4) = FourLaneFlow
                    LinkProperty(LinkNO, 3) = Int(3600 / LinkProperty(LinkNO, 4) * 10)
                Else
                    LinkProperty(LinkNO, 4) = TwoLaneFlow
                    LinkProperty(LinkNO, 3) = Int(3600 / LinkProperty(LinkNO, 4) * 10)
                End If
            End If

            If InputExiting(x, 0) = LinkProperty(LinkNO, 0).id Then
                If LinkProperty(LinkNO, 0).attvalue("numlanes") = 4 Then
```

```
                    LinkProperty(LinkNO, 2) = FourLaneSpeed
                Else
                    LinkProperty(LinkNO, 2) = TwoLanespeed
                End If
            End If
        Next
    End If
Next Link0

For Link0 = 1 To LinkNO
    For Link1 = 1 To 30
        If LinkProperty(Link0, 0).id = InputEntering(Link1, 0) Then
            Out(Link0, 0, 0) = "1" & LinkProperty(Link0, 0).id
            Exit For
        ElseIf LinkProperty(Link0, 0).id = InputExiting(Link1, 0) Then
            Out(Link0, 0, 0) = "2" & LinkProperty(Link0, 0).id
            Exit For
        ElseIf Link1 = 30 Then
            Out(Link0, 0, 0) = "3" & LinkProperty(Link0, 0).id
        End If
    Next
Next Link0

While KeepRunning = 0
    TRTI_getOneMessage()
    If MessageNO = MessageNOCompleted Then
        If MessageNOCompleted > ProcessNO Then
            If Message(ProcessNO + 1, 0) = 7777 Then
                ProcessNO = ProcessNO + 1
                KeepRunning = 1
            End If
        End If
    End If
    SimTime = Simulation.AttValue("elapsedtime")

    TRTI_getOneMessage()

        Dim ComError As Integer = 1
```

```
                        Try
                            Simulation.RunSingleStep()
                        Catch ex As Exception
                            Console.WriteLine(ex.Message & "     " & SimTime & "       " & Now())
                            ComError = 0
                        Finally
                        End Try

                        If ComError = 1 Then
                            SimTime = Simulation.AttValue("elapsedtime")

                            For Link0 = 1 To LinkNO
                                If LinkProperty(Link0, 2) > 0 And LinkProperty(Link0, 2) < 470 Then
                                    For Each Vehicle In LinkProperty(Link0, 1)
                                        If Vehicle.AttValue("linkcoord") > 100 And
Vehicle.AttValue("linkcoord") < 300 Then
                                            Vehicle.AttValue("desiredspeed") = LinkProperty(Link0, 2) / 10
                                        ElseIf Vehicle.AttValue("linkcoord") >= 300 Then
                                            Vehicle.AttValue("desiredspeed") = 48
                                        End If
                                    Next
                                End If

                                If LinkProperty(Link0, 3) > 0 And SimTime = Int(LinkProperty(Link0, 3) /
10) Then
                                        Vehicle = Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate(100, 48,
LinkProperty(Link0, 0).id, Int(Rnd() * LinkProperty(Link0, 0).attvalue("numlanes")) + 1, 0)

                                    If SimTime > IncreaseBegin And SimTime < IncreaseEnd Then
                                        For Link1 = 1 To 30
                                            If LinkProperty(Link0, 0).id = VolIncrease(Link1) Then
                                                Vehicle =
Vissim.Net.Vehicles.AddVehicleAtLinkCoordinate(100, 48, LinkProperty(Link0, 0).id, Int(Rnd() *
LinkProperty(Link0, 0).attvalue("numlanes")) + 1, 0)
                                            End If
                                        Next
                                    End If
```

```
    End If
Next

For Link0 = 1 To LinkNO
    If SimTime > IncidentBegin And SimTime < IncidentEnd Then 'incident
        If LinkProperty(Link0, 0).id = IncidentLink Then
            For Each Vehicle In LinkProperty(Link0, 1)
                If Vehicle.AttValue("linkcoord") > 100 Then
                    Vehicle.AttValue("desiredspeed") = IncidentSpeed / 10
                End If
                If Vehicle.AttValue("linkcoord") > 300 Then
                    Vehicle.AttValue("desiredspeed") = 48
                End If
            Next
        End If
    ElseIf SimTime = IncidentEnd Then
        If LinkProperty(Link0, 0).id = IncidentLink Then
            For Each Vehicle In LinkProperty(Link0, 1) '
                Vehicle.AttValue("desiredspeed") = 48
            Next
        End If
    End If
Next

If SimTime > 61 And (SimTime - RBProcessNO - 1 + SameRB) Mod 60 = 0 Then
    IntNO = IntNO + 1
    Simulation.SaveSnapshot("C:\ " & InpNum & "_" & IntNO * 60 & ".snp")

    Dim MessageInOne As Integer = 0
    If MessageInOne = 0 Then
        SendOutput = ""
        FourDigit(ClientNum)
        SixDigit(RBProcessNO)
        FiveDigit(IntNO * 60)
    End If
    For Link0 = 1 To LinkNO
        If Int(Out(Link0, 0, 0) / 1000000) = 3 Then
            If IntNO Mod 2 = 0 Then
```

```
                                                        Out(Link0, 5, IntNO) = Int(TTime(Link0).GetResult(IntNO * 60,
"traveltime", "", 0))
                                                        Out(Link0, 6, IntNO) = Int(Delay(Link0).GetResult(IntNO * 60,
"delay", "", 0))
                                                        Out(Link0, 7, IntNO) = Int(Queue(Link0).GetResult(IntNO * 60,
"mean"))
                                                    End If
                                                End If

                                                If Detec(Link0).GetResult("speed", "mean", 0) > 0 Then
                                                    Out(Link0, 1, IntNO) = Detec(Link0).GetResult("speed", "mean", 0)
                                                End If

                                                If Detec(Link0).GetResult("nvehicles", "sum", 0) > 0 Then
                                                    Out(Link0, 3, IntNO) = Detec(Link0).GetResult("nvehicles", "sum",
0) * 60 / LinkProperty(Link0, 0).attvalue("numlanes")
                                                End If
                                            Next

                                            If SimTime >= 600 And IntNO * 60 >= Rollback(RBProcessNO, 2) Then
                                                If (Wait_on_send) Then
                                                    While
(TRTI_reliable_sendMsgToGroupAt(Marshal.StringToHGlobalAnsi(ServerIP),
Marshal.StringToHGlobalAnsi("Server"), Marshal.StringToHGlobalAnsi(SendOutput)) <= 0)
                                                    End While
                                                Else

TRTI_reliable_sendMsgToGroupAt(Marshal.StringToHGlobalAnsi(ServerIP),
Marshal.StringToHGlobalAnsi("Server"), Marshal.StringToHGlobalAnsi(SendOutput))
                                                End If
                                            End If
                                        End If
                                    End If
                                End If

                                If RBMessageNO > RBProcessNO And RBProcessNO = RBCompleteNO Then
                                    RBProcessNO = RBProcessNO + 1
                                    Simulation.Stop()
```

```
                        Console.WriteLine("Simulation.Stop      " & Now())

                        For Link0 = 1 To LinkNO                          ''''''''''''''''''''''''''''''''''''''''
                            For Link1 = 1 To 30
                                If InputEntering(Link1, 0) = Rollback(RBProcessNO, 1) And
InputEntering(Link1, 0) = LinkProperty(Link0, 0).id And Rollback(RBProcessNO, 4) < 999 Then
                                    LinkProperty(Link0, 4) = Rollback(RBProcessNO, 4) * LinkProperty(Link0,
0).attvalue("numlanes")
                                    IntNO = Rollback(RBProcessNO, 2) / 60 - AggregateMin
                                End If
                                If InputExiting(Link1, 0) = Rollback(RBProcessNO, 1) And
InputExiting(Link1, 0) = LinkProperty(Link0, 0).id And Rollback(RBProcessNO, 3) < 999 Then
                                    LinkProperty(Link0, 2) = Rollback(RBProcessNO, 3)
                                    IntNO = Rollback(RBProcessNO, 2) / 60 - AggregateMin
                                End If
                            Next
                        Next

                        Vissim.LoadNet
                        Simulation.LoadSnapshot("C:\TMP\client" & InpNum & "_" & IntNO * 60 & ".snp")
                        SimTime = Simulation.AttValue("elapsedtime")

                        Next
                    End If
                End While

                Simulation.Stop()
                Vissim.Exit()
            End If
        End While
    End Sub
    Public Sub TRTI_OnLineReceived(ByVal a As Integer, ByVal Data As String, ByVal c As Integer)

        If Data.Substring(0, 4) = 7777 Then
            MessageNO = MessageNO + 1
            Message(MessageNO, 0) = Data.Substring(0, 4)
            MessageNOCompleted = MessageNOCompleted + 1
        ElseIf Data.Substring(0, 4) = 9999 Then
```

```
            MessageNO = MessageNO + 1
            Message(MessageNO, 0) = Data.Substring(0, 4)
            MessageNOCompleted = MessageNOCompleted + 1
        ElseIf Data.Substring(0, 4) = 8888 Then
        Else
            RBMessageNO = RBMessageNO + 1
            Rollback(RBMessageNO, 1) = Int(Data.Substring(4, 6))
            Rollback(RBMessageNO, 2) = Int(Data.Substring(10, 5))
            Rollback(RBMessageNO, 3) = Int(Data.Substring(15, 3))
            Rollback(RBMessageNO, 4) = Int(Data.Substring(18))
            If Rollback(RBMessageNO - 1, 2) <= Rollback(RBMessageNO, 2) Then
                SameRB = SameRB + 1
            End If
        End If
    End Sub
End Class
```

REFERENCES

1. U.S. Census Bureau. The 2010 Statistical Abstract. Available from: http://www.census.gov/compendia/statab.

2. FHWA. Highway Statistics 2010. Available from: http://www.fhwa.dot.gov/policyinformation/statistics/2010/index.cfm.

3. Lomax, T. D. Schrank, and B. Eisele. 2011 Annual Urban Mobility Report. Texas Transportation Institute. 2012.

4. ITSAmerica, National ITS Program Plan: A Ten-Year Vision. 2002.

5. Vehicle Infrastructure Integration Proof of Concept Executive Summary – Vehicle. 2009, VII Consortium.

6. Brainstorming the Potential Impact of VII on System Operations. 2005; Available from: http://www.ntoctalks.com/icdn/vii_roundtable_itsa05.php.

7. USDOT. IntelliDrive. 2009.

8. FHWA, Public-Private Agreements/Arrangements Associated with VII Implementation. 2005.

9. USDOT, Advanced Wireless Communications for the Transportation Sector: A Roundtable Discussion. 2008.

10. Broadhurst, A., S. Baker, and T. Kanade, Monte Carlo Road Safety Reasoning. IEEE Intelligent Vehicles Symposium, 2005.

11. Yang, X., J. Liu, F. Zhao and N. Vaidya. A Vehicle-to-Vehicle Communication Protocol for Cooperative Collision Warning. First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous'04). 2004.

12. Chu, L. and W. Recker. Micro-Simulation Modeling Approach to Applications of On-Line Simulation And Data Fusion. University of California, Berkeley. 2004.

13. Kosonen, I. and A. Bargiela. Simulation based traffic information system, 7th World Congress on Intelligent Transport Systems. Turin, Italy. 2000.

14. Wang, H., D.-H. Lee, and R. Cheu. Dynamic Routing Decisions for Commercial Vehicle Operations in Real-Time Traffic Conditions. Transportation Research Record 1882. 2004.

15. Katwijk, R., P. Koningsbruggen, and J. Hellendoorn, A Test Bed for Multi-Agent Control Systems in Road Traffic Management. Applications of Agent Technology in Traffic and Transportation. 2005.

16. Spelberg, R., H. Toetenel, and R. Vermeijs. Freeway Traffic Congestion Management Through Real-Time Simulation, IASTED International Conference: Modelling and Simulation. 1995.

17. Nishikawa, I., T. Iritani, and K. Sakakibara. Improvements of the Traffic Signal Control by Complex-Valued Hopfield Networks. International Joint Conference on Neural Networks. 2006.

18. Gorgorin, C., V. Gradinescu, R. Diaconescu, V. Cristea and L. Iftode. Adaptive Traffic Lights using Car-to-Car Communication. IEEE Vehicular Technology Conference. 2007.

19. Kosonen, I., Multi-Agent Fuzzy Signal Control Based On Real-Time Simulation. Transportation Research Part C: Emerging Technologies, 11(5): p. 389-403. 2003.

20. Lee, D.-H., and P. Chandrasekar A Framework for Parallel Traffic Simulation Using Multiple Instancing of a Simulation Program. Intelligent Transportation Systems Journal. 7. p. 279 – 294. 2004.

21. Liu, H., W. Ma, R. Jayakrishnan, and W. Recker. Large-Scale Traffic Simulation Through Distributed Computing of Paramics. PATH Research Report. 2004.

22. Junchaya, T. and G.-L. Chang. Exploring real-time traffic simulation with massively parallel computing architecture. Transportation Research Part C: Emerging Technologies. 1(1): p. 57-76. 1993.

23. Klefstad, R., Y. Zhang, M. Lai, R. Jayakrishnan, and R. Lavanya. A distributed, scalable, and synchronized framework for large-scale microscopic traffic simulation. Intelligent Transportation Systems. Proceedings: p. 813 - 818. 2005.

24. Cetin, N., A. Burri, and K. Nagel. A Large-Scale Agent-Based Traffic Microsimulation Based On Queue Model. Swiss Transport Research Conference. 2003.

25. Bononi, L., M. Felice, G. D'Angelo, M. Bracuto, and L. Donatiello. MoVES: A framework for parallel and distributed simulation of wireless vehicular ad hoc networks. Computer Networks. 52(1): p. 155-179. 2008.

26. Bononi, L., M. Bracuto, G. D'Angelo, and L. Donatiello Scalable and Efficient Parallel and Distributed Simulation of Complex, Dynamic and Mobile Systems. IEEE Workshop on Techniques Methodologies and Tools for Performance Evaluation of Complex Systems. 2005.

27. K. Nagel and N. Cetin. Parallel Queue Model Approach to Traffic Microsimulations. Transportation Research Board Annual Meeting. Washington D.C. 2003.

28. Ferscha, A., and S. Tripathi. Parallel and distributed simulation of discrete event systems. University of Maryland at College Park. 1994.

29. Innamaa, S., and I. Kosonen. Online Traffic Models - A Learning Experience. Traffic Engineering & Control. 45(9): p. 338-343. 2004.

30. Erol, K., R. Levy, and J. Wentworth. Application of Agent Technology to Traffic Simulation. 2003.

31. Sigurđur F. Hafstein, R.C., Andreas Pottmeier, Michael Schreckenberg & Florian C. Mazur A High-Resolution Cellular Automata Traffic Simulation Model with Application in a Freeway Traffic Information System. Computer-Aided Civil and Infrastructure Engineering, 2004. 19(5): p. 338-350.

32. Choffnes, D., and F. Bustamante. An integrated mobility and traffic model for vehicular wireless networks. International Conference on Mobile Computing and Networking, 2nd ACM international workshop on Vehicular ad hoc networks. 2005.

33. Gorgorin, C., V. Gradinescu, R. Diaconescu, V. Cristea and L. Iftode. An Integrated Vehicular and Network Simulator for Vehicular Ad-Hoc Networks. 20th European Simulation and Modelling Conference. 2006.

34. Nekovee, M. Sensor networks on the road: the promises and challenges of vehicular ad hoc networks and vehicular grids. Workshop on Ubiquitous Computing and e-Research. 2005.

35. Ukkusuri, S. L. Du, and S. Kalyanaraman. Geometric connectivity of vehicular ad hoc networks: Analytical characterization Transportation research. Part C, Emerging technologies. 16: p. 615-634. 2008.

36. Yousefi, S., Vehicular Ad Hoc Networks (VANETs): Challenges and Perspectives. 6th International Conference on ITS Telecommunications Proceedings. 2006.

37. FleetNet - Internet on the Road. Available from: http://www.et2.tu-harburg.de/fleetnet/english/about.html.

38. Network on Wheels. Available from: http://www.network-on-wheels.de/about.html.

39. CAR 2 CAR Communication Consortium. Available from: http://www.car-2-car.org/index.php?id=1.

40. Global System for Telematics (GST). Available from: http://www.gstforum.org/.

41. Cooperative Vehicle-Infrastructure Systems. Available from: http://www.cvisproject.org/.

42. Oyama, S. DSRC and Ad-hoc Network development in Japan. The Second ACM International Workshop on Vehicular Ad Hoc Networks (VANET 2005). 2005.

43. MASHITA, K., Development of ASV in Japan -ASV Promotion Project Phase 3. IATSS Research. 27. 2003.

44. Wani, K. The Fourth Phase of Advnaced Safety Vehicle Project. ITS World Congress London. 2006.

45. Fujimoto, R. Parallel and Distributed Simulation Systems. Wiley-Interscience. 2000.

46. Fujimoto, R PARALLEL AND DISTRIBUTED SIMULATION. Proceedings of the 1999 Winter Simulation Conference. 1999.

47. Jefferson, D. Virtual time ACM Transactions on Programming Languages and Systems. 7(3). 1985.

48. Fujimoto, R. Parallel Discrete Event Simulation. Communications of the ACM. 10: p. 30-53. 1990.

49. Park, A., and R. Fujimoto. A Scalable Framework for Parallel Discrete Event Simulations on Desktop Grids, 8th Grid Computing Conference. 2007.

50. Lin, Y.-B., and P. Fishwick. Asynchronous parallel discrete event simulation. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans. 26(4): p. 397-412. 1996.

51. Lin, Y.-B., B. Preiss, W. Loucks, and E. Lazowska. Selecting the Checkpoint Interval in Time Warp Simulation. Proceedings of the seventh workshop on Parallel and distributed simulation 1993.

52. Ben-Akiva, M., M. Bierlairey, H. Koutsopoulosz, and R. Mishalanix. Real Time Simulation of Traffic demand-supply interactions within DynaMIT. 2000.

53. Jayakrishnan, R., P. Sheu, T. Wang, and M. Xu. Database Environment for Fast Real-Time Simulation of Urban Traffic Networks with ATMIS. California Partners for Advanced Transit and Highways (PATH). 2000.

54. Yang, Q. and H. Koutsopoulos. A microscopic traffic simulator for evaluation of dynamic traffic management systems. Transportation Research Part C: Emerging Technologies. 4(3): p. 113-129. 1996.

55. Nagel, K., and M. Rickert. Parallel implementation of the TRANSIMS micro-simulation. Parallel Computing. 27(12): p. 1611-1639. 2001.

56. Rickert, M. and K. Nagel. Dynamic traffic assignment on parallel computers in TRANSIMS. Future Generation Computer Systems. 17(5): p. 637-648. 2001.

57. Gu, Y., Integrating a Regional Planning Model (TRANSIMS) With an Operational Model (CORSIM). Virginia Polytechnic Institute and State University: Civil Engineering. 2004.

58. Barceló, J., J. Ferrer, D. García and R. Grau. Microscopic Traffic Simulation for ATT Systems Analysis – A Parallel Computing Version.

59. Algers, S., E. Bernauer, M. Boero, L. Breheret, C. Taranto, M. Dougherty, K. Fox and J. Gabard. Review of Micro-Simulation Models. 1997.

60. Cameron, G., B. Wylie, and D. McArthur. Paramics: Moving Vehicles on the Connection Machine. 1994 ACM/IEEE conference on Supercomputing Conference on High Performance Networking and Computing. 1994.

61. G Cameron, and G. Duncan. PARAMICS—Parallel Microscopic Simulation of Road Traffic. The Journal of Supercomputing. 10(1): p. 25-53. 1996.

62. PADS. Parallel and Distributed Simulation (PADS) Research Group.

63. Bononi, L., M. Bracuto, G. D'Angelo and L. Donatiello. A New Adaptive Middleware for Parallel and Distributed Simulation of Dynamically Interacting Systems. Eighth IEEE International Symposium on Distributed Simulation and Real-Time Applications. 2004.

64. Yoginath, S., and K. Perumalla. Parallel Vehicular Traffic Simulation using Reverse Computation-based Optimistic Execution. 22nd Workshop on Principles of Advanced and Distributed Simulation. 2008.

65. Yoginath, S., and K. Perumalla. Reversible Discrete Event Formulation and Optimistic Parallel Execution of Vehicular Traffic Models. Int. J. Simulation and Process Modeling, 2009.

66. Bernstein, P., Middleware: A Model for Distributed System Services. Communications of the ACM, 39(2): p. 86–98. 1996.

67. Bakken, D., Middleware. Encyclopedia of Distributed Computing. 2003.

68. Norman, R., CORBA and DCOM: Side by Side. Distributed Computing,. 1(5): p. 41-46. 1998.

69. Gokhale, A., and D. Schmidt. Measuring the Performance of Communication Middleware on High-Speed Networks ACM SIGCOMM Computer Communication Review. 26(4): p. 306-317. 1996.

70. Everything You Need To Know About Middleware. Available from: http://www.pacificdataworks.com/pub/talarianmiddleware.pdf.

71. Yau, S., and F. Karim. A Lightweight Middleware Protocol for Ad Hoc Distributed Object Computing in Ubiquitous Computing Environments. Proceedings of the Sixth IEEE International Symposium on Object-Oriented Real-Time Distributed Computing. 2003.

72. Microsoft. COM: Component Object Model Technologies. Available from: http://www.microsoft.com/com/default.mspx.

73. IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Framework and Rules. 2000.

74. Dahmann, J., R. Fujimoto, and R. Weatherly. The Department of Defense High Level Architecture. 29th Winter Simulation Conference. 1997.

75. PTV, VISSIM 5.10 User Manual. 2010.

76. PTV, VISSIM 5.10-06 COM Interface Manual. 2010.