A Simulation Laboratory for Evaluation of Dynamic Traffic Management Systems

Qi Yang

M.S.T., Massachusetts Institute of Technology (1993)
M.S., Peking University (1985)
B.S., Peking University (1983)

Submitted to the Department of Civil and Environmental Engineering
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Transportation

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1997

Author
. . . .
Department of Civil and Environmental Engineering
February 1997

Certified by
. . . .
Moshe E. Ben-Akiva
Professor of Civil and Environmental Engineering
Massachusetts Institute of Technology
Thesis Supervisor

Certified by
. . . .
Haris N. Koutsopoulos
Associate Professor of Civil and Environmental Engineering
Carnegie Mellon University
Thesis Supervisor

Accepted by
. . . .
Joseph M. Sussman
Chairman, Departmental Committee on Graduate Studies

A Simulation Laboratory for Evaluation of Dynamic Traffic Management Systems

Qi Yang

Submitted to the Department of Civil and Environmental Engineering
on February 1997, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Transportation

Abstract

Advanced Traffic Management Systems (ATMS) and Advanced Traveler Information Systems (ATIS) are promising technologies for achieving efficiency in the operation of transportation systems. This research develops a traffic simulation laboratory (SIMLAB) capable of evaluating integrated ATMS and ATIS. SIMLAB can simulate the operations of integrated traffic networks including freeways, arterials, and urban streets. It explicitly incorporates traffic prediction, time variant traffic information, and dynamic route choice. Its modular design enables testing of a wide range of ATIS and ATMS concepts and facilitates the improvement of traffic management strategies.

SIMLAB consists of a microscopic traffic simulator (MITSIM) and a traffic management simulator (TMS). MITSIM represents the "real world." It accepts as input signal control and route guidance from TMS, and models the movement of individual vehicles in the network. TMS receives traffic surveillance data from MITSIM and generates signal controls and route guidance according to the ATMS and ATIS logic under evaluation. TMS supports simulation of a wide range of signal control, route guidance and incident management schemes. A mesoscopic traffic flow simulator is used within TMS to predict traffic conditions. This element is designed to model the ATMS/ATIS which generates traffic controls and route guidance based on anticipatory – as opposed to historical or the most recent measurement of – traffic conditions. The generic structure of SIMLAB allows the effectiveness and robustness of any traffic control and surveillance system to be evaluated in a computer-based laboratory environment.

The laboratory is implemented in C++ using object-oriented programming and a distributed environment. A graphical user interface allows users to visualize the simulation process, including animation of vehicle movements, measurement of surveillance sensors, state of traffic signals and signs, etc.

SIMLAB was validated using several networks, including a 6-mile stretch of I-880 around Hayward, California. The applicability of the system was demonstrated in a case study using the A10 beltway in Amsterdam, The Netherlands. SIMLAB was used to evaluate the performance of two alternative approaches in providing real-time route guidance: one based on latest measurements and the other on traffic prediction. The results provide useful insights for the design of ATIS, indicate that the simulation laboratory operates as designed, and support its value as a tool for evaluation of advanced traffic management systems.

Thesis Supervisor: **Moshe E. Ben-Akiva**
Professor of Civil and Environmental Engineering
Massachusetts Institute of Technology

Thesis Supervisor: **Haris N. Koutsopoulos**
Associate Professor of Civil and Environmental Engineering
Carnegie Mellon University

Thesis Committee

Professor Moshe E. Ben-Akiva (Chairman)

Department of Civil and Environmental Engineering

Massachusetts Institute of Technology

Professor Nathan H. Gartner

Department of Civil and Environmental Engineering

University of Massachusetts Lowell

Professor Haris N. Koutsopoulos

Department of Civil and Environmental Engineering

Carnegie Mellon University

Professor Joseph M. Sussman

Department of Civil and Environmental Engineering

Massachusetts Institute of Technology

Acknowledgments

I take this opportunity to thank my advisors, Professor Moshe Ben-Akiva and Professor Haris Koutsopoulos, for their invaluable advice and guidance throughout my doctoral study. Their constant support, encouragement and friendship made my stay at MIT an enjoyable experience. Professor Koutsopoulos has been my advisor since I came to MIT. He spent a lot of time working with me and made my journey through the master and Ph.D. programs much easier.

I would also like to thank the other members of my doctoral committee – Professor Nathan Gartner and Professor Joseph Sussman – for their advice, inspiration, and interest in my research.

My deepest gratitude goes to Michel Bierlaire, Anthony Hotz, Mithilesh Jha, and Rabi Mishalani. Their technical expertise and help were always available when needed. I would like to express special thanks to Peter Welch for his invaluable help in the implementation and debugging of the computer programs. Thanks also to:

- Kalidas Ashok for providing the time-dependent origin-destination flows used in the case study. Our many discussions were always useful and inspirational to my research, especially at the most difficult times.
- Nanne van der Zijpp from the Netherlands and Karl Petty from the University of California at Berkeley, for providing the data set used in this study.
- Hariharan Subramanian and Kazi Ahmed for their contributions in the development of the microscopic traffic simulator. Owen Chen for sharing his expertise on traffic control and on computer software.
- James Sturdy and Winston Guo for their work on the graphical road network editor I used in this study.
- My fellow students and friends – Adriana, Amalia, Andras, Chris, Daeki, Dave, David, Dinesh, Emily, Francisco, Haiping, Jiang, Jin Hong, Joan, John, Jon, Nagi, Prodyut, Gao Qian, Oliver, Scott, Shenoi, Sridevi, Susan, Xujun, and Dong Yan – for making my long tenure at MIT enjoyable.

- Professors Cynthia Barnhart, David Bernstein, Ismail Chabini, Robert Logcher, Karen Polenske, Yosef Sheffi, and Nigel Wilson; and staff including Julie, Lisa, Paula, Cheryl, and many others, for their friendship and help.

The financial support of MIT, Draper Laboratory, Bechtel/Parsons Brinckerhoff, and the Massachusetts Highway Department are deeply appreciated.

Finally I wish to express my heartfelt thanks to Xiangming, my wife, for her love, understanding, and constant support, and to my parents for their affection and encouragement.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The growth of urban automobile traffic has led to serious and worsening traffic congestion problems in most cities around the world. Since travel demand increases at a rate often greater than the addition of road capacity, the situation will continue to deteriorate unless better traffic management strategies are implemented.

One of the most attractive remedial measures for addressing the congestion problem is the development of Intelligent Transportation Systems (ITS). A wide range of technological developments, fall under the ITS umbrella, including Advanced Traveler Information Systems (ATIS), Advanced Traffic Management Systems (ATMS), Commercial Vehicle Operations (CVO), Advanced Public Transportation Systems (APTS), and Automatic Vehicle Control Systems (AVCS). Each of these systems is receiving substantial public and private attention. However, at this point in time, ITS's potential is uncertain. Decisions on whether to support the development of various systems requires data and careful analysis.

1.1 Motivation and Research Objective

While advanced technologies made it possible to develop more sophisticated traffic management strategies, experience has shown that such strategies do not always result in improved performance (Gartner et al., 1995). When designing a particular traffic management system, many possible architectures and alternatives may be chosen.

Evaluation is, therefore, an important element in the design process. Similarly, in assessing the impact of proposed changes to an existing system, the analyst faces various "what if" questions.

If a traffic management system is already fully operational, field operational tests can be used in assessing its performance and evaluating many of the practical aspects of systems. However, field tests tend to be expensive, and as a result, typically few alternatives can be tested. In addition, the test results depend on uncontrollable elements of the environment (e.g. weather conditions, travel demand, incidents).

In recent years simulation has become a tool commonly used to understand the characteristics of a traffic system and to select an appropriate design. However, most of these simulation models are not capable of evaluating integrated dynamic traffic management systems and other ITS applications. The need for a more advanced evaluation tool has been identified by a number of researchers such as (Lin, 1993; Santiago and Kanaan, 1993). Recently various approaches have been proposed and models are under development (Leonard II, 1993; Jayakrishnan et al., 1995; Jayakrishnan and Rindt, 1996). The needed ability to date is unfortunately still lacking in the state of the art, particularly for integrated evaluation of ATIS and ATMS. This problem stems in part from the complex nature of modeling traffic flow in an integrated network, and in part from the lack of basic research concerning the dynamic interaction between traffic flows and ATIS/ATMS.

In this research, a methodology for evaluation of dynamic traffic management systems is developed. The methodology utilizes a simulation laboratory (SIMLAB) for testing and evaluating designs of ATMS and ATIS. SIMLAB is a computer-based modeling system that integrates a microscopic traffic simulator (MITSIM) and a traffic management simulator (TMS) (see Figure 1-1). MITSIM simulates in detail the "state of the network". Vehicles in the network are moved from their origins to destinations and respond to the various traffic controls and guidance while interacting with each other. The vehicle movements are recorded by a surveillance system module that represents traffic sensors and probe vehicles. TMS models the candidate traffic control and routing logic under evaluation. Using the data obtained

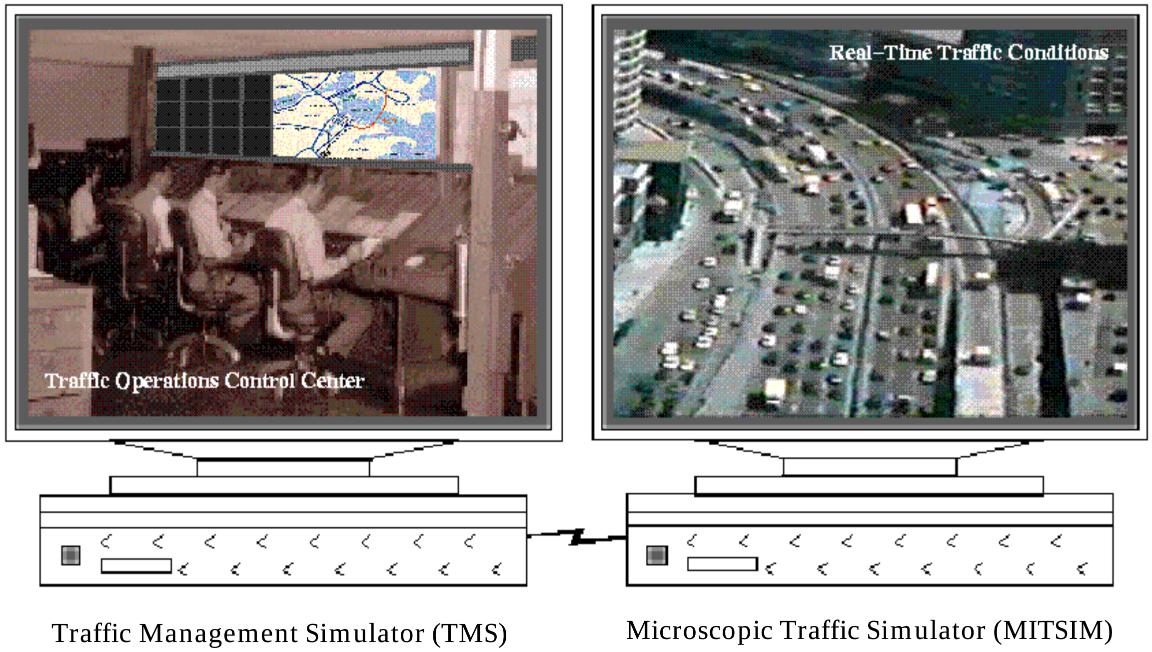


Figure 1-1: A computer-based simulation laboratory for evaluating dynamic traffic management systems

by the surveillance system, TMS generates control and routing strategies, which, in turn, are fed into MITSIM through the traffic control and route guidance devices and hence influence the traffic flows in the simulated network. The core of SIMLAB is modeling the dynamic interaction between the traffic management system (simulated by TMS) and the traffic flows (simulated by MITSIM). This interaction mimics the relationship between a traffic operation control center and the traffic flows in the road network.

The design features a separation of the simulation of surveillance and control system from the simulation of traffic flows. This modular structure provides flexibility for representing a variety of control and route guidance systems and facilitates distribution of tasks to multiple computer processors. In addition, a prototypical system that generates route guidance based on predicted traffic conditions is implemented in TMS and is evaluated as an application of the simulation laboratory.

The main contribution of this research lies in the development of the laboratory environment for evaluating dynamic traffic management systems. This simulation laboratory supports integrated ATIS and ATMS operations in general networks including freeways, arterial, and urban streets. One of its unique features is the use of a mesoscopic traffic simulation model for traffic prediction inside TMS, which provides a fundamental function necessary for evaluation of proactive dynamic traffic management systems. In addition, this default traffic predictor can be replace by other appropriate traffic predictors such as dynamic traffic assignment models. The modeling system developed in this research is the first of its kind that has been implemented and whose usage has been demonstrated though a case study using a real network.

1.2 Literature Review

The goal of this section is to review the latest developments in dynamic traffic management systems and methods and tools for evaluating such systems.

1.2.1 Advances in Dynamic Traffic Management Systems

Recently, increasing attention has been given to the development of dynamic route guidance systems and integrated and adaptive traffic control strategies.

Dynamic traffic assignment (DTA) has been one of the most recent developments receiving extensive attention in the transportation research communities worldwide (DYNA, 1992-1995; FHWA, 1995; Mahmassani et al., 1994; MIT, 1996). DTA aims at providing route guidance based on predicted rather than historically measured traffic conditions. While various DTA systems are still undergoing theoretical research as the state of the art, a variety of route guidance systems are being subjected to experiment in real-world traffic networks. For example, dynamic route information panel (DRIP) is being used to provide short-term traffic forecasts for several major urban arterials in Amsterdam, the Netherlands (von Toorenburg et al., 1996). ADVANCE, an real-time in-vehicle route guidance system, is under field test in a suburban area in Chicago, Illinois.

Frameworks for hierarchical or multi-level traffic control have been suggested by several researchers including Head et al. (1992); and Gartner et al. (1995).

Gartner et al. (1995) reviewed the past experiences with “advanced” traffic signal control strategies and suggested a multi-level design of real-time, adaptive signal control strategies. They emphasized that a system should offer varying degrees of responsiveness, depending on particular network and traffic characteristics. In addition, the system should allow selection of a control strategy when such a strategy could provide the greatest benefits and maximize the overall effectiveness of the particular system. To meet these requirements, a real-time traffic adaptive control system (RT-TRACS) is under development (Tarnoff and Gartner, 1993). Frameworks for integrating dynamic traffic assignment with real-time traffic adaptive control have been proposed (Gartner and Stamatiadis, 1997; Chen and Hsueh, 1997). A decentralized scheme for real-time route guidance was also proposed (Hawas and Mahmassani, 1995).

The EUROCOR project (Middelham et al., 1994a) is another recent example of development and application of a sophisticated traffic control system. EUROCOR, with special attention to ramp metering, is part of the European multi-national DRIVE-2 project, which includes the development and evaluation of integrated urban corridor control models. EUROCOR’s main objectives were to develop and apply dynamic traffic control models for effective management of traffic in freeway corridors using traffic signals, ramp metering, and variable message signs (VMS). The system has undergone extensive field trials on a limited set of corridors around certain cities (Diakaki and Papageorgiou, 1995; Middelham et al., 1994a).

1.2.2 Evaluation Methods

The two primary methods used in evaluation of traffic management systems are field tests and computer simulation.

Field tests have been an important element in evaluating the operations and performance of traffic surveillance and control systems. For example, a “floating car” survey was used in evaluating the performance of SCOOT (Hunt et al., 1981; Clowes, 1983), an areawide adaptive signal control system developed in the U.K. and implemented in many cities worldwide. Similar surveys have also been conducted by the California PATH program in its Freeway Service Patrol project (Petty et al., 1996). In these evaluations, vehicle trip time and distance traveled are recorded before and after the implementation of the system, and compared. Another example of operational tests is the evaluation of INFORM, a traffic management system consisting of integrated signal control and driver information systems, in a 40-mile stretch of highway corridor in Long Island, New York (Smith and Perez, 1992). The INFORM control elements include traffic monitoring in the control center, ramp metering, variable message signs, and traffic signals at intersections. Motorist response to and the effectiveness of ramp metering and variable message sign strategies was evaluated, using extensive field data and motorist perception surveys.

An evaluation based on field test was also designed for the EUROCOR project. Corridor Peripherique in Paris and A10 West in Amsterdam were selected as the test sites. The strategies evaluated were composed of a combination of ramp metering (with or without ALINEA ramp metering, see (Papageorgiou et al., 1997)) and variable message signs (display queue or travel time information). Data such as volume, occupancy, speed, travel time, and queues in the testing networks were collected and used as indicators for the performance of the system. The evaluation for each strategy was planed to be conducted during the evening peak hours (4 hours) for at least eight days.

The ADVANCE project is a recent example of using field test to evaluate and demonstrate the benefit of ATIS (Bowcott, 1993; Saricks et al., 1997). Vehicles equipped with a Mobile Navigation Assistant (MNA) act as probes, sending real time travel information about arterial and local streets to a traffic management center, which in term transmits the information to other equipped vehicles to aid drivers' dynamic route planning. A field test for evaluating the ADVANCE dynamic route guidance system was conducted on a suburban area in Chicago, Illinois (Schofer et al., 1996). Three ADVANCE-equipped vehicles were driven from five predefined origin to destination pairs. Two vehicles follow MNA-provided routes, generated based on real-time traffic information collected by a fleet of 18 equipped vehicles driven on designed routes. The third follows routes based on historical information. Experienced travel times were collected and compared.

Although operational tests are necessary, they are expensive. Furthermore, the system has to be fully operational in order to conduct such tests. On the other hand, computer simulation models allow for testing alternative system designs under a controlled environment, before conducting operational tests and refining original designs, thus resulting in more effective implementation.

Simulation based evaluations have been extensively used in studying the performance of traffic control and route guidance strategies. In the literature, various control elements of traffic management systems have been studied individually using simulation. Considerable literature exists, for example, on evaluating design of ramp metering for freeway traffic control. Payne (1973) developed a ramp metering regulator and evaluated its performance using a macroscopic traffic flow model. The objective of the control was to minimize the deviation of traffic conditions from some nominal conditions. Feedback optimal control was obtained by solving a linear quadratic cost problem. A simulation study was conducted for a 7.8-mile section of the San Diego freeway in Los Angeles. Similar approaches for ramp metering have also been used recently by Papageorgiou et al. (1990), Stephanedes et al. (1992), Shepherd (1993), Hellinga and van Aerde (1995), Diakaki and Papageorgiou (1995), etc.

Simulation-based studies have also been conducted for other traffic management strategies such as urban traffic signal controls (Sibley, 1985; Yauch et al., 1988), route diversion (Stephanedes et al., 1989; Barcelo and Ferrer, 1995), variable speed limit signs (Smulders, 1990), and mainline metering (Haboian, 1995). Common characteristics of these studies are a focus on a single element of the traffic management system and the use of small networks of a particular type (i.e. freeway, arterial, or urban streets).

Only a few simulation studies evaluated *integrated* traffic management systems. For example, Reiss and Gartner (1991) conducted simulation studies to assess the performance of IMIS (an earlier version of INFORM). Nine scenarios were studied by varying the traffic demands and the locations and severity of incidents.

New developments in adaptive traffic control, dynamic traffic assignment, and route guidance systems require more powerful integrated simulation tools for evaluation and design refinement. For example, these systems focus on the collection and utilization of real-time traffic information to adjust signal timing and route advisory. One can hypothesize that their success depends in large part on whether or not reliable information can be obtained and used in time to modify control and guidance to best satisfy the needs of changing traffic conditions. A modeling system capable of realistically simulating the traffic flow in the network and its dynamic interrelationship with the control and route guidance system under a candidate design is a necessity for both the development and evaluation of such systems. SIMLAB is a simulation model designed to meet this need (Ben-Akiva et al., 1995).

1.2.3 Traffic Simulation Tools

Traditionally, traffic simulation models were developed independently for different facilities (e.g. freeways, urban streets, etc.). A wide variety of simulation models exists for various applications (see Koutsopoulos and Yang, 1992, for a review). Representative models, classified according to their representation of traffic flow, are:

- Macroscopic: FREFLO (Payne, 1978), NETVACI (Sheffi et al., 1982), KRONOS (Michalopoulos and Plum, 1986), AUTOS (Gilmore et al., 1995), METANET (Messmer and Papageorgiou, 1990), and METACOR (Haj-Salem et al., 1994).
- Mesoscopic: METROPOLIS (de Palma et al., 1996), DYNASMART (Mahmassani et al., 1994), DYNAMIT (MIT, 1996) and INTEGRATION (van Aerde and Yagar, 1988; van Aerde, 1992)
- Microscopic: INTRAS (FHWA, 1980), FRESIM (FHWA, 1994), NETSIM (Sibley, 1985), CORSIM (FHWA, 1996), THOREAU (Codelli et al., 1992), FLEXSYT-II (Middelham et al., 1994b), and AIMSUM2 (Barcelo and Ferrer,

1995).

Most of these models were developed for evaluation; and few for real-time support of ATIS operation and traffic prediction (e.g. DYNASMART, DYNAMIT). Most of the old generation models (for example, NETSIM, FRESIM, FLEXSYT-II, and earlier versions of AIMSUM2) do not represent vehicle paths. They use intersection turning proportions to determine vehicle movements and hence are not suitable to simulate route guidance. In some simulation models (for example, NETSIM), control elements such as pretimed and adaptive signal controls exist. However, the control logic is "hard coded" in the same simulation program. One exception is FLEXSYT-II, a DOS-based microscopic simulation tool for intersection design and traffic signal control studies. It allows users to specify the control algorithm in a specially designed formula language (Middelham et al., 1994b). Recent development of FLASH – A FLEXSYT Application Shell – is underway in the Netherlands[1]. FLASH includes a MS-Windows based network editor and improved user interface. It also has the ability to program a tested control logic into an EPROM to be used in a real-world operational controller unit.

In conclusion, these microscopic models cannot effectively meet the requirements of evaluating integrated ATIS and ATMS applications at an operational level, because they have limited representation of travel behavior or are inflexible in modeling more advanced surveillance and control systems.

A new generation of traffic simulation models have been developed (or are under development) for ITS applications. Examples are AUTOS, METROPOLIS, DYNASMART, DYNAMIT, INTEGRATION, THOREAU, and AIMSUN2. AUTOS is a macroscopic traffic model developed for ATMS applications at traffic management centers such as testing signal optimization, emergency vehicle response management, and human factors. The simulation model in METROPOLIS, DYNASMART, DYNAMIT, and INTEGRATION are mesoscopic, designed mainly for dynamic traffic assignment applications. THOREAU is a microscopic model developed for ITS

[1]For more information on FLASH, see http://193.79.206.33/id/fla_proj.html or contact the Transport Research Center of AND Identification.

evaluation. However, it has a very long running time. AIMSUM2 is also a recently developed microscopic traffic simulator for ITS analysis which has a k-shortest path based routing algorithm and limited simulation of pretimed signal controls Barcelo and Ferrer (1995); Barcelo et al. (1996). Other microscopic simulation models are also under development for modeling Automated Highway Systems (Eskafi et al., 1996; Gollu and Varaiya, 1996).

While these models have all been successfully applied in particular studies, a common shortcoming is the relatively limited range of applications. Some of them are designed for particular applications and useful only for specific purposes, while others do not support advance surveillance and control systems or integrated networks. No model has the integrated componentry and functionality required for evaluation of dynamic control and route guidance strategies on general networks.

The emerging ITS technologies add new functionalities to traffic management systems, such as mainline traffic control, real-time route guidance, and incident management. The lack of an integrated simulation environment with realistic user behavior for "real time" traffic management studies has become a bottleneck in ATMS and ATIS research and development. The need for the development of a more sophisticated methodology and simulation tools has been pointed out by a few researchers, including Santiago and Kanaan (1993); Lin (1993); Underwood and Gehring (1994). Enhancement of existing models and development of new models was motivated by these requirements. For example, FHWA has been revising its simulation tools to satisfy the requirements of ITS applications (Santiago and Kanaan, 1993). Two of its popular microscopic traffic simulators for freeways (FRESIM) and urban networks (NETSIM) have been combined into a more general traffic simulation framework called CORSIM (FHWA, 1996) to which vehicle path processing is being added. Leonard II (1993) and Jayakrishnan and Rindt (1996) described a simulation approach called "Testbed Simulation Workbench (TSW)", a framework for a hybrid simulation environment which communicates with real-time traffic surveillance and control systems.

The simulation laboratory developed in this research seeks to overcome overcoming some of the limitations mentioned above. Its major differences from the existing approaches are:

(1) it supports simulation of advanced traffic surveillance and control systems;

(2) it provides the capability of modeling traffic prediction using a mesoscopic traffic simulation/assignment approach;

(3) it supports the generation of consistent route guidance; and

(4) it facilitates implementation of a wide range of scenarios and systems by separating the simulation of traffic flows from the simulation of traffic management systems.

1.3 Thesis Outline

The remaining chapters of this dissertation are organized as follows. Chapter 2 discusses the evaluation methodology and outlines the overall framework of the simulation laboratory. Chapter 3 describes a microscopic simulator (MITSIM) designed for modeling traffic operations and system performance in integrated traffic networks. Chapter 4 presents a traffic management simulator (TMS) capable of modeling a wide range of traffic management schemes and generating predictive route guidance. Chapter 5 describes a mesoscopic traffic simulator (MesoTS) used as traffic predictor in TMS. In Chapter 6 the developed simulation methodology is demonstrated: a prototypical predictive route guidance system is evaluated using the Amsterdam A10 beltway. Finally, Chapter 7 concludes this dissertation by highlighting the main contributions and discussing directions for future research.

Chapter 2

Evaluation Framework

While there exist different approaches to evaluate and refine the design of dynamic traffic management systems, a simulation based approach has distinct advantages. In a computer-based laboratory environment, complicated interactions between the components of the system can be studied in a controlled environment. In this dissertation, a simulation laboratory that integrates traffic control and routing strategies with driver behavior and network performance models is developed. The system is designed to facilitate the evaluation and refinement of traffic management systems under various conditions, taking into account the dynamic and stochastic nature of transportation systems. By obtaining suitable MOE's under a wide range of scenarios, one can improve the design of traffic management strategies. In this chapter, we describe a framework of applying the simulation approach for evaluation and design refinement of dynamic traffic management systems.

2.1 Process for Evaluation and Design Refinement

The evaluation and design refinement of an integrated traffic management system is a complex process. The first step of this process is to identify the objectives of the traffic management system. Alternative traffic management systems should be evaluated based on how successful they are in achieving the desired objectives. Potential objectives include: improving the level of service, reducing emissions, improving safety, maintaining control stability, and assuring prompt response to emergency situations.

A large number of diverse and complex interactions between the various objectives are expected. Moreover, the number of control devices, the different types of control strategies, and the various ways the devices can be used (independently or in combinations) allow for many possible management strategies. The use of simulation allows engineers to test many potential system designs and control strategies under various hypothetical conditions.

Figure 2-1 illustrates the overall framework for evaluation and design refinement. The process begins with the specification of the traffic control and routing systems. The system specification, one of the two inputs to the simulation laboratory, defines the elements (e.g. surveillance and control devices) and the logic that determines how the system operates. The other input is the scenarios representing the operational environment under which the system will be evaluated. Scenarios are determined by the following factors: (i) traffic demand; (ii) events (including incidents and surveillance and control device failures); and, (iii) driver behavior and vehicle characteristics.

In this evaluation process, a set of candidate designs are usually considered and compared against a "base" case. Alternatives or candidate designs are defined by specific logic and parameters and then tested over a range of scenarios. The initial scenarios represent expected conditions under which the system operates. The simulated performance measures quantify the effectiveness of the system and illuminate potential design shortfalls. This knowledge is used to formulate additional scenarios in order to further challenge the system and test its robustness. In this way the scenarios under which each design is tested are generated iteratively, and will likely involve an engineer's professional judgment. These results are then analyzed to suggest modifications to improve the original design.

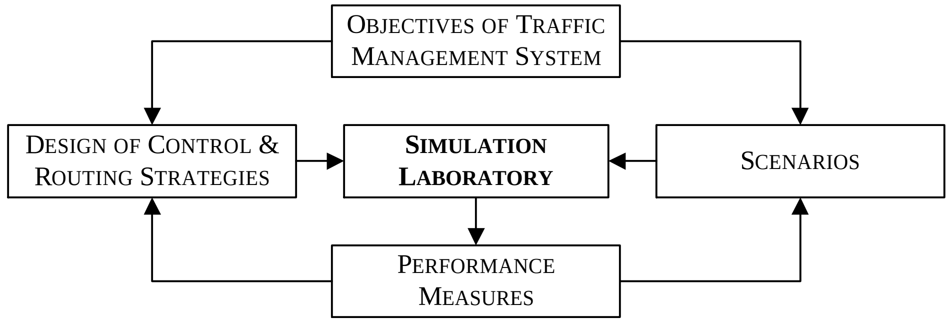


Figure 2-1: Overall design-evaluation framework

2.2 Traffic Simulation Laboratory

The core of the evaluation process is the traffic simulation laboratory (SIMLAB) which consists of a traffic flow simulator (MITSIM) and a traffic management simulator (TMS). The traffic flow simulator models the driver behavior and vehicular flow in the network, while the traffic management simulator mimics the control and routing functions performed by the particular system under evaluation. The simulation laboratory is designed to:

(a) represent a wide range of traffic management systems;

(b) model drivers' response to real-time traffic information and controls; and,

(c) incorporate dynamic interactions between the traffic management system and the drivers in the network.

The interaction between the traffic flows in the network and the control and route guidance generated by the system is a critical element for modeling dynamic traffic management systems. Traffic control and route guidance affects the behavior of individual drivers and, hence, traffic flow characteristics. The changes in traffic flows are in turn measured and potentially anticipated by the system and consequently influence present and future control and routing strategies. This simulator provides laboratory environment for the coupling of traffic control and routing logic with traffic flow. This system is a useful tool for testing and evaluating new concepts, algorithms, and technologies in research and development of ITS applications.

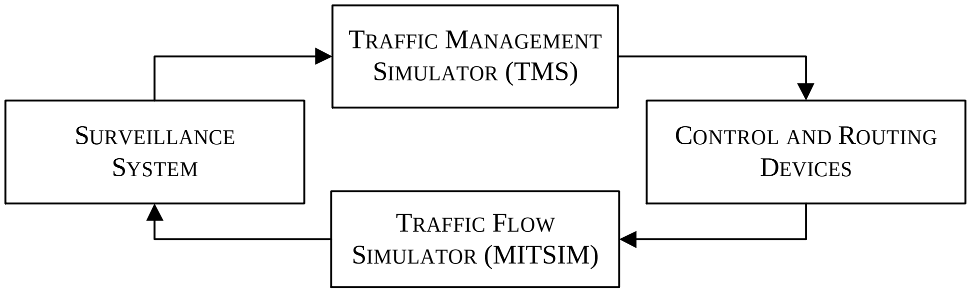


Figure 2-2: Simulation framework for dynamic traffic management

Figure 2-2 presents the main components of the simulation laboratory and their relationships. The functionality of the traffic simulation laboratory is provided by the following elements:

(1) traffic flow simulator;
(2) traffic management simulator;
(3) surveillance system; and,
(4) control and routing devices.

A microscopic simulation approach, in which movements of individual vehicles are represented, is adopted for modeling traffic flow. This level of detail is able to model the stochastic nature of traffic flow and drivers' response to route guidance. Such modeling is necessary for evaluating dynamic traffic management systems at the operational level.

The traffic management simulator represents the candidate traffic control and routing logic under evaluation. The control and route guidance provided by the traffic management system feed into the traffic flow simulator via various traffic control and routing devices in the simulated network. Drivers in the network respond to the traffic controls and guidance while interacting with one another. The outcome of the drivers' behavior is observed by the surveillance system module representing traffic sensors and probe vehicles. This module provides the traffic management simulator with the measurement of real-time traffic conditions.

2.2.1 Traffic Flow Simulator

In SIMLAB a microscopic traffic simulator (MITSIM) is developed to model the traffic in the network and represent the "real world". In MITSIM the traffic flow and network elements are represented at a level of detail sufficient to capture the sensitivity of vehicular flow to real-time control and route guidance and to simulate the surveillance system that supports dynamic traffic management. The model is designed to replicate traffic flow as accurately as possible. The main elements of MITSIM and their characteristics are outlined below. Technical details are presented in Chapter 3.

Network components The road network along with the traffic surveillance and control devices are represented in detail. The road network consists of nodes, links, segments, and lanes. The surveillance system consists of various detectors that collect traffic data, which can be either recorded in files and/or sent to the traffic management simulator. Traffic signal devices at intersections, speed limit signs, message signs, lane use signs, toll booths and special facilities (e.g. tunnels) are also represented. Their states are dynamically updated by the traffic management simulator.

Travel demand The traffic simulator accepts as input a vehicle trip table[1] and/or time-dependent origin to destination (OD) flow tables. These vehicle trip and OD flow tables represent the expected traffic demands and are defined as part of a scenario for evaluation.

Driving behavior and vehicle movement The OD flows specified in a scenario are translated into individual vehicles wishing to enter the network at a specific time. Behavioral parameters (e.g. desired speed, impatience in following a slower vehicle, critical gaps in changing lanes, compliance rates to traffic signals and signs, etc.), information accessibility, and vehicle characteristic parameters are randomly

[1] A vehicle trip table consists of records that provide information on individual vehicle trips, including departure time, origin and destination node. Each record can also optionally provide the vehicle type and path to take.

assigned to each vehicle/driver combination. MITSIM moves vehicles according to car-following and lane-changing models. The lane changing model distinguishes between mandatory and discretionary lane changing. Once a decision to change lanes is made, a probabilistic gap acceptance model is used to determine whether the desired change can be executed. MITSIM also represents drivers' responses to traffic signals, speed limits, incidents, and delays at toll booths. Merging and turning movements at intersections are also modeled.

Each vehicle in MITSIM has its own destination and a path leading to that destination. Vehicles are moved at a fixed step size along their paths in accordance with various constraints. The information provided by the route guidance system affects drivers' pre-trip and en-route route choice decisions through probabilistic route choice models. These models calculate the probabilities of choosing alternative routes at each intersection (or decision point) as a function of the perceived or provided information.

2.2.2 Traffic Management Simulator

The control and route guidance system to be evaluated is represented in the traffic management simulator (TMS). This simulator receives as input from MITSIM real-time traffic measurements, as reported by the surveillance systems. Based on the received surveillance data, TMS generates control and route guidance according to the implemented logic and updates the state of traffic signals and signs in the network.

For the traffic simulation laboratory to be useful in evaluating advance traffic management systems, it is important that TMS has a generic structure to model different types of systems. In general, control and route guidance systems can be classified as *pre-timed* and *adaptive* systems. In a pre-timed system, control and route guidance are pre-determined based on historical traffic information using off-line analysis; in an adaptive system, control and route guidance are generated on-line based on real time traffic information obtained from surveillance sensors and environmental conditions such as weather, scheduled construction work, etc.

Adaptive systems can be further divided into reactive and proactive systems. Most of the existing adaptive systems are reactive, whereas control and route guidance are provided based on prevailing traffic conditions. Proactive systems are the most recent development. A proactive system requires two main components:

- Network state estimation, and
- Prediction-based control and routing strategy generation

Therefore, a generic algorithm for a proactive system can be represented as follows:

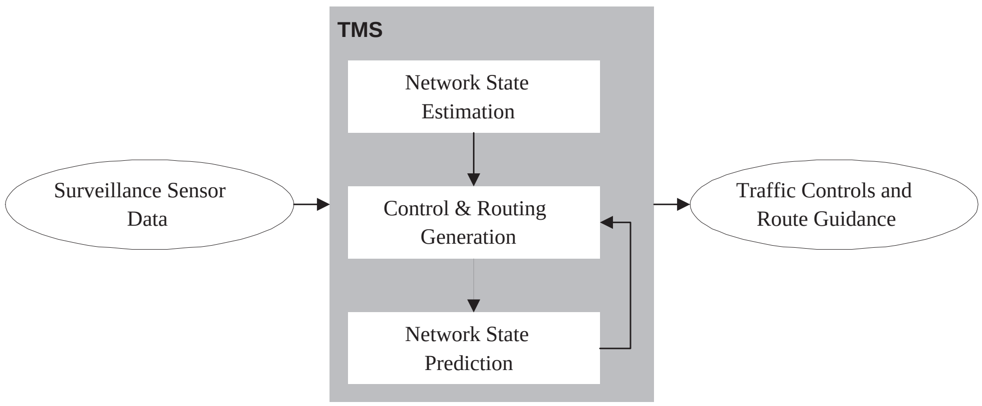


Figure 2-3: Structure of proactive traffic control and routing systems

The role of the Network State Estimation module is to obtain the best estimate of the current network state, represented by link flows and travel times, origin and destination matrix, and incidents using the available information (e.g. surveillance, reports, etc.). The Control and Routing Generation module generates signal control and route guidance for the next time period. The Network State Prediction module is used to predict future traffic conditions; it can use various approaches. The default traffic prediction module is based on a mesoscopic traffic simulator (MesoTS). MesoTS is described in detail in Chapter 5. Through the interaction between the Control and Routing Generation and the Network State Prediction, this system optimizes traffic control and generates consistent route guidance.

2.2.3 Surveillance System

The surveillance system module bridges MITSIM and TMS. SIMLAB can simulate the following types of sensors:

- *Traffic sensors,* which collect traffic counts, occupancy, and speed data at fixed points in the network (e.g. loop detectors);
- *Vehicle sensors,* which extract information on individual vehicles such as vehicle type, over-height attributes, vehicle identification, etc.;
- *Point to point data sensors,* which extract information from probe vehicles and/or detectors (such as travel time from one point in the network to another); and
- *Area wide sensors,* which collect information on vehicles (such as positions and speeds) within a wide detection zone (e.g. radar detectors and video cameras).

The readings from all active sensors are communicated to TMS and appropriate errors are then added to represent detector and communication errors. The magnitude of the errors can be controlled in the scenario definition. This allows for studying the sensitivity of the system performance with respect to the accuracy of its surveillance system.

2.2.4 Control and Routing Devices

SIMLAB is designed to support a wide range of traffic control logics, including pre-timed signal controls, traffic adaptive controls, metering controls, and control strategies in response to incidents. Traffic regulation and route guidance are conveyed to individual drivers in the simulated network via various devices and information means (e.g., VMS, in-vehicle units, etc.). When vehicles in the network view a signal or sign, they respond to the given control and route guidance according to the behavioral models. For example, delay at a particular downstream bottleneck can be shown on an upstream variable message sign. Drivers who view the message will either change their path or continue on their current path.

2.3 Simulation Output and Measures of Effectiveness

The output provided by MITSIM allows calculation of a rich set of measures of effectiveness (MOE) useful for evaluation of a wide range of traffic management systems. Typical simulation output includes:

(i) Vehicle specific information, such as total travel time, distance traveled, and average speed, reported when vehicles reach their destination nodes;

(ii) Readings from traffic sensors, such as traffic counts, occupancy, speeds, point to point travel times, and vehicle classification, reported at either a fixed frequency or upon recording by the sensor;

(iii) Segment specific traffic data such as density, average speed, and travel times, reported at a fixed frequency or when simulation is completed;

(iv) Time-dependent link travel times and vehicle counts, reported at the end of the simulation; and

(v) Graphical display of the vehicle movements, segment specific traffic data (e.g. average density and speed), and sensor measurements (e.g. counts, average flows, speeds and occupancies).

By collecting disaggregate data on vehicle movements and applying appropriate models, MOEs concerning fuel consumption, emissions, safety, etc. can also be obtained.

Since MITSIM is a stochastic simulation model, simulation runs using identical input may produce different outcomes. To obtain statistically significant evaluation results, multiple simulation runs should be conducted for each evaluation under a given scenario. Information of interest is extracted and averaged from the output of individual simulation runs.

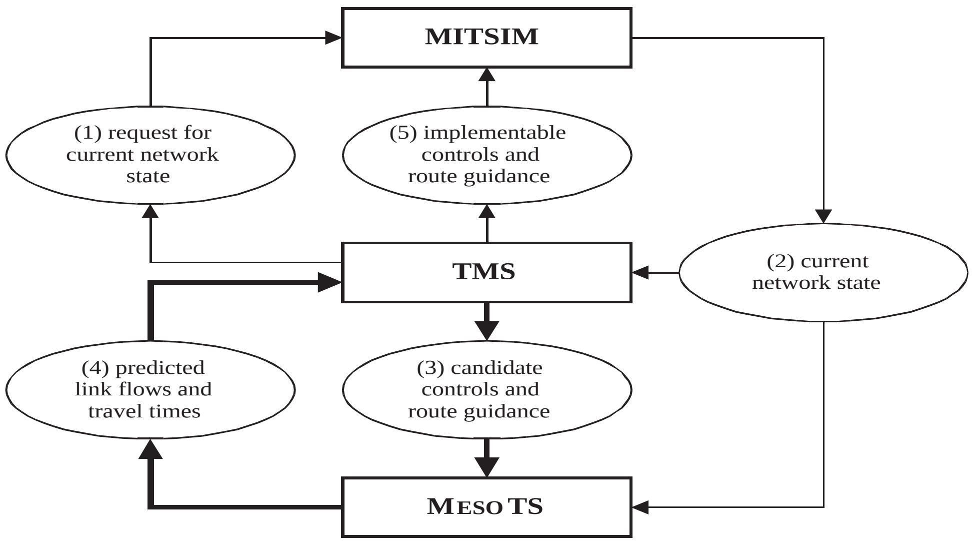


Figure 2-4: Iterative guidance generation and traffic prediction

2.4 System Integration

A prototypical predictive control and route guidance generation system is implemented in SIMLAB to demonstrate the use of the methodology and simulation tools developed in this research. In this particular application, MesoTS, the mesoscopic traffic simulator presented in Chapter 5, is used for traffic prediction.

The addition of the real-time traffic prediction model significantly complicates the synchronization of modules in the simulation laboratory. The coordination among the simulators for a typical traffic prediction cycle and testing of a candidate strategy is depicted in Figure 2-4 and summarized below:

(1) When TMS needs a new prediction of future traffic conditions (e.g. next 45 minutes), it sends a message to MITSIM to request the current network state as captured by the surveillance system. To simplify the discussion, in the following sections, we assume that the surveillance system provides complete information on the network state. Hence, a model for estimating current network state is not used.

(2) As MITSIM receives this request, it scans and writes the current network

state into a file. Network state is represented by the positions, speeds, and destinations of vehicles currently in the network and those waiting to enter the network. When MITSIM finishes reporting the current network state, it sends a message to TMS and continues the simulation.

(3) TMS also sends the current signal controls and route guidance to MesoTS in order to start the simulation for the given prediction horizon.

(4) When MesoTS receives the message from TMS, it utilizes the network state from MITSIM in step (2) and predicts future traffic conditions for a given prediction horizon (e.g., 45 minutes). When the mesoscopic simulation for the given prediction horizon is complete, MesoTS writes the predicted time dependent link flows and travel times to a file. Then it sends a message to TMS to signal the completion of the prediction task.

Having received this message, TMS may proceed in two ways: (i) if the candidate solution is not satisfactory it restarts a mesoscopic simulation with the same initial network state but updated route guidance and signal controls; otherwise, (ii) it terminates the iterations and accepts the route guidance and signal controls. In the first case, steps (2)-(4) are repeated. In the second case, step (5) is executed.

(5) TMS sends a message to MITSIM to announce that new guidance and signal control strategies are available for implementation. After a pre-specified *computation delay* – the difference between the time the network state is reported and the expected completion time for the generation of control strategies and route guidance – MITSIM switches to the updated controls and route guidance and continues the simulation.

2.5 Software Architecture

SIMLAB is implemented in C++ using object oriented programming. The software architecture of this system is depicted in Figure 2-5. The system consists of three simulation programs, namely MITSIM, TMS, and MesoTS, and a master controller

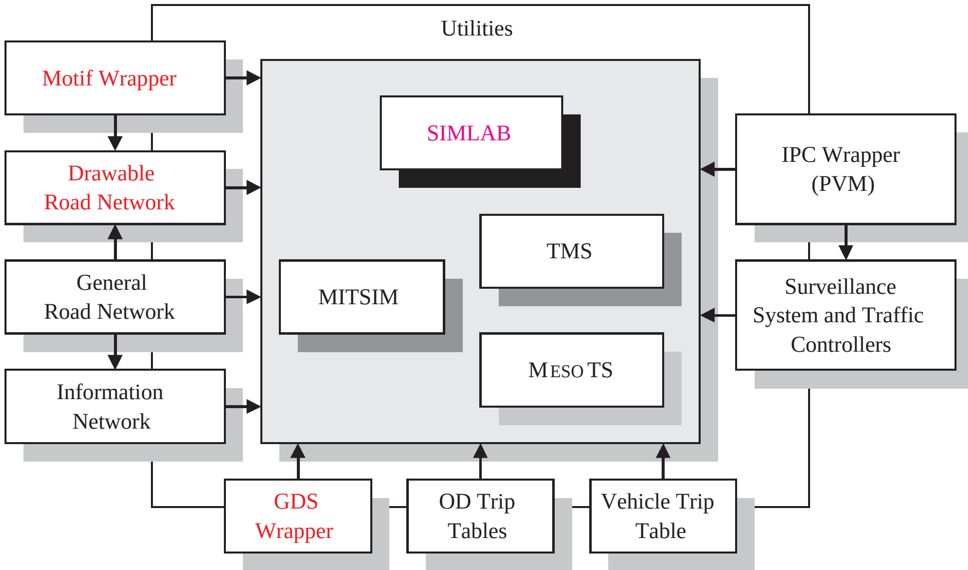


Figure 2-5: Software architecture

called SIMLAB. The master controller is a program that launches and synchronizes the execution of the SIMLAB modules (as described in the example of Section 2.4). Information is exchanged between these programs using interprocess communication and data files.

2.5.1 Software Elements

The implementation of SIMLAB can be divided into the following software elements.

(a) *Utilities* is a collection of functions and classes convenient to all programs in the simulation laboratory (e.g. *random number generators*, *simulation clock*, etc.).

(b) *Motif Wrapper* includes the base classes for building the graphical user interface. Some of these classes are developed using the Silicon Graphics IRIS ViewKit library.

(c) *GDS Wrapper* consists of several classes that load and draw GDS[2] objects.

(d) *Interprocess Communication (IPC) Wrapper* is the interface for communication between modules. The current system is implemented using PVM (Geist et al.,

[2]GDS is a major computer aided design (CAD) and mapping software package used in the transportation industry.

1994).

(e) *General Road Network* contains the implementation of various network objects, including the base classes for network, nodes, links, segments, lanes, sensors, signals and signs, toll booths, etc. Some of these classes also have a derived drawable equivalent to provide geometric information necessary for drawing the objects.

(f) *Drawable Road Network* consists of classes derived from the base network objects. It provides additional information and methods for drawing network objects. It also contains base classes for graphical objects (e.g., windows, drawing area, mouse and key interactors, etc.) common to all SIMLAB simulators.

(g) *Information Network* contains the classes that implement information and models used for vehicle routing. Examples are link travel time table, graph representation of the network and shortest path calculation, static and dynamic routes, etc.

(h) *Surveillance System and Traffic Controllers* are classes that implement surveillance modules and signal controllers common to most traffic management systems.

(i) *OD Trip Table* consists of classes that implement the time dependent OD flows and methods for generating vehicles.

(j) *Vehicle Trip Table* is a class that reads input data on trip information for *individual vehicles* scheduled at particular departure times.

FLEX, a lexical analyzer generator (Levine et al., 1992), and BISON++, a parser generator[3], are used in developing the parsers for reading various data files (e.g., network database, OD tables, signal plans, parameters files, etc. See Appendix F). Based on the grammar that specify the format of a particular input data file, the C++ code required to read the data is generated by FLEX and BISON++.

[3]BISON++ is based on GNU version of BISON (Donnelly and Stallman, 1992) and created by Alian Coetmeur (coetmeur@icdc.fr). The package is available via anonymous ftp at *ftp.cso.uiuc.edu* in the directory */pub/lang/C++/tools/flex++bison++/LATEST*.

2.5.2 Distributed Implementation

The simulation laboratory is implemented as a distributed system using PVM. Its components may run on the same or different processors. A typical configuration uses 2-3 processors, with the most powerful processors assigned to MITSIM and MesoTS. The modules in the system communicate with each other via message passing, shared memory, and data files. In every iteration of the simulation, each module in the simulation laboratory checks for messages received from other partners. All messages are processed before the simulation is advanced to the next iteration. When all processes are "checked in," SIMLAB advances the simulation clock by 1 step and broadcasts the new time to every module in the system. Each module then proceeds until it reaches the master clock time received from SIMLAB.

2.5.3 Graphical User Interface

SIMLAB can be used in batch or graphical mode. The graphical version has a user interface that provides additional features such as:

- *Interactive control of simulation*: Menus, keys, mouse buttons and tool-bar icons are used to provide easy access to software capabilities. For example, the user can load a particular scenario, choose output of interest, zoom and navigate over the network, pause and resume a simulation run, and set check points at particular times to examine the intermediate results or network states.
- *Display traffic network*: All simulators (MITSIM, TMS, and MesoTS) can display network geometry, traffic data (e.g. average speed and density of each segment), measurements of surveillance detectors, and state of traffic signals and signs. Also, vehicle movements are animated in MITSIM and MesoTS (see Figure 2-6).
- *Display OD flows*: OD flows in the entire network and OD flows for a selected node can also be graphically displayed.
- *Query network objects*: All network objects (nodes, links, segments, lanes, sensors, signals and signs, paths, and OD pairs) can be queried by their ID

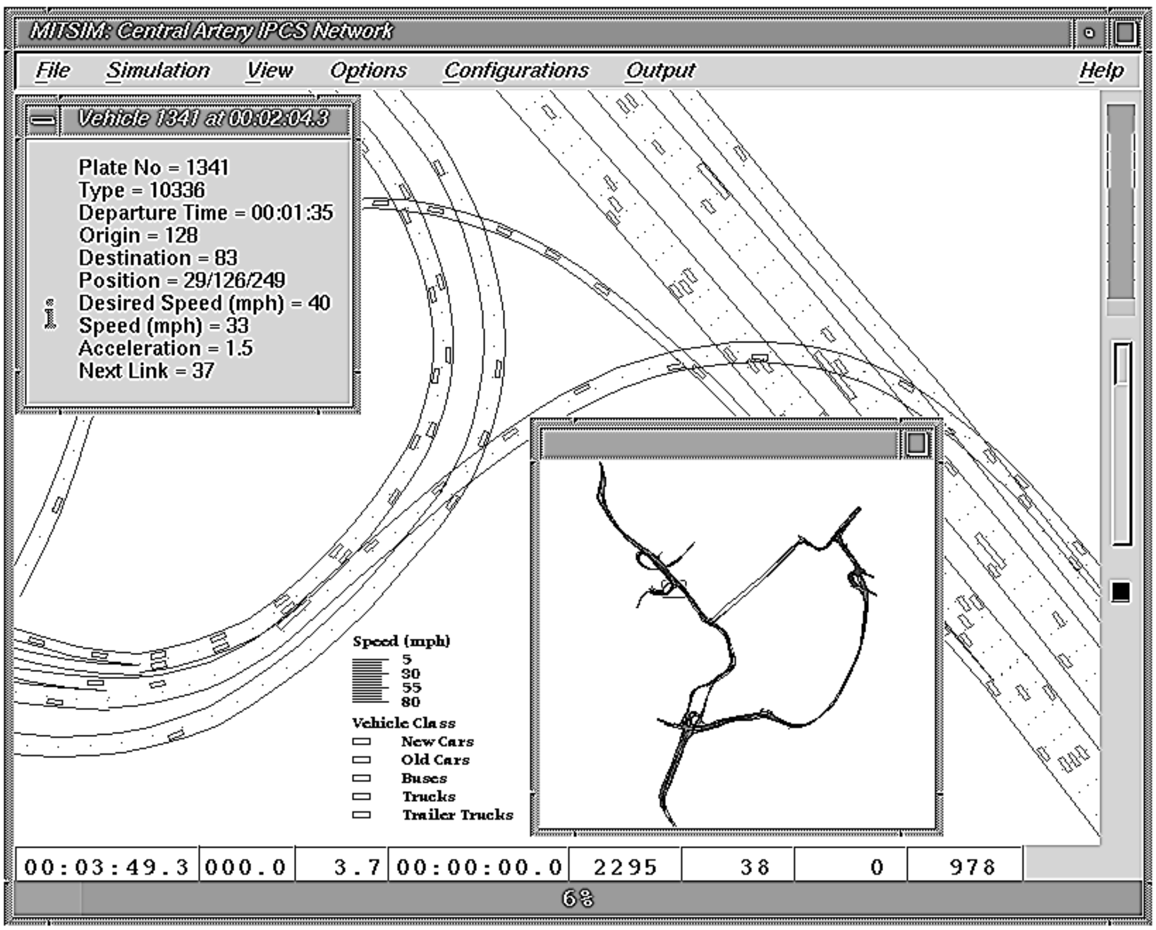


Figure 2-6: Graphical interface

and drawn with dedicated symbols.

The graphics mode, although slower than batch mode, provides an essential tool for checking the correctness of input data and visualizing the simulation process.

Chapter 3

Microscopic Traffic Simulator

A MIcroscopic Traffic SIMulator (MITSIM) has been developed for modeling traffic flows in networks involving advanced traffic control and route guidance systems (Yang and Koutsopoulos, 1996). MITSIM represents networks at the lane level and simulates movements of individual vehicles using car-following, lane changing, and traffic signal response logic. Probabilistic route choice models are used to capture drivers' route choice decisions in the presence of real time traffic information provided by route guidance systems. MITSIM is designed as a testbed for evaluating traffic management systems with the following characteristics:

- it provides real-time sensor data that mimics the surveillance capacities of the traffic management systems in an ITS environment;
- it accepts traffic controls and routing information as input from traffic management systems and maintains the state of traffic signals and signs in the simulated network; and
- it calculates a set of measures of effectiveness (MOE) that represent the performance of the systems to be evaluated.

In this chapter we first present the overall design of MITSIM. Then we describe the details of its major elements: network, traffic controls and surveillance sensors, incidents, travel demand, vehicle routing, and vehicle movements. We conclude by discussing the main simulation output.

3.1 Overall Design

The main structure of the traffic simulation model is an iteration of functions at specified frequencies (time-based) or when certain events occur (event-based). The simulation logic is summarized in the flowchart in Figure 3-1.

The simulation starts with the loading of simulation parameters, the road network description, and the scenario definition. Initialization of communication with the traffic surveillance and management modules also occurs. Once the simulator is initialized, an iterative procedure begins. The tasks performed within each iteration include:

1. Update the state of traffic signals, signs, and incidents.
2. Update routing tables and calculate shortest paths to all destinations.
3. Read new origin-destination (OD) trip tables and generate corresponding virtual vehicle queues at the origin nodes.
4. Load vehicles from the virtual queues into the network.
5. Update vehicles acceleration rates and make necessary lane changes.
6. Advance vehicles to new positions and update their speeds. If a vehicle activates a sensor, the corresponding measures (speed, occupancy, etc.) are recorded by the surveillance system module. At the end of a lane, a vehicle is either removed from the network (if it arrives at its destination) or handed to the downstream lane.
7. Update the display if the graphical user interface (GUI) is enabled.
8. Calculate MOE and/or send network states to external MOE or GUI modules.
9. Update the simulation clock and move to next iteration.

In general, MITSIM uses a time-based simulation approach in processing the vehicle movements. The car-following, lane changing, and event and signal response

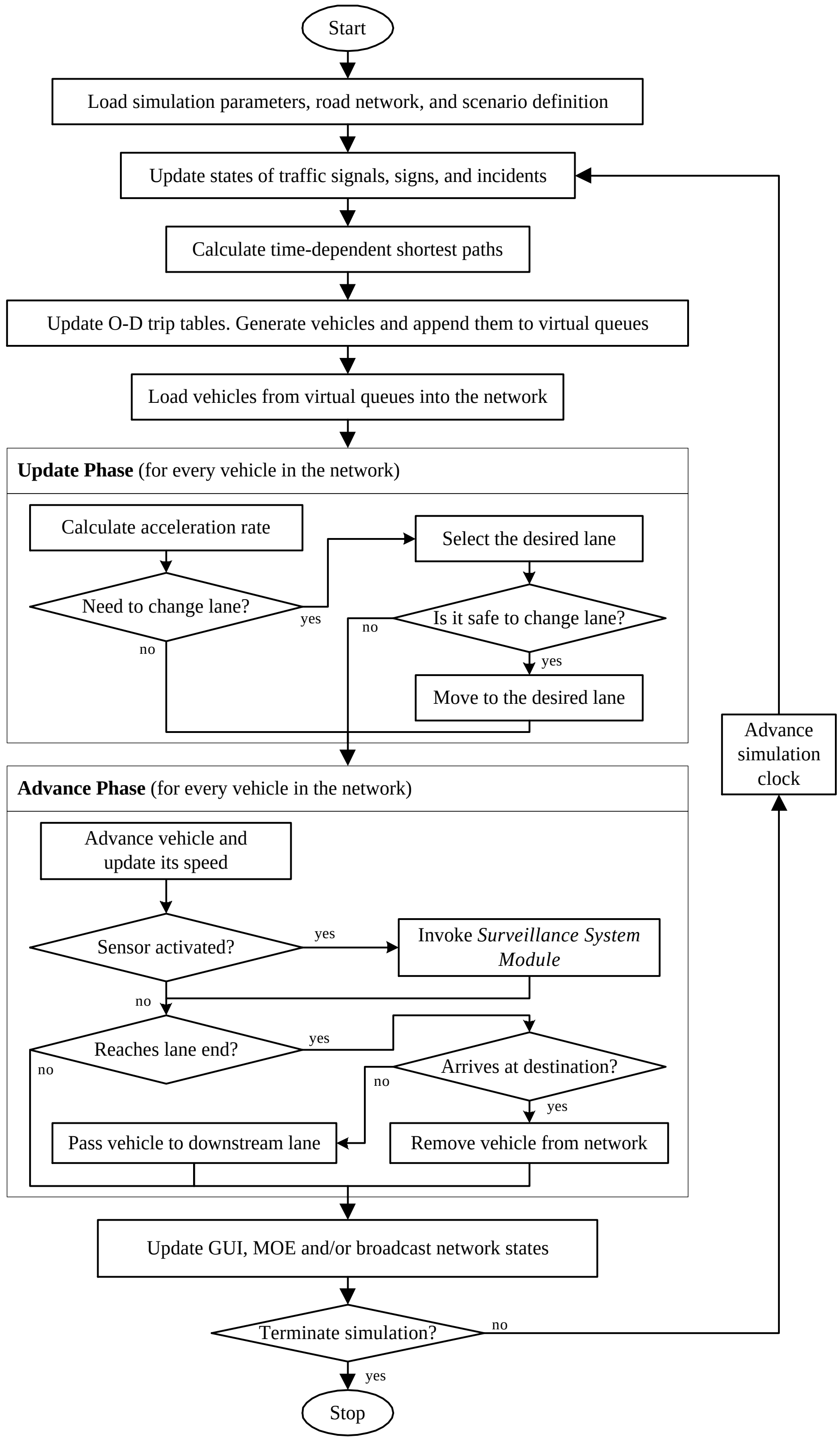


Figure 3-1: Flow chart of the traffic simulation model

functions are invoked for each vehicle at a specified interval (e.g. 1 second). Speeds and positions of the vehicles and states of surveillance sensors are updated at a higher frequency specified by the user (e.g. 1/10 or 1/2 second). This step size is subsequently used to advance the simulation clock. The step size ω, which is drawn from a given distribution and randomly assigned to individual vehicles, has to be greater than or equal to the step size τ and a modulus of τ. The actual step size applied for a particular vehicle at a given time may vary (but must be a modulus of τ) during the simulation as drivers' reaction times differ in some special conditions (e.g. too close to the leading vehicle, making a lane change, or delayed at toll booth and red signals, etc.).

3.2 Network Representation

MITSIM represents road networks with nodes, links, segments, and lanes. It allows the simulation of traffic operations in integrated networks of freeways and urban streets. The data that describes the network is read from a network database file, which can be created using an interactive graphical *Road Network Editor* (RNE) developed for MITSIM. The network database includes description of all network objects, lane connections, lane-use privilege, regulation of turning movements at intersections (no left-turns, for example), traffic sensors, control devices, and toll plazas.

Nodes: A node is either an intersection of several roadways or a origin and/or destination where traffic flows enter or leave the simulated network. Each node is represented by its *type* (e.g. intersection, origin/destination), a unique *identification number*, and an optional *name*.

Links: Links are directional roadways that connect nodes. Each link consists of one or more segments and is characterized by its *type* (freeway, ramp, urban street, tunnels, etc.), an *identification number*, *starting* and *end nodes*, and the *segments* it contains. An inbound link and an outbound link of a node are *connected* if there exists at least one lane connection between the two. Turning restrictions from one link to another are specified by a *turning prohibition table*.

Segments: Segments are road sections with uniform geometric characteristics such as number of lanes, grade, curvature, design speed, etc. Each segment is characterized by its *speed limit*, *design speed*, *grade*, *geometry*[1], a unique *identification code*, and the *lanes* it consists of.

Lanes: MITSIM represents the network at the lane level. Each lane is described by two data items: (i) *lane code*, a unique identification code; and (ii) a *lane change regulation* and *lane-use privilege* code. Lane change regulations determine whether a lane change between adjacent lanes is allowed. Lane use privilege specifies the classes of vehicles that are or are not allowed to use the lane. For example, a lane may be assigned to high occupancy vehicles (HOV), electronic toll collection (ETC) vehicles, and/or commercial vehicles only. It may also exclude the use of the lane by particular types of vehicles such as over-height vehicles, trucks, etc. The lane-use privilege code for a particular lane can be any consistent combination of predefined basic types.

Each lane can be connected to one or more upstream and downstream lanes (e.g. at toll plaza, merging, and fork area) or has no lane connection at all (e.g. lane drop or network boundary). A *lane connection table*, which contains a pairwise list of lane codes, is used to represent the connections between upstream and downstream lanes.

3.3 Toll Plazas

Each toll plaza is described by *visibility*, *segment code*, *location* in the segment, and a list of data items for each toll booth, including *lane use privilege code*, *default state* (i.e., free, open, or closed), *speed limit*, and *toll collection delay*. The lane use privilege code is used to restrict the types of vehicles allowed to use a particular toll booth. For example, a toll booth can be designated to cars, ETC, HOV, or any vehicle. The

[1]The geometry of a segment is referenced to the left curb line by *coordinates* of the end points and the *bulge* of the arc.

speed limit sets an upper bound on the speed of the vehicles passing through the toll booth (for booths with manual toll collection, this speed limit is zero). Toll collection delay is the average time it takes for a vehicle to stop at the booth to pay the toll.

A vehicle will stop if it enters a booth using manual collection or slow down to the speed limit if it is on an ETC lane. The service time at a manual toll booth is assumed to have a negative exponential distribution with mean μ and truncated at a maximum time, i.e.:

$$d_n = \min\{-\mu \ln(r_n), d_{max}\} \tag{3.1}$$

where:

d_n = time vehicle n stops for paying toll;

d_{max} = maximum stop time;

μ = average service time at the toll booth (a function of toll booth type and vehicle class);

r_n = a random number uniformly distributed between (0,1].

3.4 Traffic Surveillance and Control Devices

3.4.1 Surveillance Sensors

Surveillance sensors represent the various means used to extract data on traffic flows in the network. Sensors in MITSIM are represented by their technical capabilities rather than by simulating their operations in detail. For example, a measurement error is assumed instead of being derived by the physics that govern the operations of a sensor. When a vehicle passes a detector the occupancy and speed are interpolated based on its previous and current positions over the update phase (e.g. 0.1 second), assuming constant acceleration.

Sensors of the same type at the same longitudinal position in a segment are grouped into a *sensor station*. Each sensor station is characterized by *sensor type*, *sensor task* (i.e. the types of data collected by the sensor), length of *detection zone*, longitudinal *position*, and the *detectors* it consists of (only a subset of the lanes may be equipped). In addition, each sensor is assigned a working probability, which specifies the probability that the sensor is operational. The measurement errors for various types of sensors and data items are determined by parameters of the surveillance system module in TMS.

Simulation of other surveillance methods, such as external agency reports, is also possible in MITSIM. For example, when passing an incident, vehicles equipped with a cellular phone will report the incident to the traffic management simulator with a pre-specified probability.

3.4.2 Traffic Control Devices

MITSIM supports the simulation of a wide range of traffic control and route guidance devices, including intersection traffic signals (TS), yield and stop signs, ramp meters, lane use signs (LUS), variable speed limit signs (VSLS), portal signals at tunnel entrances (PS), variable message signs (VMS), and electronic route guidance devices.

Control devices can be either link-wide or lane specific. A link-wide device (e.g. VSLS, VMS) controls all the approaching traffic, while a lane specific device (e.g. LUS) controls only the approaching traffic in a particular lane. Control devices are characterized by their *location* (segment, lane, and longitudinal position), *type*, *initial state*, and *visibility*. Visibility represents the maximum distance from which the device is visible by the approaching drivers.

At any given time, the state of a traffic signal (TS, PS, ramp meters, etc.) can be *blank*, *red*, *yellow*, *green*, or *blank out*[2]. The state may also indicate whether the signal is *flashing* and/or an *arrow*, representing the priority (protected, yield, or stop) of a particular movement. Stop signs are represented by flashing red and yield signs by flashing yellow.

The state of VMS (including the fixed message signs) is represented by *message type* (e.g. lane use and lane change regulation for trucks, ETC and HOV vehicles; lane use recommendation for local and through traffic; route guidance, etc.). Depending

[2]A blank out sign indicates no vehicles is allowed to enter the link or lane

on the type of message, more information is specified regarding the lane and link to use, and the type (class and group) of vehicles to which the given message is relevant.

In the network database only the initial (or default) states for signal and signs are specified. The real-time state of signals and signs are managed by controllers in TMS.

3.5 Incidents

Incidents are lane blockages or capacity reductions caused by disabled vehicles, road construction, etc. Incidents can be placed anywhere in the network and activated and cleared at any particular time. Each incident may affect one or multiple lanes. Incident data is read from a scenario definition file. The information for an incident includes: *visibility*, *number of lanes affected*, *location* (segment and longitudinal position). Lane specific information for an incident includes the *severity* of the incident and its *length*, *maximum speed*, *start time*, and expected *duration*. The maximum speed of an incident sets the upper bound on the speed of the vehicles passing by, which can be used to simulate the rubber-necking effect of partial blockage incidents. The start time of an incident may differ from the time that the incident is detected by the traffic management system. The clearance time of an incident is its start time plus duration by default, but it can be shortened by TMS to a time before the "scheduled" clearance time. This allows for evaluation of different incident detection and management schemes.

3.6 Travel Demand

MITSIM accepts as input *time dependent* origin to destination (OD) tables. Each OD table contains a *time tag* – the time that this table becomes effective – and travel demands for individual OD pairs. The travel demand for each OD pair is characterized by the following data items: an *average departure rate* and its *standard deviation*, a *distribution factor*, and a list of *paths* that connect this OD pair. The last 3 data items are optional and default values are supplied.

The time intervals that individual OD tables remain effective may have different lengths. This allows the use of shorter time intervals during peak periods and longer time intervals during the off-peak. The OD tables are sorted chronologically in the input file according to their time tags. The simulator stores only the current OD tables and reads the new ones when the current one expires.

OD tables can be specified individually for each vehicle type or, alternatively, the simulator can randomly assign a type to each generated vehicle based on a *fleet-mix*[3] specified in the parameter file (See Table C.1). Vehicle type is a combination of vehicle *class* (e.g. high performance passenger cars, low performance passenger cars, buses, trucks, trailer trucks, etc.), and *group* based on *lane-use privilege* (e.g. HOV and ETC) and *access to information* (e.g. guided and unguided).

The standard deviation of the *average* departure rate represents the randomness of total travel demand for the corresponding OD pair. The departure rate used in a particular simulation run is randomly sampled based on the given average departure rate and its standard deviation (default is to sample from a normal distribution). The distribution factor, which is a value between 0 and 1, determines the percentage of vehicles departing deterministically. For example, a distribution factor of 0.4 indicates 40% of vehicles will depart according to constant headways and the remaining 60% of vehicles according to Poisson distribution (random). The choice of distribution factor should be made based on traffic conditions. A distribution factor of close to 0, for example, can be used for low congestion traffic conditions and a value close to 1 for high congestion (see May, 1990).

The inter-departure time between two vehicles is the reciprocal of the departure rate if the vehicle's departure is deterministic. For Poisson departures, inter-departure time is randomly drawn from negative exponential distribution, i.e:

$$t_{n+1} = t_n - \frac{\ln(r_n)}{\lambda} \qquad \lambda > 0,\ 0 < r \leq 1 \tag{3.2}$$

[3]Fleet-mix is the percentages of vehicles in each vehicle class (e.g. cars, buses, trucks, etc.)

where:

$t_{n+1} =$ departure time for the next vehicle;

$t_n =$ departure time for the previous vehicle;

$\lambda =$ departure rate;

$r_n =$ a random number uniformly distributed between $(0, 1]$.

An alternative method for specifying travel demand is to use a *vehicle trip table* file. The vehicle trip table contains a list of scheduled vehicle departures, sorted chronologically according to departure times. Each vehicle departure is described by its *origin*, *destination*, *type*, and a predetermined *path*. Type and path are optional and can be determined based on the fleet mix and the route choice model respectively.

After a vehicle is generated, a set of vehicle and driver characteristics and a pre-trip path are assigned. The details are described in the following two sections.

3.7 Vehicle Characteristics

The parameters representing vehicle performance (e.g. maximum acceleration and speed) are deterministic and type specific, while driver behavior parameters (e.g. desired speed) are randomly assigned when a vehicle enters the network according to pre-specified distributions.

Maximum acceleration rate: The maximum acceleration rate is defined for each vehicle class as a function of the grade of road segment and the speed of the vehicle (Pline, 1992), i.e.:

$$a^+_{GV} = a^+_{LV} - \frac{G\,g}{100} \tag{3.3}$$

where:

$G =$ gradient (%);

$a^+_{GV} =$ maximum acceleration rate at speed V on grade G;

$a^+_{LV} =$ maximum acceleration rate at speed V in level terrain;

$g =$ acceleration of gravity (32.2 ft/sec^2).

The default values of a^+_{LV} for five vehicle classes and five speed categories are based on FHWA (1980), FHWA (1994) and Pline (1992). These are documented in Table C.2 in Appendix C.

Maximum deceleration rate: The maximum deceleration rate is defined for each vehicle class as a function of its speed (see Table C.3).

Normal deceleration rate: Normal deceleration rate is used to: (i) smoothly slow down a vehicle in non-emergency situations; and, (ii) compute the *normal stopping distance* required for responding to a downstream event (e.g. traffic signal, incident, and exit). The normal deceleration rate is a function of vehicle class and its speed. The default values for normal deceleration rates are based on ITE (1982) (see Table C.4).

Desired speed: A driver's desired speed is randomly assigned when a vehicle enters the network and updated whenever the driver views a different speed limit sign or moves into a segment with a different *design speed*. The desired speed is calculated as follows:

$$v_n^0 = \min\{v_n^{sign} + \nu_n,\ \tilde{v}_n\} \tag{3.4}$$

where:

v_n^0 = desired speed of driver n;

v_n^{sign} = speed limit of the segment (or the value the driver viewed on a VSLS);

ν_n = driver behavior parameters randomly assigned based on the desired speed distribution; and

$\tilde{v}_n$ = maximum speed the vehicle can achieve in the segment (FHWA, 1980), a function of vehicle class and grade of the segment (see Table C.5).

The distribution of drivers with various ν_n are obtained from the parameter file (see Table C.7 for an example).

Speed distribution across lanes: To capture the speed difference across lanes, a maximum speed is also defined for each *lane* based on the *design speed* and the number of lanes in the segment, i.e.:

$$V_i^{max} = \kappa_{mi} \, V^{design} \tag{3.5}$$

where:

V_i^{max} = *maximum speed* vehicles can travel in lane i;

V^{design} = design speed of the segment; and

κ_{mi} = speed factor for the ith lane in a segment consisting of m lanes.

The parameters κ_{mi} capture the fact that, in segments with multiple lanes, the vehicles in the central and left lanes tend to travel faster than the vehicles in the right lanes. Suggested values of κ_{mi} are based on FHWA (1980) (see Table C.6 in Appendix C).

Target speed: The target speed vehicles actually try to achieve in a lane is constrained by both the desired speed and maximum speed in the lane, i.e.:

$$v_n^{target} = \min\{v_n^0, V_i^{max}\} \tag{3.6}$$

where v_n^0 and V_i^{max} are driver's desired speed and the maximum speed of lane i, as defined in Eq. 3.4 and 3.5 respectively. A driver may seek a discretionary lane change (see Section 3.9.4) if v_n^{target} is constrained by V_i^{max} instead of the driver's desired speed.

3.8 Vehicle Routing

MITSIM maintains two sets of travel time information: historical and real-time. Real-time link travel times are updated periodical or when information is received from TMS. The updated travel times are "transmitted" to the vehicles equipped with on-board route guidance devices when vehicles enter the range of a communication beacon. Upon receiving the new information, guided drivers select their routes based on the updated travel times. Any vehicle may respond to VMS according to a pre-specified compliance rate.

Route choice models are used to capture drivers' route choice decisions and response to traffic information. Regarding access to in-vehicle information, two driver groups are assumed: *informed* and *uninformed* drivers. Drivers may have either predefined or dynamically computed paths. Depending on the situation, a route choice model or a route switching model may be used to update a vehicle's paths.

3.8.1 Route Choice Model

The routes for vehicles without pre-specified paths are generated at each intersection using a *route choice model* (*see* Figure 3-2):

$$p(l|j,t) = \frac{\exp(V_l(t))}{\sum_{m \in L_j} \exp(V_m(t))} \tag{3.7}$$

where:

$p(l|j,t)$ = probability to choose link l for a vehicle that expects to arrive at node j at time t;

L_j = set of outgoing links at node j;

$V_l(t)$ = systematic utility of choosing a route with link l as the next link.

In the default model, the utility is a function of:

$c_l(t)$ = perceived travel time on link l at time t;

$C_k(t)$ = perceived travel time on the *shortest path* from node k (the downstream node of link l) to the destination if the vehicle arrives at node k at time t; and

z_l = penalty that captures freeway bias.

The perceived travel times, $c_l(t)$ and $C_k(t)$, are time dependent and calculated either from historical or real-time link travel times, depending on whether the vehicle is guided or not. For sophisticated ATIS/ATMS systems, for example, predicted travel times obtained from the *traffic management center* (TMS) are used and the shortest paths are updated periodically. The frequency at which shortest paths are updated depends on the capabilities and nature of the system to be evaluated.

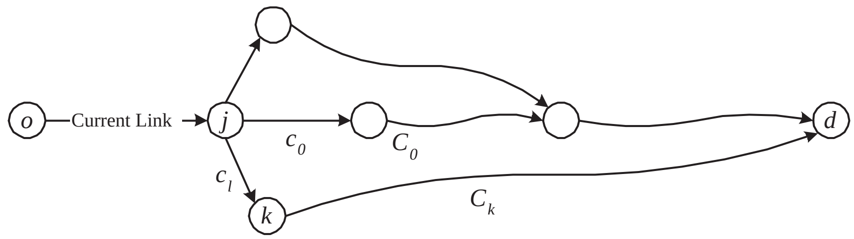


Figure 3-2: Route choice

MITSIM calculates shortest paths from each link to all destination nodes for each driver group. A dynamic shortest path algorithm, which is a modified version of the *label correcting algorithm* (Ahuja et al., 1993), is used (see Appendix B for details). The algorithm takes into account: (i) link travel times perceived by a particular driver group; (ii) delays and regulations of turning movements at intersections; and (iii) possible penalties for using certain links (e.g., freeway bias).

Due to the link-based implementation of the route choice model and the fact that a network may have loops, the choice set (L_j) should be carefully defined in order to prevent drivers from choosing paths with cycles and circulating in the network. The following tests may be used:

- A link l must satisfy $C_k(t + c_l(t)) \leq C_j(t)$ in order to be included in the choice set at an intersection j. In other words, a link must take the vehicle *closer* to its destination. This constraint is modified from Dial (1971).
- The predecessor information (see Appendix B) obtained from the calculation of link-to-node shortest travel times can be used for an explicit test of cyclic path. However, this test can be computationally expensive and unless it is absolutely necessary it should be avoided.

3.8.2 Route Switching Model

Instead of choosing their paths at each intersection, vehicles can also be assigned pre-specified paths to their destinations. However, they can switch to alternative routes (e.g., an incident is viewed or delay information is obtained from ATIS) at intermediate nodes. The simulator reads a list of predefined paths from a *path table* file. Each path is defined by an unique *ID* and a *list of links* it consists of. The *expected travel time* on the path is also included (see Appendix F.4 for an example). The *route switching model* defined below is used for choosing an initial route and en-routing vehicles:

$$p(s|r,t) = \frac{\exp(V_s(t))}{\sum_{i \in S_r} \exp(V_i(t))} \tag{3.8}$$

where:

$p(s|r,t) =$ probability to choose path s for a vehicle that expects to arrive at the decision node at time t using path r;

$S_r =$ set of available paths from node j to the driver's destination;

$V_i(t) =$ systematic utility of choosing path i which is a function of:

- $\hat{C}_i(t) =$ expected time on path i at time t; and,
- $Z_i =$ diversion penalty if path i is different from the driver's current path r (or habitual path).

Based on the probabilities calculated from Eq (3.8), a random number is drawn to determine whether the vehicle should stay on its current path or switch to an alternative path.

3.8.3 Properties and Extensions

In the simulator the route choice model and the route switching model can be used either individually or jointly. The route choice model requires no explicit storage of vehicle paths, thus allows the simulator to be used effectively with minimal memory requirements for large networks. The route switching model is designed to support the simulation of ITS applications with ATIS elements. Since a path table is pre-specified in this approach, calculation of time-dependent shortest paths can be avoided. The route switching model requires enumeration of all usable paths. To define the paths considered as choice alternatives by the users, two approaches may be followed

(Cascetta et al., 1996):

- In an "exhaustive" approach, all loop-less paths connecting the OD pairs belong to the choice set.
- In a "selective" approach, only a subset of feasible paths is included in the set of perceived choice alternatives.

The "exhaustive" approach can generate unrealistic routes and is computationally expensive, especially for large networks. In the "selective" approach this drawback can be avoided if the conditions and rules for selecting path subsets are well defined.

More sophisticated models that incorporate other considerations in making route choice decisions can also be used (Cascetta et al., 1992, 1996, see, for example). The modular design of the simulator allows for easy substitution of the default models.

3.9 Vehicle Movements

Interactions with vehicles ahead, response to traffic controls, desired speed, and lane-use preference determine the movements of vehicles. These interactions are manifested in lane change decisions and acceleration and deceleration rate applied at any given time. The simulator maintains a linked list of vehicles in each lane and moves individual vehicles according to the *car-following*, *lane changing*, and *event response* models described in this section. The car-following model computes the acceleration or deceleration rate of a vehicle in terms of its relationship with the leading vehicles; the lane changing model represents the behavior of switching lanes; and the event response model captures drivers' responses to traffic signals and signs, incidents, and toll booths.

3.9.1 Reaction Time

For computational reasons, the current version of MITSIM does not simulate *explicitly* driver's reaction time (except for vehicles at traffic signals). The acceleration rate applied for the next interval and decisions on lane changes are calculated based on the situation at the end of the previous simulation interval. Therefore, the reaction time is 0 at the beginning of a simulation interval and equals to the update simulation interval at the end. To incorporate the fact that reaction time varies among drivers and under different traffic conditions for the same driver, the update simulation interval can vary from driver to driver and from one condition to another for the same driver. When a vehicle n enters the network, it is randomly assigned a simulation interval ω_{nk} for situation k, where $k = \{d, a, c, s\}$ indicating **d**ecelerating, **a**ccelerating, **c**ruising at uniform speed, or **s**topped respectively. ω_{nk} is drawn from a truncated normal distribution with the following parameters:

$\mu_{\omega k}$ = mean value;

$\sigma_{\omega k}$ = standard deviation;

$\omega_k^{lower}, \omega_k^{upper}$ = lower and upper bounds.

The step sizes determined according to the above discussion are used in refreshing a vehicle's acceleration rate and lane change maneuver in the four cases under "normal" conditions. To make the simulated vehicles behavior more realistically, the following exceptions are made:

(a) When a vehicle is in the emergency regime at any time, an emergency deceleration rate (see Eq (3.10)) is applied immediately.

(b) As a vehicle changes lanes, the car-following and event response models are invoked in the next simulation interval (i.e. τ seconds later) for this vehicle and the vehicle that follows in the target lane.

(c) During the advance phase (*see* Figure 3-1), if the speed of a vehicle reaches the driver's desired speed or becomes zero, then the acceleration rate is immediately set to zero.

(d) If the leading vehicle is slowing down, the following vehicle updates its acceleration rate and lane change state at the minimum of the previously scheduled update time and current clock time plus half of the driver's update step size.

3.9.2 Vehicle Loading

Each vehicle enters the network from the upstream end of the first link on its path. Its initial position and speed are determined by the simulation step size, the driver's desired speed, and the traffic conditions on the loading segment. More specifically, if the necessary leading spaces are not available in the entrance links, vehicles are stored in a FIFO queue and wait to enter the network during subsequent time intervals; when vehicles can enter the network, they are pushed to positions with appropriate initial speeds that do not cause disturbance and artificial queues at the network "entrances".

3.9.3 Acceleration Rate

A vehicle accelerates (or decelerates) in order to:

- react to the vehicles ahead (car-following);
- perform a lane changing or merging maneuver;
- respond to events (e.g. red signals and incidents); and
- achieve its desired speed.

The most constraining of these situations determines the acceleration (or deceleration) rate to be implemented.

Car following

The car-following model calculates a vehicle's acceleration rate, taking into consideration its relationship with the leading vehicle. In several circumstances, this model is also used as a sub-model for calculating appropriate acceleration rates: (i) preparing to follow another vehicle if two or more lanes merge into a single downstream lane; and (ii) yielding to another vehicle shifting into the same lane. The car-following model used in MITSIM draws upon previous research (see, for example Herman et al., 1959; Herman and Rothery, 1963; Wicks, 1977). The model is based on the headway and relative speed of the leading and the following vehicles. Depending on the magnitude of this headway, a vehicle is classified into one of three regimes: *free flowing*, *car following*, and *emergency decelerating*.

Free flowing regime: If the time headway is larger than a pre-determined threshold h^{upper}, the vehicle does not interact with the leading vehicle. In this case, if the vehicle's current speed is lower than its target speed (*see* Eq 3.6), it accelerates at the *maximum acceleration rate* to achieve its target speed as quickly as possible; if the current speed is higher than the target speed, the vehicle decelerates with the *normal deceleration rate* to slow down:

$$a_n = \begin{cases} a_n^+ & \text{if } v_n < v_n^{target} \\ 0 & \text{if } v_n = v_n^{target} \\ a_n^- & \text{if } v_n > v_n^{target} \end{cases} \tag{3.9}$$

where:

a_n = acceleration rate;

a_n^+ = maximum acceleration rate;

a_n^- = normal deceleration rate;

v_n = current speed;

v_n^{target} = target speed in the current lane.

Emergency regime: If a vehicle has a headway smaller than a pre-determined threshold h^{lower}, it is in the emergency regime. In this case the vehicle uses an appropriate deceleration rate to avoid collision and extend its headway. Let x_n and x_{n-1} be the longitudinal positions of the two vehicles (position is measured as the distance from the downstream end of the segment, *see* Figure 3-3); L_{n-1} the length of the leading vehicle; $g_n = x_n - x_{n-1} - L_{n-1}$ the gap distance from the leading vehicle; and a_{n-1} the acceleration rate of the leading vehicle. The acceleration rate a_n for the following vehicle n is given by:

$$a_n = \begin{cases} \min\{a_n^-,\ a_{n-1} - 0.5\,(v_n - v_{n-1})^2\ /\ g_n\} & v_n > v_{n-1} \\ \min\{a_n^-,\ a_{n-1} + 0.25\ a_n^-\} & v_n \leq v_{n-1} \end{cases} \tag{3.10}$$

which guarantees that the following vehicle will always decelerate to extend the headway toward a safe range.

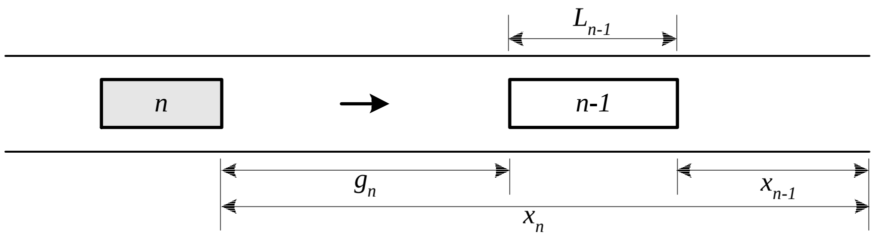


Figure 3-3: Car following

Car-following regime: Finally, if a vehicle has a headway between h^{lower} and h^{upper}, it is in the car-following regime. In this case the acceleration rate is calculated based on Herman's general car-following model (Herman et al., 1959):

$$a_n = \alpha^{\pm} \frac{v_n^{\beta^{\pm}}}{g_n^{\gamma^{\pm}}} (v_{n-1} - v_n) \tag{3.11}$$

where $\alpha^{\pm}$, $\beta^{\pm}$ and $\gamma^{\pm}$ are model parameters. α^{+}, β^{+} , γ^{+} are used for accelerating ($v_n \leq v_{n-1}$), and α^{-}, β^{-} , γ^{-} for decelerating ($v_n > v_{n-1}$) cases. Default values for these parameters are based on Subramanian (1996) and listed in Table C.8.

Merging

When two or more upstream lanes are connected to a single downstream lane, a *merging area* is defined (see Figure 3-4). In the merging area, additional constraints are considered in calculating acceleration rate because vehicles from adjacent upstream lanes may need to coordinate with each other. For the purposes of the simulator, merging is classified into: (i) priority merging; and (ii) non-priority merging. Priority merging includes merging from ramps to freeways, and from minor streets to major streets. Non-priority merging occurs at the downstream of toll plazas, and lane drops on freeways.

For priority merging, a vehicle *without* right-of-way must check whether there is any vehicle from the competing upstream lanes. The vehicle executes the merge only if the projected *headway gap* is acceptable. If the headway gap is not acceptable, the vehicle either calculates the car-following acceleration rate by treating the vehicle which has the right-of-way as leader or it prepares to stop at the end of the lane (depending on which case is the critical one).

For non-priority merging, a "first come, first serve" principle determines right-of-way. In other words, among all the vehicles coming from competing upstream lanes, the one closest to the downstream lane is chosen as the first vehicle to merge. Other vehicles will either follow appropriate leaders or prepare to stop at the end of their current lanes.

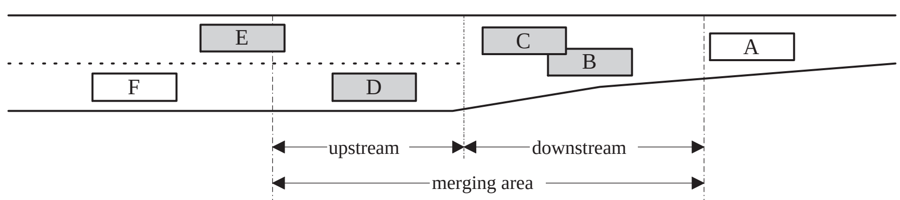


Figure 3-4: Merging area

A merging area consists of an upstream area (e.g. 100 feet) and a downstream (140 feet) area. The downstream area is also characterized by the maximum number of vehicles allowed in. An upstream merging area vehicle (i.e. vehicles D or E) is tagged as a *merging vehicle* if it has yet to be tagged and the number of downstream merging area vehicles have not reached the maximum. All vehicles in the downstream merging area (i.e. vehicles B and C) are tagged as merging vehicles. The acceleration rates for merging vehicles is calculated by relaxing the car-following constraint (negative gap from the immediate leading vehicle is allowed) and incorporating a merging constraint (no overtaking is allowed).

Vehicles are untagged when they leave the merging area, and the merging constraints are then relaxed.

Event Responding

The car-following and merging models described above capture behavior in response to the leading vehicle or vehicles from competing upstream lanes. In making acceleration decisions, another set of constraints is imposed by various downstream events within driver's view. Such events include: (i) traffic signal and signs; (ii) incidents; (iii) making a connection to the next link at the downstream node; and (iv) yielding to another vehicle switching into the same lane. These events may also influence drivers' lane change decisions, which are discussed in Section 3.9.4.

Traffic signals and signs: Each control device in the network is assigned a visibility parameter (based on roadway geometry, weather conditions, etc.). When calculating the acceleration rate and making decisions on lane changes, a vehicle checks all the downstream signals and signs (including those in the next link on its path) within its view. Signals and signs are considered as acceleration constraints if the distances within which vehicles have to stop, as mandated or suggested by the signals and signs, are less than or equal to the *normal stopping distance*:

$$y_n = \max\{-\frac{v_n^2}{2a_n^-}, y_{min}\} \tag{3.12}$$

where y_n is the distance required for a vehicle with speed v_n to decelerate to a stop by applying the normal deceleration rate a_n^-; y_{min} is a lower bound of the normal stopping distance.

If a viewed control device is within the above distance, the signal is *red* or *yellow* and the travel time to the signal is longer than the *expected remaining yellow time*, the vehicle prepares to stop using the deceleration rate given by:

$$a_n = -v_n^2/(2\, x_n) \tag{3.13}$$

where:

$a_n =$ deceleration rate to be applied in order to stop before the stop line;

$x_n =$ distance from the stop line;

$v_n =$ vehicle n's current speed.

If the state of the control device is *blank*, *green*, or *yellow* and the time headway between the vehicle and the device is shorter than the *expected remaining yellow time*, the constraint defined by Eq (3.13) is ignored. For a *flashing red* control (including stop signs), this constraint is considered as the vehicle is approaching the stop line and ignored after the vehicle stops at the stop line. For a *flashing yellow* control (including the yield sign), a vehicle may need to slow down using the following acceleration rate:

$$a_n = \frac{(v_n^{turn})^2 - v_n^2}{2\ x_n} \tag{3.14}$$

where v_n^{turn} is the target speed the driver wishes to use at the given location. v_n^{turn} is either set to the target speed in the lane (see Eq (3.6)) or determined based on the angle of the turning movement if the signal or sign is at an intersection.

The corresponding acceleration rate for all the viewed signals and signs within the normal stop distance (y_n) are calculated and the one with the minimum value is used as the *signal constrained acceleration rate.*

When the state of a traffic signal changes from red to green, a startup delay is added. This delay is a function of the vehicle's *position in the queue.*

The simulator allows the user to specify probabilities of driver compliance with each type of traffic control. The response to signals discussed above applies only to compliant drivers.

Incidents: An incident is either a complete blockage of a lane (assigned a speed of zero) and/or a rubber necking effect where vehicles slow down to a particular speed. When an incident is within view and the vehicle can not make a lane change to bypass it, the vehicle slows down and prepares to stop when the distance from the incident is less than the *normal stopping distance* (y_n) according to formula (3.12).

Connection to downstream link: This case occurs when a vehicle has to change lanes to follow the path leading to its destination (for example, to take an exit ramp, make a left turn, etc.). If the immediate situation prevents a vehicle from shifting into the desired lane, the vehicle will decelerate and prepare to stop before the downstream node to avoid missing its exit or moving into a wrong link.

Courtesy yielding: Courtesy yielding refers to the cases where a driver decelerates to make space for another vehicle switching into its lane. The courtesy yielding probabilities used in the simulator are a function of traffic conditions, the previous state of the vehicle and the state of the proceeding vehicle. When a driver has decided to yield, it decelerates using the vehicle attempting to change lanes as the leader. The yielding state is maintained until the lane change is completed or cancelled after a maximum amount of time has elapsed.

3.9.4 Lane Changing

The lane changing model in MITSIM is implemented in three steps: (a) checking if a change is necessary and defining the type of the change; (b) selecting the desired lane; and, (c) executing the desired lane change if gap distances are acceptable. This model is based on Gipps (1986).

Lane Change Decisions

The decision to look for a lane change depends on traffic conditions, driver's destination and behavior characteristics (e.g. from conservative to aggressive). MITSIM classifies lane changes into two types: *mandatory* and *discretionary*. Mandatory lane changing occurs when drivers have to change lanes in order to:

(a) connect to the next link on their path;
(b) bypass a lane blockage downstream;
(c) avoid entering and using a restricted lane; or,
(d) respond to LUS or VMS (e.g. warning of lane drop).

Discretionary lane changing refers to cases in which drivers change lane in order to increase speed, bypass a slower or heavy vehicle, avoid the lane connected to a ramp, etc. In Figure 3-5 the freeway diverges into two routes. White vehicles are going to route 1 and black vehicles to route 2. Drivers may begin to consider lane connections at different times, some earlier and some later. For example, vehicle C and 3 are in the wrong lane and looking for a mandatory lane change. Vehicles D and 4 are in the correct lane but are looking for a discretionary lane change into the lanes connected to their path. Vehicle J, since it is far from the downstream node and is not concerned about lane connection, is making a discretionary lane change to the opposite direction.

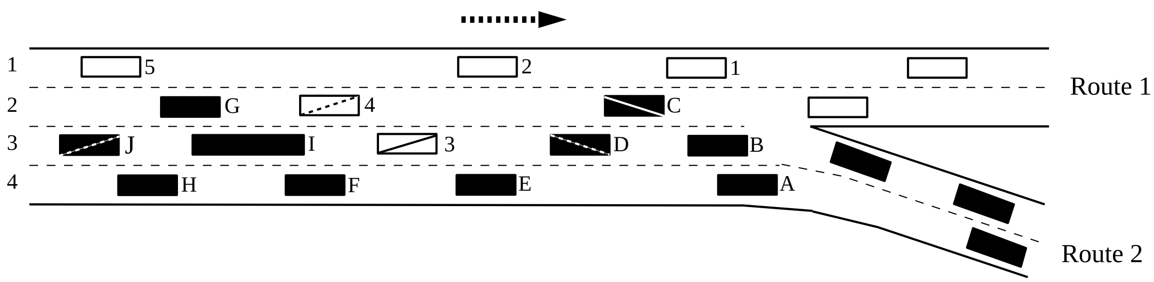


Figure 3-5: Lane changes

If a downstream node has multiple outgoing links, vehicles approaching that node are tagged as having a *mandatory state* according to a certain probability. Before a vehicle is tagged with mandatory state, both the left and right lanes can be the target lane of a discretionary lane change. After being tagged to mandatory state, the vehicle can seek two types of lane changes: (i) if the current lane is not connected to the next link on its path, it will make a mandatory lane change or a series of mandatory lane changes so as to move onto a lane leading to the next link; and (ii) if the current lane is connected to its next link on the path, then it can only make discretionary lane changes onto lanes that also connect to its path. The types of lane changes and their relationship with mandatory state are depicted in Figure 3-6. In the following, we describe how the probability used to tag vehicles to mandatory state is determined.

Assume that, at a particular position and congestion level, the probability a vehicle should be tagged to mandatory state can be written as:

$$f_n = \begin{cases} \exp(-(x_n - x_0)^2/\sigma_n^2) & x_n > x_0 \\ 1 & x_n \leq x_0 \end{cases} \tag{3.15}$$

where:

$f_n =$ probability of vehicle n has been tagged to mandatory state at point x_n;

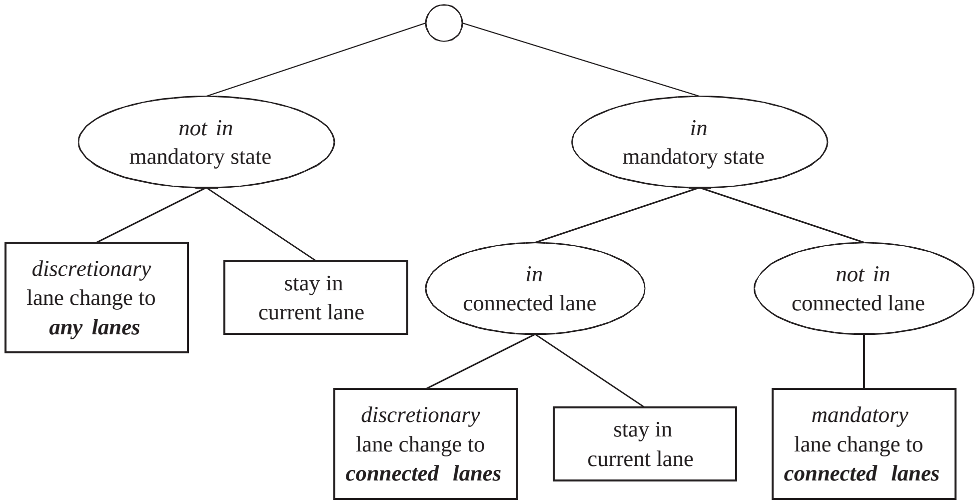


Figure 3-6: Mandatory state and lane change types

x_n = distance from the vehicle to the downstream node (or lane drop);

x_0 = distance of a critical location, which may be associated to the position of a particular message sign (such as final exit warning);

and σ_n is a variable defined as follows:

$$\sigma_n = \alpha_0 \, (1 + \alpha_1 \, m_n + \alpha_2 \, K) \tag{3.16}$$

where:

m_n = number of lanes that the vehicle needs to cross in order to be in the target lane;

K = indicator of traffic congestion, which is defined as the density of the segment divided by the jam density;

α_i = model parameters.

As depicted in Figure 3-7(a), the probability given by Eq (3.15) increases to 1 as the vehicle approaches the critical position. When the number of necessary lane changes and traffic congestion increases, this probability also goes up as shown by the different curves in Figure 3-7(a).

The simulator draws a random number in each iteration to decide whether an untagged vehicle should be tagged to mandatory state. Let p_{ni} be the probability

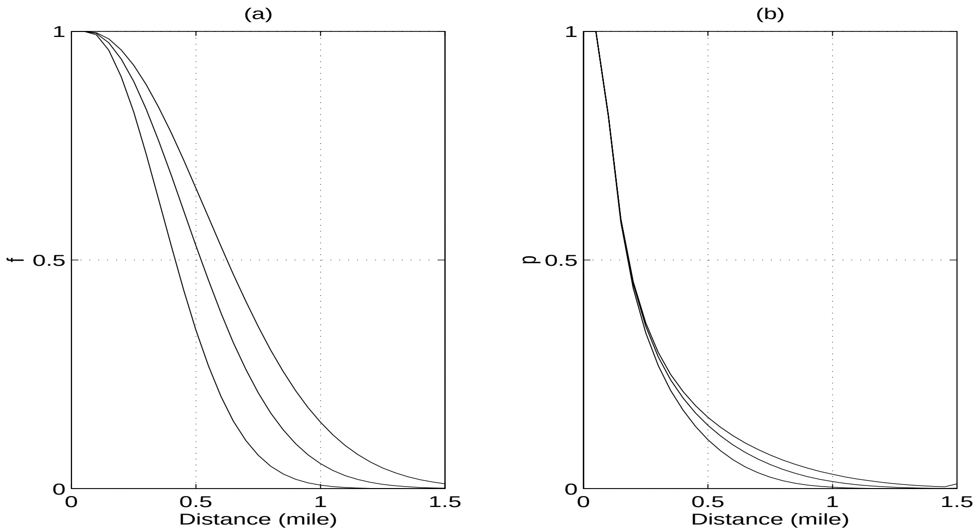


Figure 3-7: Probability of tagging vehicles to mandatory state

to tag a vehicle at time interval i. Then its relationship with f_{ni} – the probability that the vehicle should *have been tagged* upto ith time interval – can be represented recursively as the following:

$$
\begin{aligned}
f_{n1} &= p_{n1} \\
f_{n2} &= 1 - (1 - p_{n1})(1 - p_{n2}) \\
&\vdots \\
f_{ni} &= 1 - (1 - p_{n1})(1 - p_{n2}) \cdots (1 - p_{ni})
\end{aligned}
\tag{3.17}
$$

Since p_{n1}, p_{n2}, $\cdots$, $p_{n\ i-1}$ and f_{ni} are known at the ith time interval, p_{ni} can be written as:

$$
p_{ni} = 1 - \frac{1 - f_{ni}}{(1 - p_{n1})(1 - p_{n2}) \cdots (1 - p_{n\ i-1})}
\tag{3.18}
$$

An example of the results obtained from Eq (3.18) is illustrated in Figure 3-7(b). The curves in this figure will shift down if the simulation step size decreases, and vice versa. In other words, P_{ni} obtained from Eq (3.18) depends on the *sampling rate*. This is to guarantee that the probability a vehicle has been tagged at a given condition is independent from the simulation step size.

When a vehicle has been tagged to mandatory state, it keeps that state until it has performed the desired lane change or moved into the downstream link.

For discretionary lane changing, the decision to stay or change is based on traffic conditions of both the current lane and adjacent lanes. If a vehicle has a speed lower than the driver's desired speed due to a slow vehicle in front or the maximum speed of that lane, it checks the neighboring lanes for opportunities to increase its speed. Several parameters, including an *impatience factor* and a *speed indifference factor*, are used to determine whether the current speed is low enough and the speeds in adjacent lanes are sufficiently high to consider making a lane change.

In order to decide on the lane to move into, the vehicle first determines the set of admissible lanes. A lane is defined as *admissible* based on several criteria including lane changing regulation, lane use privilege, lane connectivity, incident, lane use signs, message signs, prevailing traffic conditions, driver's desired speed, lane's maximum speed V_i^{max}, and whether the vehicle is in mandatory state.

Gap acceptance

Once a vehicle has decided to change lanes, it examines the lead and lag gaps, g_{na} and g_{nb} respectively in Figure 3-8, in the target lane to determine whether the desired change can be executed. If both the lead and lag gaps are acceptable, the desired lane change is executed instantaneously.

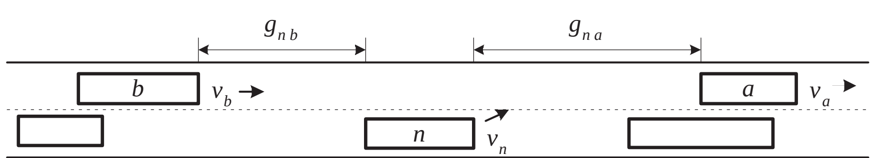


Figure 3-8: Lead and lag gaps for lane changing

The minimum acceptable gaps take into account the speed of the subject vehicle (v_n), speed of the lead and lag vehicles (v_a and v_b), and the type of the lane change (i.e., discretionary or mandatory). For discretionary lane changes, the critical gap is given by:

$$\begin{aligned} g_{na}^D &= \max\{\ g_a^D,\quad g_a^D + \beta_{a1}^D\, v_n + \beta_{a2}^D\, (v_n - v_a) + \epsilon_{na}\ \} \\ g_{bn}^D &= \max\{\ g_b^D,\quad g_b^D + \beta_{b1}^D\, v_b + \beta_{b2}^D\, (v_b - v_n) + \epsilon_{nb}\ \} \end{aligned} \tag{3.19}$$

where the subscript a indicates the lead vehicle, b the lag vehicle, and n the subject vehicle (see Figure 3-8). In addition:

g_{na}^{D} = critical lead gap;
g_{bn}^{D} = critical lag gap;
g_{a}^{D} = minimum lead gap;
g_{b}^{D} = minimum lag gap;
v_a = speed of the lead vehicle;
v_b = speed of the lag vehicle;
v_n = speed of the lane changer;
β = parameters (see Table C.10); and,
$\epsilon_{na}, \epsilon_{nb}$ = error terms.

For mandatory lane changes the critical gap is also a function of the distance of the vehicle from the downstream node (or incidents and lane drops). In other words, it is assumed that drivers tend to accept smaller gaps as they get closer to the last location where the lane change has to take place:

$$\begin{aligned} g_{na}^{M} &= \max\{\, g_{a}^{M},\;\; g_{a}^{M} + [\beta_{a1}^{M}\, v_n + \beta_{a2}^{M}\,(v_n - v_a)]\,[1 - \exp(-\gamma\, x_n^2)] + \epsilon_{na} \,\} \\ g_{bn}^{M} &= \max\{\, g_{b}^{M},\;\; g_{b}^{M} + [\beta_{b1}^{M}\, v_b + \beta_{b2}^{M}\,(v_b - v_n)]\,[1 - \exp(-\gamma\, x_n^2)] + \epsilon_{nb} \,\} \end{aligned} \tag{3.20}$$

where:

x_n = distance from the position where the lane change has to take place (ft);
β,γ = model parameter.

A more sophisticated lane change gap acceptance model has been proposed by Ahmed et al. (1996) and parameters are being estimated using field data. This model will eventually be used in MITSIM.

Nosing and Yielding

In heavily congested traffic, gaps for lane changing are difficult to find. In these cases drivers may nose into the target lane and force the lag vehicle to yield. For this reason the concept of nosing has been introduced. If a vehicle has been tagged as in mandatory lane change state and the current lead or lag gaps are not acceptable, a vehicle may nose in with some probability. In this case the vehicle is tagged as in *nosing state*. The nosing state is maintained until the desired lane change is made. A nosing vehicle then conducts a feasibility check in each iteration and tags the lag vehicle as in *yielding state* if the nosing is feasible. The nosing and yielding vehicles are pairwise coordinated and apply appropriate acceleration rates to create the necessary gaps.

Probability of nosing: The probability of nosing is a function of the vehicle's position, the number of necessary lane changes, and the time the vehicle has been waiting for the lane change. It is calculated by:

$$f_n = f_0 \, \frac{1 + \cos(\pi \; y_n^{z_n})}{2} \tag{3.21}$$

where:

f_n = probability of nosing for vehicle n;

f_0 = maximum probability of nosing, and it is a function of traffic conditions;

π = $3.14159 \cdots$;

y_n = relative position in the link or *view range*[4], i.e. $y_n = \frac{x_n}{L}$, where x_n is the vehicle's position and L the length of the link or view range.

z_n = a scaling variable defined as the following:

$$z_n = \beta_0 + \beta_1 \; N_n + \beta_2 \; K + \beta_3 \; T_n \tag{3.22}$$

where:

N_n = number of necessary lane changes in order to connect to the next link on its path (or to an open lane in case of incidents);

K = indicator of traffic condition (i.e. density divided by jam density);

T_n = time since the vehicle has been in *mandatory lane change* state; and

β_i's = model parameters (see Table C.11).

The probability given by Eq (3.21-3.22) (see Figure 3-9 (a)) has the following characteristics:

[4]View ranges is defined for lane drops, incidents, red lane use signs based on their visibility parameters. A view area begins from the point where a driver can see the signs or event and ends at the point where a lane change must be completed.

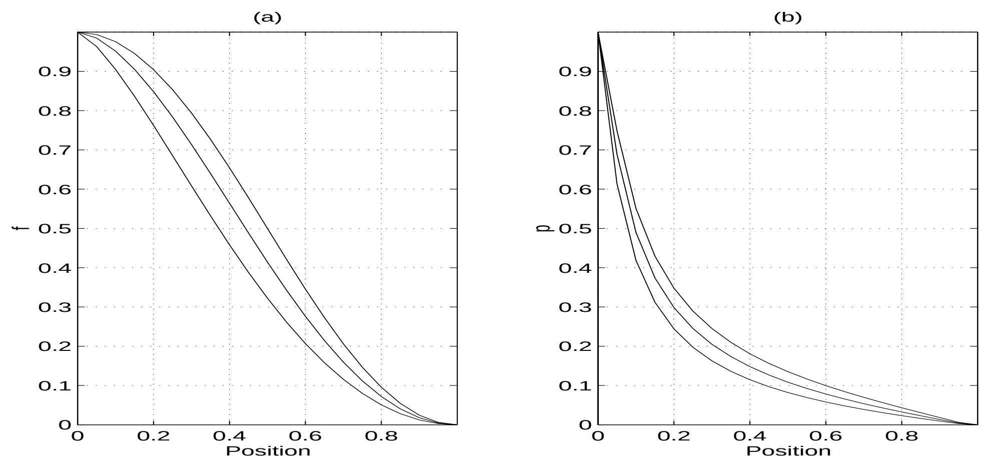


Figure 3-9: Probability of nosing

- increases from 0 to 1 as the driver moves from the beginning to the end of the link, and;
- increases with the number of lanes that the vehicle must cross and the time it has been looking for a gap.

This probability is then converted to the *probability to tag* a vehicle to *nosing state* (see Figure 3-9 (b)) using Eq (3.18).

Feasibility of nosing: The differences in speed between the subject vehicle and the vehicles in the target lane are examined to determine whether a nosing is feasible. A nosing is *feasible*[5] only if the subject vehicle is capable of reaching a speed, by either accelerating or decelerating within its capability (i.e. maximum acceleration or deceleration rates) at the current speed, faster than the lag vehicle and slower than the lead vehicle in the target lane. The requirement of this "speed match" can be represented by the following two constraints:

$$a_{n-1} + \frac{v_n - v_{n-1}}{T} \geq a^-_{n\,max} \tag{3.23}$$

$$a^-_{n+1} + \frac{v_{n+1} - v_n}{T} \leq a^+_n \tag{3.24}$$

[5]Feasibility of nosing does not guarantee that the desired lane change can be achieved. A lane change can be executed only when both the minimum lead and lag gaps are satisfied.

where:

a_{n-1},v_{n-1} = acceleration rate and speed of the lead vehicle in the target lane;

a^-_{n+1} = normal deceleration rate of the lag vehicle in the target lane;

a^+_n, $a^-_{n\ max}$ = subject vehicle's maximum acceleration and deceleration rates respectively;

v_n = current speed of the subject vehicle; and

T = time period, a parameter to be estimated.

The above conditions are examined in every iteration for the vehicles in mandatory lane change state. If nosing is feasible, the lag vehicle is tagged to *yielding state* and the subject vehicle is tagged to *nosing state*. A nosing vehicle applies the maximum acceleration rate for following the leading vehicle in the target lane (subject to a minimum lead gap constraint). A yielding vehicle applies the car-following acceleration rate with the target vehicle as the leader (subject to the normal deceleration constraint).

3.10 Simulation Output

MITSIM can provide a very detailed set of output, which can be used at various levels of aggregation for evaluating ITS applications. The output can be classified into three categories: sensor readings, MOE's, and animation graphics.

3.10.1 Sensor Readings

As one of the SIMLAB components, MITSIM provides TMS with real time surveillance data for simulated ATIS and ATMS operations. More specifically, at the end of each iteration or when a vehicle passes a particular detector, MITSIM checks whether any data should be sent to TMS and other supporting modules.

Sensor data such as traffic count, occupancy, and speed collected by point data sensors are reported at a given frequency (e.g. every 5 or 10 minutes) and logged into output files. Sensor ID and vehicle information such as vehicle ID, speed, etc. are

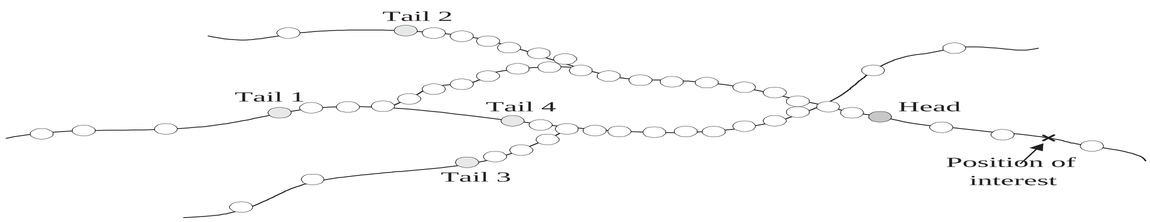


Figure 3-10: Queue

reported each time a probe vehicle passes a point to point data sensor.

3.10.2 Measures of Effectiveness (MOE)

Detailed data on vehicle trajectory and trip information can be recorded during the simulation. Input and output flows as well as average link and path travel time can also be reported. These data are useful in calculating MOE's necessary for performance evaluation of traffic management systems.

Snapshots of queue lengths at selected locations[6] can be reported at a given frequency. A queue is defined as a collection of vehicles whose speeds and gaps are less than some predefined thresholds. Every queue has one *head* and one or more *branches* that end with particular *tails* (see Figure 3-10). The queue head is the first vehicle upstream of the location of interest whose speed is less than the speed threshold (e.g. 0 or 5 mph). Queue tail is the last vehicle whose speed and gap distance from the leader is less than predefined thresholds.

3.10.3 Graphical User Interface

MITSIM uses graphics to visualize the input data as well as the simulation output, including the display of the road network, location and measurements of surveillance sensors, states of control devices, volume of travel demand (OD trip table), incidents, and vehicle movements.

[6]A location for reporting queue length is specified by *lane id* and longitudinal *position* in that lane.

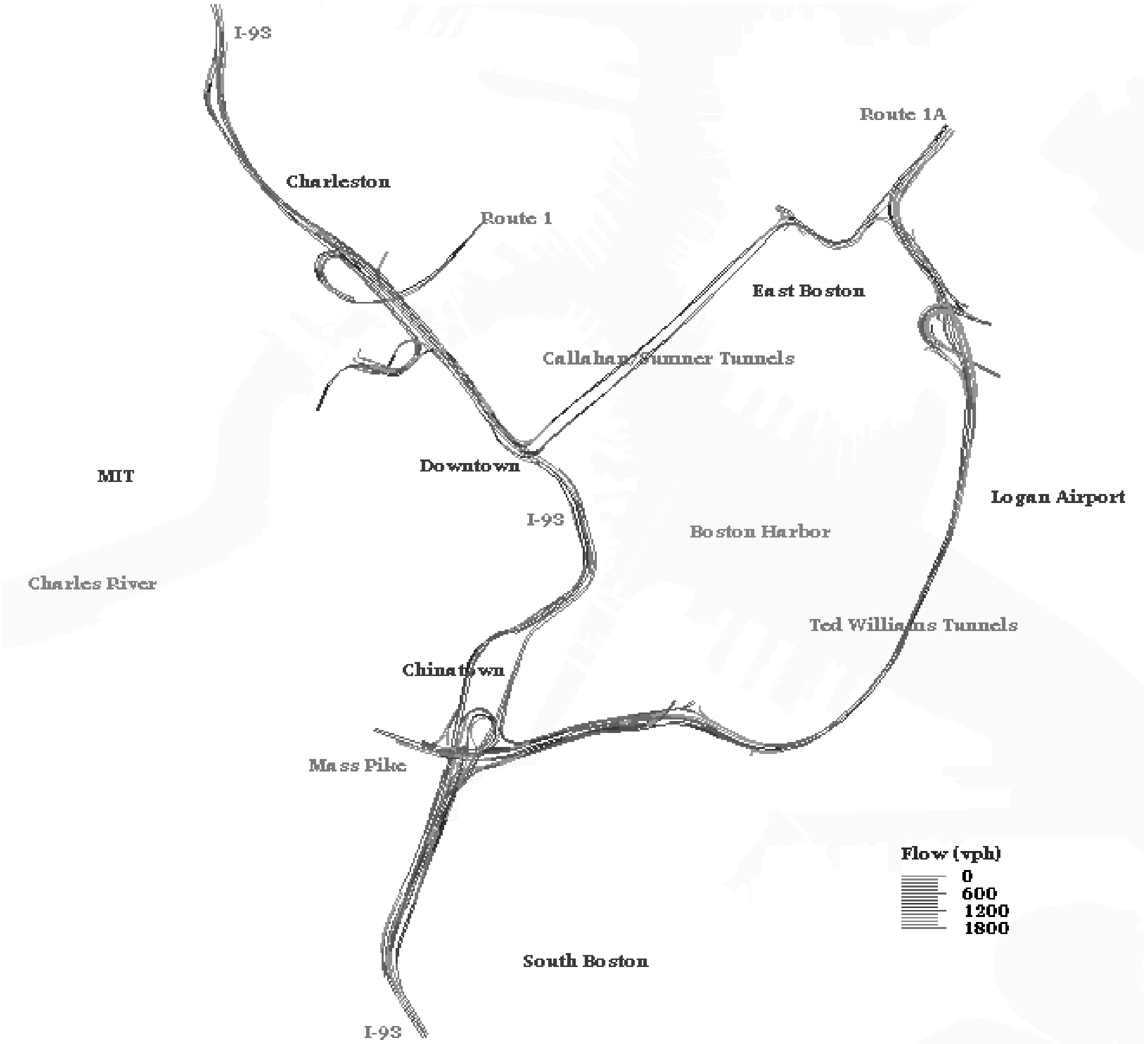


Figure 3-11: A macroscopic view of the CA/T network

Traffic network: The road network is shown color coded by direction, facility type (e.g. tunnel, freeway, ramp, or urban streets), and average density, speed or flow of segments (see Figure 3-11). Lane marks are shown in colors to indicate lane change regulations. Dedicated symbols drawn in lane indicate the lane-use privilege (e.g. HOV, ETC, etc.). Information that changes dynamically (e.g. speed, density, or flow) is updated at a fixed frequency (e.g., 60 seconds).

Traffic sensors: The measurement of surveillance sensors (counts, average occupancy, speed and flow rate) is displayed with colors and refreshed at fixed time intervals (see Figure 3-12).

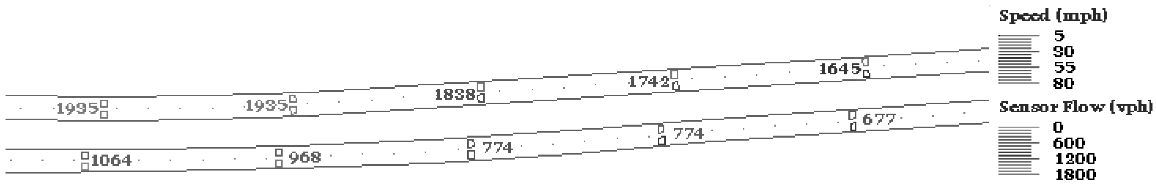


Figure 3-12: Flow rates measured by traffic sensors

Traffic controls and incidents: The traffic control devices are shown by dedicated symbols indicating their current state. Incidents are also shown with dedicated symbols and are color coded by their state (active or cleared).

Animation of vehicle movements: Vehicles are shown as colored rectangles with dimensions proportional to their size. The following information can be displayed: *vehicle type* (vehicle classes, or driver groups such as information availability, lane-use privilege, etc), *car-following regime*, *speed* and *acceleration* (accelerating, cruising, decelerating, or stopped), *lane changing* (left, right and whether in mandatory state or not), and *turning movement* (e.g. turn right, straight, left, etc.) at the downstream intersection. The user can also pause the simulation at any time to query information on individual vehicles. The animation of vehicle movements is updated at a fixed time interval. Figure 3-13 shows in detail the I-90 and I-93 interchange (at the lower left corner of the network shown in Figure 3-11).

3.11 Validation

MITSIM is tested and calibrated on a number of networks with varying structure and complexities. Aided by animation graphics of vehicle movements, unrealistic behavior caused by limitations in models and errors in parameter values as well as implementation mistakes were detected and corrected. In this section, we describe a validation study conducted using a data set acquired from 16 detector stations on a 5.9-mile stretch of I-880 around Hayward, California. The network contains 4 on-ramps and 6 off-ramps (see Figure 3-14). The left lane is a HOV lane. The traffic

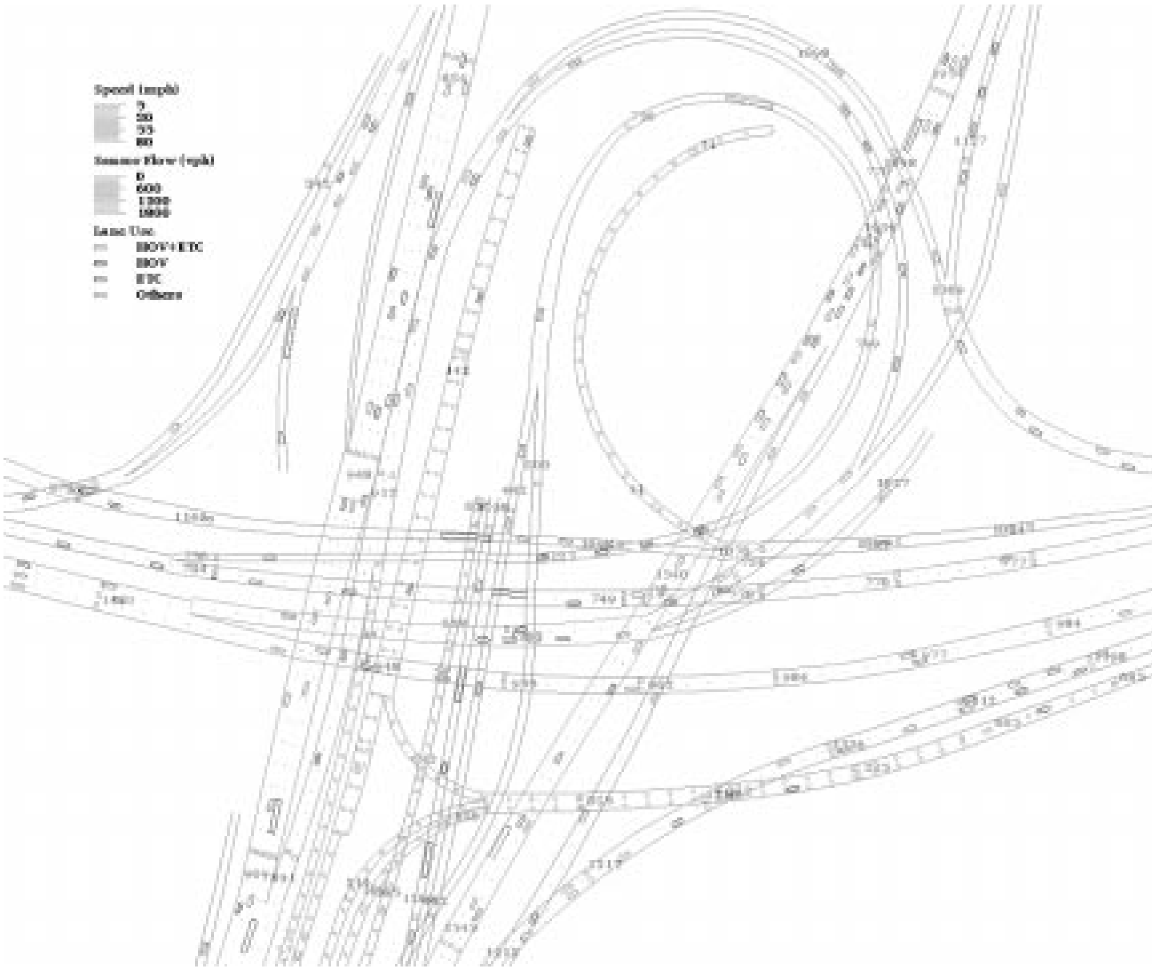


Figure 3-13: Animation of vehicle movements

counts, speeds, and occupancies aggregated over 5-minute time intervals are used in this study.

Using the observed traffic counts and speeds during a 4-hour time period for a number of days, time dependent O-D matrices were first estimated using the method of Ashok and Ben-Akiva (1993). The average departure rate was 9,430 vehicles per hour (vph). 20% of these vehicles were classified as HOV, 1.5% buses, and 3.5% trucks. Traffic counts, speeds, and occupancy for the period from 6:50 to 9:10am were collected at each detector station for 5-minute intervals and averaged over the 5 simulation runs. The first 10 minutes (i.e. two time intervals) are treated as "warm-up" periods and excluded from the data collection.

Figures 3-15(a)-(c) show scatter plots of the simulated and actual data. The

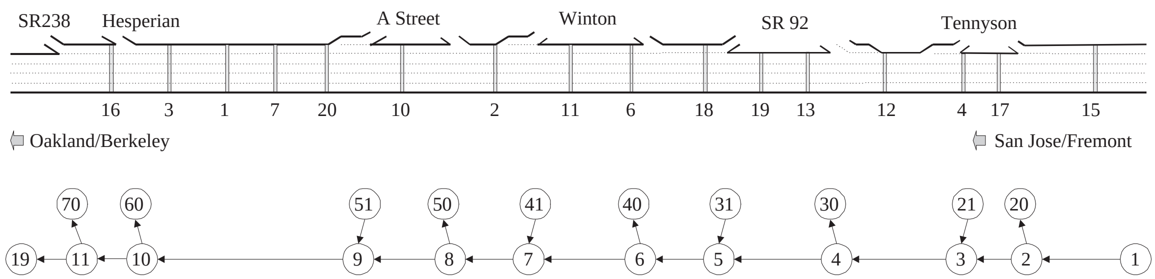


Figure 3-14: I-880 north freeway network

points in these scatter plots indicate that the simulated traffic counts fit the actual data reasonably well. Simulated speeds and occupancies exhibit poor fit in some cases. The contour plots in Figure 3-16 show the evolution of speed and occupancy over time and space. A closer look into these data reveals that most of the outliers in Figures 3-15 (b) and (c) occur at two particular sections and may be caused by the following:

- *Section between Tennyson ramps:* For detector stations 17 and 4 located in this area there is a significant over-estimation of the speeds and under-estimation of the occupancies between 8:00 and 8:40 am. The variance of the simulation output over the 5 runs is also high during this time period. The animation graphics show that the weaving traffic in this section is heavy and the queues upstream of the merging area occasionally back up to the detector stations. As a result the sensor output is inaccurate.
- *Section between A-Street On-ramp and SR238 Off-ramp*: Five detector stations (20, 7, 1, 3 and 16) are located in this area. In the first hour of the simulation, there is significant over-estimation of speeds and under-estimation of occupancies at these stations. The differences between simulated and field data are more persistent at the downstream end. This is probably due to congestion occurring at the downstream section outside the network boundary, which is not taken into account in the simulation. Indeed, there are 3 more ramps located within 1 mile downstream of detector station 16 and the freeway changes from 5-lanes to 3-lanes. Unfortunately, data in this area is not available for simulating the operations and incorporating their impact.

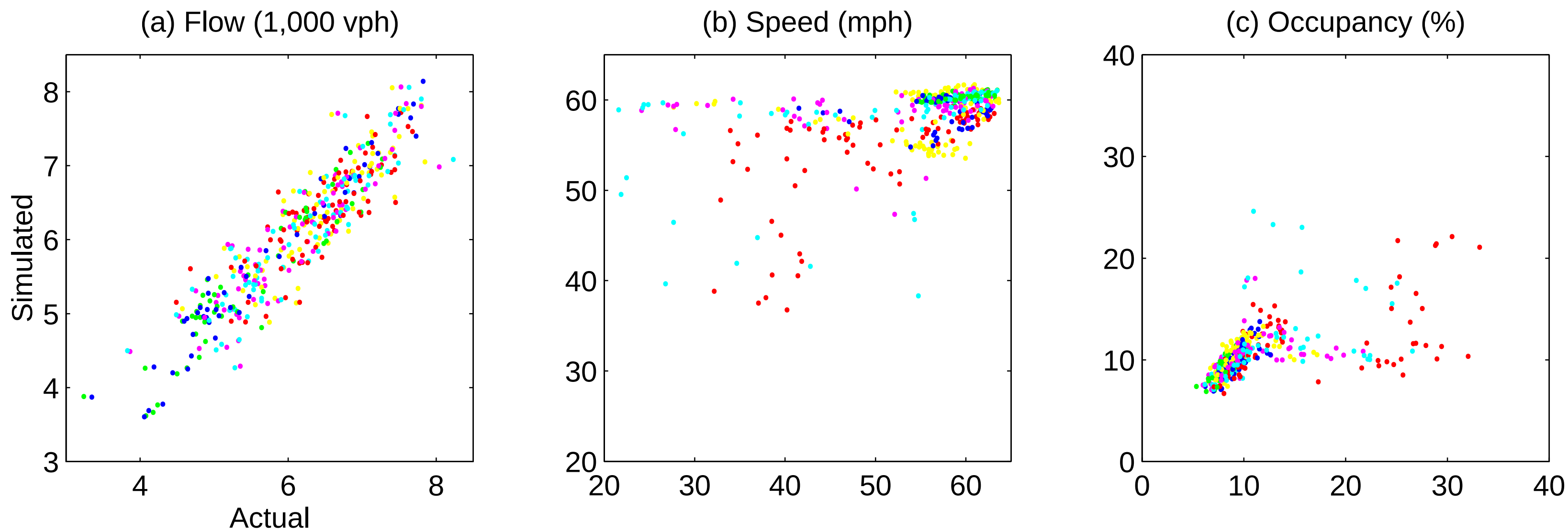


Figure 3-15: Comparison between simulation output and field data

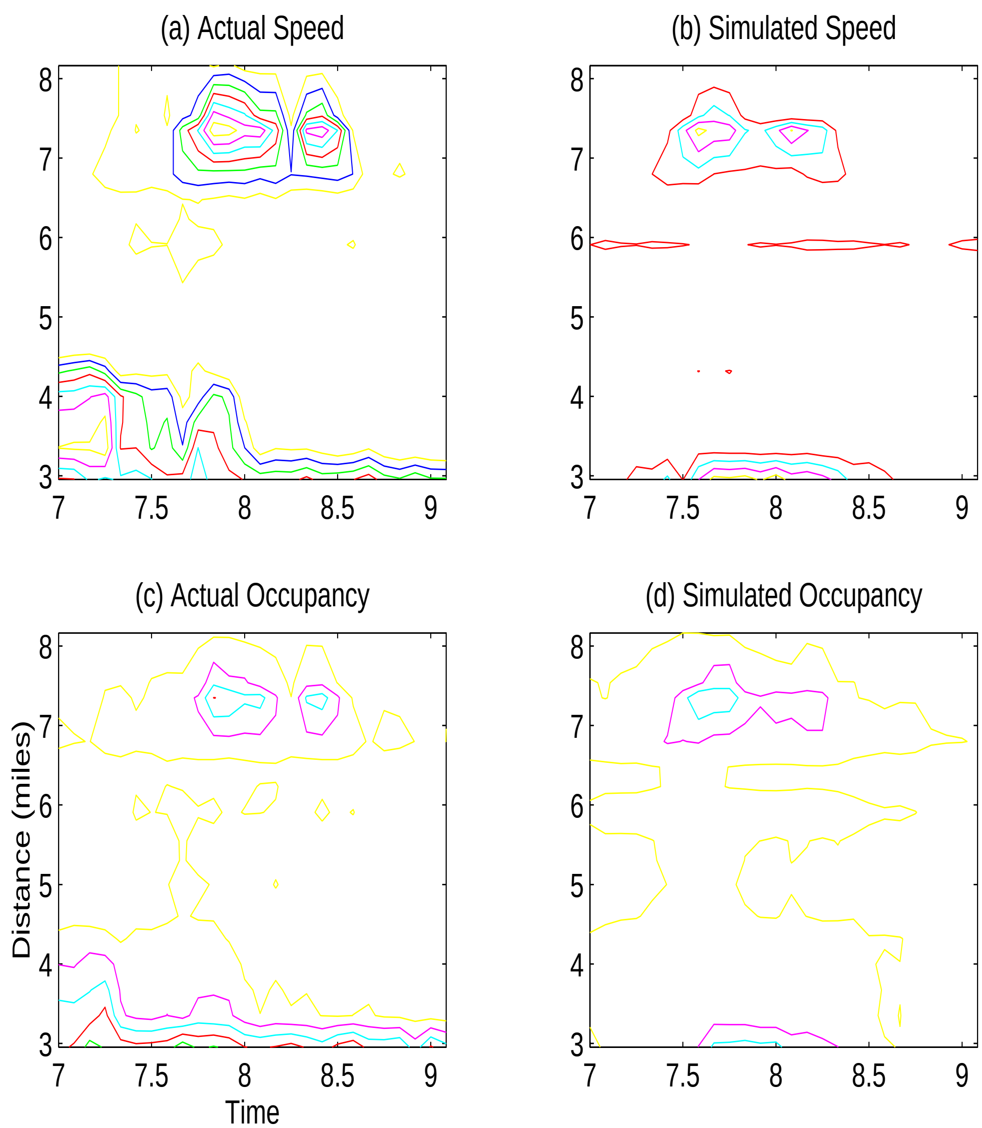


Figure 3-16: Contour plots of field and simulated speeds and occupancies

Further inspection of the data set also revealed that detector station 16 may be malfunctioning. Figure 3-17 shows the relationship between speeds and occupancy based on data from all sensor stations. The data points corresponding to station 16 (points marked in the circle) stand out from the rest of the points. The reason for this special case is yet to be investigated, but it explains most of the outliers of the occupancy fit shown in Figure 3-15 (c).

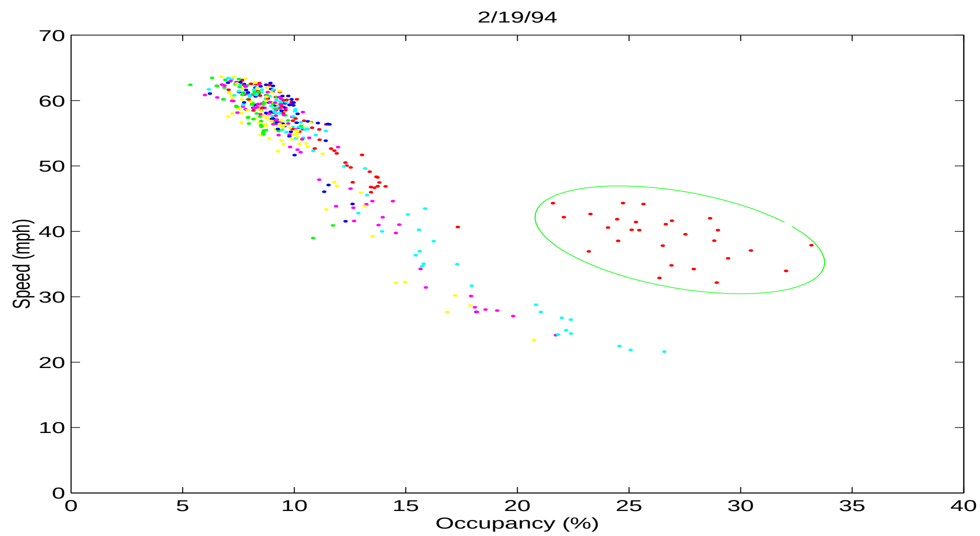


Figure 3-17: Relationship between speed and occupancy in field data

The overall performance of the simulation can also be evaluated using the statistics listed in Table 3.1. These statistics are based on Pindyck and Rubinfeld (1991) and documented in Appendix E. Detector station 16 is excluded in the calculation of these statistics.

The above results were obtained using the default values for various parameters in the simulation model. The only calibration attempt was made with respect to several parameters in the car-following model. The parameters $\alpha^{\pm}$, $\beta^{\pm}$ and $\gamma^{\pm}$ of the car-following model in Eq (3.11) are based on Subramanian (1996), who estimated these parameter values using disaggregated data on vehicle trajectory collected from a freeway section of I-10 near Washington D.C. (Smith, 1985). These values are $\alpha^{+} = 2.15$, $\beta^{+} = -1.67$, $\gamma^{+} = -0.89$, and $\alpha^{-} = 1.55$, $\beta^{-} = 1.08$, $\gamma^{-} = 1.65$ (distance is measured in meters, speed in m/sec, and acceleration in m/sec^2). The h^{lower} and h^{upper} are set to 0.5 and 1.36 seconds respectively. The step size for advancing vehicles

Table 3.1: Simulation errors in the I-880N network

Performance Measure		Flow[a]	Speed[b]	Occupancy[c]
RMS error		30.99	8.82	2.80
Mean error		-3.94	3.21	0.39
Mean percent error (%)		-0.62	9.83	7.81
RMS percent error (%)		6.45	29.36	21.13
Correlation Coefficient		0.92	0.34	0.51
Theil's inequality coefficient		0.0303	0.0770	0.1324
Proportions of inequality	U_M	0.0162	0.1329	0.0192
	U_S	0.0016	0.4017	0.1117
	U_C	0.9822	0.4655	0.8691
Number of data points		390		

[a] Flow is the number of vehicles passing a detector station in 5-minute intervals;

[b] Speed is the harmonic mean speed in mph;

[c] Occupancy is measured in percentage of time that a detector is activated by passing or stopped vehicles.

is set to 0.2 seconds.

The lane-changing model used in MITSIM is currently undergoing extensive calibration and validation using detailed data on driver behavior and traffic conditions on various facilities (Ahmed et al., 1996).

Chapter 4

Traffic Management Simulator

The traffic management simulator (TMS) mimics the traffic control and routing guidance logic under evaluation. It receives as input real-time traffic data from the surveillance system, generates control and routing strategies, and updates the state of traffic control and guidance devices. TMS currently supports evaluation of route guidance, signal control, and incident management functions.

4.1 Framework and Overall Structure of TMS

Advanced traffic management systems that dynamically monitor and control network traffic flows have been the focus of recent transportation research. A fundamental characteristic of dynamic traffic management systems is their ability to react to current or anticipated traffic conditions in real-time. A widely accepted approach on how to provide the above functionality does not exist. For this reason a structure that facilitates the representation of both reactive and proactive systems is employed to allow for experimentation. This structure can represent different designs of dynamic traffic management systems with varying levels of sophistication.

Figure 4-1 illustrates the main components of TMS and their interactions with MITSIM. The role of the network state estimation is to obtain the best estimate of the current network state utilizing the data obtained from the surveillance system. Once the state of the network has been estimated, the generation of the control

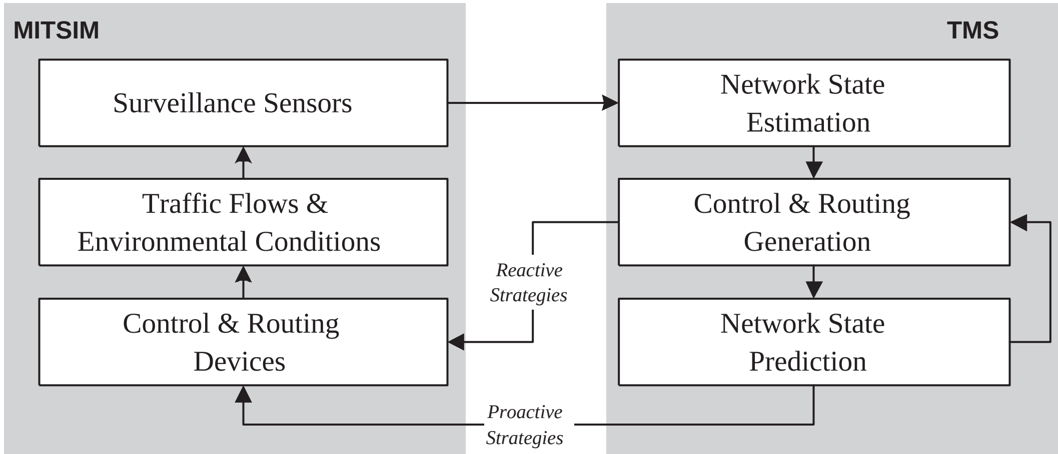


Figure 4-1: Generic structure of dynamic traffic management systems

and route guidance strategies can be done in two general approaches: *reactive* or *proactive*. The reactive approach consists of pre-determined rules or control laws that depend only on the current network state. Typical examples of such systems are the ALINEA ramp metering model by Papageorgiou et al. (1990) (see Section 4.3.4) and the CA/T incident response plans (De Leuw Cather, 1996). For the proactive case, the structure can represent a system that is able to: (i) utilize the real-time traffic information provided by the surveillance system; (ii) predict future traffic conditions; and, (iii) optimize traffic control and routing strategies.

The majority of the proactive systems proposed in the literature (see Chapter 1 for more details) are based on a sequential approach which uses an iterative process to generate control and route guidance. Examples of such systems are Gershwin et al. (1978); Gartner et al. (1980); Reiss and Gartner (1991); and Papageorgiou (1980). Most recently, the work by Mahmassani et al. (1994); FHWA (1995) and Ben-Akiva et al. (1996, 1997) attempts to develop proactive routing and/or control strategies using an iterative process. In this approach, a control and routing strategy which accounts for predicted future traffic conditions is generated. Given a proposed strategy, traffic conditions on the network are predicted and the performance of the candidate strategy is evaluated based on some convergence test. One of two actions are taken based on the evaluation: (i) if a satisfactory strategy has been identified, the strategy is implemented; or, (ii) if additional strategies need to be tested, another generation-prediction iteration is conducted.

The structure of TMS, as shown in Figure 4-1, is designed to support evaluation of the traffic management systems that are based on the above sequential process. The main functions that are supported by TMS include route guidance, traffic control, and incident management.

4.2 Route Guidance

A variety of approaches for route guidance, covering the entire spectrum from analytical to hybrid analytical-simulation models, have been proposed in the literature. Various assumptions are being made regarding the nature of information used for guidance generation, drivers' behavioral response to information and route guidance (including drivers' compliance to guidance).

With respect to the nature of information, some systems use predicted traffic conditions while others are based on current information. Most systems utilize network-wide information and a few utilize information on local traffic conditions. Assumptions on driver behavior vary widely. A common assumption, an extension of the static user equilibrium principle, is that users follow the shortest path to their destination. Some systems also assume that several classes of users exist and attempt to incorporate the effect of different behaviors (see MIT (1996) task B report for a review).

While researchers seem to be in overall agreement on the importance of basing guidance generation on predicted traffic conditions and incorporating realistic driver behavior, the performance, data and computational requirements of various approaches remain less studied. For example:

- How much improvement can be achieved if a system uses predicted rather than current information and what is the sensitivity to the quality of prediction?
- How different is the performance if a system generates guidance based on local rather than system-wide traffic conditions?

- How does the accuracy of input data (e.g., OD table and surveillance data) affect the quality of the guidance generated and the effectiveness of the ATIS?
- How robust is the system with respect to user behavior?

The answers to these questions depend on various factors, including the particular design of the route guidance system, network configuration, level of congestion, availability of alternative routes in the network, variability of travel demand, etc. The route guidance module developed in TMS attempts to provide a tool for answering some of above questions.

4.2.1 Reactive Route Guidance

Reactive route guidance refers to systems which generate guidance based on current traffic conditions. A system with reactive route guidance is simulated in TMS as follows:

(a) The average travel time for each link in the network is updated periodically (e.g. every 5 minutes) based on speed measured by the surveillance sensors or the actual speed probe vehicles experienced in the link during the last time period.

(b) For every eligible path, the travel time from each link to the destination of the path is recalculated based on the updated link travel times.

(c) The shortest paths from each link to every destination in the network are also computed based on the updated link travel times, using a modified version of the label correcting shortest path algorithm (see Appendix B).

(d) Drivers who have access to the updated information choose routes and make en-route decisions based on the updated path travel times or shortest path tables (see Section 3.8).

4.2.2 Predictive Route Guidance

A number of researchers have emphasized that guidance should be generated based on predicted traffic conditions which take into account drivers' response to the provided guidance. Predictive route guidance can minimize the inconsistency between provided information and drivers' experience and avoid problems such as over-reaction (Kaysi et al., 1993). In the literature there are 3 general approaches to guidance generation: (i) assignment/simulation; (ii) optimal feedback control; and, (iii) fuzzy-neural network. While pros and cons exist for all these approaches, the assignment/simulation based approach is receiving extensive attention because of its flexibility to incorporate behavior models and provide network-wide solutions.

In a predictive system, route guidance is provided by taking into account drivers' current position, destination and projected travel time on alternative paths. The time that a driver arrives at the decision node is very relevant in providing this guidance. In calculating the expected travel time that a driver would experience on the shortest path and alternative paths, projected time-variant link travel times are used (instead of current link travel times), i.e.:

$$C_i(t) = c_{i1}(t) + c_{i2}(t + c_{i1}(t)) + \dots \tag{4.1}$$

where:

$C_i(t) =$ travel time on path i given departure time t;

$c_{ij}(t) =$ travel time on link j of the path given that the driver enters the link at time t.

These travel times depend not only on traffic conditions at time t, but also on the past and future route choices made by all drivers.

An *idealized guidance system* based on the assignment/simulation approach for traffic prediction is implemented in TMS. The main characteristics of this guidance system are:

- simulation-based traffic prediction and guidance generation which consists of network state estimation, network state prediction, and guidance generation; and,
- periodic updates of guidance in a rolling horizon implementation.

Important model parameters, which can be varied in order to study the sensitivity of

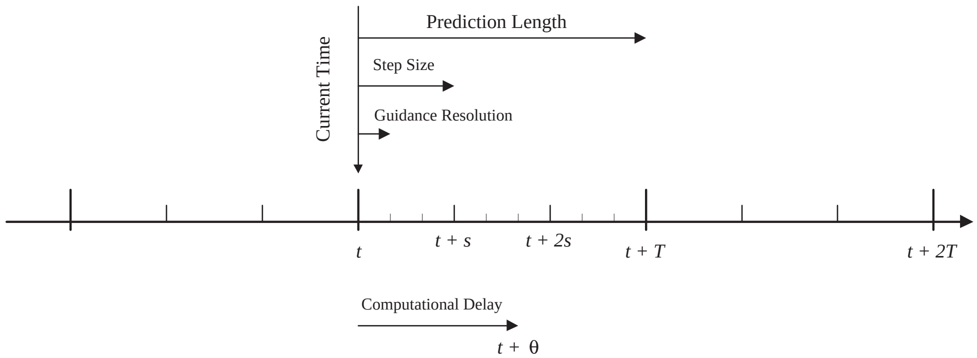


Figure 4-2: Rolling horizon implementation of guidance generation

results to the design characteristics of the ATIS, include:

- length of prediction horizon;
- frequency of guidance updates;
- time resolution of guidance between updates; and,
- computational delay.

Clearly, the characteristics of the system and its implementation allow for testing a wide range of ATIS.

Rolling Horizon Model

A rolling horizon implementation is used to provide predictive guidance. Under this model, the route guidance is periodically generated and evaluated for a given time horizon (e.g. 45 minutes) based on the latest information available. However, only the guidance until the next guidance update (e.g. 15 minutes) is actually implemented. In the implementation of an ATIS, the following parameters are of interest:

Rolling horizon length (T) is the time period for which prediction takes place. This length is a function of the maximum trip length. A rolling horizon is further divided into short time intervals (e.g., 5 minutes) which determine the resolution for representation of the time dependent travel times (see Guidance Resolution on page 91).

Rolling horizon step size (s) specifies how often guidance is renewed. This step size may be determined based on level of variation in traffic conditions over time (e.g. a longer step size can be used if traffic conditions do not change very much), timing of incidents, and available computational resources.

Guidance Resolution (Δt) is another important parameter in generating dynamic route guidance. Although guidance is renewed every s minutes, this does not mean that it has to remain constant during this step size. Within the rolling horizon step size s, different guidance may be provided every Δt minutes, where Δt ($< s$) is the guidance resolution. In other words, guidance is time dependent within each step size s. This resolution, Δt, defines the length of time intervals considered in calculation of link and path times (see Figures 4-2 and 4-3). Link and path travel times for vehicles that depart at a particular time are represented by piecewise linear functions. For example, in Figure 4-3 the expected travel time for vehicles entering the link during 7:00-7:10 is interpolated based on the predicted average travel times (see Section 5.7) in three time intervals from 6:50 to 7:20. The smaller the time intervals considered in calculation of link and path times, the more dynamic the generated guidance might be. However, smaller time intervals require more elaborate input data and computational power. Moreover, using smaller time intervals also introduces the risk of instability. The developed TMS module allows the user to experiment with different values for this parameter, and, therefore, facilitates the development of appropriate ATIS design based on the network structure, quality of the surveillance system, and available computational resources.

Computational delay (θ) may not be negligible for control and route guidance systems that require extensive computation. Suppose that the guidance generation process starts at time t and utilizes the information collected up to time t. The results may not become available until time $t + \theta$ (see Figure 4-2). Computational delay is explicitly modeled in SIMLAB and allows the assessing of trade-offs such as model complexity (for better prediction capability) and computational delay. If the delay is

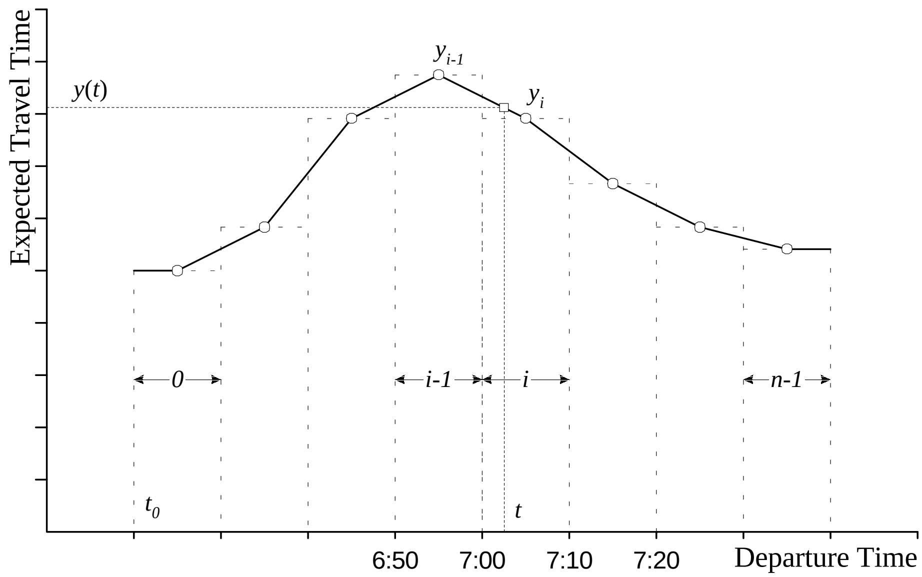


Figure 4-3: Piecewise linear time-variant travel time

significant, for example, the generated control and guidance may be obsolete when it is implemented. If the computational delay is longer than the rolling horizon step size ($\theta > s$), then the guidance generations need to be computed in parallel. Specifically, a guidance generation task is started every s minutes on the next available processor (one or more earlier tasks may be still running on other processors). This ensures a new guidance, which is based on information collected θ minutes ago, and becomes available every s minutes.

As stated earlier, the default approach to predictive guidance generation is based on two models: state estimation/prediction and guidance generation. It is an iterative process which attempts to generate a consistent guidance.

Network State Estimation

The network state estimation module – the start point for a traffic prediction and guidance generation – is used to obtain the best knowledge on the current network state. The network state to be estimated includes traffic flows, densities, speeds, travel times, and incident occurrences along with their locations and severities. These state variables can be estimated based on data obtained from the surveillance system module. For example, based on the measurements from the sensors, statistical methods may be used in estimating the traffic variables of interest. Alternatively, traffic simulation models may be used instead of or as a complement to the statistical methods. The advantage of the simulation-based approach is that it can provide network-wide estimates of the travel times on various paths. It may also have less dependency on the accuracy and completeness of the surveillance data. This is an important consideration because not all links and intersections are equipped with sensors and some sensors may not be operational. The mesoscopic traffic simulation model presented in Chapter 5 can be used in the laboratory for this purpose.

In this research, the default network state estimation module is treated as a dummy element that provides "perfect" estimation. In other words, we assume that the true network state – the state observed in MITSIM – is directly available to TMS. An estimation error can be easily added to the true network state to test a control and route guidance system's sensitivity to the quality of the network state estimation.

Network State Prediction

The network state prediction module forecasts future traffic conditions based on the current network state, the proposed control and routing strategies, and predicted OD flows.

Dynamic OD prediction: Time-dependent OD flows are a key input to the dynamic traffic model. Most assignment/simulation models in the literature assume that historical OD data can be readily used, while a few others emphasize that actual OD flows may differ from their historical counterparts and include methods to estimate dynamic OD flows online. Several methods exist for the dynamic OD estimation problem. Examples of models developed recently include the work by Ashok and Ben-Akiva (1993) and Chang and Wu (1994). These models explicitly recognize the dynamic nature of the problem and the need to operate in real-time.

For the purpose of investigating a particular control and route guidance system's robustness with respect to prediction of OD flows, a pseudo-OD prediction module is used to provide the time-dependent OD matrices at a controlled level of accuracy.

This module takes the input OD matrices specified in a scenario and randomly permutes these "true OD matrices" with noise that represents prediction errors. This approach is useful in evaluating a particular control and route guidance system's sensitivity to the accuracy of the OD prediction.

Traffic prediction: Existing methods for traffic prediction include four approaches: (i) statistical; (ii) neural networks; (iii) analytical dynamic traffic assignment; and, (iv) simulation-based traffic assignment. The developed laboratory is capable of evaluating any of these methods. Simulation-based assignment (Ben-Akiva et al., 1994; Mahmassani et al., 1994) is the only approach that can explicitly capture the impact of control and route guidance on traffic flows and represent a variety of driver behavior. A specially designed mesoscopic traffic simulator, MesoTS, is implemented in TMS and used as the default traffic prediction model. This traffic simulator assumes that the initial network state, time-variant OD matrices, control and routing guidance are given. It estimates future traffic conditions by simulating movements of individual vehicles using macroscopic speed-density relationships, drivers' route choice, and response to information models. The outcome of this simulation-based congestion prediction includes traffic flows, densities, speeds, queues, and link travel times for each time period within the rolling horizon. This information serves as the basis for evaluating a proposed control and route guidance and generating better ones. Details on this simulator are presented in Chapter 5.

Guidance generation

The guidance generation module is used iteratively with the traffic prediction module to identify potential improvements to route guidance provided in previous iterations. This module takes as input the guidance provided in the last iteration and the corresponding traffic conditions predicted by the mesoscopic traffic simulator described in Chapter 5, and generates the guidance to be used in the next iteration. The main objective is the generation of "consistent" guidance, i.e., the guidance with the smallest difference between the information drivers received and the actual condition they would experience.

The route guidance provided by the ATIS may take various forms, ranging from descriptive (i.e. traffic information such as travel times, delays, and queues, etc.) to prescriptive (e.g. route recommendation on VMS or in-vehicle units, etc.). The ATIS implemented in TMS is a generic one. It assumes that two groups of drivers exist: guided and unguided. The unguided drivers choose their routes based on historical knowledge of link travel times. Guided drivers choose their route based on information provided by the route guidance system. Travel time information provided by the route guidance system is accessible for and used by all guided drivers, regardless of their current position and what they have observed in the network.

Two sets of routing information – historical and predicted travel times – are maintained in the simulation. Drivers in each group use the corresponding table to choose their routes probabilistically according to the route choice models described in Section 3.8.

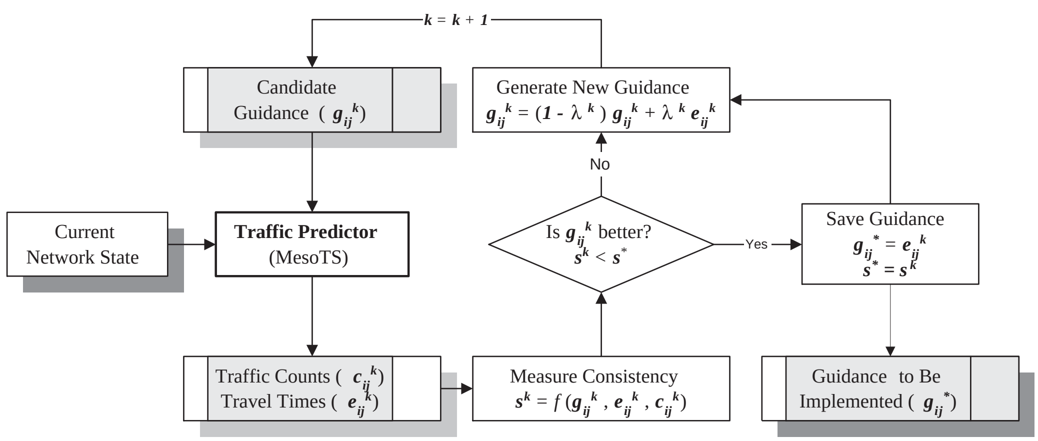


Figure 4-4: Generation of predictive route guidance

Travel times are predicted iteratively using the mesoscopic traffic simulator for a given guidance (e.g. predicted travel time table, see Figure 4-4). At the start of the simulation, estimates of link travel-times are provided to the drivers, who, in turn, make their route choices. Their choices influence the link flows and travel-times, which are recorded. These travel times are referenced as "experienced travel times" and used to determine the guidance to be provided in the next iteration. This process continues until the travel times assumed for guidance and the "experienced" travel times converge, or a pre-determined number of iterations is reached. In the latter case, the iteration with the most consistent travel time is chosen.

In the current implementation of TMS, a heuristic algorithm is used in updating the travel times in each iteration, i.e.:

$$g_{ij}^{k+1} = (1 - \lambda^k)\ g_{ij}^k + \lambda^k\ e_{ij}^k \tag{4.2}$$

where superscript k is the iteration counter; subscript i represents paths (or links) and j time interval, respectively; and:

g_{ij}^k = travel times used for the guidance information;

e_{ij}^k = "experienced" travel times; and,

λ^k = update step size;

The step size λ^k can be set to either $\frac{1}{k+1}$ or a constant parameter ($0 < \lambda^k \leq 1$). In either case the iteration loop is terminated when "experienced" travel times converge to the travel times used for guidance, or a pre-defined number of iterations has been reached. A criterion used to measure the consistency between "experienced" travel times and travel times used for guidance at a particular iteration is defined as:

$$s^k = \sum_i \sum_j f_{ij}^k\ \mid g_{ij}^k - e_{ij}^k \mid \tag{4.3}$$

where:

s^k = consistency measurement of the guidance g_{ij}^k; and,

f_{ij}^k = number of drivers who experience the travel times e_{ij}^k.

This algorithm does not guarantee that a consistent guidance, which may not even exist, can be found. In such cases, the travel time information g_{ij}^k which yields the minimum s^k, among all iterations, is defined as the most consistent guidance and implemented for the next prediction horizon.

In general, this method tends to generate guidance which becomes increasingly consistent as more iterations are performed. However, oscillation in choosing alternative routes between iterations is possible if, for example, drivers are sensitive to the shortest path information and a large percentage of drivers respond to the guidance. Further research is needed to devise good algorithms that generate consistent guidance. TMS provides the tool that facilitates such research efforts.

4.3 Traffic Control

The TMS module in SIMLAB can simulate a wide range of traffic control and advisory devices. Examples include:

- *Intersection controls:* traffic signals, yield and stop signs (TS).
- *Ramp controls:* ramp metering and speed limit signs.
- *Mainline controls:* lane use signs (LUS), variable speed limit signs (VSLS), variable message signs (VMS), portal signals at tunnel entrances (PS).

These signals and signs are controlled by four types of traffic signal controllers, namely *static*, *pre-timed*, *traffic adaptive*, and *metering* controllers. As the simulation proceeds a controller can switch from one type to another based on the implemented logic. For example, a controller may switch to traffic adaptive control in the off-peak period and to pre-timed control in the peak period.

Each controller is characterized by data items such as *controller type*, *signal type* (e.g., traffic signal at intersections (TS), variable speed limit signs (VSLS), lane use signs (LUS), etc.), number of *egresses* (i.e. the out-degree of the intersection node), *IDs of signals* it controls, timing table, etc. The representation of the control devices as network elements is described in Section 3.4.2. In the following sections, traffic signals at an intersection are used as an example to describe how traffic signal operations are represented in TMS. The same methodology is applied to other control elements (e.g., LUS, VSLS, VMS, etc.).

To represent traffic signals at intersections, dedicated indicators for every possible turning movement for each link are used. For example, in Figure 4-5, each inbound link at the intersection (links 10, 20, 30 and 40) has three indicators, which control the

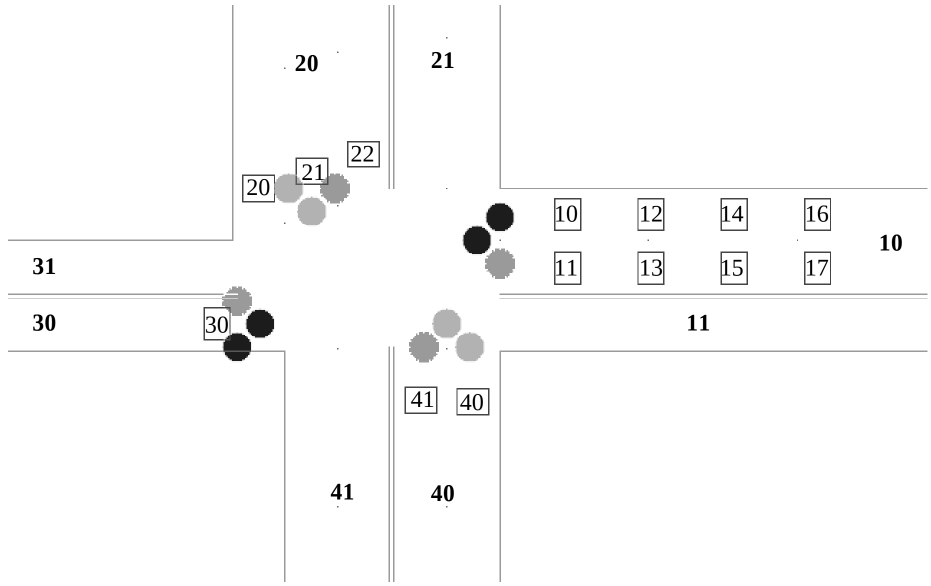


Figure 4-5: Traffic signals at an intersection

right, straight, and left turning movements respectively. These indicators facilitate the implementation and improve the communication efficiency between TMS and MITSIM.

4.3.1 Static Controllers

Static controllers represent the traffic signal and signs whose state does not change during the simulation. Examples include yield and stop signs, fixed speed limit signs and message signs. Each static controller contains a list of traffic signals or signs under its control and uses a vector of signal states to establish the right-of-way for each turning movement. This state vector does not change during the course of simulation.

4.3.2 Pretimed Controllers

Pretimed controllers simulate the operation of signals and signs whose states change according to a pre-determined sequence. This type of controller is typically used to represent pretimed intersection signal controls. The control logic for a pretimed controller is specified by an *offset* and a *timing table*, which consists of a set of *phases* and *control intervals* (see Table 4.1).

Offset: Offset is used to coordinate operations of the traffic signals between adjacent intersections in the network. It is defined as the time difference between the beginning of the first phase and a given master clock time (i.e. start time of the simulation). In order for pretimed signal controls to be coordinated, it is necessary to have the signals operate on a common cycle time (or in rare cases on the modulus of a common cycle time).

Timing Table: A set of phases represents the timing table for a pretimed controller. Each phase is further divided into *control intervals*. For traffic signals at intersections, three intervals are typically used: green, yellow and red clearance. Each control interval represents a period of time during which states of all signals remain constant. The data items describing a control interval are *length* of the interval and a vector of *signal states*. The length specifies the time duration of the interval and the signal states specify the right-of-way for various turning movements. The number of bits required to represent each signal state variable is equal to 4 × *Number of egress links*[1]. Each 4 bits of this variable represents the right-of-way for a particular movement, and may have the following values:

0 = *Blank:* The signal is not used (no regulation for the movement);

1 = *Red:* Movement is not allowed or vehicles should stop before moving again (i.e. flashing red, see below);

2 = *Yellow:* Vehicles should prepare to stop if time headway from the signal is longer than a threshold;

3 = *Green:* The movement is allowed.

One of the following two tags can be added to a signal "color value" to indicate whether the movement is protected or has a lower priority.

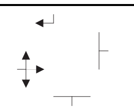
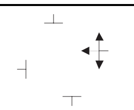
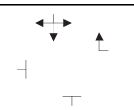
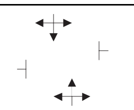
4 = *Arrow:* The movement is protected;

[1]The signal states for up to 8 movements from a given approaching link is fully represented by a single 32-bit integer number.

8 = *Flashing:* The movement has a lower priority.

The implementation allows the right-of-way for each turning movement to be represented by a single hexadecimal digit and the sophisticated signal sequencing to be coded in a very compact and organized fashion. For example, a flashing red signal (the equivalent of a stop sign) is assigned a state value of 9; a green arrow for protected turning movement is assigned a state value of 7. The signals at the intersection depicted in Figure 4-5 may have following sequence table:

Table 4.1: An example of pretimed signal plans
(see Figure 4-5)

Phase	Control Interval		Signal State			
	Diagram	Length (sec)	40 ↑	20 ↓	30 →	10 ←
1		9	111	113	333	111
		3	111	112	222	111
		1	111	111	111	111
2		15	111	111	111	333
		3	111	111	111	222
		1	111	111	111	111
3		17	111	333	111	113
		3	111	733	111	112
		1	111	333	111	111
4		18	333	333	111	111
		3	222	222	111	111
		1	111	111	111	111

Blank out sign[2] is represented by a special state value (i.e. F). States for other types of signals and signs are represented either by cardinal values (e.g. VSLS) or dedicated integer codes (e.g. message signs).

4.3.3 Traffic Adaptive Controllers

Traffic adaptive controllers use real-time data from surveillance detectors and pre-specified control laws. Examples of traffic adaptive controllers are SCOOT (Hunt et al., 1981; Bretherton, 1996) and OPAC (Gartner, 1983). Depending on the

[2]Blank out sign: No movement is allowed at a blank out sign and this state is not expected to change shortly

particular system to be evaluated, the control logic may require coding as a customized module and interfacing with the TMS, or it may be implemented as a new controller type in TMS. The modular design and object oriented implementation facilitate the addition of new types of controllers into the system with minimal effort.

The functionality of the system will be demonstrated through the implementation of the default traffic adaptive controller in TMS. The default controller is described by three data blocks of: (i) signal records; (ii) phase records; and, (iii) detector records.

Signal records: A signal record provides information on *signal ID*, *maximum red time*, and the ID of the phase to be called in the event that continuous red time for the signal has reached maximum value. As in pretimed controllers, a composite signal is used to control all movements from a particular approaching link.

Phase records: As with pretimed controllers, each traffic adaptive controller consists of a set of phases, and each phase is further divided into control intervals. The data items for the control intervals are the same as for pretimed controllers (i.e. each control interval has a length and a state vector for the signals defined in signal records). The difference is that, for the traffic adaptive controllers, a phase can be either *extendible*, *called*, or both. A phase is *extendible* if its green interval can be extended when detector data satisfies certain criteria and the continuous red time for any conflicting movement has not reached its maximum value. A phase is *called* if, after completion of an earlier phase, signal operations can be switched (shortcut) to this phase without completing the subsequent phases in the cycle. Based on the real-time traffic information obtained from the detectors and the pre-specified control logic, the green interval of a phase may be extended or terminated. The actual length of the green interval of an extendible phase and the sequence of phases may change as a result of the extension and calling logic. The logic for extending and calling a phase is specified by the detector records and the following variables:

- *Extension time:* In a traffic adaptive signal controller, the length of the green interval is the minimum green time for a phase. An extension time (e.g. 2

seconds) can be specified to extend the current phase incrementally. If the extension time is non-zero, detector measurements and continuous red time for each movement are examined at the green interval's end to decide whether the current phase should be extended. If so, the green interval continues for a number of seconds as specified by the *extension time*.

- *Maximum green time:* When the length of the green interval reaches the *maximum green time*, it switches to the next interval without further extension, regardless of traffic conditions measured by the available detectors.
- *Reset flags:* They determine whether the time of the green interval of a phase should be reset to its initial value when the subject phase is completed.

When the last interval (i.e. red clearance) of a phase is completed, the conditions for calling an alternative phase are examined. If the conditions warrant switching to a called phase, the signal states are switched to the first interval of that phase; otherwise the signal operation moves to the next phase in the cycle.

Detector records: Detector records specify the logic for extending the current phase and calling a new phase. A controller may contain any number of detector records, each corresponding to a single detector. These records contain the following information: *detector ID*, a vector of *specification flags for extendible phases*, and a vector of *specification flags for called phases.* Each specification flag indicates under what conditions the corresponding phase should be or should not be extended or called (see Table 4.2). In the current implementation, a specification flag consists of three pieces of information: *state indicator*, *selection indicator*, and *group indicator.* The **State indicator** describes the required state of the detector for the phase under consideration and may have one of the following values:

0000 = irrelevant to the phase;

1000 = idle state is required; and

2000 = activated state is required.

The selection and group indicators determine whether the detectors are considered inclusively or exclusively. The **selection indicators** may have one of the following values:

0**1**00 = preemptive (if the detector has the required state, no other records in any group will be examined further);

0**2**00 = any one in the group (the condition is satisfied if any detector in the group has the required state);

0**3**00 = every one in the group (the condition is satisfied only if all the detectors in the group have the required state).

A **group indicator**, represented by the last two digits of hexadecimal specification flag, may have a value between 1 and 255. Records in the same group are treated as a family and the conditions for whether to call (or extend) the phase or not are examined together.

Using the notation described above, a variety of adaptive signal control logic can be represented. For example, assume that, for the intersection shown in Figure 4-5, the control logic is represented by the detector records given in Table 4.2 in addition to the signal table specified in Table 4.1 (however, the length of control interval is a variable is this case). The implemented logic states that:

- Phase 2 is called (if currently it is in a different phase) or extended (if it is already in phase 2) if there is traffic from link 10 (one of the detectors 10-17 is activated) and there is no traffic from other links (detectors 20-41 are idle).
- Phase 3 is extended if there is traffic in the right lane (detector 10, 12, 14, or 16 is activated) and no traffic elsewhere (other detectors are all idle).

A logic for conflict resolution is also included. This logic assumes that the detector records are organized in descending order according to their priorities. Hence, the rules with the higher priority are listed on the top.

4.3.4 Metering Controllers

Ramp and mainline metering can be represented by either pretimed or traffic adaptive controllers if the timing table is predetermined or a direct mapping from detector state to signal state is available. Otherwise, a metering controller can be used to simulate

Table 4.2: Examples of detector records
(see Figure 4-5)

Detector	Called Phase	Extendible Phase	
	2	2	3
10	2201	2201	2201
12	2201	2201	2201
14	2201	2201	2201
16	2201	2201	2201
11	2201	2201	1301
13	2201	2201	1301
15	2201	2201	1301
17	2201	2201	1301
20	1302	1302	1302
21	1302	1302	1302
22	1302	1302	1302
30	1302	1302	1302
40	1302	1302	1302
41	1302	1302	1302

the signal operations whose metering rate is adjusted based on measured or predicted downstream traffic conditions.

The signal operations of metering controllers are implemented as single-phase "pretimed" controllers. Metering controllers uses the "desirable network state" (i.e., desirable value of occupancy at given locations) to compute the timing table. The desirable network state can either be predetermined or set dynamically by a user defined external control model. The implemented default ramp metering logic follows the ALINEA ramp metering model by Papageorgiou et al. (1990). A closed-loop feedback control law is used to update the metering rate periodically:

$$u(t+1) = u(t) + \kappa\,[\hat{x}(t+1) - x(t)] \tag{4.4}$$

where:

$u(t+1) =$ metering rate to be applied in the next time period;

$u(t) =$ metering rate applied in the last time period;

$\hat{x}(t+1) =$ *desired state* for next time period; and,

$x(t) =$ measured state during the last time period;

$\kappa =$ a parameter.

In addition, TMS allows the metering parameters to be updated by an external model. Therefore, it is capable of modeling systems with a two-level hierarchical control logic: (i) a system-wide optimization model (i.e. up-level model) calculates the desired network state; and, (ii) a local closed-loop feedback controller (i.e. lower-level model) adjusts the metering rate, by taking into account the stochastic disturbance of traffic flows, in order to minimize the difference between actual and desired network states. For example, using MATLAB (MathWorks, 1995) Chen (1996) implemented an up-level control model which interfaces with TMS to set the metering parameters.

The following parameters are used for metering controllers (their values are obtained from a control logic file and can be updated periodically):

Cycle length: This defines the time in seconds required by the metering signals to complete a green-yellow-red cycle. The time splits between green and red intervals are calculated based on the current metering rate and the given *cycle length*.

Limiting parameters: The metering rate must be chosen from a feasible range because of capacity constraints, driver behavior, safety considerations, etc. These constraints are represented by a set of limiting parameters, including *minimum green time*, *yellow time*, and *maximum red time*. These parameters are the basis for translating metering rates into an implementable timing table.

Regulators and desired state: These define the parameters used in Eq (4.4), including the regulator parameter κ, the detectors that measure the network state $x(t)$, and the default value of desirable state.

Queue detectors: To prevent queue caused by metering from spilling back to the upstream intersection and blocking cross traffic, a queue over-writing logic is often used in practice. The metering controllers in TMS can be assigned a list of *queue detectors* and an *occupancy threshold* to determine whether the queue length has become critical. When the measured occupancy exceeds the threshold, the metering rate is increased to the maximum rate in order to reduce the queue length.

Step size for updating metering rate: The metering rate and signal timing table may be updated periodically based on sensor measurements. A step size is used to specify this updating frequency. Between two updates the metering rate remains constant and ramp signals operate according to the timing table. When the metering rate changes, the corresponding timing table will not be implemented until the current phase is completed. This allows a control interval to complete its minimum length.

4.4 Incident Management

A rule-based incident management scheme is included in the TMS module. Currently this incident management scheme applies only to the operations of traffic signals and signs, not to the emergency units. Furthermore, the response plans are pre-determined and response delay is user defined.

Response Plan

Incident management in TMS is represented by response plans, which specify the states of the signals and signs in the network at various stages. In the current implementation, incident response plans are designed only for lane use signs, portal signals at the entrance to a tunnel, variable speed limit signs, and variable message signs.

Each response plan consists of one or more *response phases* and a final *clearance phase*. Each phase is assigned an *activation delay* and a set of predefined *actions* to be taken at various situations. For the first response phase the activation delay is defined as the interim time between when an incident is detected/confirmed and the actions are implemented. The delay for subsequent response phases is the elapsed time since the implementation of the previous phase. The final phase of a response plan is called clearance phase, which defines the actions to be taken when an incident is cleared. These final actions usually restore the devices to their default state. The corresponding activation delay indicates how long the system will wait before it restores the devices to their default state after the incident is cleared.

In implementing incident management in TMS the following parameters have been defined to select appropriate actions: (i) situation code; (ii) device type; (iii) affected region; and, (iv) signal/sign state.

Situation code determines whether or not the corresponding action applies to a particular situation. Situations are characterized by the severity of the incident and how emergency vehicles, if any, will access the incident site, i.e.:

- *Lane blockage:* An incident can cause a partial or full blockage.
- *Emergency vehicle access:* Emergency vehicles may approach the incident from upstream or downstream. In the case of full blockage, downstream access may be the only choice.

Device type specifies the types of signals or signs a given action affects. The following control devices can be used in the current incident management module: (i) lane use sign (LUS); (ii) variable speed limit signs (VSLS); (iii) variable message signs (VMS); and, (iv) portal signals (PS).

Affected region defines the area where signals or signs are of interest to the given action. Four variables are used in defining an affected region. Two of them define the reference points (e.g. incident position, upstream, or downstream portal signals), and the other two define the relative distance from the specified reference point to the start and end points of the region.

Signal/Sign state specifies designated state, depending on the device type, representing a signal color (LUS or PS), message ID (VMS), or a speed limit (VSLS), for each of the signals or signs in the affected region. For VSLS actions that change speed limits gradually over the space, the state is specified by two variables: an initial state and a step size. In this case, the speed limit is set to the initial state for the first sign and varies for the remaining signs in the region, according to the step size and predefined upper and lower bounds for speed limits.

Incident Detection Time

In the real world a traffic management center (TMC) detects incidents based on data obtained from the surveillance system and external agency reports (e.g., police, drivers, etc.). While it is possible to directly model the incident detection element in the simulator, the current implementation uses a simpler approach. When an incident occurs in MITSIM, a message is sent to TMS to describe the characteristics of the incident such as position, severity, duration, etc. A detection delay is added to the activation delay of the first phase of the incident response plan. This detection delay is a function of the method used for incident detection.

Selection and Activation of Response Plan

After an incident is confirmed, the activation time for the first phase of each response plan is scheduled. This response phase is activated at the scheduled time, and after its activation, the next phase is scheduled again. Such operations proceed phase by phase until all phases specified in the response plan are completed.

When a response phase is activated, all related actions are executed. In searching the signals and signs that are of interest to an action, a breadth-first search algorithm is used. The search begins from the start point and moves toward the end point(s) of the affected region. All the control devices of the given type found in the affected region are then set to the designated *state*.

Chapter 5

Mesoscopic Traffic Simulator

A mesoscopic traffic simulator (MesoTS) is used in TMS to predict traffic conditions in the network – the information needed for generating anticipatory route guidance. MesoTS starts with a given initial network state, predicted travel demand, and candidate control and route guidance and simulates vehicular flow using speed-density relationships and route choice models. Capacities for each segment and turning movements at intersections are periodically updated based on the traffic signal settings and the volume and composition of approaching traffic flows. The simulation of vehicle movements in MesoTS consists of two phases: an *update phase* and an *advance phase*. The update phase calculates speeds of *traffic cells* – groups of vehicles located in the vicinity of each other. The advance phase moves individual vehicles based on the speeds of the traffic cells they belong to. Each update phase may consist of one or more advance phases. Individual vehicles' approximate positions are tracked and as spacing between vehicles changes cells are split and combined. MesoTS outputs time variant link traffic counts, travel times, and other variables of interest.

5.1 Network Representation

MesoTS represents the road network using three types of network objects: nodes, links, and segments. Nodes and links are described by the same data items used in MITSIM. Each segment is characterized by the following data items: *free flow* *speed*, *geometry*, the *number of lanes* it contains, and an *index to a performance function*. The index to the performance function is an ID that points to the *capacity*, *jam-density*, and other parameters to be used in the speed density function (see Section 5.4) of the segment. MesoTS does not explicitly model lanes. However, the turning movements at intersections and queue spillbacks are represented by *traffic streams*.

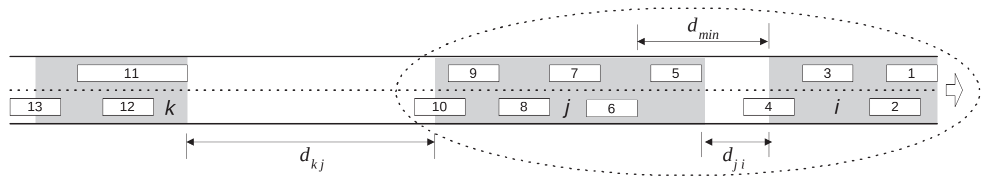
5.2 Traffic Cells and Traffic Streams

Vehicles in MesoTS are organized in groups called *traffic cells* (see Figure 5-2). Each traffic cell consists of a list of vehicles that move together according to the same traffic dynamics. A traffic cell stores the speeds and positions of the "head" and "tail" vehicles *in the update phase*. In a link connected to multiple downstream links, the last traffic cell mutates into *traffic streams*. A traffic stream is a collection of vehicles that intend to move to the same downstream link. In this case, the traffic cell maintains the speed and position not only for the tail vehicle but also for the head vehicle of each traffic stream. Speeds of individual vehicles are interpolated based on their positions and the speeds of the head and tail vehicles of the traffic streams they belong to. Vehicles in a traffic stream are moved according to the interpolated speeds until they are constrained by a leading vehicle or traffic cell. This traffic cell and traffic stream approach allows for efficient modeling of vehicle movements and queues.

Two predefined thresholds – d_{min} and d_{max} – are used to control the merging and splitting of traffic cells. When the distance between two traffic cells[1] becomes less than the parameter d_{min} (e.g., 100 feet), they are merged into a longer traffic cell as demonstrated by cells i and j in Figure 5-1(a). When the distance between two vehicles in a traffic cell is greater than the parameter d_{max} (e.g., 300 feet), this traffic cell is split into two cells between these two vehicles (see cell j in Figure 5-1(b)).

[1]The distance between two traffic cells is denoted by the gap distance between the last vehicle in leading traffic cell and the first vehicle in the following traffic cell (see d_{kj} and d_{ji} in Figure 5-1).

(a) Merge



(b) Split

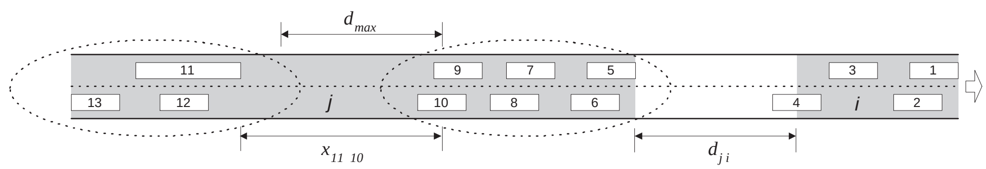


Figure 5-1: Merge and split of traffic cells

In the current implementation of MesoTS, all traffic cells are completely contained in their segments. In other words, no traffic cell will extend across segment boundaries except for the last vehicle in a traffic cell. When a vehicle moves from one segment to the next, it either joins the last traffic cell in the downstream segment or forms a new traffic cell in case the last traffic cell is not *reachable*. A traffic cell is *reachable* by a vehicle if the distance from the last vehicle in that cell is less than a predefined threshold d_0 (e.g., 200 feet). If all vehicles in a traffic cell have moved out and no vehicle is left at the end of the advance phase, the empty traffic cell is destroyed.

The longitudinal space occupied by a vehicle in a traffic cell is the vehicle length divided by the number of lanes in the segment. Thus density is correctly estimated.

5.3 Capacity Constraints

The time to be simulated is divided into short periods called *constant capacity intervals* (e.g., 15 seconds). At the beginning of each constant capacity interval, the simulator computes capacities for each segment and turning movements at intersections and equivalent time headways. The capacity of a segment is the maximum number of vehicles allowed to exit the segment during a period of time. Capacity for the turning movements at intersections are computed in two steps:

(a) The simulator scans the inbound links at each intersection for the vehicles that could arrive at the intersection during the next update interval (given their current positions and speeds). The result of this calculation is stored in a matrix whose rows correspond to the inbound links and columns correspond to the outbound links. Each element in this matrix is an estimate of the total *demand* for a particular turning movement.

(b) A *capacity translator* is then used to convert the signal setting or control logic at the intersection into the number of vehicles allowed for each turning movement. The translator takes into account both the effective green time and the interactions from conflict movements which are determined based on the results of step (a). Recommendation in HCM (1985), for example, can be used for this purpose.

Based on the capacity of a segment (or a turning movement), the simulator calculates an average time headway (a similar approach was used in INTEGRATION for queue dissipation (van Aerde et al., 1997)). This time headway is then used to determine the time at which vehicles can move to the segment:

$$t_n = t^*_{n-1} + \frac{1}{q} \tag{5.1}$$

where:

t_n = the earliest time to move next vehicle;

t^*_{n-1} = time last vehicle was moved; and,

q = capacity of the segment (or turning movement).

A vehicle is allowed to move from one segment to another or across an intersection only if the time of the simulation clock is greater than or equal to the time t_n given by Eq (5.1); otherwise the vehicle queues up.

The simulator also checks the storage capacity in the downstream segment before it moves a vehicle across segments. If the end of the last traffic cell in a segment reaches the boundary of that segment, no vehicle is allowed to enter that segment and vehicles form spillback queues at the upstream segment(s).

If a segment contains an incident, its capacity is reduced according to the number of lanes the incident blocks and the severity of the rubber-necking effect. When an incident is cleared, however, the capacity is not immediately returned to its default value but rather, restored gradually. In addition, the head speed of a queued traffic cell is constrained to a maximum value which also gradually increases to the free flow speed of the segment. This treatment reflects the fact that it takes time for a congested segment to recover because vehicles need time to accelerate to their full speed and severe interactions between vehicles slow down this process.

5.4 Traffic Dynamics

Traffic cells in a link and vehicles in a traffic cell are processed sequentially from downstream to upstream. The distance a vehicle moves in each advance step is calculated based on its speed; however, if the front vehicle is a constraint, a vehicle follows the front vehicle. Spillback only blocks vehicles that move to the direction which is the source of the spillback, unless the segment has only one lane or the density ahead exceeds the jam density. For example, in Figure 5-2, a spillback occurs for the south-bound traffic, and as a result, vehicles D and E queue up; however, the east-bound vehicles 5 and 6 can still move.

MesoTS uses two traffic models in simulating vehicle movements: a *speed-density* model and a *cell-following* model. The speed-density model calculates the speed for the last vehicle in the traffic cell; the cell-following model calculates the speeds of the head vehicles in a traffic cell (speeds of traffic streams). The speeds of the vehicles between the heads and tail are interpolated.

5.4.1 Speed-Density Model

The speed of the last vehicle in a traffic cell is determined using speed-density relationships. In the current implementation the following equation is used:

$$v_{i0} = V_{min} \ + \ (V_{max} \ - \ V_{min}) \ (1 - [\frac{K_i}{K_{jam}}]^{\alpha})^{\beta} \tag{5.2}$$

where:

v_{i0} = speed of the tail vehicle of traffic cell i;

K_i = traffic density in cell i;

K_{jam} = jam density of the segment; and,

α, β = model parameters.

V_{max} = free flow speed of the segment;

V_{min} = minimum speed of the segment.

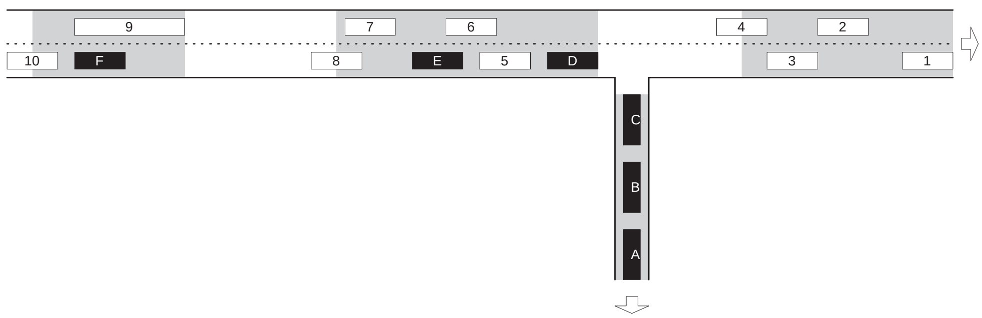
While the parameter V_{min} defines a speed lower bound, parameters α and β and the jam density K_{jam} determine the shape of the speed-density function.

Eq (5.2) is the same as the one used by May and Keller (1967) if V_{min} is set to 0. With a jam density of 210 vehicles/lane, they estimated using empirical data, $\alpha = 1.8$, $\beta = 5.0$.

5.4.2 Cell-Following Model

The cell-following model is used to compute the speeds of the head vehicles. If there is no leading traffic cell or the distance from the leading cell is greater than or equal to a predefined threshold d_{max}, free flow speed is used for all the head vehicles of the streams of a traffic cell; otherwise, the speed for a head vehicle is computed by interpolating the free flow speed of the segment and the tail speed of the leading traffic cell. Specifically, the speeds of head vehicles are calculated in two steps.

(a) Vehicles in a traffic network



(b) Corresponding representation in MesoTS

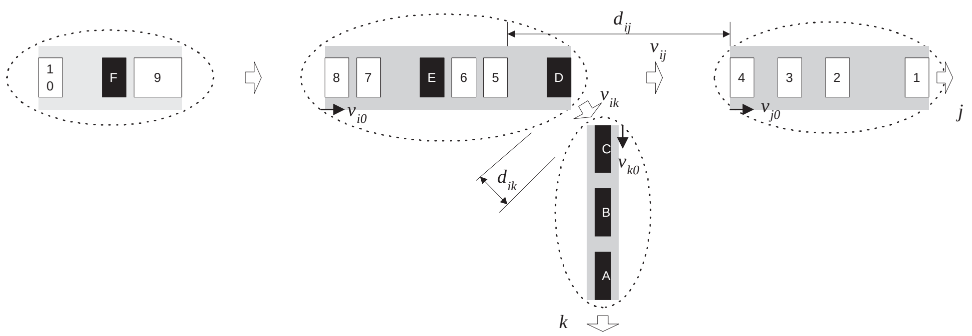


Figure 5-2: Vehicles and traffic cells
(white vehicles go east-bound; back vehicles go south-bound)

Step 1: A cell following speed is calculated as follows:

$$u_{ij} = \begin{cases} V_{max} & d_{ij} \geq d_{upper} \\ \lambda_j \, V_{max} \; + \; (1 - \lambda_j) \; v_{j0} & d_{ij} < d_{upper} \end{cases} \tag{5.3}$$

where:

u_{ij} = cell following speed of the first vehicle in traffic cell i traveling in direction j, i.e. the head speed of traffic stream ij (see vehicle 6 in Figure 5-2(b));

v_{j0} = speed of the tail vehicle in the leading traffic cell j (i.e. vehicle 4);

V_{max} = free flow speed in the segment;

d_{ij} = distance from the head vehicle ij (corresponding to the jth stream in cell i) to the last vehicle in the leading traffic cell in direction j (i.e. distances between vehicles 5 and 4, C and D);

d_{upper} = a predefined threshold distance (e.g., 300 feet); and,

λ_j = a scalar equal to d_{ij}/d_{upper}.

Step 2: The actual speed of the head vehicle ij, v_{ij}, equals u_{ij} when this vehicle is one of the first n vehicles in its traffic cell, where n is the number of lanes in the segment; otherwise, v_{ij} is the minimum of the cell-following speed u_{ij} and a density-based speed v_{i0} calculated using Eq (5.2) with K_i being the density in the section ahead of this vehicle.

Eqs (5.2-5.3) are applied upon creation of each traffic cell and in each update phase. At the same time, the positions of the tail and head vehicles are recorded. The speeds of other vehicles are simply interpolated from these speeds, calculated in the update phase, based on the vehicles' current positions and the traffic streams they belong to.

A traffic stream may shrink or expand depending on whether the tail speed of the traffic cell is higher or lower than the head speed of the traffic stream. The movement of a vehicle in a particular traffic stream is also constrained by the position of the leading vehicle in the same stream. In segments with multiple lanes, for example, over-taking is possible for vehicles that are in different traffic streams, but impossible for vehicles that are in the same traffic stream.

5.5 Vehicle Characteristics

Each vehicle is assigned a type and group attribute. *Vehicle type* (e.g. cars, buses, truck, etc.) determines vehicle length. Aggregated vehicle type information such as percentage of trucks and passenger-car-equivalent is also used in the calculation of segment and turning movement capacities and vehicle density in traffic cells. *Vehicle group* (e.g. guided or unguided) specifies whether the driver has access and/or response to real-time traffic information. Guided drivers, for example, use the route guidance provided by ATIS in choosing their route.

5.6 Vehicle Routing

The same route choice and route switching models used in MITSIM (see Section 3.8) are also used by MesoTS in routing vehicles in the network. However, the model parameters used in MesoTS can differ from those in MITSIM, representing the error in modeling drivers' route choice behavior. This also allows the assessment of the sensitivity of results to errors in modeling user behavior.

5.7 Input and Output

The input to MesoTS includes a network database, a time-variant link travel time which represents the pre-trip traffic information, and time-dependent OD matrices. In addition, MesoTS can start from a given initial network state instead of an empty network. Initial network state consists of the following information on vehicles currently in the network or waiting to enter it: vehicle ID, type, origin, destination, path, current location, and time entering the current link.

The main objective of MesoTS is to calculate flow and travel times on individual links and paths. When a vehicle leaves a link, the time it spent in the link is recorded; when a vehicle finishes its trip, the total travel time from its origin to destination and other information related to the trip (e.g. vehicle type, origin, destination, miles traveled, etc) are also recorded. The travel times of all vehicles entering a link during a given period are averaged and stored. This result is reported at the end of the simulation. In addition, vehicle path records are also available.

5.8 Computational Tests and Validation

MesoTS is tested on the same network used in Section 3.11 (see Figure 3-14). On a SGI Indy R4400 workstation (200Mhz) MesoTS requires about 65 seconds to complete a $2\frac{1}{3}$ hours of simulation, while MITSIM requires about 16 minutes for the same network.

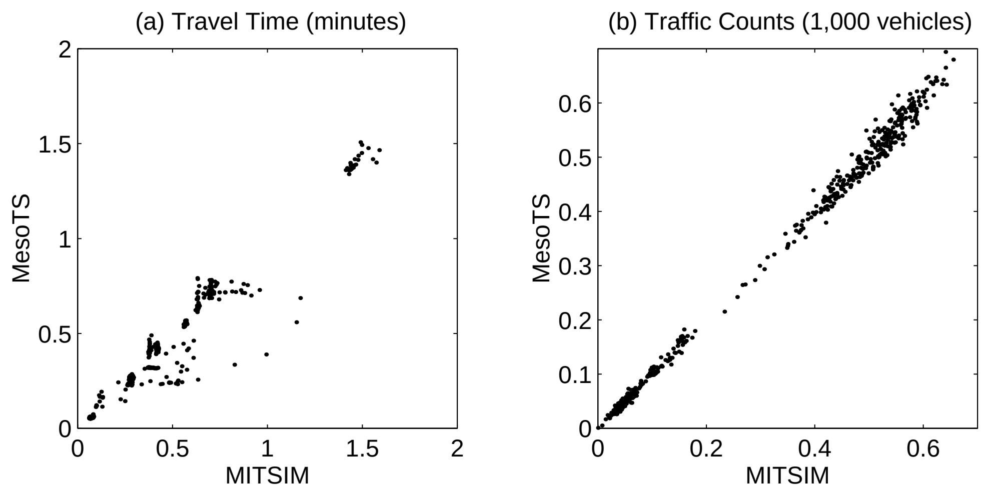


Figure 5-3: Comparison of MITSIM and MesoTS output using the I-880N network

Table 5.1 and Figure 5-3 compare the link traffic counts and travel times simulated by MITSIM and MesoTS. Link traffic count is the number of vehicles that enter a link during a given time interval (i.e. 5 minutes); link travel time is the average travel time spent by these vehicles in the link. The statistics in Table 5.1 show that the difference of link flows and travel times from the two simulators are small (see the RMS errors and Theil's inequality coefficient). The high value of U_C and low value of U_S indicate that the variation in link flows and travel times simulated in MITSIM is also well represented in MesoTS. However, the slightly higher value of U_M for link travel time indicates that there is a systematic bias. The two scatter plots in Figure 5-3 show that the simulated travel times by MesoTS are slightly lower than the results given by MITSIM at some links. Most of this "under-estimation" occurs at the ramps (where vehicles enter and leave the simulated network). For example, when merging from an on-ramp into mainline traffic, vehicles in MITSIM check for safe gaps and may queue up if a gap is not acceptable. However, this type of temporary queuing is not modeled in MesoTS, where queues are built up only if there is a spill back or the capacity of the segment or the turning movement is used.

The only calibration effort made in this study was varying the values of parameters that control traffic cells merging and splitting and the capacity of certain segments. The default values used in the simulation provide reasonable results for the network

Table 5.1: Comparison of MITSIM and MesoTS output in the I-880N network

Measure of Errors[a]		Travel Time[b]	Traffic Counts[c]
Theil's inequality coefficient		0.0795	0.0183
Proportions of inequality	U_M	0.1157	0.0049
	U_S	0.0226	0.0187
	U_C	0.8618	0.9764
Correlation coefficient		0.9778	0.9982
RMS error		0.09	13.8
Mean error		0.03	-1.0

[a] See Appendix E for definition of error measures;
[b] Travel times are in minutes; and
[c] Traffic counts are in number of vehicles in 5-minute intervals.

tested. Nevertheless, extensive validation and calibration work is needed.

Chapter 6

Case Study

In this chapter, the developed simulation laboratory uses the Amsterdam A10 beltway to evaluate the route guidance systems described in Section 4.2. As discussed earlier, route guidance can be based on two approaches: (i) naive guidance, utilizing current prevailing traffic conditions; and (ii) predictive guidance, utilizing predicted traffic conditions. The performances of these two systems are compared with the case in which no route guidance is provided. The overall objective of this study is to demonstrate the use of the developed simulation framework and to understand some important issues involved in evaluating dynamic route guidance systems in a realistic network setting.

6.1 The Network

The Amsterdam beltway (A10) is used in this case study (see Figure 6-1). A10 consists of two 32-km freeway loops which intersect with five major freeways and have 20 interchanges of various sizes (75 ramp intersections). The network serves local and regional traffic and acts as a hub for traffic entering and exiting north Holland. The network is subject to considerable recurrent congestion, especially on the north-western and southern parts of A10. Because of the network's loop structure and heavy traffic load, route choice affects traffic flow, and hence dynamic traffic management could potentially improve traffic conditions particularly when incidents occur.

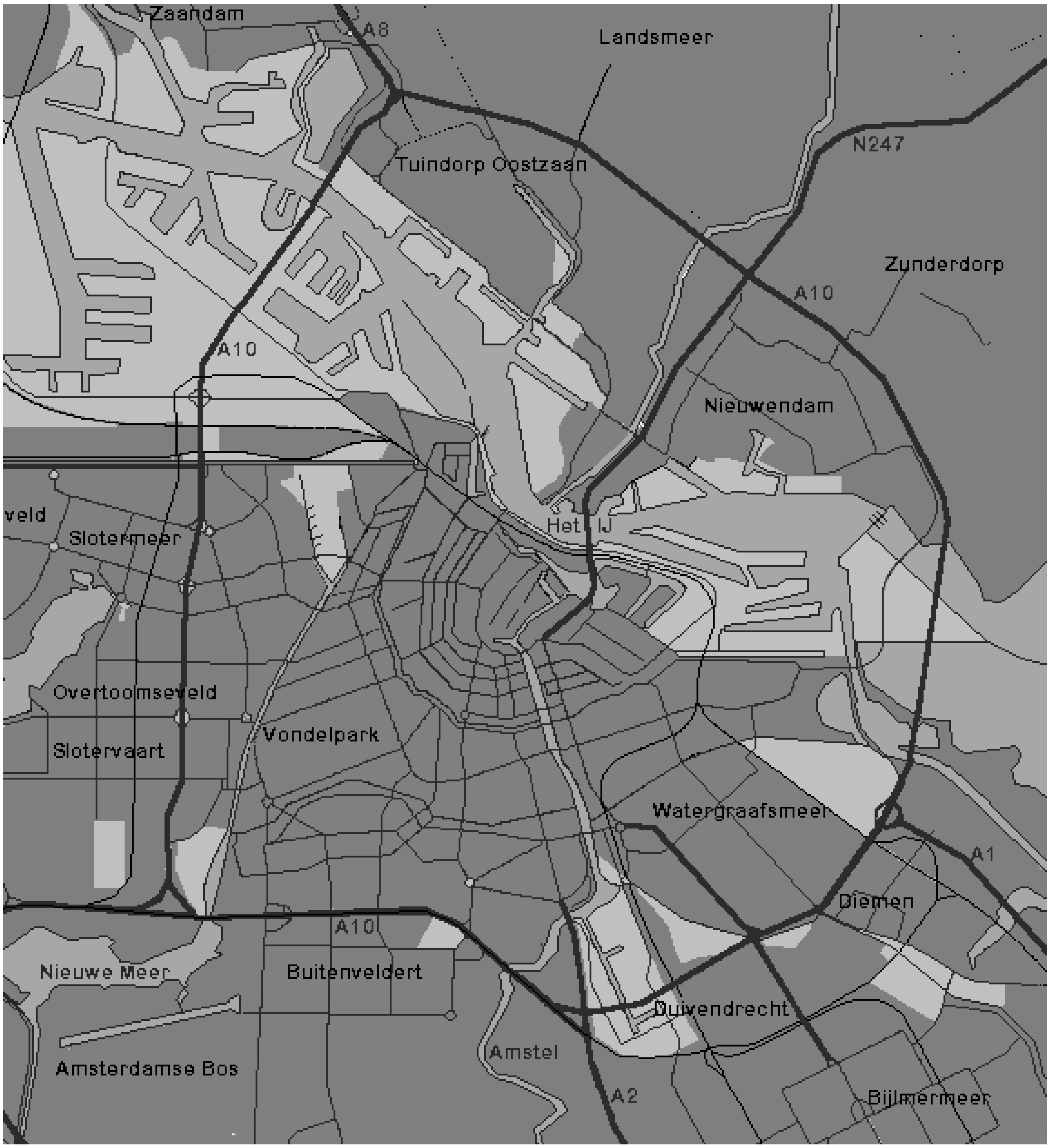


Figure 6-1: Amsterdam city map
(Source: http://www.hcg.nl/daccord/amst.htm)

The simulation network (see Figure 6-2) is created using the *road network editor* based on intersection coordinate data and highway maps obtained from the Netherlands Department of Transport and Water Management. The total length of the coded network is about 130 kilometers (344 lane-kilometers). It consists of 196 nodes, 310 links, 612 segments and 1510 lanes.

6.2 Sensor Data

Surveillance sensor data at various locations on the A10 network have been used to calculate the free flow speed of the segments in the network and to estimate the time-dependent OD flows. This data set was collected by the MARE system, a research facility affiliated with the Netherlands national Motorway Traffic Management (MTM) system. The freeway sections on which MTM is installed are equipped with dual induction loops at approximately 500-meter intervals. Traffic counts and speeds, aggregated and exponentially smoothed for 1-minute intervals, were recorded in files for a period of 3 weeks between April 12th to May 2nd, 1994. The data for 209 sensor stations in the A10 network were further aggregated into 15-minute intervals, covering the hours between 6:00-10:00 am for 19 days (data on April 12th and 25th are erroneous and dropped from further analysis). In addition, some sensor stations are omitted on days showing incomplete data or revealing the exact same value across all the time intervals (a strong indication of malfunction of the sensors).

The maximum speed observed over 19 days at each sensor station across all intervals, usually occurring at the off-peak periods and weekends, is used to determine the free flow speed distribution for the segment that contains the sensor.

6.3 OD Flows

The time dependent OD trip tables provided by Ashok (1996) are modified in order to be used as input in this case study. Ashok estimated OD flows between 20 centroids based on speeds and counts measured at 65 sensor stations. Each origin/destination

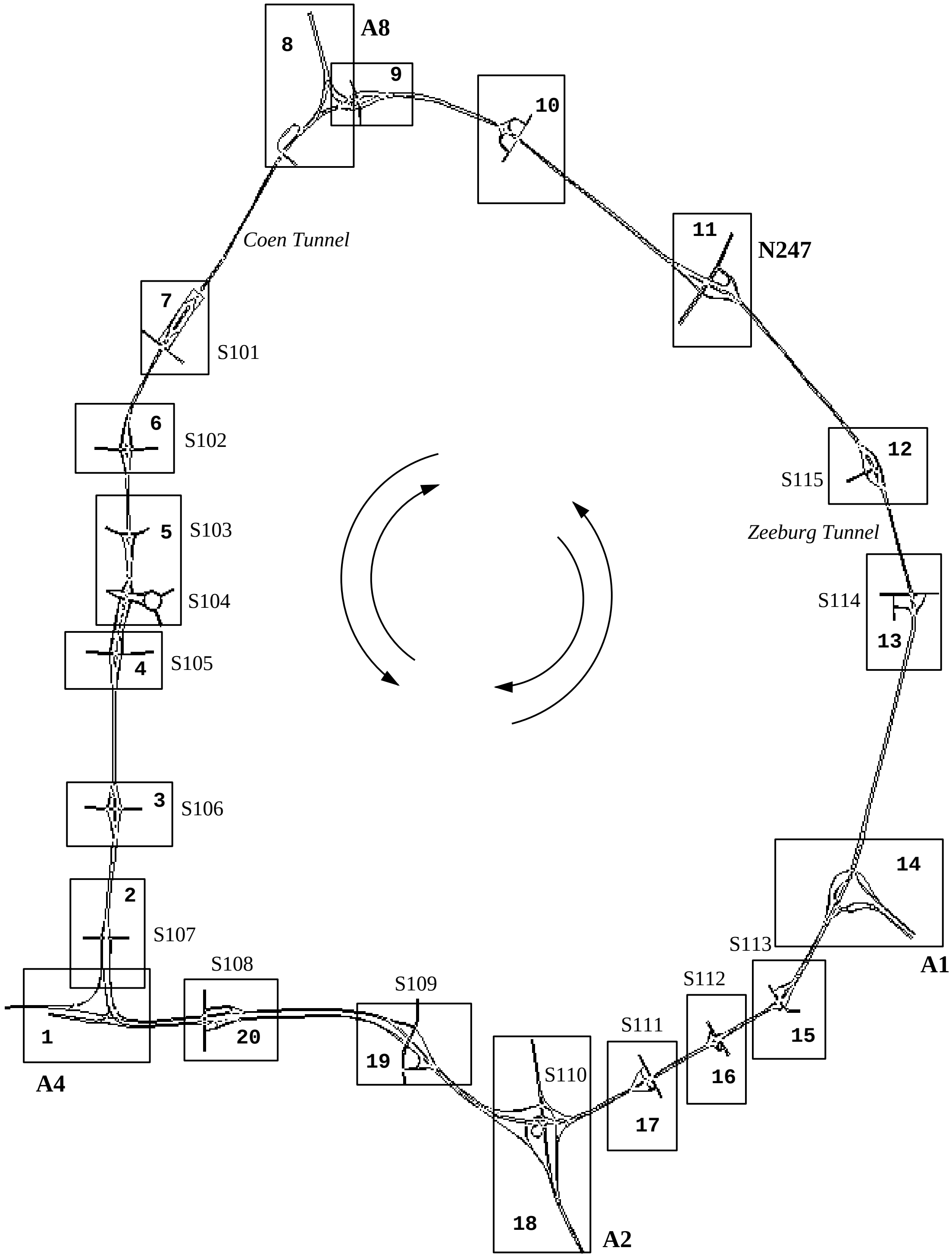


Figure 6-2: The A10 network and “origins/destinations”

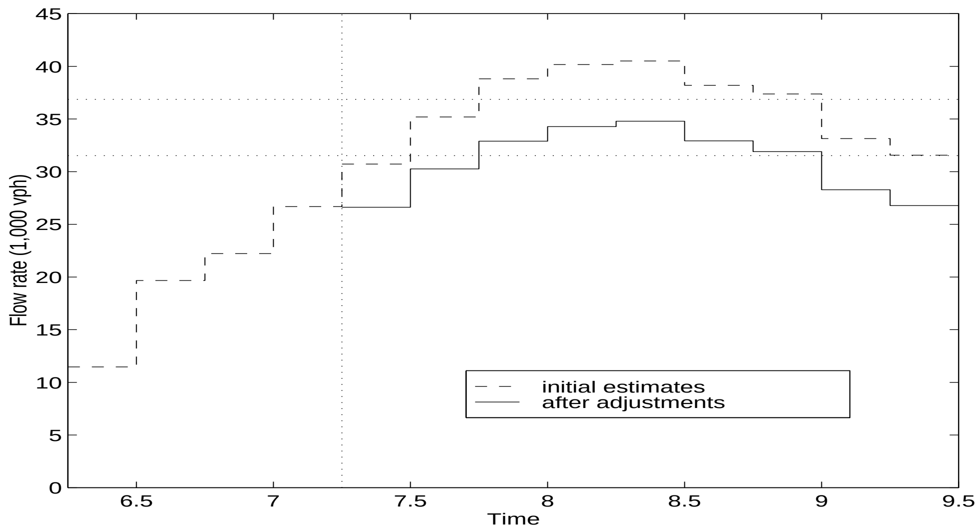


Figure 6-3: OD flow profile

represents a group of ramp intersections and freeway interchanges along A10 (see Figure 6-2). These OD flows are specified for 366 OD pairs in 15-minute intervals between the hours from 6:15 to 9:30 am. The average departure rate is 36,200 vph. Variation of departure and arrival rates across various centroids and time intervals are depicted in Figure 6-4. The total demand estimated by Ashok is shown by the dashed line in Figure 6-3.

Using these OD data, traffic flows during the $2\frac{1}{4}$ hours from 7:15 to 9:30am (morning peak) were simulated. The result from this initial simulation revealed that excessive queues and congestion formed at several locations, including the ramps at Sloten (S107) and Osdrop (S106) and the A8 interchange (see Figure 6-2). Beginning at 8:00 am, vehicles departing from and arriving at these locations formed queues at the entrance, causing severe congestion which continued throughout the remaining simulation. However, many congestion instances were not observed in the field data. A closer look at the volume of OD flows and network capacity indicates that the estimated OD flows are too high for centroid 2 (Sloten). During the peak hour of 8:00-900 am, there are 5,200 departures and 6,113 arrivals at the Sloten ramps. However, there are only 2 on-ramp lanes and 3 off-ramp lanes available at this location, with a probable total capacity of up to 3,500 departures and 5,000 arrivals per hour. Sensor data revealed that traffic counts at several sensor stations in this area were

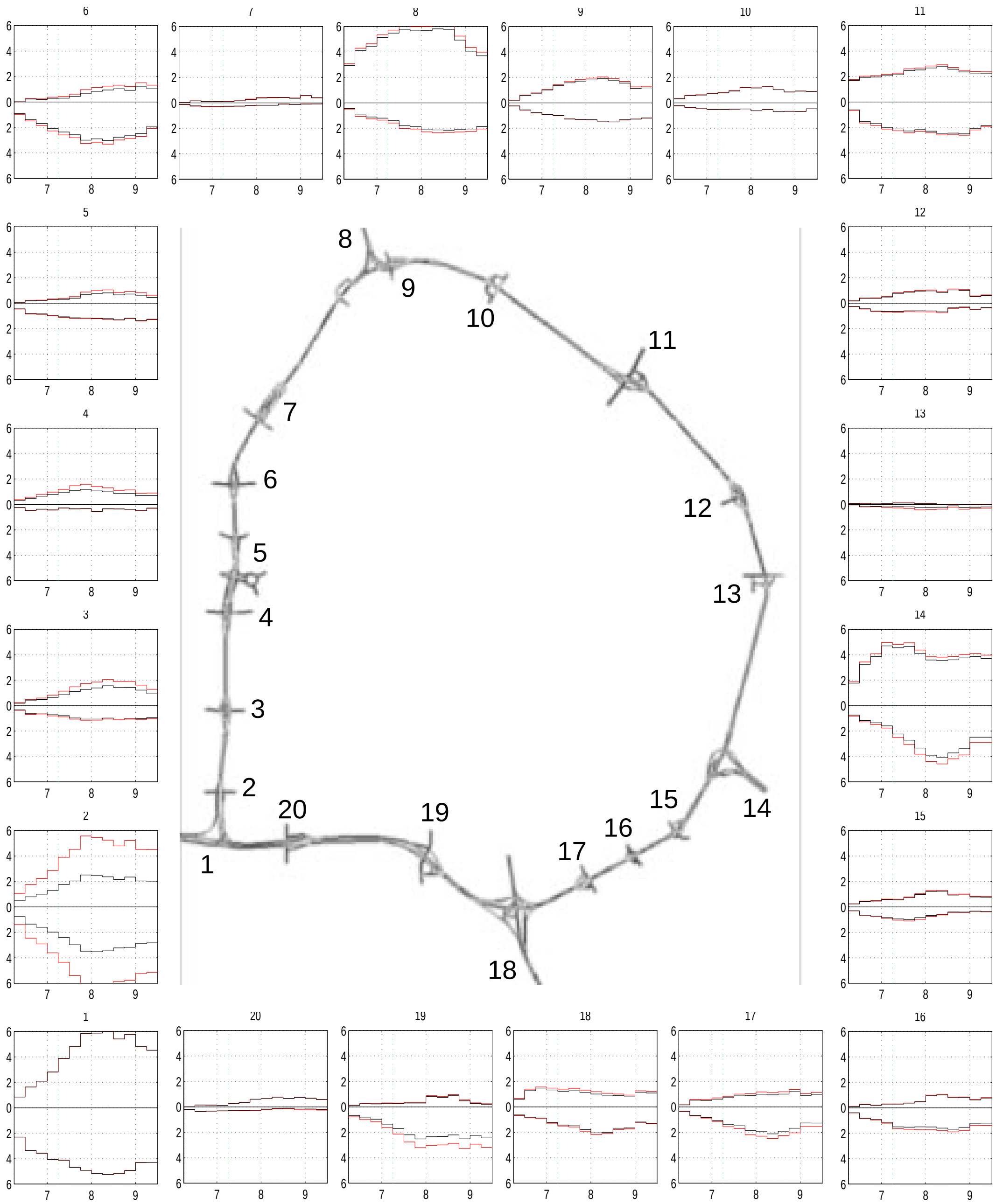


Figure 6-4: Inflow and outflow at each centroid

NOTE: Each plot corresponds to one centroid. The vertical axis is number of trips (in 1,000 vph), the horizontal axis is the time; the upper portion of the plot depicts inflows and the lower portion outflows; the dashed line represents the original estimates and the solid line the modified estimates used in simulations.

Table 6.1: Total traffic flows at the 20 centroids of the Amsterdam Beltway

	Outflows			Inflows		
Centroids	Original	Modified	Change (%)	Original	Modified	Change (%)
1	5209	5209	0.0	4756	4756	0.0
2	4857	2186	55.0	5649	3107	45.0
3	1659	1243	25.1	1066	984	7.7
4	1217	930	23.6	385	373	3.0
5	778	599	23.0	1249	1209	3.2
6	1083	820	24.3	2870	2633	8.3
7	365	336	8.0	171	161	5.7
8	5508	5209	5.4	2137	1938	9.3
9	1675	1551	7.4	1293	1291	0.2
10	1021	1002	1.8	571	568	0.5
11	2581	2446	5.2	2366	2249	5.0
12	901	839	7.0	568	511	10.0
13	47	40	13.7	333	203	39.1
14	4194	3926	6.4	3580	3148	12.1
15	960	897	6.6	684	619	9.5
16	706	673	4.7	1672	1475	11.7
17	1101	946	14.1	1964	1657	15.6
18	1205	1058	12.3	1707	1618	5.2
19	528	473	10.4	2928	2274	22.3
20	589	589	0.0	235	195	16.9

much higher than the segments' road capacities. These high link traffic counts were probably caused by malfunction of detectors and might have led to the overestimation of traffic flows.

Correcting the sensor data and estimation of OD flows is beyond the scope of this dissertation. However, in order to make the input data more realistic, the OD flows to and from the Sloten ramps were adjusted based on the number of lanes and lane capacity. Specifically, the outflows from Sloten are reduced by 55% and inflows to Sloten by 45%. The modified OD flows are summarized in Table 6.1 and Figures 6-3 and 6-4. The total OD flows after implementing these adjustments are depicted by the solid line. A total of approximately 69,700 vehicles depart during the $2\frac{1}{4}$ hour simulation period, which is equivalent to an average departure rate of 31 thousand vph. Compared with the OD flows provided by Ashok, the modified OD traffic flows are 14.4% lower.

The OD flows for these centroid-based OD pairs are then distributed to the corresponding "entrance" and "exit" nodes of the network, to derive departure rates for approximately 700 node-based OD pairs.

Because information on urban streets is unavailable, it is assumed that drivers' exit nodes are pre-determined and do not change. This assumption may be unrealistic, as drivers may change their planned exit nodes. For example, in response to observed/received traffic information, drivers may divert from A10 and instead travel urban streets which are not modeled in this case study.

6.4 Path Generation and Vehicle Routing

In this case study, MITSIM and MesoTS use the same travel behavior model and set of paths in routing vehicles. The specification of this model and its major input parameters, such as diversion penalty and freeway bias, are described. The preparation of path tables and calculation of historical travel times on these paths are also presented.

6.4.1 Specification of Route Choice Model

A multinomial logit *route choice model* is used to generate the paths connecting various OD pairs. The utility function $V_l(t)$ of the model can be specified in many ways. In this application, the following function is used:

$$V_l(t) = \beta \, \frac{c_l(t) + C_k(t + c_l(t)) \; + \; z_l}{c_0(t) + C_0(t + c_0(t)) \; + \; z_0} \tag{6.1}$$

where the subscript 0 denotes the shortest path, and (see Figure 3-2):

$c_l(t) =$ perceived travel time on link l at time t;

$C_k(t) =$ perceived travel time on the *shortest path* from node k (the downstream node of link l) to the destination if the vehicle arrives at node k at time t;

$z_l =$ penalty that captures freeway bias; and,

$\beta =$ parameter.

A similar specification is also used in the *route switching model*:

$$V_i(t) = \gamma \frac{\hat{C}_i(t) \ + \ Z_i}{\hat{C}_0(t) \ + \ Z_0} \tag{6.2}$$

where the subscript 0 denotes the shortest path, and:

$\hat{C}_i(t) =$ expected time on path i at time t; and,

$Z_i =$ diversion penalty if path i is different from the driver's current path.

Diversion penalties and freeway bias are additional costs imposed on (i) switching from the current route to an alternative route in the *route switching model*; and (ii) choosing an off-ramp as the next link in the *route choice model*. The penalty for choosing an off-ramp is introduced to capture freeway bias in route choice (see Figure F-1). A penalty of 5-10 minutes for z_l and 1-2 minutes for Z_i were used in the A10 network. A lower value for z_l may not be sufficient to prevent drivers from taking a ramp path[1] under normal conditions[2]; while a higher value may cause some vehicles to circulate if the acyclic test is not used (see Section 3.8.1).

6.4.2 Path Table

For every OD pair in the network, a set of paths was generated using MITSIM in a pre-process mode under the following conditions:

- the acyclic path test is enabled;
- the path set consists of routes generated under both normal and incident conditions;
- the path that each driver chooses is progressively generated using the route choice model;
- simulation time is sufficiently long for all vehicles to complete their trips.

[1]A ramp path is defined as a route that contains the following road segments: diverting from freeway and taking an off-ramp, passing an intersections of surface street, and returning back to freeway using an on-ramp.

[2]Ramp paths can be used under incident to bypass the blockage or make a U-turn to use the opposite loop.

All unique paths connecting various OD pairs are collected into a *path table* (see Section F.4 for an example). Most OD pairs in the A10 network have two routes; therefore, the path table generated contains a total of about 1,500 paths for the 700+ OD pairs.

6.4.3 Historical Link Travel Times

Both MITSIM and MesoTS require time-dependent link travel times to calculate shortest paths and update path travel times. While historical link travel times can be obtained in several different ways, in this case study they are estimated by simulating drivers' day-to-day travel decisions using MITSIM. The process begins with an initial estimate of time-variant link travel times (e.g., observed travel-times or free-flow travel-times). Traffic flows are simulated based on the given link travel times. During the simulation, the time each driver spends in a link is recorded according to the time he enters that link, and is used to compute the time-dependent link travel times. The expected travel time for the next day is a weighted sum of the expected and experienced travel times from the current day:

$$c_{it}^{(k+1)} = \lambda^{(k)}\, \hat{c}_{it}^{(k)} \;+\; (1-\lambda^{(k)})\, c_{it}^{(k)} \qquad k = 0, 1, \cdots \tag{6.3}$$

where:

$i =$ link index;

$t =$ time interval index;

$k =$ day;

$c_{it}^{(k)} =$ input link travel times to the kth iteration;

$\hat{c}_{it}^{(k)} =$ output link travel times from the kth iteration; and

$\lambda^{(k)} =$ a weight parameter for kth iteration.

The simulation is terminated when the expected and experienced link travel times converge or a pre-defined maximum number of iterations is reached. The evolution of the average OD trip travel time, with $\lambda^{(k)} = 0.75$, is shown as a function of the number of iterations in Figure 6-5. Each iteration represents drivers' experience for a particular day. The change in average trip travel time becomes insignificant after 11 iterations. Hence, in the case study the average link travel times for the last three iterations are used as historical link travel times.

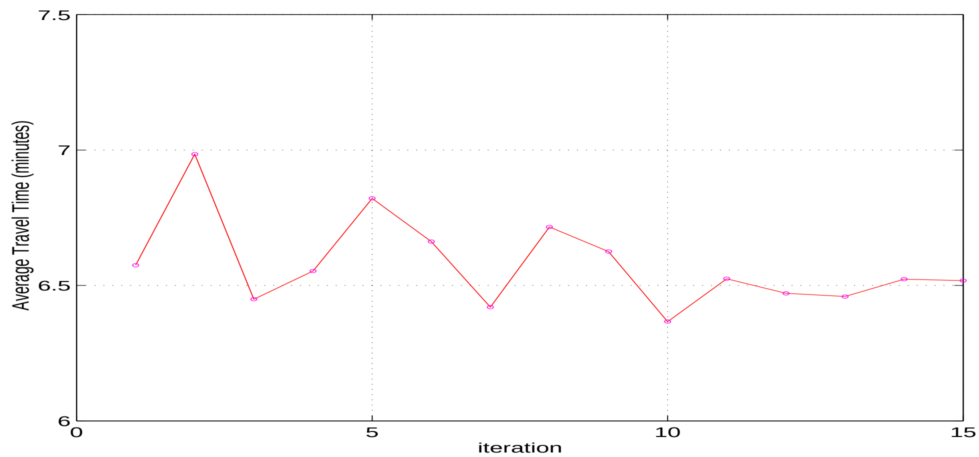


Figure 6-5: Change of average trip travel times

6.5 Calibration of MITSIM and MesoTS and Validation of MesoTS

In this case study, MesoTS is used by TMS in generating predictive route guidance. Because the "ground truth" of traffic conditions in the network is represented by MITSIM, MesoTS was calibrated against MITSIM in order to obtain accurate traffic prediction. Simulation of traffic flows in the A10 network under normal conditions (without incident) was conducted and the output was used to estimate the parameters in the speed-density functions. A limited calibration of MITSIM took place. More specifically, the free flow speeds of the segments were derived from the observed free flow speed distribution (as measured by the sensors in the A10 network).

Speeds and occupancies at 209 sensor stations simulated in MITSIM were collected for 15-minutes time intervals. Traffic densities at these sensor stations were then estimated based on the occupancies, i.e.:

$$K_{it} = \frac{5280\ O_{it}}{100\ L} \tag{6.4}$$

where the subscript i indicates the sensor station, and t the time interval during which the data is collected; and:

K_{it} = Density (number of vehicles per lane-mile);

O_{it} = Occupancy (percent of time a detector is occupied by vehicles); and

L = Average vehicle length (18.6 feet).

The speeds and densities were then used to estimate the parameters of the speed-density function in MesoTS (see Eq (5.2) on page 114 and Figure 6-6). To explain the estimation procedure, we rewrite Eq (5.2):

$$\ln(V_{it} - V_{min}) = \ln(V_{i\,max} \; - \; V_{min}) \; + \; \beta_i \; \ln(1 - [\frac{K_{it}}{K_{jam}}]^{\alpha_i}) \; + \; \epsilon_{it} \qquad (6.5)$$

where:

V_{it} = average speed at station i during time interval t;

V_{min} = the minimum speed;

$V_{i\,max}$ = free flow speed of the segment that contains sensor station i;

K_{jam} = jam density; and,

ϵ_{it} = the error term.

The parameters, α_i and β_i are then estimated using non-linear least square (Pindyck and Rubinfeld, 1991), that is:

$$[\alpha, \beta] = \text{argmin} \; (\sum_i \sum_t \epsilon_{it}^2) \qquad (6.6)$$

The results are summarized in Table 6.2.

Table 6.2: Parameters in the speed-density function

α	β	$\hat{V}_{max}$ (mph)		
		Minimum	Mean	Maximum
1.1	1.5	52	72	91

Jam density (K_{jam}) is set to 210 vehicles/lane/mile;
Minimum speed (V_{min}) is set to 5 mph.

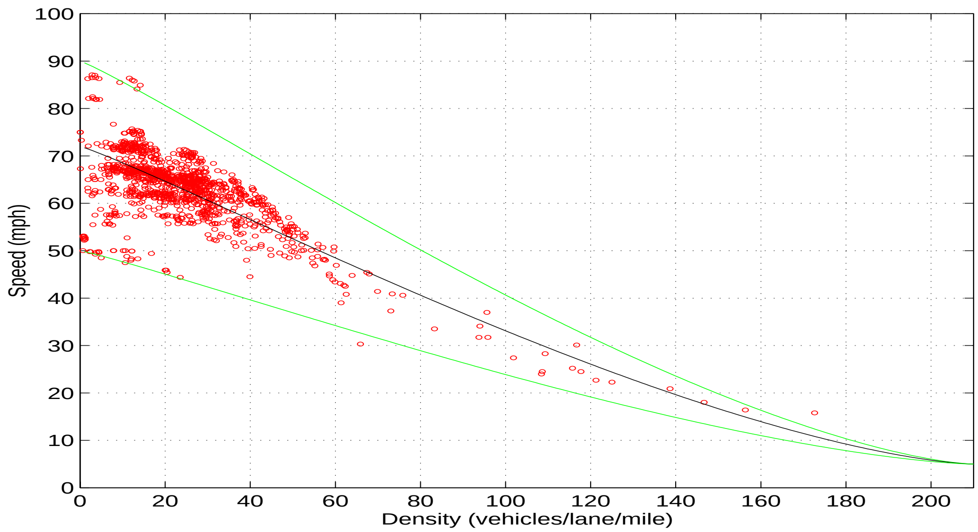


Figure 6-6: Speed-density relationship

Figure 6-6 shows the speed-density relationships based on the estimated parameters. The three curves correspond to the speed-density functions for segments with the maximum, mean, and minimum free flow speed respectively (i.e. same values for parameters α and β but different values for free flow speed $\hat{V}_{max}$).

In order to assess the accuracy of MesoTS, link traffic counts and travel times simulated by MesoTS and MITSIM are compared under two scenarios: one with the default traffic conditions (Figure 6-7) and the other with an incident which blocks the Coen tunnel exit for 20 minutes (Figure 6-8). Traffic count is the number of vehicles that enter a link during a particular time interval. Link travel time is the average time these vehicles spend on that link. A total of 8370 data points (310 links $\times$ 27 time intervals), each representing a pair of data values obtained in MITSIM and MesoTS, respectively, for a particular link and time interval (5-minute long each), are used in each scenario.

These figures reveal that traffic counts by MesoTS fit well with the counts by MITSIM (see Figures 6-7(a) and 6-8 (a)), while at some data points travel times compared poorly with the MITSIM output (see Figures 6-7(b) and 6-8 (b)). However, for the vast majority of data points, the errors fall into an acceptable range, with larger errors occurring under the incident scenario.

The relatively large travel time errors are also indicated by the high value of Theil's

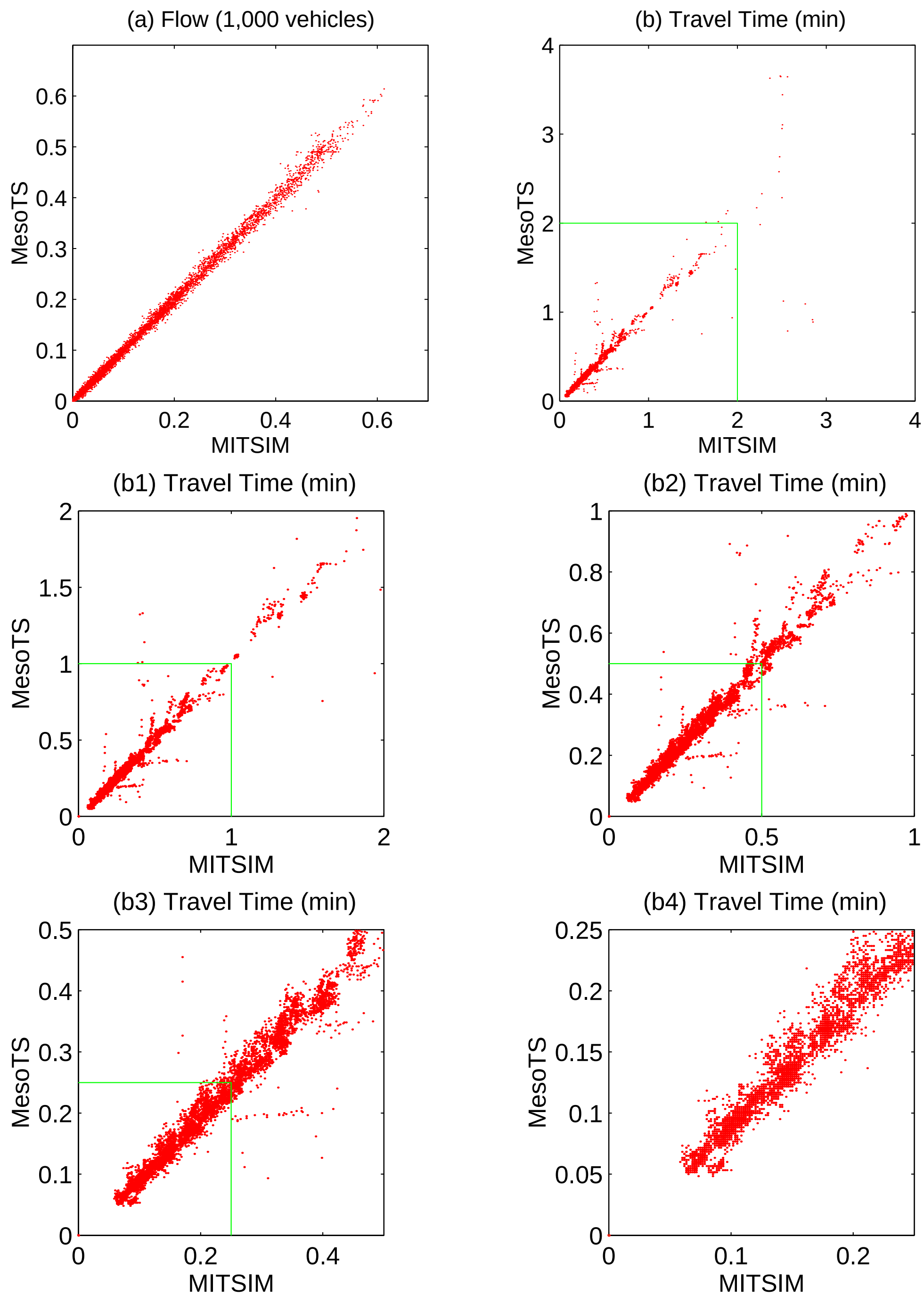


Figure 6-7: Comparison of MesoTS and MITSIM output (without incident)

NOTE: In all plots, the X and Y axes represent the values simulated in MITSIM and MesoTS respectively. Flow is the number of vehicles entering a link during a particular 5-minute time interval; travel time is the average travel time these vehicles spent in that link. The plot (b1) magnifies the boxed portion of plot (b); plot (b2) magnifies the boxed portion of plot (b1); and so on.

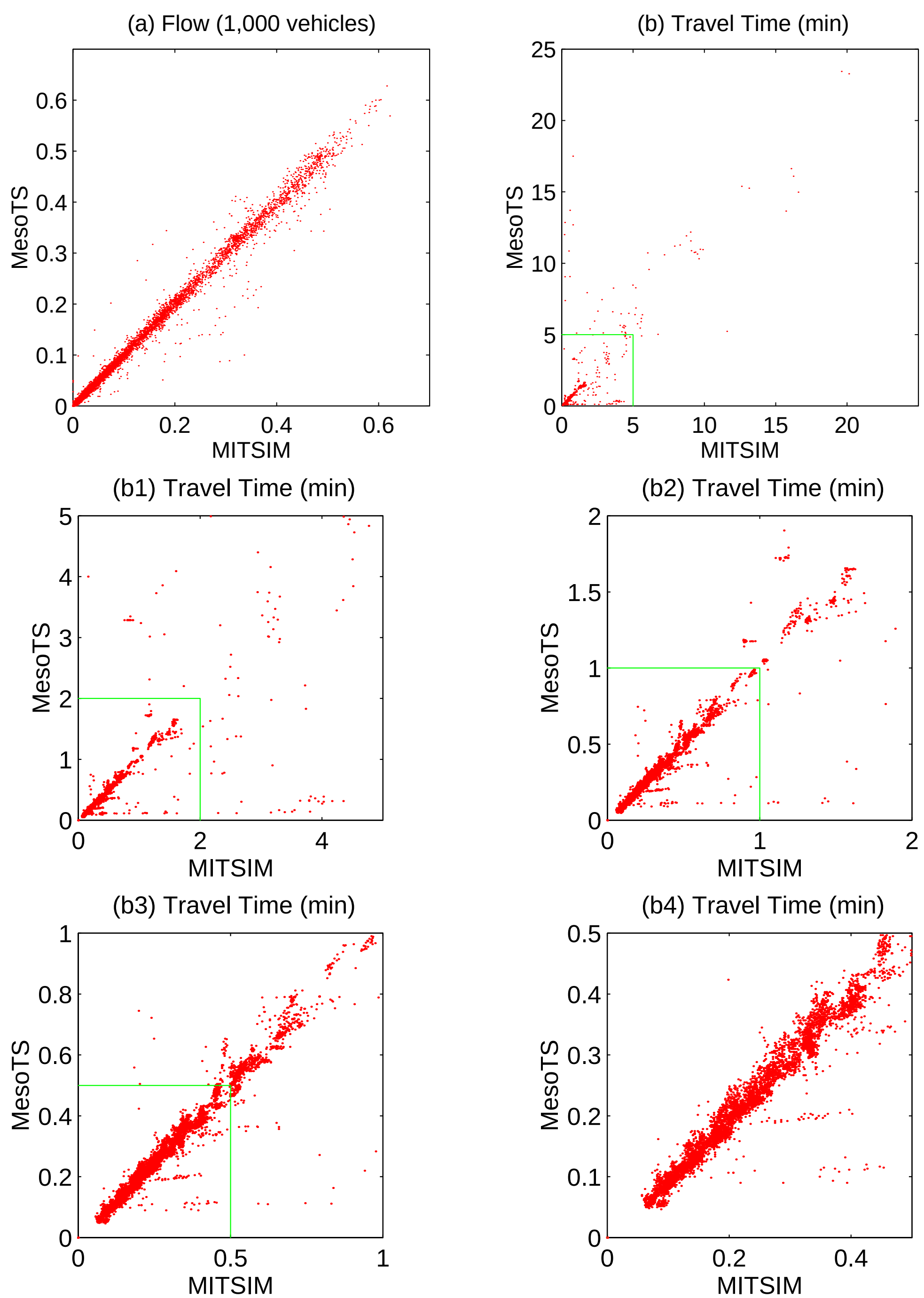


Figure 6-8: Comparison of MesoTS and MITSIM output (with incident)

Table 6.3: Comparison of MesoTS and MITSIM output in the A10 network

		w/ Incident		w/o Incident	
		Flow	Travel Time	Flow	Travel Time
Correlation coefficient		0.9986	0.9687	0.9952	0.8556
RMS error		0.0069	0.0651	0.0130	0.4933
RMS percent error (%)		24.57	14.21	26.57	89.69
Mean error		0.0003	0.0035	0.0005	-0.0150
Theil's inequality coefficient		0.0197	0.0853	0.0373	0.2653
Proportions of inequality	U_M	0.0022	0.0029	0.0012	0.0009
	U_S	0.0019	0.0276	0.0059	0.1418
	U_C	0.9959	0.9695	0.9929	0.8573

inequality coefficient and root mean square (RMS) percent error (see Table 6.3). Most of the outliers that contribute to this error correspond to the links close to the incident. Fortunately, the small U_M values indicate that there is no systematic bias in link travel times estimated by MesoTS. In addition, the size of link travel time variations observed in MITSIM is similar to that in MesoTS, as indicated by the value of U_S. The correlation between the two outputs is high. Nevertheless, the reason behind the poor fit of travel times at some links and time intervals between the two simulators requires further investigation. A preliminary investigation indicated that three types of outliers exist in comparing MesoTS and MITSIM travel times under the incident scenario: (1) under-estimation at some on-ramps when mainline is congested; (2) over-estimation in the link after the incident is cleared; and (3) over-estimation at links before fork. The first case may be caused by inadequate representation of merging behavior in MesoTS; while the last two cases may due to the inappropriate parameter values in the models that dealing with queues.

6.6 Value of Real-Time Route Guidance

In this section, we first give a brief description of the three simulated scenarios and then present the simulation results.

6.6.1 Scenarios

This case study's three simulated scenarios have identical input data in OD flows, driver behavioral parameters, network configuration, and available paths. They vary only in the availability of real-time route guidance and the type of guidance. Simulations are conducted for each of the three scenarios under 100% demand and 80% demand respectively.

- N – *No guidance:* In this base scenario no real-time traffic information is provided. Drivers use their habitual routes which are based on time-variant historical travel times.
- M – *Naive guidance:* In this scenario, guided drivers update their paths periodically using information based on the latest traffic condition measurements. Specifically, every 5 minutes the travel time of each link is estimated according to the current state of the network using the speed-density relationship. These latest measurement-based estimates of link travel times are assumed to be time-invariant in the future, and are used by guided drivers' in making their route choice decisions.
- P – *Predictive guidance:* In this case guided drivers choose routes based on predicted travel times. The rolling horizon step size is 15 minutes, length is 45 minutes, and time resolution of guidance is 5 minutes. Specifically, every 15 minutes TMS generates new route guidance and predicts traffic conditions for the next 45 minutes by calling MesoTS iteratively (see Figure 6-9). Within each of the three 5-minute intervals, the route guidance remains constant until renewed when the next prediction becomes available 15 minutes later. MesoTS iterates at most five times and the most consistent travel time prediction is then used in MITSIM for routing guided drivers. The computational delay is assumed to be 1 minute.

In all scenarios, an 20-minute incident is simulated, beginning at 7:40 am. The incident occurs at the exit of the Coen Tunnel (see Figure 6-11) and blocks the two lanes. In Scenario P, the traffic management system's detection delay of the incident

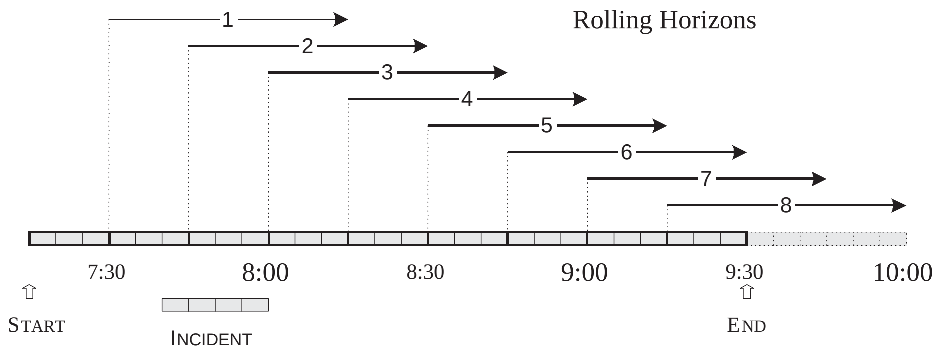


Figure 6-9: Rolling horizon diagram for traffic prediction

is set to 3 minutes. We also assume an ATIS penetration rate (percentage of drivers who are equipped and use the provided routing information) of 30%.

6.6.2 Measures of Effectiveness (MOE)

The MOEs considered in this case study include the average trip time and distance traveled for drivers who had completed their trips by the end of the simulation, as a function of their departure times. These data were collected for all trips (see Figure 6-10) and organized according to the departure time of each trip. Statistics for 6 representative OD pairs were also collected (see Figure 6-11). In Scenarios M and P, the MOE data are calculated separately for guided and unguided vehicles. Number of vehicles remaining in the network at the end of the simulation and their travel times were also calculated, since the evaluation of a particular route guidance system should take into account these incomplete trips.

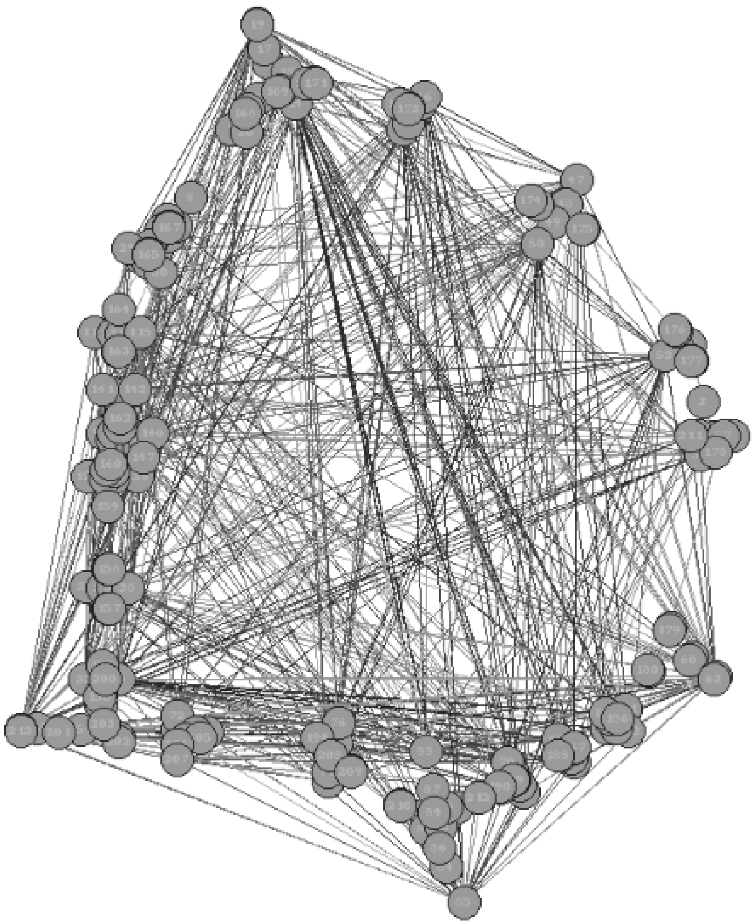


Figure 6-10: A snapshot of OD flow distribution

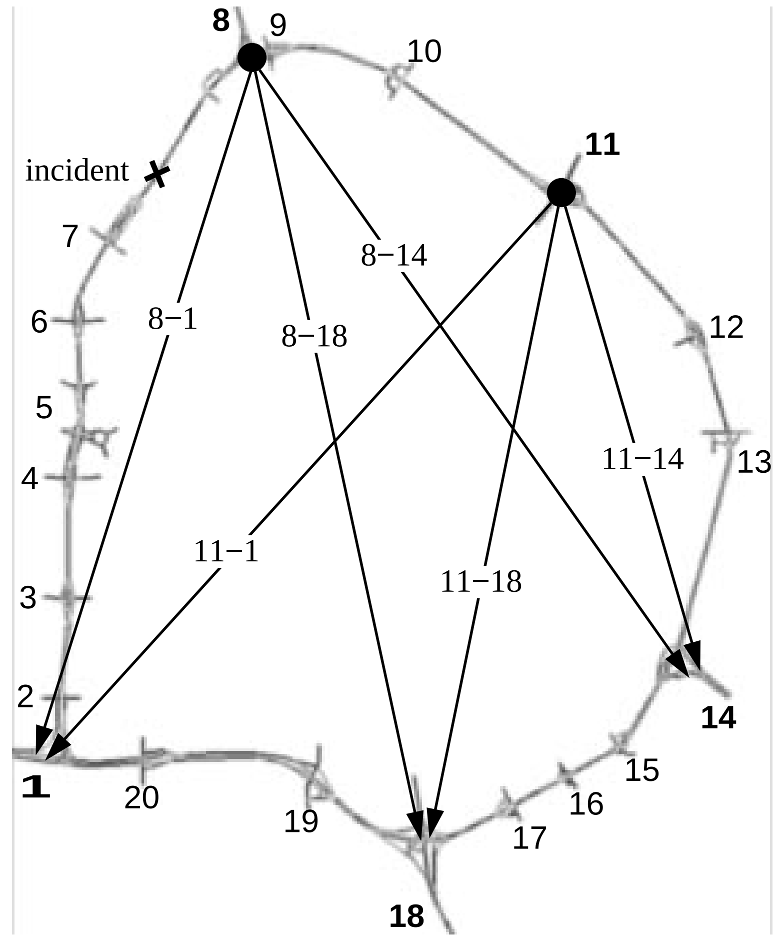


Figure 6-11: Representative OD pairs

6.6.3 Results

The measure of consistency for guidance (see Eq (4.3) on page 96) for one particular simulation run is depicted as a function of number of iterations and time period in Figure 6-12. As seen in the figure, the generated guidance tends to become more consistent as the number of iterations increases. Figure 6-12 also shows that, for a given number of iterations, the guidance used in the last iteration may not be the most consistent one.

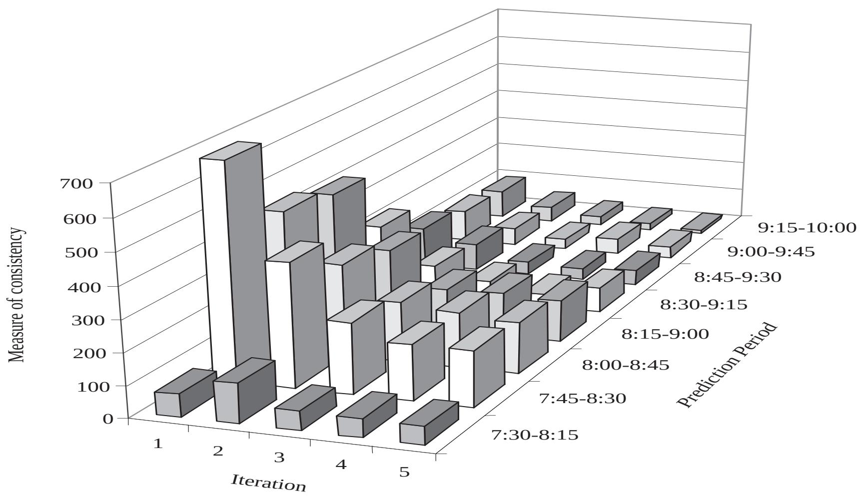


Figure 6-12: Measure of consistency of candidate route guidance

A comparison among the different scenarios of average travel times experienced and distances traveled by all vehicles is given in Table 6.4. Under normal traffic conditions, the average travel time in this network is about 6 minutes. A 20-minute incident (see Figure 6-11) caused a 1.7 minute delay and increased the average travel time by 28% in Scenario N. When drivers are given route guidance based on real-time traffic conditions, the delay is reduced. Travel time savings are about $2-3\%$ on average in both Scenarios M and P. While the guided drivers benefit more ($3-4\%$ on average) from real time information, unguided drivers also benefit ($2-3\%$ on average) because of the improved utilization of the network capacity. This delay reduction is achieved by guided drivers traveling slightly longer distances (an $1-2\%$ increase in distance traveled) using alternative routes.

Table 6.4: Comparison of average travel time and distance traveled

Performance Criteria	Driver Group	No Incident	Incident				
			w/o Guidance	w/ Guidance			
				Naive (M)		Predictive (P)	
			N	M	$M/N-1$	P	$P/N-1$
Travel Time (minutes)	All	6.06	7.77	7.52	-3.24 %	7.56	-2.69 %
	Guided			7.48	-3.79 %	7.50	-3.44 %
	Unguided			7.54	-3.01 %	7.59	-2.36 %
Distance Traveled (miles)	All	5.71	5.70	5.72	0.39 %	5.74	0.75 %
	Guided			5.79	1.54 %	5.83	2.18 %
	Unguided			5.69	-0.10 %	5.71	0.14 %

Figure 6-13 shows average travel time differences among the three scenarios as a function of departure time. The curve with circle markers is the travel time for guided drivers and the other curve is the travel time for unguided drivers. Both Figures 6-13(a) and 6-13(b) show a clear difference between travel times experienced by guided and unguided drivers. This savings in travel time for guided drivers compared to unguided drivers is more significant during the time intervals shortly after the occurrence of the incident. Also note that in the case of predictive guidance, guided vehicles are worse off at the late intervals. This may be due to the inaccuracy of predicting queue dispatching times.

Figure 6-14 (a) and (b) illustrate in more detail the travel time savings under Scenarios M and P compared with the base scenario for guided and unguided drivers respectively. These results show that in general, both guided and unguided drivers experienced shorter travel times when real-time route guidance is provided. Exceptions occur for vehicles departing during the first couple of time intervals (30 minutes into the simulation) in some of the charts in Figure 6-14. In one time interval, guided drivers under naive guidance spend significantly more time traveling than those in the base scenario (see Figure 6-14 (a)); this is probably caused by underestimation of the travel times on some paths (because the estimation is based on current measurement of traffic conditions). Under both naive and predictive guidance, unguided drivers have slightly longer travel times than in the base case in two intervals

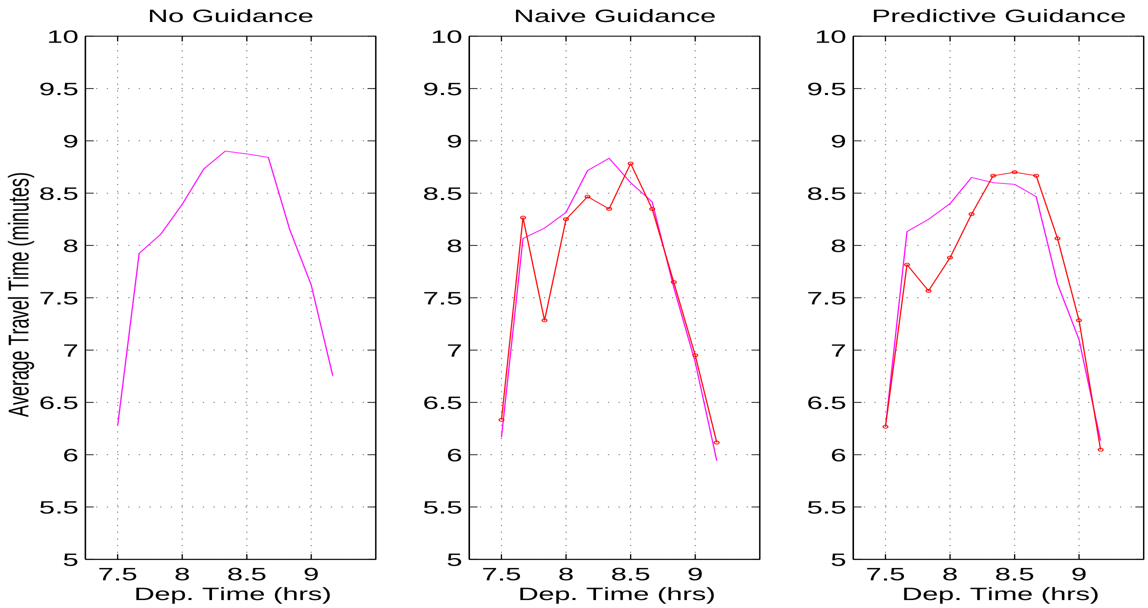


Figure 6-13: Average travel time

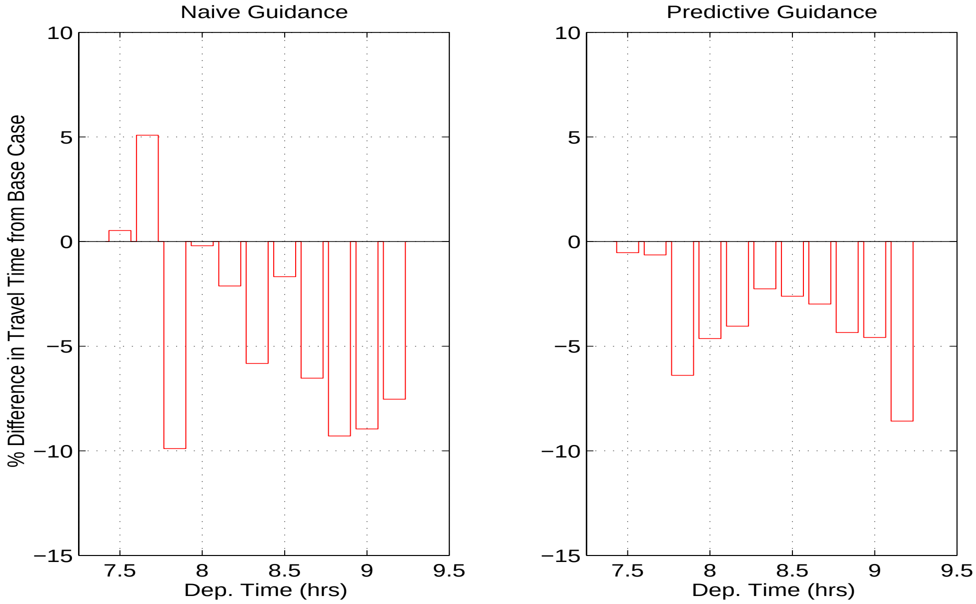
(see Figure 6-14(b)). However, these differences are insignificant and may be due to the random errors in the simulations.

Table 6.5: Comparison of average travel times for representative OD pairs

Driver Group	OD Pairs	No Incident	Incident				
			w/o Guidance	w/ Guidance			
				Naive (M)		Predictive (P)	
			N	M	$M/N-1$	P	$P/N-1$
Guided	8-14	10.0	14.7	13.5	-7.8 %	14.4	-2.3 %
	8-18	12.3	18.9	17.5	-7.1 %	17.1	-9.4 %
	8-1	7.3	20.9	17.4	-16.6 %	17.7	-15.5 %
	11-14	6.0	5.8	8.0	38.3 %	7.5	29.6 %
	11-18	9.1	9.0	12.0	33.1 %	11.1	22.4 %
	11-1	10.7	17.8	14.4	-19.0 %	14.7	-17.6 %
Unguided	8-14	10.0	14.7	13.3	-9.3 %	14.1	-4.2 %
	8-18	12.3	18.9	17.8	-5.5 %	18.2	-3.7 %
	8-1	7.3	20.9	18.0	-14.0 %	17.3	-17.4 %
	11-14	6.0	5.8	8.2	41.1 %	7.5	28.9 %
	11-18	9.1	9.0	11.8	30.6 %	10.9	21.0 %
	11-1	10.7	17.8	15.7	-11.8 %	16.1	-9.7 %

Although on average there are travel time savings when real time guidance is provided, these savings vary across OD pairs, and for certain OD pairs travel time actually increased. Figures 6-15, 6-16, and Table 6.4 illustrate the impact of information on travel time as a function of vehicles' departure time for 6 selected OD

(a) Guided Drivers



(b) Unguided Drivers

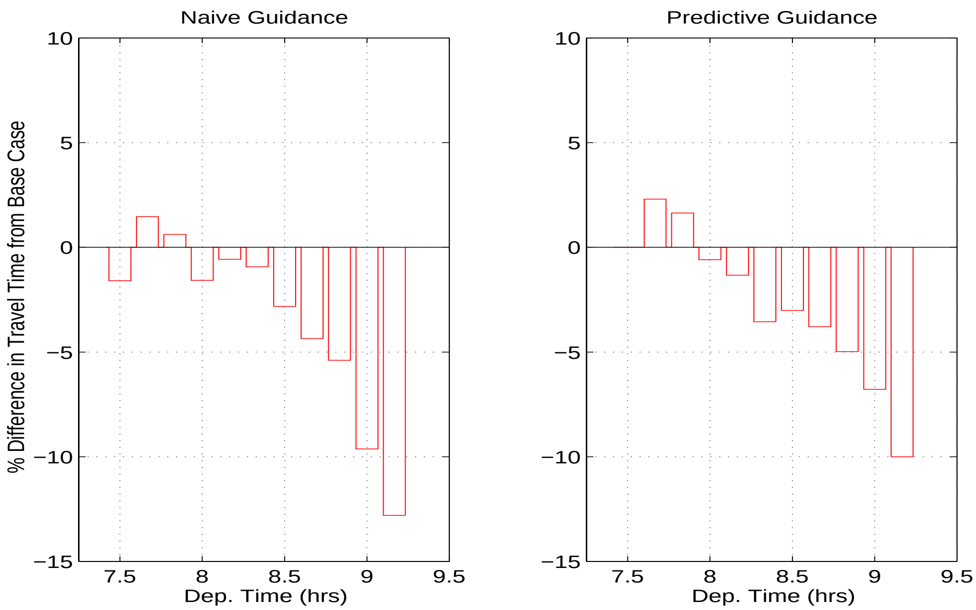


Figure 6-14: Change in average travel times compared with the base scenario by departure time interval

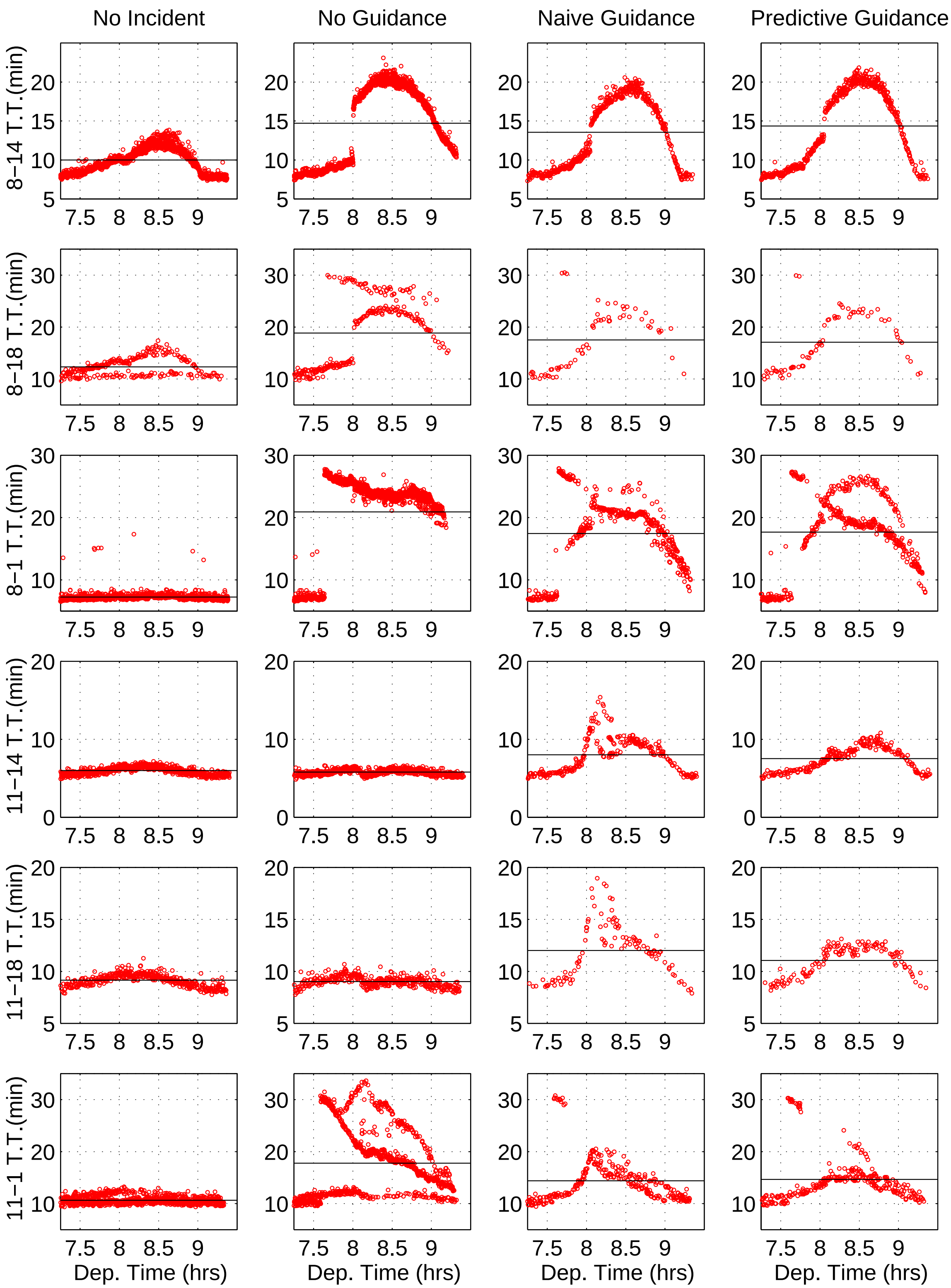


Figure 6-15: Travel time for guided vehicles of representative OD pairs

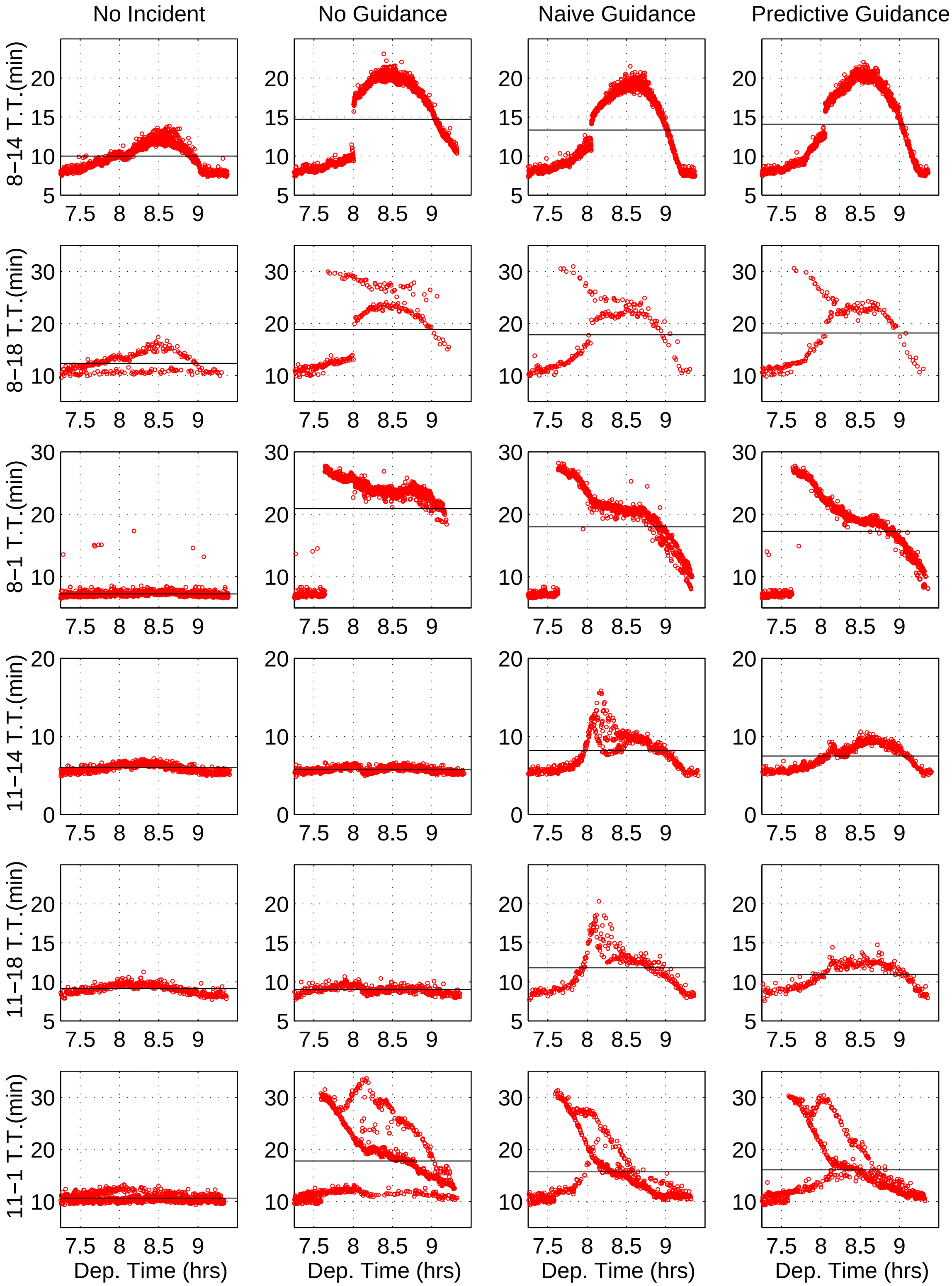


Figure 6-16: Travel time for unguided vehicles of representative OD pairs

pairs (see Figure 6-11). Each row in the figures corresponds to a particular OD pair (e.g. 8–14, 8-18, and so on) and each column to a scenario. Trip travel times for guided and unguided drivers are plotted separately in Figures 6-15 and 6-16. Each point in these figures represents a completed vehicle trip (about $64,400$ under Scenario N, and $64,900$ under Scenarios M and P). The horizontal axis is the departure time and the vertical axis the experienced travel time. The average travel time is shown by a solid horizontal line.

Vehicles from OD pairs 8–1 and 11–1 have the largest travel time savings ($15-19\%$ for guided drivers and $10-17\%$ for unguided drivers) because the incident directly affects the habitual route of most drivers from these two pairs. The alternative route is not much longer, hence guided drivers gain significant savings. Unguided drivers from these two pairs also benefit, because although they follow their habitual path (through the tunnel where the incident occurs), overall, less traffic use the route due to the switching of the guided drivers.

On the other hand, the vehicles that switched to the new path had a negative impact on some other OD pairs, especially since the alternative routes have little extra capacity. For example, vehicles from OD pairs 11–14 and 11–18 were primarily using the A10's southeastern inner loop. The diverted vehicles (which under normal conditions travel the A10's western outer loop) increased the congestion on the inner loop routes linking OD pairs 11–14 and 11–18. As a result, travel times on these OD pairs increased ($21-30\%$ under predictive guidance and $30-41\%$ under naive guidance). Since traffic in the A10 network is congested during the peak period (8:00-9:00am) and several sections operate at capacity, the increased vehicle travel time in these sections is expected and it is significant. This result shows that because users maximize their own utility while incorporating the real time guidance information, some drivers benefit while others are negatively affected. In some cases (i.e. alternative routes are already operating at their capacities), this negative impact of real-time route information on some users can offset the benefits others have achieved.

The results in Table 6.4 also highlight the effect of prediction. Travel times under naive and predictive guidance did not significantly differ for the OD pairs that experienced improvement (e.g. 8–14, 8–18, 8–1, etc.). However, for pairs that generally experienced an increase in travel time (e.g. 11–14, 11–18, etc.) the percent increase is lower under the predictive guidance scenario.

Table 6.6: Comparison of average travel time and distance traveled (with 80% demand)

Performance Criteria	Driver Group	No Incident	Incident				
			w/o Guidance	w/ Guidance			
				Naive (M)		Predictive (P)	
			N	M	$M/N-1$	P	$P/N-1$
Travel Time (minutes)	All	5.61	6.30	6.07	-3.66 %	6.05	-3.96 %
	Guided			6.04	-4.19 %	6.01	-4.65 %
	Unguided			6.09	-3.44 %	6.07	-3.66 %
Distance Traveled (miles)	All	5.71	5.71	5.74	0.41 %	5.75	0.56 %
	Guided			5.83	1.97 %	5.83	2.06 %
	Unguided			5.70	-0.26 %	5.71	-0.09 %

In order to investigate the effect of overall congestion levels on the conclusions drawn above, we now discuss the results from the application of the same scenarios but with demand uniformly reduced by 20%. Under this demand level, alternative routes do not operate at capacity.

Table 6.6 compares average travel times and distances traveled for all vehicle trips with travel demand reduced to 80%. In this case, the average travel time under normal traffic conditions is about 5.6 minutes. The incident caused a 40-second average delay and increased the average travel time by 12% in the base case (no guidance). Compared to the base case (N), the average travel time decreased by 3.4 – 4.2% under the naive guidance scenario (M) and by 3.7 – 4.7% under the predictive guidance scenario (P). As with 100% demand, the guided drivers benefit more than the unguided from the real-time information, but unguided drivers also experience some travel time savings.

In contrast to the cases with 100% travel demand, when guided drivers are given real time guidance, its negative impacts on drivers from certain OD pairs (e.g. OD pair 11–14 and 11–18) are insignificant (see Table 6.7 and Figures 6-17 and 6-18).

Table 6.7: Comparison of average travel time for representative OD pairs (with 80% demand)

Driver Group	OD Pairs	No Incident	Incident				
			w/o Guidance	w/ Guidance			
				Naive (M)		Predictive (P)	
			N	M	$M/N-1$	P	$P/N-1$
Guided	8-14	8.0	8.5	8.1	-4.8 %	8.1	-4.2 %
	8-18	10.9	13.7	12.1	-11.9 %	11.2	-17.9 %
	8-1	7.1	13.2	10.8	-18.0 %	10.8	-18.6 %
	11-14	5.5	5.5	5.5	-0.3 %	5.6	1.1 %
	11-18	8.5	8.5	8.5	0.4 %	8.5	0.9 %
	11-1	10.3	13.8	11.7	-15.1 %	11.6	-15.4 %
Unguided	8-14	8.0	8.5	8.1	-5.0 %	8.1	-4.3 %
	8-18	10.9	13.7	12.2	-10.9 %	12.5	-8.8 %
	8-1	7.1	13.2	11.5	-12.9 %	11.0	-16.6 %
	11-14	5.5	5.5	5.5	-0.1 %	5.5	0.9 %
	11-18	8.5	8.5	8.5	0.6 %	8.5	0.3 %
	11-1	10.3	13.8	13.0	-5.4 %	12.9	-6.4 %

This is due to sufficient excess capacity on the alternative routes. Furthermore, in this case, the predictive guidance has a better overall performance compared to the guidance without prediction, both in terms of vehicles that benefit and vehicles that experience higher travel times (i.e. negative impact is negligible).

6.7 Computational Performance

MITSIM and MesoTS are computationally the most expensive components in the simulation laboratory and dictate the computing time for the simulations. The former has a small simulation step size and the detailed simulations of vehicle movements are time consuming; the latter is used iteratively in TMS to generate predictive route guidance.

MITSIM: In the stand alone mode, the graphical version of MITSIM takes about 150-160 minutes CPU time to complete a 135-minute simulation in the A10 network. The time factor of the simulator, therefore, is about 1.2. When computing MOEs for the evaluation, the batch version of the simulator is used. The batch version runs about 30% faster than the graphical version and 20% faster than real time for the

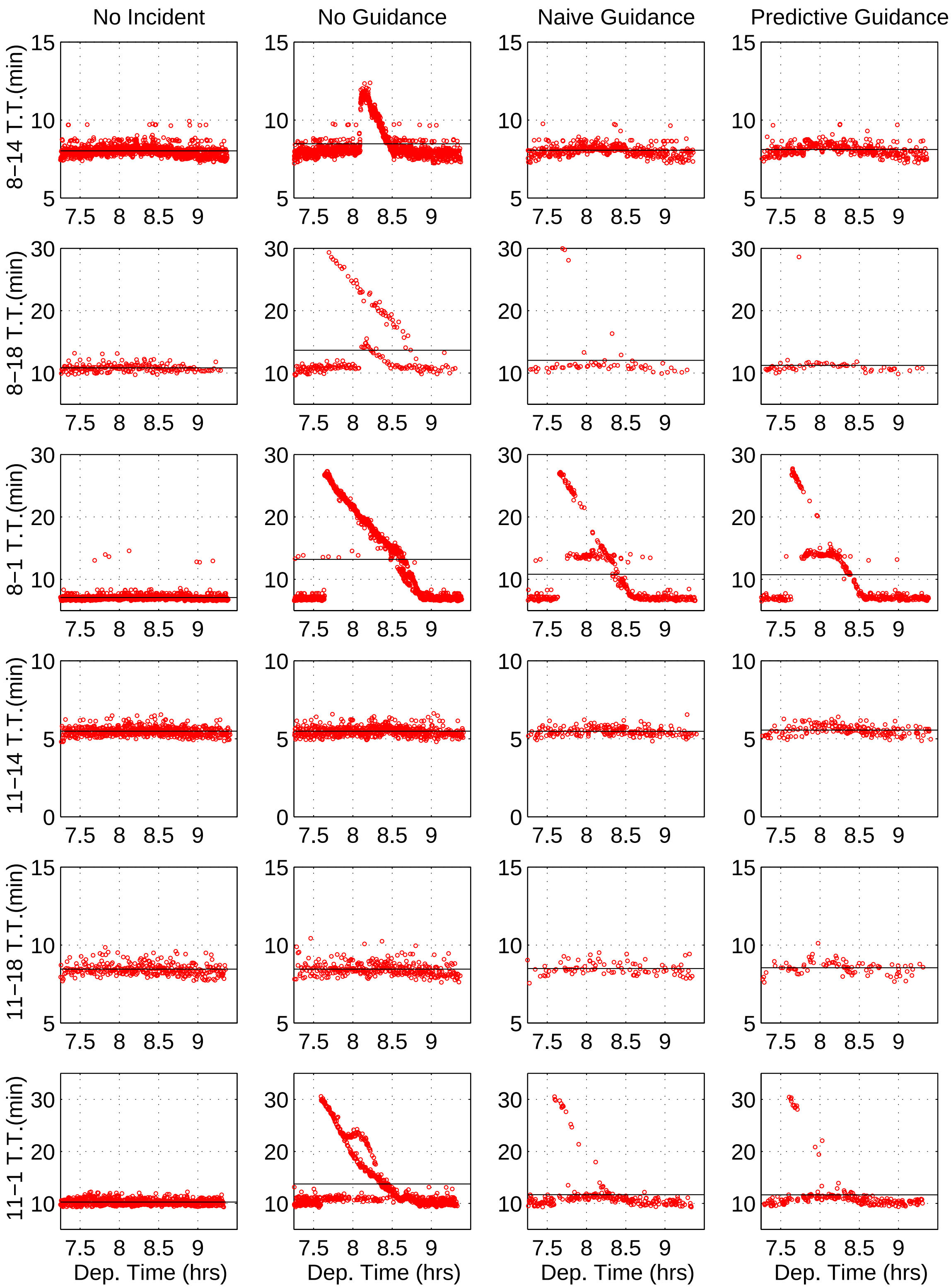


Figure 6-17: Travel time for guided vehicles of representative OD pairs (with 80% demand)

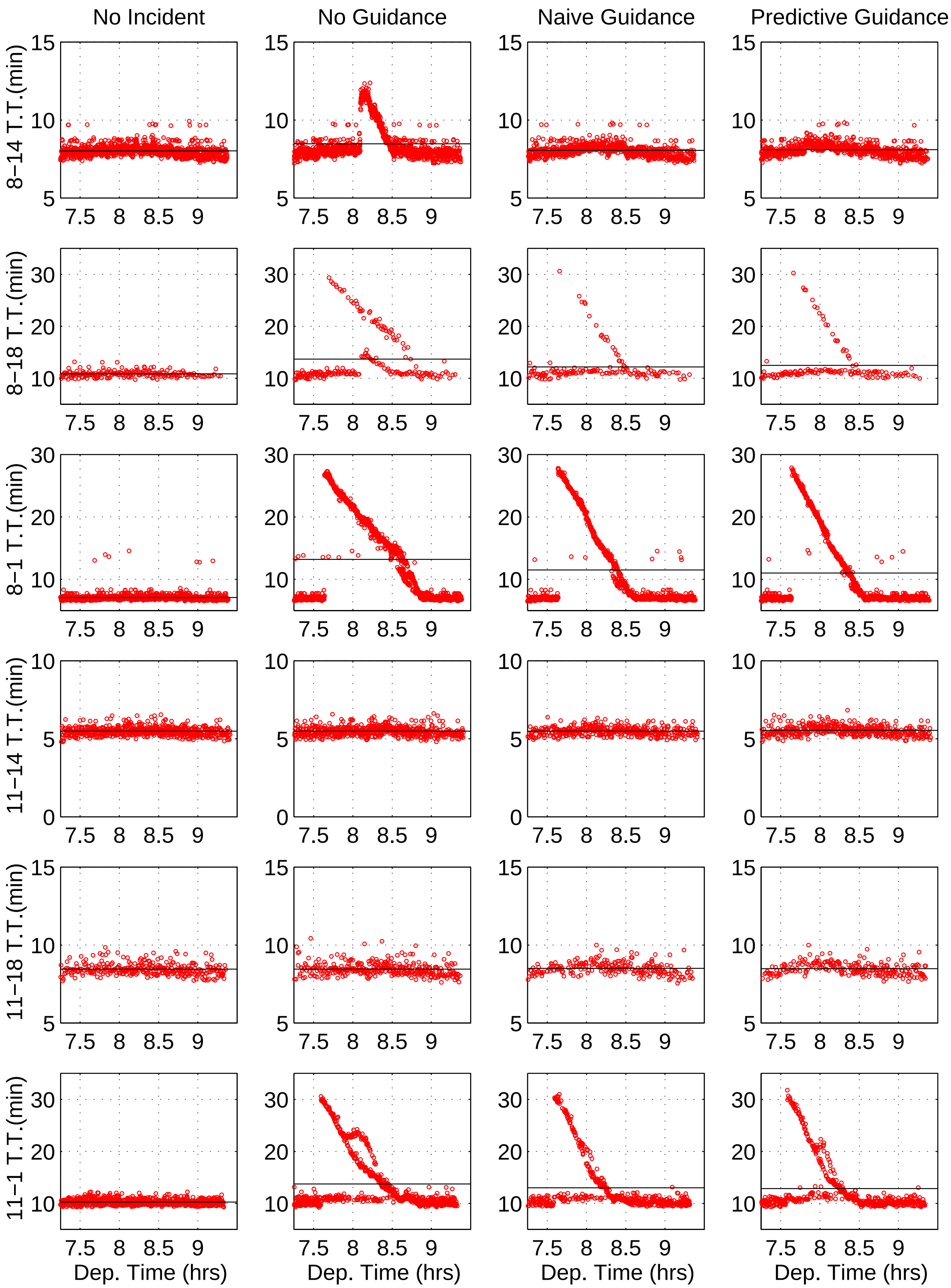


Figure 6-18: Travel time for unguided vehicles of representative OD pairs (with 80% demand)

A10 network (the time factor of the simulator is about 0.8).

Simulating scenarios with incidents is computationally more expensive because of the increased number of vehicles simultaneously in the network due to the incident induced congestion. Under Scenario N, the batch version of MITSIM takes about 120-130 minutes of CPU time to simulate 135 minutes of operations (a time factor of about 0.9).

MesoTS: The batch version of MesoTS takes about 6 minutes of CPU time in the stand alone mode to complete a 135-minute simulation (time factor less than 0.05).

SIMLAB: In the integrate mode, the CPU time taken by MITSIM and MesoTS rises slightly, but the total time used for the simulation increases significantly due to communication and idle times. The running time for simulating Scenario P is about 300 minutes using two processors (MITSIM on one computer, TMS and MesoTS on another computer), equivalent to a time factor of 2.2. This increase in computation time is due to the following:

- Every 15 minutes, MITSIM pauses and waits for new guidance from TMS, which takes about 20-25 minutes (5 MesoTS runs, each taking about 4-5 minutes). Because the iterations of MesoTS must be conducted in sequence, the distributed processing does not provide a great advantage.
- Communication delay.

All computational performance measures described above is obtained on SGI Indy R4400 workstations with 200 MHZ of speed and 64 MB of RAM.

6.8 Conclusion

In conclusion, the results of the case study show that the simulation laboratory operates as designed. In the A10 network with non-recurrent congestion caused by a 20-minute incident, the case study demonstrates that on average $2-4\%$ of travel time savings is achieved when real-time traffic information is provided to 30% of drivers.

For drivers having viable alternative routes, real time route guidance is very effective, creating travel time savings of up to 18%. In addition, the system that provides predictive route information performs better overall than the one that provides route information according to current traffic conditions. In addition, the computational time was reasonable and allowed for testing a large number of scenarios and cases.

Chapter 7

Conclusion

7.1 Research Contribution

The main contribution of this research lies in three areas: (1) the creation of a computer-based traffic simulation laboratory capable of evaluating dynamic traffic management systems; (2) the development of a prototypical "real-time" route guidance system that provides predictive routing information; and, (3) the demonstration of the use of the developed simulation laboratory and the value of real-time traffic information in a considerable-sized real network. By modeling driver behavior, providing traffic prediction capability, and supporting a wide range of traffic surveillance and control systems, the laboratory can evaluate integrated ATMS and ATIS operations in integrated traffic networks.

The main elements of the traffic simulation laboratory are a microscopic traffic flow simulator (MITSIM) and a traffic management simulator (TMS). Because MITSIM models the vehicular flows at a microscopic level, it can compute the performance measures necessary for a detailed system evaluation. It represents the "reality" that occurs in the traffic network and can provide TMS with surveillance information necessary for developing and testing ITS applications. TMS can simulate a wide range of traffic control and route guidance strategies using a generic structure that can represent both reactive and proactive systems. Such strategies include the traditional technologies (e.g., UTCS type of signal controls and freeway ramp metering) and dynamic traffic management strategies such as adaptive signal controls and predictive route guidance. The system is distributed and features a friendly graphical user interface. The simulation laboratory provides a unique tool for off-line system evaluation and benefit assessment of ATMS and ATIS at an operational level.

To demonstrate the use of the simulation laboratory, a predictive route guidance system has been implemented in TMS. A mesoscopic traffic simulator (MesoTS) was used to predict traffic congestion. Simulation-based traffic prediction has been identified in the literature as an important alternative in generating anticipatory route guidance and has been adopted in several DTA systems (FHWA, 1995; MIT, 1996; Mahmassani et al., 1994). The results of this SIMLAB application demonstrates the promise of simulation-based traffic prediction.

The developed simulation laboratory and predictive route guidance were tested on the Amsterdam A10 beltway. The evaluation under a representative scenario (with 30% guided drivers and non-recurrent congestion caused by a 20-minute incident) demonstrates that both guided and unguided drivers may benefit from real-time information. Three scenarios were tested: no guidance, guidance based on current traffic conditions (naive guidance), and anticipatory "consistent" guidance (predictive guidance). Compared to the case without guidance (drivers choose routes based on historical information), on average drivers save $2-4\%$ of travel time when predictive route information is provided (to guided drivers). These savings vary across OD pairs and departure times. While drivers from some OD pairs saved up to 19% of travel time compared to the "no guidance" scenario, drivers from certain OD pairs actually experienced increased travel time (up to 40% under naive guidance and 30% under predictive guidance) because of increased traffic caused by the diverted drivers from OD pairs directly impacted by the incident. However, this negative impact is not observed when travel demand is reduced to 80%. This shows that excess capacity on alternative routes is a necessary condition to improve the performance of the system with provision of real-time route guidance. This case study also shows that, although MeosTS is still limited in terms of its accuracy in traffic prediction, predictive route guidance has the potential to improve traffic conditions. This result is not surprising and furthermore is similar to that found by Kaysi (1992). However, this case study is significant. It is based on a realistic network and a general methodology that can be applied to networks of varying complexity. This application also differs in its approach from the work done by Mahmassani and Peeta (1995). Their evaluation results were obtained by using the same model (mesoscopic) that generates the guidance. In contrast, this case study separates system evaluation from guidance generation and the evaluation is based on microscopic traffic simulation.

It should be noted that the main purpose of the case study was to demonstrate the applicability of the simulation laboratory. Since many input parameters had not been calibrated and the experimental design was not complete, the results of case study were only indicative and should be interpreted with caution.

In conclusion, the methodology and simulation tools developed in this research can be used for evaluating other ATIS and ATMS strategies in general networks comprised of both freeways and urban streets. Because of the software's modular structure and an object-oriented programming approach, various SIMLAB default models can be easily substituted with user specified models and new components can be added as required by particular applications.

7.2 Future Work

The research presented in this dissertation can be extended in the following directions:

Validation and calibration

Both MITSIM and MesoTS used a rich set of parameters to represent drivers' behavior, vehicle performance, and the characteristics of road networks. These parameters provide the flexibility to customize the laboratory for use in various environments. However, many of the parameters should be calibrated based on field traffic data. Furthermore, the travel behavior models used in the simulators (e.g. route choice and route switching models) have not been calibrated. As the state of the art in the area of driver response to information advances, better and more appropriately calibrated models can replace the default models used in SIMLAB.

The validation and calibration of the traffic simulation models require collection of empirical data at various levels of aggregation. For example, vehicle trajectory data are necessary for estimating parameters such as those in car-following, lane-changing models, etc. (Ahmed et al., 1996; Subramanian, 1996; Benekohal, 1986). Sensor data such as flow, speed, occupancy, and headway are needed to validate the simulation model as a whole and to identify the defects of simulation models (see, for example, Cassidy, 1995). Various approaches and techniques for validation of simulation models are described by Law and Kelton (1991) and Neelamkavil (1986).

Methodology for design refinement and optimization

When designing a traffic management system, SIMLAB can be used as a tool to understand the "behavior" of the system and to assist in refining the design (see Guo, 1997, for a lane control system evaluation example). Using the developed SIMLAB, for example, researchers can do factorial experimental design and construct response surface to analyze the relationship between the system's input parameters (or structural assumptions) and the output (i.e., response). The "input-response" relationships obtained from well designed simulation experiments can be useful in choosing the system's optimal design parameters (see Law and Kelton (1991) for a thorough discussion on experimental design and optimization using simulation). Further work, however, is necessary to identify the controllable parameters to which the system is most sensitive, efficient and systematic experimental design, and guidelines for effective use of SIMLAB in optimization and design refinement.

Applications

The evaluation methodology and simulation tools developed in this research provide a unique laboratory environment for testing and evaluating different designs and alternative ITS technologies (see Law and Kelton (1991), pg. 582-611, for methods of comparing alternative system configurations). Researchers and engineers can also use it to test new ideas and obtain a better understanding of algorithms and system parameters. For example, the simulation laboratory can be used in the following ATMS and ATIS applications:

Evaluation of alternative dynamic traffic assignment (DTA) models: DTA research is expected to produce models that generate predicted traffic conditions for real-time traffic control and route guidance (MIT, 1996; Mahmassani et al., 1994). SIMLAB can serve as a test-bed for calibrating the DTA model components and evaluating alternative designs and approaches.

Evaluation of various traffic management strategies: Many traffic management strategies, such as adaptive signal controls, ramp and mainline metering, incident detection and management (using LUS, VSLS, VMS, etc.), dedicated lane assignment (HOV, ETC, etc.), reversible lanes, etc, can be evaluated using the developed framework and simulation tools.

Operator training: Operators in a traffic management center (TMC) monitor and supervise the operation of the traffic system. Their prompt and correct responses to certain situations are essential for appropriate functioning of the system. SIMLAB can be used in developing "operator-in-the-loop" environment for training operators to manage real time traffic control systems (Chao, 1994).

Laboratory travel behavior studies: Improved driving simulators are currently used in data collection for understanding travel behavior related to provision of real-time traffic information (Bonsall, 1995; Koutsopoulos et al., 1994). SIMLAB can be used to simulate the background traffic in the network and a "realistic" driving environment. Therefore, more reliable data on safety and human factors (such as response time and driver's ability to process viewed information) can be collected. Furthermore, such a driving simulator can also function as a tool to collect data necessary for validating and calibrating various models of the microscopic traffic simulator itself.

Several models have been proposed to describe drivers' dynamic travel choice behavior with respect to departure time choices (de Palma et al., 1983; Ben-Akiva et al., 1984; Vythoulkas, 1990) and route choices (Cascetta, 1989; Cascetta and Cantarella, 1991; Cantarella and Cascetta, 1995). Simulation studies have been conducted to analyze various information strategies (Jayakrishnan et al., 1995), route choice behavior (Lotan and Koutsopoulos, 1993), and information perception (Jha et al., 1996) in the presence of real-time information. SIMLAB can be used to study day-to-day dynamic travel behavior in a laboratory environment. An example of such an application is the demand simulation for dynamic traffic assignment by Antoniou (1997).

Enhancement of SIMLAB with additional components

The following components can be added into the simulation laboratory to enhance its capability:

Network state estimation: In the current version of SIMLAB, the default traffic prediction model uses the actual network state obtained from MITSIM. Although this approach is appropriate for evaluating the sensitivity of traffic prediction and guidance generation to the accuracy of network state estimation, a network state estimator based on surveillance data will enhance the capabilities of SIMLAB and make simulation of traffic management systems more realistic.

OD estimation and prediction: Time-variant travel demand is a fundamental input to SIMLAB. However, in most applications, time-dependent OD data is not readily available. Methods for estimating and predicting time-dependent OD flows have been proposed by several researchers such as Cascetta et al. (1993), van der Zijpp (1996), and Ashok (1996). Evaluating these methods and developing an OD estimation and prediction module, as the default option in the simulation laboratory, would be helpful for using SIMLAB in real world applications.

Capacity translator: Traffic prediction methods based on MesoTS or similar models often assume that capacities of link/segments and turning movements at intersections are known in advance or can be estimated in real-time. However, the output of traffic control systems is usually the specification of state transition of traffic signals and signs. To evaluate these traffic control systems, a capacity translator is needed to convert the signal control into the throughput capacity.

Fuel consumption and emissions models: The microscopic simulation approach in MITSIM allows vehicle performance and attributes to be represented in detail. It can also provide detailed vehicle trajectory data. This data can then be used in calculating fuel consumption and emissions.

Software improvements

Although the simulation laboratory has been implemented and tested in several networks of various sizes and complexity, from a software development point of view the system is still at an early stage of development. Input, output, and interface with user-developed external modules need to be standardized. There is also need and potential for improvement of the laboratory's computational performance. Approaches that can help the computational performance of SIMLAB include:

Hybrid simulation: MITSIM and MesoTS can be integrated into a hybrid traffic flow simulator to facilitate simulation of large networks. One approach for integrating these two simulators is to define *Micro links* in the parts of the network where more detail is needed and *macro links* in the parts where less detail is needed. Vehicles in micro links can be modeled based on the microscopic simulation logic used in MITSIM; those in macro links can be modeled using the traffic cells and traffic stream concepts developed in MesoTS. When crossing the boundary between a micro and macro link, a vehicle is switched from one type to another.

A similar approach has been proposed by Jayakrishnan and Rindt (1996) for integrating INTRAS and DYNASMART. However, much less work would be needed for integrating MITSIM and MesoTS, because these two simulators were developed using the same network representation and implemented using object-oriented programming.

Distributed and parallel simulation: Jha et al. (1995) proposed a framework for distributed traffic simulation and described the possibility to parallelize DYNASMART. Junchaya et al. (1992) reported a traffic simulation work developed on a massive parallel machine. These efforts represent two possible approaches of parallelism. In the first approach, a large network is decomposed into several smaller subnetworks, and simultaneous simulations on multiple processors are conducted for these subnetworks. In the second approach, the simulator is implemented on a massively parallel computing architecture, such as the Connection Machine by Thinking Machine Corp [1].

[1]A CM-5 system, for example, contains tens, hundreds or even thousands of processing nodes, each with up to 128 Mflps of 64-bit of floating-point performance and 8, 16, or 32 Mbytes of memory. See Thinking Machine Corporation.

Bibliography

K. Ahmed, M. Ben-Akiva, H. Koutsopoulos, and R. Mishalani. Models for freeway lane changing and gap acceptance behavior. In *Proceedings of 13th International Symposium on Transportation and Traffic Theory*, Lyon France, 1996.

R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, Englewood Cliffs, New Jersey, 1993.

C. Antoniou. Demand simulation for dynamic traffic assignment. Master's thesis, Massachusetts Inst. of Tech., Cambridge, MA, 1997.

K. Ashok. *Estimation and prediction of time-dependent origin-destination flows*. PhD thesis, Massachusetts Inst. of Tech., Cambridge, MA, 1996.

K. Ashok and M. E. Ben-Akiva. Dynamic O-D matrix estimation and prediction for real-time traffic management systems. In C. F. Daganzo, editor, *Transportation and Traffic Flow Theory*, pages 465–484. Elsevier Science Publishing Company Inc., 1993.

J. Barcelo, J. Ferrer, D. Garcia, R. Grau, M. Fiorian, I. Chabini, and E. L. Saux. Microscopic traffic simulation for ITS analysis. Contribution to the 25th Anniversary of Centre de Recherche sur les Transports, Universite de Montreal, 1996.

J. Barcelo and J. L. Ferrer. A simulation study for an area of Dublin using the AIMSUN2 traffic simulator. Technical report, Department of Statistics and Operation Research, Universitat Politecnica de Catalunya, Spain, 1995.

M. E. Ben-Akiva, M. Bierlaire, J. Bottom, H. N. Koutsopoulos, and R. G. Mishalani. Development of a route guidance generation system for real-time application, 1997. to be presented at 8th IFAC Symposium on Transportation Systems.

M. E. Ben-Akiva, M. Bierlaire, H. N. Koutsopoulos, and R. G. Mishalani. DynaMIT: Dynamic network assignment for the management of information to travelers. In *Proceedings of the 4th Meeting of the EURO Working Group on Transportation*, Newcastle, 1996.

M. E. Ben-Akiva, M. Cyna, and A. de Palma. Dynamic models of peak period traffic congestion. *Transportation Research*, 18(4):339–355, 1984.

M. E. Ben-Akiva, A. F. Hotz, R. G. Mishalani, and N. R. V. Jonnalagadda. A design-evaluation framework for dynamic traffic management systems using simulation. In *WCTR Proceedings*, Sydney, Australia, 1995.

M. E. Ben-Akiva, H. N. Koutsopoulos, and A. Mukandan. A dynamic traffic model system for ATMS/ATIS operations. *IVHS Journal*, 1(4), 1994.

R. F. Benekohal. *Development and validation of a car following model for simulation of traffic flow and traffic wave studies at bottlenecks*. PhD thesis, The Ohio State University, Columbus, Ohio, 1986. UMI Dissertation Services.

P. W. Bonsall. Simulator experiments and modeling to determine the influence of route guidance on drivers' route choice. In *Proceedings of the World Congress on Applications of Transport Telematics and Intelligent Vehicle-Highway Systems – Towards an Intelligent Transport System, Vol. 2*, pages 964–972, Boston, 1995. Artech House.

S. Bowcott. The ADVANCE project. In S. Yagar and A. Santiago, editors, *Large Urban Systems – Proceedings of the Advanced Traffic Management Conference*, pages 59–70, St. Peterburg, Florida, 1993.

D. Bretherton. SCOOT current developments: version 3. In *Proceedings of TRB 75th Annual Meeting*, Washington, D.C., 1996.

C. Cantarella and E. Cascetta. Dynamic process and equilibrium in transportation network: towards a unifying theory. *Transportation Science*, 25A(4):305–329, 1995.

E. Cascetta. A stochastic process approach to the analysis of temporal dynamics in transportation networks. *Transportation Research*, 23B(1):1–17, 1989.

E. Cascetta and C. Cantarella. A day-to-day and within day dynamic stochastic assignment model. *Transportation Research*, 25A(5):277–291, 1991.

E. Cascetta, D. Inaudi, and G. Marquis. Dynamic estimators of origin-destination matrices using traffic counts. *Transportation Science*, 27(4):363–373, 1993.

E. Cascetta, A. Nuzzolo, and L. Giggiero. Analysis and modeling of commuters departure time and route choices in urban networks. In *Proceedings of the Second International CAPRI Seminar on Urban Traffic Networks*, Capri, Italy, 1992.

E. Cascetta, A. Nuzzolo, F. Russo, and A. Vitetta. A modified logit route choice model overcoming path overlapping problems: Specification and some calibration results for interurban networks. In *Proceedings of 13th International Symposium on Transportation and Traffic Theory*, Lyon France, 1996.

M. J. Cassidy. Validation and evaluation of freeway simulation models. Technical Report Final Report to US DOT, Purdue University, West Lafayette, IN, 1995.

D. L. Cather. Traffic control strategy. Design Document Volume 6, Parsons Transportation Group, De Leuw, Cather, 1996.

G. Chang and J. Wu. Recursive estimation time-varying O-D flows from traffic counts in freeway corridors. *Transportation Research*, 28B(2):141–160, 1994.

T. C. Chao. Human factor in the design of display for traffic operations control centers. Master's thesis, Massachusetts Inst. of Tech., Cambridge, MA, 1994.

H. K. Chen and C. F. Hsueh. Combining signal timing plan and dynamic traffic assignment. 76th Transportation Research Board Annual Meeting, 1997.

O. J. Chen. A dynamic traffic control model for real time freeway operations. Master's thesis, Massachusetts Inst. of Tech., Cambridge, MA, 1996.

D. J. Clowes. Real-time wide area traffic control – the user's viewpoint. *Traffic Engineering & Control*, pages 194–196, 1983.

D. K. Codelli, W. P. Niedringhaus, and P. Wang. The Traffic and Highway Objects for REsearch, Analysis, and Understanding (THOREAU) IVHS model, Vol. I. Technical Report MTR 92W208V1, MITRE, McLean, Virginia, 1992.

A. de Palma, M. Ben-Akiva, C. Lefevre, and N. Litinsa. Stochastic equilibrium model of peak period traffic congestion. *Transportation Science*, 17(253-274), 1983.

A. de Palma, F. Marchal, and Y. Nesterov. METROPOLIS: A modular system for dynamic traffic simulation. http://www.ceic.com/metro/, 1996.

C. Diakaki and M. Papageorgiou. Design and simulation test of coordinated ramp metering control (METALINE) for A10-west in Amsterdam. ATT-Project Eurocor V2017, Technical University of Crete, Department of Production Engineering and Management, Chania, Greece, 1995.

R. B. Dial. A probabilistic multipath traffic assignment algorithm which obviates path enumeration. *Transportation Research*, Vol. 5(2):83–111, 1971.

C. Donnelly and R. Stallman. *Bison – The YACC-compatible parser generator.* the Free Software Foundation, Cambridge, MA, 1992.

DYNA. DYNA – A dynamic traffic model for real-time applications – DRIVE II project. Annual review reports and deliverables, Commision of the European Communities - R&D programme telematics system in the area of transport, 1992-1995.

F. Eskafi, D. Khorramabadi, and R. Varaiya. An automated highway system simulator. *Transportation Research*, Vol. 3(1):1–17, 1996.

FHWA. Development and testing of INTRAS: a microscopic freeway simulation model – Vol. 1-3. Report FHWA/RD-80/106-108, Federal Highway Administration, US-DOT, McLean, Virginia, 1980.

FHWA. FRESIM User Guide. Technical Report Version 4.5, Federal Highway Administration, US-DOT, McLean, Virginia, 1994.

FHWA. DTA RFP. Technical report, Federal Highway Administration, US-DOT, McLean, Virginia, 1995.

FHWA. CORSIM User Guide. Technical Report Version 1.0, Federal Highway Administration, US-DOT, McLean, Virginia, 1996.

N. Gartner, S. Gershwin, J. Little, and P. Ross. Pilot study of computer-based urban traffic management. *Transportation Research*, Vol. 14B:203–217, 1980.

N. H. Gartner. OPAC: A demand-responsive strategy for traffic signal control. *Transportation Research Record*, (906):75–81, 1983.

N. H. Gartner and C. Stamatiadis. Integration of dynamic traffic assignment with real time traffic adaptive control. 76th Transportation Research Board Annual Meeting, 1997.

N. H. Gartner, C. Stamatiadis, and P. J. Tarnoff. Development of advanced traffic signal control strategies for IVHS: A multi-level design. *Transportation Research Record*, (1494), 1995.

A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek, and V. Sunderam. *PVM: Parallel Virtual Machine – A users' guide and tutorial for networked parallel computing*. The MIT Press, Cambridge, MA, 1994.

S. B. Gershwin, J. D. C. Little, and N. Gartner. Computer-assisted traffic engineering using assignment, optimal signal setting, and modal split. Final report prepared for US DOT DOT-TSC-RSPA-78-10, Massachusetts Inst. of Tech., Cambridge, MA, 1978.

J. F. Gilmore, S. P. Roth, H. C. Forbes, and K. A. Payne. ATMS universal traffic operation simulation. 5th IVHS America Annual Meeting, Washington, D.C., 1995.

P. G. Gipps. A model for the structure of lane changing decisions. *Transportation Research*, Vol. 20B(5):403–414, 1986.

A. Gollu and P. Varaiya. SmartAHS: An object oriented simulation framework for highway systems. Path,University of California at Berkeley, 1996.

W. D. Guo. Evaluation of the effectiveness of lane control systems using microsimulation. Master's thesis, Massachusetts Inst. of Tech., Cambridge, MA, 1997.

K. A. Haboian. A case for freeway mainline metering. 74th Transportation Research Board Annual Meeting, Washington, D.C., 1995.

H. Haj-Salem, N. Elloumi, S. Mammar, M. P. J. Chrisoulakis, and F. Middelham. METACOR: A macroscopic modelling tool for urban corridor. In *Proceedings of the First World Congress on Applications of Transport Telematics and Intelligent Transportation Systems*, Paris, 1994.

Y. E. Hawas and H. S. Mahmassani. A decentralized scheme for real-time route guidance in vehicular traffic networks. In *Proceedings of the Second World Conference on ITS*, volume Vol. IV, Yokohama, 1995.

HCM. *Highway Capacity Manual.* Transportation Research Board, 1985. Special Report 209.

K. L. Head, P. B. Mirchandani, and D. Sheppard. A hierarchical framework for real-time traffic control. 71st Transportation Research Board Annual Meeting, Washington, D.C., 1992.

B. Hellinga and M. van Aerde. Examining the potential of using ramp metering as a component of an ATMS. 74th Transportation Research Board Annual Meeting, Washington, D.C., 1995.

R. Herman, E. W. Montroll, R. Potts, and R. W. Rothery. Traffic dynamics: Analysis of stability in car-following. *Operation Research*, Vol. 1(7):86–106, 1959.

R. Herman and R. W. Rothery. Car-following and steady state flow. In *Theory of Traffic Flow Symposium Proceedings*, pages 1–13, 1963.

P. B. Hunt, D. I. Robertson, R. D. Bretherton, and R. L. Winton. SCOOT: A traffic responsive method of coordinating signals. *TRRL Report*, (1014), 1981. Transport and Road Research Laboratory, Crowthorne.

ITE. *Transportation and Traffic Engineering Handbook.* Prentice-Hall, Inc., Englewood Cliffs, N.J., 2nd edition, 1982. Institute of Transportation Engineers.

R. Jayakrishnan, H. Mahamassani, and T. Hu. An evaluation tool for advanced traffic information and management systems in urban networks. *Transporation Research C*, Vol. 2C(3):129–147, 1995.

R. Jayakrishnan and C. R. Rindt. A distributed computing platform and hybrid simulation environment in a real-time research testbes for advanced traffic management and information systems. Civil and Environmental Engineering, University of California, Irvine, 1996.

M. Jha, A. Joshi, and K. Sinha. A framework for dynamic traffic simulation on distributed systems. In *Proceedings of applications of advanced technologies in transportation*, Capri, Italy, 1995.

M. Jha, S. Madanat, and S. Peeta. Day-to-day travel choice dynamics and perception updating in traffic networks with information provision. Submited for publication in Transportation Research C, 1996.

T. Junchaya, G. Chang, and A. Santiago. ATMS: Real-time network traffic simulation methodology with a massively parallel computing architecture. In *Proceedings of 71st TRB Annual Meeting*, Washington, D.C., 1992.

I. Kaysi. *Framework and models for the provision of real-time driver information.* PhD thesis, Massachusetts Inst. of Tech., Cambridge, MA, 1992.

I. Kaysi, M. E. Ben-Akiva, and H. N. Koutsopoulos. Integrated approach to vehicle routing and congestion prediction for real-time driver guidance. *Transporation Research Record*, 1408, 1993.

H. Koutsopoulos, T. Lotan, and Q. Yang. A driver simulator for data collection and its application to the route choice problem. *Transportation Research C*, 2C(2), 1994.

H. N. Koutsopoulos and Q. Yang. Modeling requirements for emerging transportation system operating environments. Report DTRS-57-88-C-0078, TTD 29, John A. Volpe National Transportation Systems Center, Cambridge, MA, 1992.

A. M. Law and W. D. Kelton. *Simulation Modeling and Analysis.* McGraw-Hill, Inc., 2nd edition, 1991.

J. D. Leonard II. Demonstration of prototyping tools for atms design. In S. Yagar and A. Santiago, editors, *Large Urban Systems – Proceedings of the Advanced Traffic Management Conference*, pages 35–46, St. Peterburg, Florida, 1993.

J. R. Levine, T. Mason, and D. Brown. *Lex & Yacc.* O'Reilly & Assoc, Cambridge, MA, 2nd edition, 1992.

F.-B. Lin. Need for improved evaluation models for signal coordination in adaptive control. In S. Yagar and A. Santiago, editors, *Large Urban Systems – Proceedings of the Advanced Traffic Management Conference*, pages 151–156, St. Peterburg, Florida, 1993.

T. Lotan and H. Koutsopoulos. Approximate reasoning models for route choice behavior in the presence of information. In C. Daganzo, editor, *Proceedings of the 12th international symposium on transporation and traffic theory (ISTS)*, pages 71–88, Berkeley, CA, 1993.

H. S. Mahmassani, T.-Y. Hu, S. Peeta, and A. Ziliaskopoulos. Development and testing of dynamic traffic assignment and simulation procedures for atis/atms applications. Report DTFH61-90-R-00074-FG, U.S. DOT, Federal Highway Administration, McLean, Virgina, 1994.

H. S. Mahmassani and S. Peeta. Network performace under system optimal and user equilibrium dynamic assignments: Implications for advances traffic information systems. *Transportation Research Record*, (1408):83–93, 1995.

I. MathWorks. *The student version of MATLAB - version 4 User Guide.* Prentice Hall, Englewood Cliffs, New Jersey, 1995.

A. May. *Traffic Flow Fundamentals.* Prentice Hall, Englewood Cliffs, New Jersey, 1990.

A. May and H. E. Keller. Non-integer car-following models. *Highway Research Board*, (199):19–32, 1967.

A. Messmer and M. Papageorgiou. METANET: A macroscopic simulation progrm for motorway networks. *Traffic Engineering & Control*, 31(8/9):466–477, 1990.

P. G. Michalopoulos and R. Plum. KRONOS-4: Final report and user's manual. Minnesota Department of Transportation, 1986.

F. Middelham, W. J. Schouten, J. Chrisoulakis, M. Papageorgiou, and H. Haj-Salem. Eurocor and a10-west – coordinated ramp metering near amsterdam. In *Proceedings of the First World Congress on Applications of Transport Telematics & Intelligent Vehicle-Highway Systems*, pages 1158–1165, Paris, France, 1994a.

F. Middelham, T. Wang, R. Koeijvoets, and H. Taale. FLEXSYT-II: Manual – Vol. 1-2. Technical report, Transport Research Center (AVV), Ministry of Transport, Public Works and Water Management, the Netherlands, 1994b.

MIT. Development of a deployable real-time dynamic traffic assignment system. Technical Report Task B-C, Massachusetts Inst. of Tech., Intelligent Transportation Systems Program and Center for Transportation Studies, Cambridge, MA, 1996. Interim reports submitted to the Oak Ridge National Laboratory.

F. Neelamkavil. *Computer Simulation and Modelling.* John Wiley & Sons, 1986.

M. Papageorgiou. A new approach to time-of-day control based on a dynamic freeway traffic model. *Transportation Research*, Vol. 14B:349–360, 1980.

M. Papageorgiou, H. Haj-Salem, and J.-M. Blosseville. ALINEA: A local feedback control law for on-ramp metering. *Transportation Research Record*, 1320:58–64, 1990.

M. Papageorgiou, H. Haj-Salem, and F. Middelham. ALINEA local ramp metering: summary of filed results. 76th Transportation Research Board Annual Meeting, Washington, D.C., 1997.

S. K. Park and K. W. Miller. Random number generators: Good ones are hard to find. *CACM*, 31(10):1192–1201, 1988.

H. J. Payne. Freeway traffic control and surveillance model. *Trasportation Engineering Journal*, 99(TE4):767–783, 1973.

H. J. Payne. FREFLO: A macroscopic simulation model of freeway traffic: Version 1 – User's guide. Technical report, ESSOR Report, 1978.

K. Petty, H. Noeimi, K. Sanwal, D. Rydzewski, A. Skabardonis, P. Varaiya, and H. Al-Deek. The freeway service patrol evaluation project: database, support programs, and accessibility. http://www.path.berkeley.edu/FSP/, 1996.

R. S. Pindyck and D. L. Rubinfeld. *Econometric models and economic forecasts.* McGraw-Hill, Inc., 3rd edition, 1991.

J. L. Pline. *Traffic Engineering Handbook.* Prentice Hall, Englewood Cliffs, New Jersey, 4th edition, 1992.

R. A. Reiss and N. H. Gartner. Dynamic control and traffic performance in a freeway corridor: A simulation study. *Transportation Research*, Vol. 25A(5):267–276, 1991.

A. J. Santiago and A. Kanaan. ATMS laboratories: A requirement for program delivery. The 3rd Annual Meeting of IVHS America, 1993.

C. L. Saricks, J. L. Schofer, S. Soot, and P. A. Belella. Evaluating effectiveness of a real-time ATIS using a small test vehicle fleet. In *Proceeding TRB 76th Annual Meeting*, 1997. Paper No: 970585.

J. L. Schofer, F. S. Koppelman, R. G. Webster, S. Berka, and T. shiou Peng. Field test of the effectiveness of ADVANCE dynamic route guidance on a suburban arterial street network. ADVANCE Project Document #8463.01, Northwest University, Transportation Center, 1996.

Y. Sheffi, H. Mahmassani, and W. B. Powell. A transportation network evacuation model. *Transportation Research*, Vol 16A(3), 1982.

S. Shepherd. Metering strategies applied to signalized networks. In *Proceedings of the ASCE Third International Conference on Applications of Advanced Technologies in Transportation Engineering*, 1993.

S. W. Sibley. NETSIM for microcomputers – simulates microscopic traffic flow on urban streets). *Public Roads*, 49, 1985.

S. A. Smith. Freeway data collection for studying vehicle interactions. Technical Report FHWA/RD-85/108, Federal Highway Administration, Office of Research, Washington D. C., 1985.

S. A. Smith and C. Perez. Evaluation of inform: Lessons learned and application to other systems. *Transportation Research Record*, TRR 1360, 1992.

S. Smulders. Control of freeway traffic flow by variable speed signs. *Transportation Research B*, Vol. 24B(2):111–132, 1990.

Y. J. Stephanedes, E. Kwon, and K. K. Chang. Control emulation method for evaluating and improving traffic-responsive ramp metering strategies. *Transportation Research Record*, TRR 1360, 1992.

Y. J. Stephanedes, E. Kwon, and P. G. Michalopoulos. Demand diversion for vehicle guidance, simulation and control in freeway corridors. *Transportation Research Record*, TRR 1220, 1989.

H. Subramanian. Estimation of a car-following model for freeway simulation. Master's thesis, Massachusetts Inst. of Tech., Cambridge, MA, 1996.

P. J. Tarnoff and N. Gartner. Real-time, traffic adaptive signal control. In S. Yagar and A. Santiago, editors, *Large Urban Systems – Proceedings of the Advanced Traffic Management Conference*, pages 157–168, St. Peterburg, Florida, 1993.

S. E. Underwood and S. G. Gehring. Framework for evaluating intelligent vehicle highway systems. 74th Transportation Research Board Annual Meeting, Washington, D.C., 1994.

M. van Aerde. INTEGRATION: A dynamic traffic simulation/assignment model. Paper presented at the IVHS Dynamic Traffic Assignment and Simulation Workshop, Federal Highway Administration, 1992.

M. van Aerde, B. Hellinga, M. Baker, and H. Rakha. INTEGRATION - A overview of traffic simulation features. In *Proceedings of 75th TRB Annual Meeting*, Washington, D.C., 1997.

M. van Aerde and S. Yagar. Dynamic integrated freeway/traffic signal networks: A routing-based modeling approach. *Transportation Research A*, Vol. 22A(6):445–453, 1988.

N. van der Zijpp. *Dynamic origin-destination matrix estimation on motorway networks*. PhD thesis, Delft University of Technology, The Neitherlands, 1996.

J. von Toorenburg, R. van der Linden, and B. van Velzen. Predictive control in traffic management. Final Report AV-2468, Ministry of Transport, Public Works and Water Management, The Netherlands, 1996.

P. Vythoulkas. A dynamic stochastic assignment model for the analysis of general networks. *Transportation Research*, 24B(6):453–469, 1990.

D. A. Wicks. INTRAS - a microscopic freeway corridor simulation model. In *Overview of Simulation in Highway Transportation*, volume 1, pages 95–107, 1977.

Q. Yang and H. N. Koutsopoulos. A microscopic traffic simulator for evaluation of dynamic traffic management systems. *Transportation Research C*, 4(3):113–129, 1996.

P. J. Yauch, J. C. Gray, and W. A. Lewis. Using NETSIM to evaluate the effects of drawbridge openings on adjacent signalized intersections. *ITE Journal*, 58(3): 35–44, 1988.

A. K. Ziliaskopoulos and H. S. Mahmassani. Note on least time path computation considering delays and prohibitions for intersection movements. *Transportation Research B*, 30B(5):359–367, 1996.

Appendix A

Abbreviation

ATIS	Advanced Traveler Information Systems
ATMS	Advanced Traffic Management Systems
CA/T	Central Artery and Tunnels, Boston, Massachusetts.
ETC	Electronic toll collection.
HOV	High occupancy vehicles.
ITS	Intelligent Transportation Systems.
LUS	Lane use signs.
mph	Miles per hour.
mps	Meters per second.
MITSIM	Microscopic traffic simulator.
MesoTS	Mesoscopic traffic simulator.
OD	Origin to Destination.
OOP	Object Oriented Programming.
RMSE	Root Mean-Square Error.
TMS	Traffic management simulator.
VMS	Variable Message Signs.
vph	Number of vehicles per hour.
VSLS	Variable Speed Limit Signs.

Appendix B

Calculation of Time-Dependent Shortest Paths

A dynamic shortest path (SP) algorithm has been implemented in SIMLAB to calculate time-dependent shortest paths. This is a modified version of the dequeue implementation of the label correcting algorithm described in Ahuja et al. (1993). The implemented algorithm finds the shortest path from a given link (more precisely, the upstream end of the link) to every destination node in the network. It calculates links to destination nodes instead of node to node shortest paths because it allows an efficient consideration of turning penalties at intersections and provides the shortest path data needed in the simulation (e.g., the shortest path for a vehicle from its current position to its destination).

The modifications in the implementation of the algorithm, compared to the standard algorithms in the literature, are the following:

(a) Labeling links instead of nodes; and,

(b) Considering penalties for turning movements.

The data structures that support implementation of the algorithm are the described below:

- Each node has pointers to the incoming and outgoing links at the node.

- Each link has pointers to its upstream and downstream nodes. It also has a *predecessor* which keeps track of the shortest path from the source link to the subject link (i.e. the previous link on the shortest path), and a *label* that stores the cost from the source link to the downstream node of the subject link on the path.
- An integer flag is used for each link to indicate whether the turning movements to downstream links are allowed or not.
- A link has an (optional) array to store the penalties for the turning movements to the downstream links.

The implementation of the SP algorithm is presented as follows:

(1) *Initialization*

(1.1) Set *predecessors* to UNSET and *label* to INF for all links.

(1.2) Set the *predecessor* of the source link to START and the *label* to the cost to travel the source link.

(1.3) Put the source link into the queue.

(2) *Label Correcting Loop*

If the queue is empty, go to the *post process* (the labeling is done); otherwise, do:

(2.1) Get a candidate link u by dequeuing. Find its downstream links and turning penalties to these links.

(2.2) For each downstream link v of link u calculate the potential path cost:

$$cost = path\_cost[u] + link\_cost(entry\_time, v) + penalty[u, v] \qquad \text{(B.1)}$$

If $cost$ is less than $path\_cost[v]$, goto step (2.3); otherwise, check next downstream link of link u.

(2.3) Set the *predecessor* of link v to u and $path\_cost[v]$ to $cost$. If link v has never been in the list, append link v to the end of the queue; if link v

was in the list, insert link v in the beginning of the queue. Check next downstream link of link u.

(3) *Post Process*

After the label correcting loop is completed, the shortest paths from the source link to every link in the network are available and the post process starts. For each node k in the network, its predecessor and shortest path cost are calculated by choosing the incoming link with the smallest cost among all the links entering node k.

This algorithm does not need to "explode" the intersections which have turning penalties. A similar approach has be reported in the literature by Ziliaskopoulos and Mahmassani (1996) and is considered to be more efficient than the alternative approach that "explodes" nodes.

This algorithm is used in MITSIM and MesoTS to assist the vehicle routing. In this application, the time dependent shortest paths are calculated once for unguided vehicles and periodically for guided vehicles when projected link travel time data becomes available. When applying the route choice model described in Section 3.8 (see Eqs (3.7) and (3.8)), a vehicle needs to access the shortest travel time to its destination from each downstream link (or path) at the decision node. Since the vehicle may not arrive the downstream link in current time interval, the shortest paths need to be computed not only for the current time periods, but several periods ahead. Therefore, the algorithm needs should be repeated for a number of time periods. The results of this computation are two 3-dimensional matrices: one stores predecessors and the other stores travel times on the shortest paths. The size of each matrix is the product of (i) number of time periods projected ahead; (ii) number of links; and (iii) number of nodes. For large networks the size of these two matrices can be significant. However, since route choice models are usually invoked only once per link, the shortest path data is stored in database files and tagged with the input file names and their modification time.

Appendix C

Simulation Parameters

Simulation parameters are read from data files. The values used in the case study are documented in this appendix.

C.1 Vehicle Characteristics

Vehicle characteristics for following vehicle classes are listed in Table C.1–C.5:

1 High performance passenger cars.

2 Low performance passenger cars.

3 Buses

4 Heavy single unit trucks

5 Trailer trucks

Table C.1: Vehicle fleet and attributes

Class	Length (ft)	Width (ft)	Fleet Mix (%)	ETC (%)	HOV (%)	Over Height (%)	Toll booth delay factor
1	18	6	50	40	10	0	1.0
2	18	6	48	20	15	0	1.0
3	40	8	1	50	100	0	1.5
4	50	8	1	50	0	0	1.5
5	70	8	0	50	0	0	2.0

Table C.2: Maximum acceleration rates on level terrain (ft/sec^2)

Class	Speed (ft/s)				
	<20	20-40	40-60	60-80	≥80
1	10.00	7.90	5.60	4.00	4.00
2	8.71	5.17	4.43	2.89	2.00
3	7.00	5.00	4.00	1.50	1.00
4	2.80	2.50	1.50	1.00	0.50
5	1.60	1.45	0.89	0.47	0.40

Source: Adjusted based on FHWA (1980), Pline (1992), FHWA (1994).

Table C.3: Maximum deceleration rates (ft/sec^2)

Class	Speed (ft/s)				
	<20	20-40	40-60	60-80	≥80
1	10.0	9.5	9.0	8.5	8.0
2	10.0	9.5	9.0	8.5	8.0
3	na	na	na	na	na
4	na	na	na	na	na
5	na	na	na	na	na

Table C.4: Normal deceleration rates (ft/sec^2)

Class	Speed (ft/s)				
	<20	20-40	40-60	60-80	≥80
1	7.8	6.7	4.8	4.8	4.8
2	7.8	6.7	4.8	4.8	4.8
3	na	na	na	na	na
4	na	na	na	na	na
5	na	na	na	na	na

Source: ITE (1982).

Table C.5: Maximum speed (ft/sec)

Class	Grade (%)				
	<-2	-2-0	0-2	2-4	≥4
1	200	200	200	200	200
2	200	200	200	200	200
3	150	125	100	80	60
4	130	105	80	65	45
5	100	90	80	60	40

See: Eq (3.4) on page 54; Source: Modified based on FHWA (1980).

Table C.6: Differences in speed across lanes

Number of Lanes	Lane Index (from left to right)				
	1	2	3	4	5
1	1.00	-	-	-	-
2	1.06	0.94	-	-	-
3	1.06	1.01	0.93	-	-
4	1.05	1.05	0.97	0.93	-
5	1.04	1.06	1.01	0.95	0.94

See Eq (3.5) on page 55; Source: FHWA (1980).

C.2 Driver Behavior

Table C.7: Distribution of desired speed

Percentage of drivers (%)	ν (mph)
5.0	0
25.0	5
45.0	10
20.0	15
5.0	20

See Eq (3.4) on page 54.

Table C.8: Parameters in car-following model
(distance in meter, speed in meter/sec, and acceleration in meter/sec^2)

	α	β	γ
accelerating	2.15	-1.67	-0.89
decelerating	1.55	1.08	1.65

See Eq (3.11) on page 63; Source: Subramanian (1996).

Table C.9: Startup delay when traffic signals change from red to green

Position in Queue	Mean (seconds)
1	1.5
2	1.2
3	1.0
4	0.9
5	0.8
6	0.6
7	0.5
8	0.4

(Maximum delay is set to 2 seconds)

Table C.10: Parameters in gap acceptance model for lane change
(Distance in feet, speed in feet/sec)

Type	Gap	Scale	γ	g_{min}	β_1	β_2
Discretionary	Lead	1.0	0.0	3.0	0.05	0.15
	Lag	1.0	0.0	5.0	0.15	0.40
Mandatory	Lead	1.0	2.5e-5	3.0	0.05	0.15
	Lag	1.0	2.5e-5	5.0	0.15	0.40

See Eqs. (3.19-3.20) on page 71.

Table C.11: Miscellaneous Parameters

Category	Parameter Name (Unit)	Value
Merging	Probability of aggressive merge from ramp	0.5
Mandatory Lane Change (MLC), see Eq (3.15-3.16)	Distance lower bound, x_0, (feet)	330
	α_0 (feet)	1320
	Coef. for number of lanes, α_1	0.5
	Coef. for congestion level, α_2	1.0
	Minimum time in lane (seconds)	1.0
Discretionary Lane Change (DLC)	Prob. to look for a DLC	0.50
	Prob. to continue a DLC	0.90
	Speed impatient lowerbound	0.80
	Speed impatient upperbound	1.00
	Slower leader threshold	0.85
	Speed threshold	0.10
	Acceleration threshold	0.85
	Speed ahead looking distance	300
	Min time in lane, in same direction (seconds)	3
	Min time in lane, in different direction (seconds)	10
Yielding	Probability to start yielding	0.8
	Probability to continue yielding	1.0
	Max yielding time (seconds)	300
Nosing, see Eq (3.21-3.22)	T, time period (seconds)	1.0
	β_0, constant	0.5
	β_1, coef. for number of lanes	0.6
	β_2, coef. for waiting time (in minutes)	0.2
	Max prob. at ramp lane drop, inc., etc.	0.5
	Max prob. in connection to next link	1.0
	Max stuck time (seconds)	60
Compliance Rates	Traffic signals only	1.0
	Portal signals and traffic signals	1.0
	Ramp meters, portal signals and traffic signals	0.9
	Red LUS only	0.9
	Yellow and red LUS	0.5
	Lane use privilege	1.0
	Lane change regulation and lane use privilege	0.9
	Vehicle type related lane use message	1.0
	Path related lane use message	1.0
	Route guidance message	0.7

Appendix D

Random Number Generator

Random number generation is a fundamental element in stochastic simulation. A modified version of the linear congruential random number generator proposed by Park and Miller (1988) is used to generate uniformly distributed pseudo-random number in the range from 0 to $2^{31} - 1$, i.e.:

$$\hat{r}_n = A\,(r_{n-1} \bmod Q) - R\,(r_n/Q)$$
$$r_n = \begin{cases} \hat{r}_n \ + \ M & \hat{r}_n < 0 \\ \hat{r}_n & \hat{r}_n \geq 0 \end{cases} \tag{D.1}$$

where r_n $(n = 1, 2, \cdots)$ is the recursively generated random numbers; r_n is the random seed, which is set to the clock time if the simulator is run in randomized mode or a hard coded arbitrary value otherwise; $M = 2^{31}$ is the maximum value of integer; $A = 48271$ is constant chosen, after follow up work to Park and Miller (1988), to provide better statistical properties of the generator; and $Q = M/A = 44488$ and $R = M\%A = 3399$ are constants.

Random numbers r_n^0 uniformly distributed in the range from 0 to 1 are given by:

$$r_u = r/M; \tag{D.2}$$

where r is the random number calculated from Eq (D.1).

Appendix E

Statistics for Evaluating Simulation Models

In this section we describe the statistical measures used to compare simulation output value with "observed" values. This discussion follows the detailed discussion in Pindyck and Rubinfeld (1991).

Root-mean-square (rms) error is a measure of the deviation of the simulated variable from its actual value and defined as:

$$\text{rms error} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2} \tag{E.1}$$

where:

$\hat{y}_i =$ simulated value;

$y_i =$ actual value; and,

$n =$ number of data points

Rms percent error is defined as:

$$\text{rms percent error} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\frac{\hat{y}_i - y_i}{y_i})^2} \tag{E.2}$$

Mean error is defined as:

$$\text{mean error} = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i) \tag{E.3}$$

and the *mean percent error* is defined as:

$$\text{mean percent error} = \frac{1}{n}\sum_{i=1}^{n}\frac{\hat{y}_i - y_i}{y_i} \tag{E.4}$$

Mean error and mean percent error are useful as an indication of systematic bias.

The *correlation coefficient* can be used as an indication of how the simulated and actual values are related. This statistic is defined as:

$$\rho = \frac{\sum_{i=1}^{n}(\hat{y}_i - \overline{\hat{y}})\ (y_i - \overline{y})}{n\ \hat{\sigma}\ \sigma} \tag{E.5}$$

where:

$\overline{\hat{y}}, \hat{\sigma} =$ mean and standard deviation of the simulated values;

$\overline{y}, \sigma =$ mean and standard deviation of the actual values.

The closer the value of ρ to 1, the better the performance of the simulation model.

The *rms percent error* and *mean percent error* can only be used for the cases where the value of actual data points are all non-zero. This condition may not be satisfied for some scenarios (e.g. speeds and flow can be zero in case of queue; occupancy and flow can be zero if there is no traffic). A useful simulation statistic, related to the rms percent error, that does not have this restriction is *Theil's inequality coefficient*, which is defined as:

$$U = \frac{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}\hat{y}_i^2}\ +\ \sqrt{\frac{1}{n}\sum_{i=1}^{n}y_i^2}} \tag{E.6}$$

The numerator in Eq (E.6) is just the rms error. U always has values between 0 and 1. If $U = 0$, $\hat{y}_i = y_i$ for every data point and there is a perfect fit to the actual data. If $U = 1$, the simulation fit is as bad it could be.

The Theil inequality can be decomposed into three components called the *proportions of inequality* which break the simulation error down into its characteristic sources:

$$
\begin{aligned}
U_M &= n\ (\overline{\hat{y}} - \overline{y})^2\ /\ \textstyle\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 \\
U_S &= n\ (\hat{\sigma} - \sigma)^2\ /\ \textstyle\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 \\
U_C &= 2\ (1 - \rho)\ n\ \hat{\sigma}\ \sigma\ /\ \textstyle\sum_{i=1}^{n}(\hat{y}_i - y_i)^2
\end{aligned} \tag{E.7}
$$

such that:

$$U_M + U_S + U_C = 1.0 \tag{E.8}$$

where:

$U_M =$ *bias proportion*, a measure of systematic error;

$U_S =$ *variance proportion*, an indication of the model's ability to replicate the degree of variability in the variable of interest;

$U_C =$ *covariance proportion*, a measure of unsystematic error.

Ideally U_M, U_S should be close to 0 and U_C close to 1. A high value of U_M (e.g. > 0.1 or 0.2) implies the existence of a systematic bias. A high value of U_S implies that there exists a large fluctuation in the simulated outcome while the actual data has little fluctuation, or vice versa. The value of U_C is less worrisome since it is unreasonable to expect an exact match between the simulated and actual outcomes.

Appendix F

Examples of Data Files

The following sections provide examples of the data files representing the road network and travel demand. Unless otherwise specified, the data is based on the I-880N network (see Figure 3-14) and shown in the format used by the simulators.

F.1 Network Database

```
/*
 * This network file was created based on the data downloaded from the
 * FSP Project of Path Program at University of California, Berkeley.
 * For more information on the data set, see:
 *             http://www.path.berkeley.edu/FSP/.
 */

[Title]: "I-880 North"

[Link Labels] : 7 # number of link labels
{
# { Label-id "Street-name" }
  { 1 "I-880 North"}     | { 4 "Winton"}      | { 7 "Leweling"}
  { 2 "Tennyson"}        | { 5 "A Street"}    |
  { 3 "SR-92"}           | { 6 "Hesperian"}   |
}

[Nodes] : 22 # Number of nodes
{
# { Node-id Type "Name" }
```

```
  { 1 1 "San Jose/Fremont"}
  { 2 0 "Tennyson Off"}
  { 3 0 "Tennyson On"}
  { 4 0 "SR 92 Off"}
  { 5 0 "SR 92 On"}
  { 6 0 "Winton Off"}
  { 7 0 "Winton On"}
  { 8 0 "A Street Off"}
  { 9 0 "A Street On"}
  {10 0 "Hesperian Off"}
  {11 0 "Leweling Off"}
  {19 1 "Oakland/Berkeley"}
  {20 1 "Tennyson Off-ramp"}
  {21 1 "Tennyson On-ramp"}
  {30 1 "SR 92 Off-ramp"}
  {31 1 "SR 92 On-ramp"}
  {40 1 "Winton Off-ramp"}
  {41 1 "Winton On-ramp"}
  {50 1 "A Street Off-ramp"}
  {51 1 "A Street On-ramp"}
  {60 1 "Hesperian Off-ramp"}
  {70 1 "Leweling Off-ramp"}

}

[Links] : 21 : 24 : 79 # Total number of links, segments, and lanes
{
# Links are listed in arbitrary order; segments within each link are
# listed in order from upstream to downstream; lanes in each segment
# are listed in order from left to right.
#
# { Link-id Link-type Upstream-node Downstream-node Link-label
#   { Segment-id Default-speed-limit Free-flow-speed Grade SD-index
#     { X1 Y1 Bulge X2 Y2 }
#     { Lane-id Lane-change-and-lane-use-rules }
#   }
# }

  { 100 1 1 2 1 # I880 North
    { 110 55 65 0
    {44000 0 0 40600 0}
      { 111 0x81}
      { 112 0x03}
      { 113 0x03}
      { 114 0x03}
      { 115 0x02}
    }
  }

  { 200 1 2 3 1 # I880 North
    { 210 55 65 0
    {40600 0 0 38600 0}
      { 211 0x81}
      { 212 0x03}
      { 213 0x03}
      { 214 0x02}
    }
  }

  { 300 1 3 4 1 # I880 North
    { 310 55 65 0
    {38600 0 0 38000 0}
      { 311 0x81}
      { 312 0x03}
      { 313 0x03}
      { 314 0x03}
      { 315 0x02}
    }
    { 320 55 65 0
    {38000 0 0 35800 0}
      { 321 0x81}
      { 322 0x03}
      { 323 0x03}
      { 324 0x02}
    }
    { 330 55 65 0
    {35800 0 0 35200 0}
      { 331 0x81}
      { 332 0x03}
      { 333 0x03}
```

```
    { 334 0x03}
    { 335 0x02}
  }
}

{ 400 1 4 5 1 # I880 North
  { 410 55 65 0
  {35200 0 0 31500 0}
    { 411 0x81}
    { 412 0x03}
    { 413 0x03}
    { 414 0x02}
  }
}

{ 500 1 5 6 1 # I880 North
  { 510 55 65 0
  {31500 0 0 30100 0}
    { 511 0x81}
    { 512 0x03}
    { 513 0x03}
    { 514 0x03}
    { 515 0x02}
  }
}

{ 600 1 6 7 1 # I880 North
  { 610 55 65 0
  {30100 0 0 27100 0}
    { 611 0x81}
    { 612 0x03}
    { 613 0x03}
    { 614 0x03}
    { 615 0x02}
  }
}

{ 700 1 7 8 1 # I880 North
  { 710 55 65 0
  {27100 0 0 25700 0}
    { 711 0x81}
    { 712 0x03}
    { 713 0x03}
    { 714 0x03}
    { 715 0x03}
    { 716 0x02}
  }
}

{ 800 1 8 9 1 # I880 North
  { 810 55 65 0
  {25700 0 0 23500 0}
    { 811 0x81}
    { 812 0x03}
    { 813 0x03}
    { 814 0x03}
    { 815 0x02}
  }
}

{ 900 1 9 10 1 # I880 North
  { 910 55 65 0
  {23500 0 0 22900 0}
    { 911 0x81}
    { 912 0x03}
    { 913 0x03}
    { 914 0x03}
    { 915 0x03}
    { 916 0x02}
  }
  { 920 55 65 0
  {22900 0 0 16000 0}
    { 921 0x81}
    { 922 0x03}
    { 923 0x03}
    { 924 0x03}
    { 925 0x02}
  }
}

{1000 1 10 11 1 # I880 North
  {1010 55 65 0
  {16000 0 0 14700 0}
    {1011 0x01}
    {1012 0x03}
    {1013 0x03}
    {1014 0x03}
    {1015 0x02}
  }
}

{1100 1 11 19 1 # I880 North
  {1110 55 65 0
  {14700 0 0 12800 0}
```

```
      {1111 0x01}
      {1112 0x03}
      {1113 0x03}
      {1114 0x02}
    }
  }

  {2000 2 2 20 2 # Tennyson off
    {2010 35 45 0 2
    {40600 48 0 40300 72}
      {2011 0x00}
    }
  }

  {2100 2 21 3 2 # Tennyson on
    {2110 35 45 0 2
    {38900 72 0 38600 48}
      {2111 0x00}
    }
  }

  {3000 2 4 30 3 # SR 92 off
    {3010 35 45 0 2
    {35200 48 0 34900 72}
      {3011 0x01}
      {3012 0x02}
    }
  }

  {3100 2 31 5 3 # SR 92 on
    {3110 35 45 0 2
    {31800 72 0 31500 48}
      {3111 0x00}
    }
  }

  {4000 2 6 40 4 # Winton off
    {4010 35 45 0 2
    {30100 60 0 29800 84}
      {4011 0x00}
}
  }

  {4100 2 41 7 4 # Winton on
    {4110 35 45 0 2
    {27400 84 0 27100  60}
      {4111 0x00}
    }
  }

  {5000 2 8 50 5 # A Street off
    {5010 35 45 0 2
    {25700 60 0 25400 84}
      {5011 0x00}
    }
  }

  {5100 2 51 9 5 # A Street on
    {5110 35 45 0 2
    {23800 84 0 23500 60}
      {5111 0x00}
    }
  }

  {6000 2 10 60 6 # Hesperian off
    {6010 35 45 0 2
    {16000 60 0 15700 84}
      {6011 0x00}
    }
  }

  {7000 2 11 70 7 # Lewelling off
    {7010 35 45 0
    {14700 48 0 14400 72}
      {7011 0x00}
    }
  }

}

[Lane Connections] :
{
# {Up-lane Downstream-lane}
```

```
  { 111  211}
  { 112  212}
  { 113  213}
  { 114  214}
  { 115 2011}
  { 211  311}
  { 212  312}
  { 213  313}
  { 214  314}
  {2111  315}
  { 311  321}
  { 312  322}
  { 313  323}
  { 314  324}
  { 321  331}
  { 322  332}
  { 323  333}
  { 324  334}
  { 324  335}
  { 331  411}
  { 332  412}
  { 333  413}
  { 334  414}
  { 334 3011}
  { 335 3012}
  { 411  511}
  { 412  512}
  { 413  513}
  { 414  514}
  {3111  515}
  { 511  611}
  { 512  612}
  { 513  613}
  { 514  614}
  { 515  615}
  { 515 4011}
  { 611  711}
  { 612  712}
  { 613  713}
  { 614  714}
  { 615  715}
  {4111  716}
  { 711  811}
  { 712  812}
  { 713  813}
  { 714  814}
  { 715  815}
  { 716 5011}
  { 811  911}
  { 812  912}
  { 813  913}
  { 814  914}
  { 815  915}
  {5111  916}
  { 911  921}
  { 912  922}
  { 913  923}
  { 914  924}
  { 915  925}
  { 921 1011}
  { 922 1012}
  { 923 1013}
  { 924 1014}
  { 925 1015}
  { 925 6011}
  {1011 1111}
  {1012 1112}
  {1013 1113}
  {1014 1114}
  {1015 7011}

}

[Turn Prohibitors] : 0
{
# {From-link To-link}
}

[Sensors] : 16 # Total number of sensors
{
# { Type-code, Task-code, Zone-length, Segment-id, Position-in-segment
#   { Sensor-id Working-probability Lane-id } # Lane specific sensors
#   or
#   { Sensor-id Working-probability }         # Area wide sensors
# }

  {257 7 6  110 0.3902
    {15 1}
  }
  {257 7 6  210 0.4000
    {17 1}
  }
  {257 7 6  210 0.9000
    { 4 1}
  }
  {257 7 6  320 0.9545
    {12 1}
  }
  {257 7 6  410 0.2702
    {13 1}
  }
  {257 7 6  410 0.6216
    {19 1}
  }
  {257 7 6  510 0.2142
    {18 1}
  }
  {257 7 6  610 0.2666
    { 6 1}
  }
  {257 7 6  610 0.7000
    {11 1}
  }
  {257 7 6  710 0.5000
    { 2 1}
  }
  {257 7 6  810 0.6363
    {10 1}
  }
  {257 7 6  920 0.0144
    {20 1}
  }
  {257 7 6  920 0.2608
    { 7 1}
  }
  {257 7 6  920 0.5072
    { 1 1}
  }
```

```
  {257 7 6  920 0.7536
    { 3 1}
  }
  {257 7 6 1010 0.3076
    {16 1}
  }

}

[Control Devices] : 4 # Total number of signals and signs
{
# { Type Visibility-distance Segment-id Position-in-segment
#   { Device-id Initial-state } or
#   { Device-id Initial-state Lane-id }
# }

  {12 328 2110 0.99
    {1 0x3 2111}
  }
  {12 328 3110 0.99
    {2 0x3 3111}
  }
  {12 328 4110 0.99
    {3 0x3 4111}
  }
  {12 328 5110 0.99
    {4 0x3 5111}
  }

}

[Toll Plazas] : 0
{
# { Visibility-distance Segment-id Position-in-segment
#    { Booth-id Initial-state Lane-id Lane-use-rules Speed-limit
#      Delay
#    }
# }
}
```

F.2 Time Dependent OD Trip Tables

```
/*
 * This file contains two OD tables.  The first one becomes in effect
 * at 7am and the second at 8am.  All the optional fields assume the
 * default values.  The data format for each table is as the following:
 *
 * Start-time Vehicle-type Scaling-factor
 * {
 *   { Origin-node Destination-node Departure-rate Variance
 *     Distribution-factor { Path-1 Path-2 ... Path-N }
 *   }
 *   ... ...
 * }
 *
```

```
 * where the fields Variance, Distribution-factor and Path-set are
 * optional and have the following default values:
 *
 *   Variance             = 0.0 (Deterministic demand)
 *   Distribution-factor  = 0.0 (0=Random 1=Deterministic)
 *   Path-set             = None (use route choice model)
 */

07:00:00 0 1
{
  {  1  20   414 }
  {  1  30  1146 }
  {  1  40   426 }
  {  1  50   312 }
  {  1  60   432 }
  {  1  19  2898 }
  { 21  30     6 }
  { 21  40     6 }
  { 21  50    18 }
  { 21  60    18 }
  { 21  19   750 }
  { 31  40    12 }
  { 31  50    24 }
  { 31  60    30 }
  { 31  19  1668 }
  { 41  19   528 }
  { 51  60     6 }
  { 51  19   402 }
}
08:00:00 0 1
{
  {  1  20   690 }
  {  1  30  1236 }
  {  1  40   498 }
  {  1  50   420 }
  {  1  60   510 }
  {  1  19  2862 }
  { 21  30    42 }
  { 21  40    30 }
  { 21  50    48 }
  { 21  60    42 }
  { 21  19   810 }
  { 31  40    12 }
  { 31  50    66 }
  { 31  60    60 }
  { 31  19  1806 }
  { 41  50    24 }
  { 41  60    24 }
  { 41  19   696 }
  { 51  60     0 }
  { 51  19   588 }
}

<END>
```

F.3 Vehicle Trip Table

```
/*
 * This example shows an alternate way to provide travel demand.  Five
 * vehicles of type 3 and 4 are scheduled to travel from the node 1 to
 * node 19, three of them depart at 7:30am and the others at 8:00am.
 * The data format for each block (vehicles that depart at a given time)
 * is:
 *
 *   Time {
 *     ID { Origin-node Destination-node Type Path }
 *   or
```

```
 *     Count * { Origin-node Destination-node Type Path }
 *     ... ...
 *   }
 *
 * where ID is a unique vehicle identification which is verbosely copied
 * into the output.  Count represents the number of vehicles with the
 * same information.  ID, Count, and the last two columns for each record
 * are optional and their default values are:
 *
 *    Type  = 0 (randomly assigned based global vehicle mix)
 *    Path  = None (choose path based on route choice model)
 *    Count = 1
 */

07:30:00
{
  { 1 19 }          # type and path yet to be determined
  2 * { 1 19 4 }    # two type 4 vehicles, beware of the "*"
}
08:00:00
{
  412 { 1 19 4 }    # vehicle with ID=412
  { 1 19 0 8 }      # using path 8, type yet to be determined
}
<END>
```

F.4 Vehicle Path Table

A small hypothetical network (Figure F-1) with the similar structure as the Amsterdam A10 network is used to illustrate the path table. This network is a 2-mile stretch of two freeway loops consisting of two interchanges with an arterial road. The network has 16 nodes, 28 links, and 12 OD pairs. There are 20 paths for this network.

```
/*
 * OD Path Table
 *
 * Each record in the path table specifies one path connecting a
 * particular OD pair and contains the following information:
 *
 *   { Path-id Origin-node Destination-node
 *     { Link-1 Link-2 ... Link-N } Path-travel-time
 *   }
 *
```

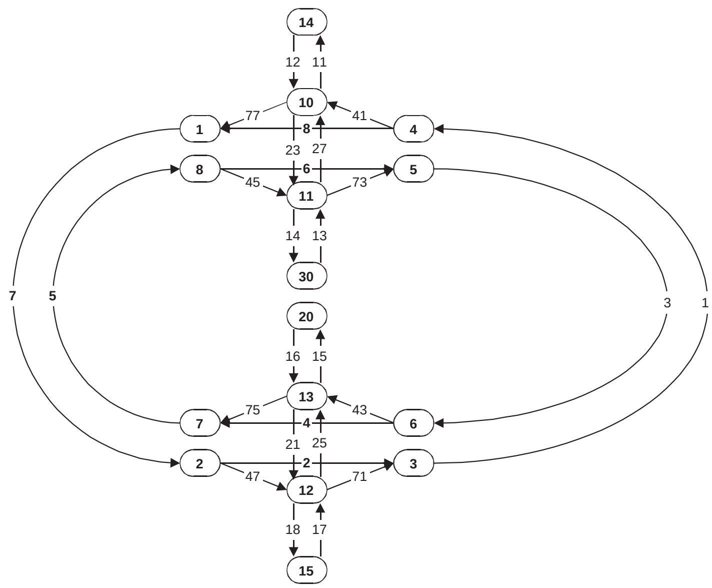


Figure F-1: A hypothetical network with loops and interchanges

```
 * This file can be generated from the path records and vehicle trip
 * records produced by either MITSIM or MesoTs.
 */

[OD Path Table] : 20
{
  {  1 14 15 { 12 77 7 47 18 } 94 }
  {  2 14 15 { 12 23 73 3 43 21 18 } 117.3 }
  {  3 14 20 { 12 23 73 3 43 15 } 109.6 }
  {  4 14 20 { 12 77 7 47 25 15 } 110.9 }
  {  5 14 30 { 12 23 14 } 28.9 }

  {  6 15 14 { 17 71 1 41 11 } 93.3 }
  {  7 15 14 { 17 25 75 5 45 27 11 } 115.5 }
  {  8 15 20 { 17 25 15 } 26.3 }
  {  9 15 30 { 17 25 75 5 45 14 } 110.8 }
  { 10 15 30 { 17 71 1 41 23 14 } 111.3 }

  { 11 20 14 { 16 21 71 1 41 11 } 109.7 }
  { 12 20 14 { 16 75 5 45 27 11 } 109.8 }
  { 13 20 15 { 16 21 18 } 27.9 }
  { 14 20 30 { 16 21 71 1 41 23 14 } 128.3 }
  { 15 20 30 { 16 75 5 45 14 } 105.4 }

  { 16 30 14 { 13 27 11 } 28.9 }
  { 17 30 15 { 13 27 77 7 47 18 } 112.4 }
  { 18 30 15 { 13 73 3 43 21 18 } 113.1 }
  { 19 30 20 { 13 73 3 43 15 } 105 }
```

```
  { 20 30 20 { 13 27 77 7 47 25 15 } 126.9 }
}
```

F.5 Incidents

```
/*
 * Incident File
 *
 * { Visibility-distance  Segment-ID  Position-in-segment
 *   { Severity-code Capacity-factor Start-time Duration Speed-limit
 *     Lane-ID Incident-ID }
 * }
 *
 * Note: Severity of an incident is defined by a severity code
 * (0=ignore 1=minor 2=major) and a factor of the deduction of the
 * lane capacity.
 */
{ // start at 7:40 and stay for 20 minutes
  150 610 0.95
  { 1 0.25 27600 1200   0 613 911 }
  { 2 1.00 27600 1200   0 614 912 }
  { 2 1.00 27600 1200   0 615 913 }
}
```