1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
SWUTC/04/167823-1		

4. Title and Subtitle	5. Report Date
Development and Validation of a Flexible, Open Architecture, Transportation Simulation with an Adaptive Control Implementation	June 2004
	6. Performing Organization Code
7. Author(s)	8. Performing Organization
Michael Hunter and Randy B. Machemehl	Research Report 167823-1
9. Performing Organization Name and Address	10. Work Unit No. (TRAIS)
Center for Transportation Research University of Texas at Austin 3208 Red River, Suite 200 Austin, Texas 78705-2650	11. Contract or Grant No. 10727 10727
12. Sponsoring Agency Name and Address	13. Type of Report and Period Covered
Southwest Region University Transportation Center Texas Transportation Institute The Texas A&M University System College Station, Texas 77843-3135	14. Sponsoring Agency Code

15. Supplementary Notes

Supported by general revenues from the State of Texas.

16. Abstract

Simulation has been utilized in the planning and development of almost all sectors of the transportation field. The practicing transportation community primarily relies on simulation packages. When a practitioner (or end user) uses a simulation package, most of the simulation development efforts have already been completed. Unfortunately, the use of these simulation packages has several disadvantages, most notably the "black box" phenomenon and reduced modeling flexibility.

The simulation model described in this research lays the foundation for a transportation simulation, OPEN-TS3, that minimizes the black box problem and increases modeling flexibility, while still providing an easy to use package in which highly capable models may be quickly and accurately built. The approach to simulation in this research is to develop a platform that allows for the use of existing constructs where applicable, while still retaining the flexibility for the user to incorporate new basic modeling constructs. OPEN-TS3 utilizes SIMAN (a simulation language) and ARENA (a simulation development tool), both commonly found in manufacturing applications.

In intersection and arterial validation studies, comparing the OPEN-TS3 to CORSIM, a high level of agreement was seen in the volumes, delays, queues, and speeds simulated by both models. Some differences were seen between the models in overcapacity demand situations, most significantly in left turn operations. Agreement was also seen in a comparison to real-world delays measured for a twelve-intersection downtown Chicago network. The real-world data validation effort also highlighted some important issues regarding validation with real-world data, in particular the difficulties in obtaining data and the potential pitfalls of GPS probe vehicle studies.

As a demonstration of the flexibility of OPEN-TS3 two adaptive signal control strategies are also successfully implemented. The adaptive control strategies are tested on three different networks under varying volume conditions. Based on the OPEN-TS3 simulations it was seen that adaptive control can provide superior overall performance, but can have a significantly greater range of variability than that of pre-timed control.

17. Key Words	18. Distribution Statement
Simulation, Validation, Adaptive Control	No restrictions. This document is available to the public through NTIS: National Technical Information Service 5285 Port Royal Road Springfield, Virginia 22161

19. Security Classif.(of this report)	20. Security Classif.(of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	276	

DEVELOPMENT AND VALIDATION OF A FLEXIBLE, OPEN ARCHITECTURE, TRANSPORTATION SIMULATION WITH AN ADAPTIVE TRAFFIC SIGNAL CONTROL IMPLEMENTATION

By

Michael Hunter

Randy B. Machemehl

Research Report SWUTC/04/167823-1

Southwest Region University Transportation Center

Center for Transportation Research

The University of Texas at Austin

Austin, Texas 78712

June 2004

DISCLAIMER

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated under the sponsorship of the Department of Transportation, University Transportation Centers Program, in the interest of information exchange. Mention of trade names or commercial products does not constitute endorsement or recommendation for use.

ABSTRACT

Simulation has been utilized in the planning and development of almost all sectors of the transportation field. The practicing transportation community primarily relies on simulation packages. When a practitioner (or end user) uses a simulation package, most of the simulation development efforts have already been completed. Unfortunately, the use of these simulation packages has several disadvantages, most notably the "black box" phenomenon and reduced modeling flexibility.

The simulation model described in this research lays the foundation for a transportation simulation, OPEN-TS3, that minimizes the black box problem and increases modeling flexibility, while still providing an easy to use package in which highly capable models may be quickly and accurately built. The approach to simulation in this research is to develop a platform that allows for the use of existing constructs where applicable, while still retaining the flexibility for the user to incorporate new basic modeling constructs. OPEN-TS3 utilizes SIMAN (a simulation language) and ARENA (a simulation development tool), both commonly found in manufacturing applications.

In intersection and arterial validation studies, comparing the OPEN-TS3 to CORSIM, a high level of agreement was seen in the volumes, delays, queues, and speeds simulated by both models. Some differences were seen between the models in overcapacity demand situations, most significantly in left turn operations. Agreement was also seen in a comparison to real-world delays measured for a twelve-intersection downtown Chicago network. The real-world data validation effort also highlighted some important issues regarding validation with real-world data, in particular the difficulties in obtaining data and the potential pitfalls of GPS probe vehicle studies.

As a demonstration of the flexibility of OPEN-TS3 two adaptive signal control strategies are also successfully implemented. The adaptive control strategies are tested on three different networks under varying volume conditions. Based on the OPEN-TS3 simulations it was seen that adaptive control can provide superior overall performance, but can have a significantly greater range of variability than that of pre-timed control.

ACKNOWLEDGEMENTS

The authors recognize that support was provided by a grant from the U.S. Department of Transportation, University Transportation Centers Program, to the Southwest Region University Transportation Center which is funded 50% with general revenue funds from the State of Texas.

EXECUTIVE SUMMARY

Over the years, several trade-offs have been made to gain the speed and efficiency required for simulations to be cost-effective and gain general acceptance in transportation practice. Transportation simulation packages have become "black boxes" and little knowledge of the underlying principles, strengths, and weaknesses of the models is required for a user to generate simulations. Even when detailed, up-to-date documentation is desired, it is often difficult to obtain and comprehend. Also, today's transportation simulation packages are often inflexible in terms of the type of scenarios they are able to evaluate. The likelihood of an end user being able to conceptualize, design, and develop a simulation beyond the bounds set by the simulation package is extremely low. This research effort investigates the feasibility of constructing a transportation simulation approach that minimizes the black box problem and increases modeling flexibility while still providing an easy-to-use package in which typical, straightforward models may be quickly and accurately built. The developed approach is Open-TS3 (Open Architecture Transportation Simulation, 2003).

The first part of this effort was a review of existing transportation simulations and signal control methodologies. This review provides an understanding of the strengths and weaknesses of transportation simulation and signal control (including pre-timed, semi-actuated, actuated, and adaptive control) as practiced today. Importantly, this review also presents general simulation concepts, beyond strict transportation applications. It is from this general review of simulation that the foundation for Open-TS3 arises.

Open-TS3 utilizes the SIMAN simulation modeling language, and ARENA, a graphical user interface for SIMAN. Open-TS3 is an intuitive and flexible simulation modeling architecture, utilizing a hierarchical object-oriented approach to simulation. Open-TS3 may be briefly described as follows:

Transportation – Currently individual intersections, arterials and networks may be analyzed with Open-TS3, under pre-timed, actuated, and adaptive signal control.

Hierarchical – Open-TS3 consists of three tiers of blocks (objects). The top tier blocks may be combined to create an intersection, arterial, or network simulation model. Each of these tier 1 blocks is constructed from hierarchy tier 2 and tier 3 blocks. Tier 3 blocks (the lowest tier) are the basic SIMAN constructs.

Event-Based – Time progresses from event to event, not continuously or through pre-selected time steps.

Object-Oriented – In this approach, one first considers the simulation in terms of objects and how those objects interact with each other, rather than coding structures or program implementation.

Stochastic – While an Open-TS3 model may be specified to be completely deterministic, it is also possible for a user to introduce randomness.

Simulation – Open-TS3 is a simulation, i.e. a computer representation of a real world system (currently existing or planned) that attempts to mimic real world behavior.

As a result of its approach to simulation, Open-TS3 has several advantages. Firstly, within the limitations to be discussed, it is readily expandable to other aspects of the transportation system. Much of this expansion potential is a result of Open-TS3's hierarchical, object-oriented structure. Users may develop their own blocks and employ them with the existing

Open-TS3 blocks. This "open architecture" approach frees a user from a dependence on Open-TS3 developers.

The hierarchical nature of Open-TS3 also allows for a minimal learning curve to initial model construction. One may quickly become efficient with tier 1 blocks, learning as little or as much as desired about the underlying logic (tiers 2 and 3) and still be able to construct a realistic, usable model. As users desire to expand beyond the default tier 1 blocks, they can learn and experiment with tier 2 and 3 blocks, performing more unique analyses. Finally, the object-oriented approach to modeling represents a more "common sense" approach to simulation, utilizing people's existing natural mechanisms for viewing their surroundings.

As part of the development process Open-TS3 was successfully compared against both the well-respected CORSIM model and real-world data. The CORSIM validation studies included comparisons of simulated volumes, delays, queues, and speeds. Agreement was seen between Open-TS3 and CORSIM, except in two instances: overcapacity demand scenarios and left-turn operations. When compared to real-world data Open-TS3 was able to reasonably capture conditions for a twelve-intersection network in downtown Chicago.

As a final step in this initial Open-TS3 development, the openness and flexibility of Open-TS3 was highlighted by incorporating adaptive signal control. The feasibility of implementing an existing adaptive control strategy and enhancements into Open-TS3 was demonstrated. The adaptive control was shown to be capable of providing superior average network performance when compared to pre-timed control. However, adaptive control can also have a significantly greater range of performance variability.

The development and testing of Open-TS3 has led to an understanding of limitations in the model, both in real-world traffic features (either not captured or poorly captured), and in general limitations to the simulation approach. Open-TS3 is still far from the capabilities of commercial transportation simulation packages, such as CORSIM and VISSIM, and should not be considered a competitor in the commercial market place. The primary area of current usage for Open-TS3 is in simulation research and development. Some limitations include;

Traffic features - Currently Open-TS3 is limited to typical arterial networks. Features not modeled include: permissive phasing, sign control, vehicle overtaking (i.e. vehicles may not pass a slower moving vehicle in the same lane), and pedestrian or alternative modes. Also, the lane changing and shared lane logic are based upon idealized assumptions.

Vehicle Queuing - Open-TS3 utilizes vertical queuing which can fail to adequately capture several aspects of realistic system performance including; not limiting the queue length by the approach link length, upstream crossing movements are not be blocked by spillback, and vehicles may enter a turn bay when in actuality the turn bay may be full or the vehicle may be trapped in its current queue, upstream of the turn bay.

Event-Based - While event-based simulation is well suited to modeling signal control, it is not nearly as apt at capturing the interactions that take place at a microscopic level in traffic flow. This weakness will become particularly constraining when attempting to model freeways and to implement detailed car-following models.

Scalability - Both the computational time and potential maximum model size must be considered. The additional overhead of the ARENA simulation package will not allow for the construction of simulations of the size and speed that may be generated directly by lower level languages such as C++.

Open Architecture - While this approach attempts to open the "black box" by allowing the user to add to and alter the underlying objects, it must not be assumed that this will be a simple task. To fully understand the model constructs, the user will have to devote time and effort into gaining an understanding of general simulation development and the underlying SIMAN language. Without this effort a user may still construct complex models using the tier 1 blocks, but the model will be no less a "black box" than the other available simulation packages.

This research aids academics, researchers, and potentially practitioners. Open-TS3 opens the "black box" of transportation simulation modeling while maintaining the capabilities for which the box was initially created. The capability to expand the abilities of Open-TS3 is not limited to any particular developer. Open-TS3 provides an intuitive, flexible, and open simulation modeling architecture that may be applied to problems other than those studied in this report, outside of traditional transportation analysis. Finally, as additional features are generated transportation engineers and modelers will be increasingly benefited. Using the Open-TS3 approach, transportation professionals will be able to readily and efficiently construct simulation models (such as those built with current "black box" models) using the existing Open-TS3 blocks, while maintaining the ability to develop model objects for transportation features not currently included.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1 INTRODUCTION

BACKGROUND

Full utilization of all elements of our existing transportation infrastructure is a primary concern, particularly as traffic demands grow and funding continually falls further behind needs. The majority of urban transportation networks are composed of arterial and lower functionally classified streets that are controlled by traffic signals. These signalized intersections act as capacity constraints (potential or actual bottlenecks) on most streets.

Simulation is increasingly becoming the primary tool by which traffic engineers, planners, researchers, and other practitioners analyze signal systems. Numerous transportation-specific simulation models are in existence today, with existing models in a seemingly endless cycle of development and new models continually being produced. Simulation to many practitioners has become a "connect-the-dots" task, where one is sheltered from the inner workings of the simulation model. Rarely does a practitioner develop a simulation from scratch to address a specific problem, instead the problem is made to fit the bounds of the existing simulation packages. CORSIM, VISSIM, INTEGRATION and TEXAS are excellent examples of simulation packages experiencing wide usage today.

Over time, the analysis required to gauge the potential of arterial traffic system operations has grown increasingly complex, further increasing the demand for simulation. Significant effort has been placed in developing traffic signal control strategies that adequately address intersection traffic demand beyond what is capable with pre-timed, semi-actuated, and actuated control. Adaptive control is one of the most promising control methods. It is a category of control that has been developed with the continuing advancement in hardware, software, and traffic flow theory. In a general sense, adaptive control includes any control strategy that attempts to respond to fluctuations in traffic demand in a manner that provides signal operation superior to that of isolated actuated control or coordinated off-line control. Many adaptive control strategies attempt to provide superior performance by "adapting" to traffic conditions in a real-time manner through the utilization of some type of traffic predictive model. Clearly, adaptive control strategies can be very complex, resulting in their being extremely difficult to test outside of a simulation environment, short of actual field implementation.

MOTIVATION AND PROBLEM STATEMENT

Coordinated and adaptive signal control strategies typically require some level of testing and optimization through simulation. Several trade-offs have been made to gain the speed and efficiency afforded by these simulation packages. The transportation simulation packages have created what has become widely know as the "black box" phenomenon. A modeler can enter data and receive results with little understanding of how a simulation operates and its inherent assumptions. Even if a modeler desires to learn these details it can be a nearly impossible task. Detailed, up-to-date documentation is often difficult (if not impossible) to obtain and difficult to comprehend. Also, the user is allowed to skip the fundamental learning that is required in "ground up" simulation development, which results in a murkier understanding of the weaknesses and strengths of simulation. Even the more knowledgeable user finds modeling flexibility greatly reduced. It is virtually impossible for an end user to conceptualize, design and develop a simulation beyond the bounds set by the simulation package. Clearly there exists the need for a simulation modeling approach that minimizes the black box problem and increases modeling flexibility while still providing an easy to use package in which typical, simple models may be quickly and accurately built. The approach proposed in this research, hierarchical object-oriented simulation, offers great potential for building such a model.

Beyond the general "black box" and minimal flexibility drawbacks to existing transportation simulations is the inability of these simulations to model advanced signal timing control strategies. Implementation of signal coordination is a common approach jurisdictions take when they wish to improve traffic signal operations. Often this coordination is based on signal settings that are determined off-line, utilizing one or several of the many computerized aids to achieve some desired objective function (minimize delay or stops, maximize bandwidth, etc). This approach to signal coordination does work well under certain circumstances:

- there is little cycle-to-cycle variation in traffic demand
- there is minimal platoon dispersion (i.e. little variation in vehicle speeds)
- coordinated arterials are either one-way or have intersection spacing conducive to coordination (i.e. for two-way progression, travel time between intersections is a half multiple of cycle length)

The off-line approach does have drawbacks. Even though available off-line software continues to become more user friendly, ability to utilize these software programs and implement the results is still limited by available staff time of agencies maintaining the signals. Also, traffic varies with time. While some general predictability does exist, such as higher volumes during the evening peak, there is also random, or stochastic, volume variation during these times. Over the short term (i.e. several cycle lengths) traffic variation is stochastic. Over longer periods (months to years), general trends change, making once optimal off-line plans no longer applicable to the network for which they were originally designed. Adaptive control has the potential to efficiently assign green time accounting for these variations. Adaptive control should, in a real time manner, adjust to the immediate changes in traffic flow while also adjusting to the changing trends in traffic flow over time. Thus, any simulation must have the ability to incorporate adaptive signal control.

OBJECTIVES AND RESEARCH TASKS

This research effort consists of four main components.

1) *Review of existing transportation simulations and signal control methodologies.* Prior to the development of a transportation simulation, a general review of simulation and simulation practice in transportation is performed. Specific focus is dedicated to the flexibility and openness of the existing simulation approaches. A review of existing research in adaptive control is also undertaken.
2) *Development of a simulation test bed with intuitive user interfaces and transparent, ready access to internal simulation model operations.* A simulation is developed that addresses the "black box" phenomenon, allowing the user flexibility in simulation model development, while maintaining intuitive and efficient model construction capabilities. The platform for this simulation is SIMAN, a simulation modeling language, and ARENA, a graphical user interface for SIMAN. Both of these products are commonly utilized in the manufacturing industry but have not been utilized to their full potential in the transportation field.
3) *Validation of simulation through the use of real world data.* While it is important that a simulation be open and flexible, the primary objective of simulation is to reasonably

model real-world events. To demonstrate that the developed simulation may be used to reliably reflect real world traffic operations, a twelve-intersection grid in downtown Chicago is simulated. Performance measures from the simulation are then compared against actual performance measures collected in the field.

4) *Development and implementation of adaptive control strategies in the simulation and testing of the strategies on several networks under several traffic flow conditions.* Development of an implementable adaptive control strategy is a large and formidable undertaking that may incorporate many different features and enhancements in attempts to best manage a traffic control system. This task demonstrates the feasibility of implementing an existing adaptive control strategy into the simulation and also incorporating some additional enhancements. Future extensions of this work may include incorporation of additional phasing options, alternative vehicle detector placement, and transit interference (or priority) into the adaptive control along with the possibility of integrating adaptive control with route choice and traffic assignment systems.

   The principal focus of the adaptive control strategy in this research is near and under-capacity systems. In these flow regimes strategies that minimize delay, queue length, or other changing traffic parameters are typically the most likely to improve traffic operations. Several adaptive control strategies were considered for implementation into the developed simulation. Controlled Optimization of Phases at an Intersection (Sen and Head, 1997) was selected.

   As part of this task, the adaptive control strategy is tested in the developed simulation. The strategy is tested on three different networks: a two-way arterial, two intersecting arterials, and a grid of intersections.

OVERVIEW OF THIS DOCUMENT

This document contains seven chapters. Chapter one provides an introduction to transportation simulation and adaptive control, outlines the motivations for this study, and provides a brief overview of the tasks undertaken. Chapter two provides a review of transportation simulation, the SIMAN simulation modeling language, and the ARENA simulation modeling tool. Chapter three describes how SIMAN and ARENA are utilized in the development of Open-TS3, an Open architecture Transportation Simulation, version date 2003. Chapter four presents validation of the simulation utilizing data for a real world network in downtown Chicago. Chapter five provides a review of traffic signal control. Chapter six presents the implementation of adaptive control into the developed simulation and the results for three different test networks. Finally, chapter seven summarizes the effort, highlighting both successes and drawbacks, and presents guidance for future direction of this research.

CHAPTER 2 TRANSPORTATION SIMULATION, SIMAN SIMULATION LANGUAGE, AND THE ARENA MODELING ENVIRONMENT

INTRODUCTION

Simulation Defined

Simulation is a widely utilized analysis tool, limited to no one field of study or area of expertise. Manufacturing, microchip development, chemical engineering, operations research, transportation engineering, and many other areas have all benefited from the use of simulation. While each discipline may approach simulation differently, Schriber, as referenced by Pegden et al. (1995), offers an excellent universally applicable definition of simulation; "Simulation involves the modeling of a process or system in such a way that the model mimics the response of the actual system events that take place over time." Pegden et al. further clarifies the definition of simulation to include the model construction and experimental use, describing simulation modeling as "an experimental and applied methodology that seeks to accomplish the following: describe the behavior of the systems, construct theories or hypotheses that account for the observed behavior, and use the model to predict future behavior."

Simulation In Transportation

The vast majority of traffic engineers, planners, researchers, and other practitioners view simulation as an important tool in the analysis of transportation systems. Numerous general and transportation-specific simulation models are in existence today, with existing models in a seemingly endless cycle of development and new models continually being produced. Simulation has been utilized in the planning and development of almost all sectors of the transportation field, including urban and rural areas, isolated intersections, arterials, freeways and entire roadway systems, pedestrian movements, airline/airport planning, evaluation of control strategies (from congestion management to new signal timings), and evaluation of toll plazas. Clearly, simulation will continue to be an important part of the transportation field. (May 1990; Pedgen, Shannon, and Sadowski, 1995; List and Troutbeck, 1999).

Chapter Overview

Taking the continued utilization of simulation as a given, the purpose of this chapter and the next is not to debate the merits of simulation, but to review transportation simulation as it currently exists and to present a new approach to future transportation simulation. For this research, simulation is generally limited to intersections and street networks. Based on these efforts, future research will hopefully expand the proposed simulation approach to other areas in the transportation field.

The first half of this chapter will briefly present some of the main transportation simulation packages in use today, provide a general discussion of simulation methodologies, discuss advantages and disadvantages of simulation in general and simulation as practiced in the transportation community, and state what the proposed new approach to simulation hopes to achieve. The second half of this chapter includes a discussion of a simulation language, SIMAN, and a general simulation package, ARENA, both of which are commonly used in the manufacturing industry and together, make up the foundation upon which the simulation developed for this research is built. Chapter three will present the developed simulation.

SIMULATION PACKAGES AND METHODOLOGIES

Before discussing simulation as a transportation analysis tool, it is important to clarify the difference between simulation packages and simulation methodologies. First discussed in this section will be simulation packages and then simulation methodologies. This discussion will show that simulation packages are a direct outgrowth of simulation methodologies, with the general tradeoff between the "ground up" methodological approach and the "black box" package approach being greater flexibility and understanding versus significantly reduced time and effort to final simulation execution.

Simulation Packages

Typically practitioners (engineers, planners, etc.) utilize simulation packages. When a practitioner (or end user) uses a simulation package, most of the simulation development (to be discussed in the methodological approach) efforts have already been completed. The user is performing more of a "connect-the-dots" task, sheltered from the details of simulation development. The user is unconcerned with (or at least unable to alter) the distributions, employed sampling techniques, car-following equations, etc., embedded within the simulation. The simulation itself is often a "black box" into which geometrics, traffic control, and demand are input and measures of effectiveness (such as speed, stops, emissions, etc.) are output. A few of the more well-known simulation packages are CORSIM, WATSim, INTEGRATION, TRANSYT-7f, TEXAS, and EVIPAS. The following provides a brief introduction to each of these models.

- *CORSIM* is considered the "grandfather of all [simulation] programs" (Sabra and Stockfish, 1994). It is a microscopic (i.e. simulates individual vehicles), stochastic simulation integrating an urban-network simulation (NETSIM) and a freeway simulation (FRESIM) (Wang and Prevedourous, 1998). CORSIM is a fixed-step time-based model, updating simulation elements (vehicle location, signal indications, vehicle speeds, etc.) every second. Vehicle behavior is controlled by car-following, queue-discharge and lane switching rules (Helali and Khan, 1994). The stochastic portions of the model include driver behavior (i.e. levels of aggressiveness), vehicle characteristics, vehicle paths, free-flow speed, and queue discharge headways. CORSIM can model isolated intersections, arterials, networks, and freeways and it supports both fixed time and actuated control. Measures of effectiveness (MOEs, i.e. outputs) include delays, queue length, queue time, number of stops, stop time, travel time, and speeds. CORSIM also provides animation of the simulation.
- *WATSim* (Wide Area Traffic Simulation) is a microscopic, integrated simulation, essentially an extension of NETSIM to include freeways and ramps. WATSIM applies different car-following and lane-changing logic to freeway and surface links. Additional features beyond those found in NETSIM include high-occupancy vehicles, light-rail, ramp metering, and toll plaza modeling. (KLD Associates, 2002; Wang and Prevedourous, 1998)
- *INTEGRATION* is mesoscopic, modeling aggregate speed-volume interactions of traffic. It does not include any lane changing, car-following behavior, or actuated signal control. It is a route-oriented model, with vehicle origin-destination pairs and departures required as input to the model.

- *TRANSYT-7f* (Traffic Network Study Tool) is unique in that from its inception it has strived to provide both optimization and simulation. The simulation is macroscopic (considers platoons rather than individual vehicles) and deterministic (Rakha and Van Aerde, 1995). A detailed review of TRANSYT-7f is offered in the literature review in chapter four.
- *TEXAS* is a microscopic simulation. TEXAS can be used to model isolated intersections, diamond interchanges, and exchange ramps. The model consists of three main parts, a pre-simulation geometry processor (computes paths and plots through the intersection), a pre-simulation driver vehicle processor (produces list of driver/vehicle pairs based on several classes of driver and vehicle types and input traffic volumes), and a simulation processor which forecasts vehicle positions, velocity and acceleration at each time step (Lee, Rioux, and Copeland, 1977; Lee, Grayson, and Copeland, 1977).
- *DYNASMART-P* (Mahmassani et. al., 2003) combines both network assignment models and traffic simulation models allowing for its use in both network planning and operations decisions. Urban and metropolitan networks that experience congestion are the primary area of focus. The simulation is a hybrid traffic simulation approach, utilizing macroscopic traffic flow relations to move individual particles. Individual trip-maker decisions, including route, departure time, and mode are simulated. Also included are the ability for trip chaining and multiple user classes.
- *TRANSIMS* (TRansportation ANalysis and SIMulation System) (Los Alamos National Laboratory, 2002) is an integration of travel forecasting models and simulation. TRANSIMS utilizes micro-simulation to simulate movement in one-second intervals. Entire travelers trips (including multiple stops) may be simulated including pedestrian, transit, and vehicle movements trip portions. A Cellular Automata (particle-hopping) model is utilized to model the vehicle movements and interactions.
- *VISSIM (VISSIM, 2003)* is a traffic and transit microscopic, time based simulation model, originally developed in Germany. VISSIM is capable of modeling arterials and freeways (including roundabout and transit operations), incorporates multiple modes (car, bus, pedestrian, trucks, and light and heavy rail), and incorporates several intelligent transportation features (variable message signs, ramp metering, transit signal priority, etc.) One of the strongest features of VISSIM is it exceptional graphical representation of traffic.

There are numerous similarities and differences between the models described above. Each model has its own strengths and weaknesses, levels of detail, and approach to simulation. Many studies have been performed that highlight the strengths and weaknesses of the different models and propose the best model to use under different circumstances. Many papers, texts, and manuals have been written on these models. The reader is referred to the reference list for more detailed information on any particular model.

Simulation Methodologies

In contrast to a simulation package, a simulation methodology is a basic approach to simulation development (List and Troutbeck, 1999). Using a simulation methodology, or paradigm, developers (programmers, researchers) build simulations "from the ground up." A simulation methodology can be applied to a wide variety of simulation problems. Simulation methodologies start from a very basic level where the issues addressed include event-based vs. time-based simulation, distribution selection and implementation, underlying vehicle movement (i.e. car-following equations, Newtonian mechanics, acceleration / deceleration parameters, etc.), selection of programming language, and programming technique. List and Troutbeck list four basic paradigms for simulation development: program code, flowcharts, pseudo code, and worksheets, although in practice initial development of any simulation will most likely involve a combination of two or more of these approaches. In practice a simulation developed in this ground up fashion is then sometimes converted in a simulation package, where the users interact with the simulation through interfaces constructed by the developer. As stated early, at this point the user may not access the code and is confined to the simulation as constructed by the developer.

An example of a methodological simulation development process could proceed as follows. First, the developer uses a flowchart to diagram the logical flow of the simulation. This graphical presentation shows the logical structure, decision points, input/output and processes. The next step is the development of pseudo code. Pseudo code transforms the flow chart into a statement-by-statement description of the simulation process in its logical order, including loops, decisions, and sub routines. Ideally, pseudo code is computer language independent. It does not specifically address steps but describes process thoughts. That is, in pseudo code one would write "create a vehicle," but would not list the particular computer code required to complete this action. Pseudo code may include items such as drawing numbers from a particular distribution and using car following equations. The final step in the initial development of a simulation is the conversion of pseudo code into executable computer code (FORTRAN, C++, etc). List and Troutbeck offer an excellent discussion of paradigms and steps in the simulation development process.

In their efforts List and Troutbeck also offer one other intersecting new approach to simulation, Petri Nets (new at least to transportation simulation). Petri Nets provide a concise, unambiguous, graphical presentation of simulation logic. While a discussion of Petri Nets is beyond the scope of this report, several comments are warranted (detailed explanations of Petri Nets may be found in references). In many ways the Petri Net approach is similar to the simulation approach proposed in this research. It is possible that the overall characteristics found desirable in the proposed simulation could have been developed utilizing the Petri Nets concepts, as opposed to SIMAN concepts. The primary advantage to SIMAN is the well developed ARENA software, which offered an existing, supported simulation platform from which to build upon. This allowed shorter development time than would have been required with petri nets. The trade-off, as recognized by List et al., is that programs such as ARENA are proprietary in nature. The financial costs associated with the acquisition of ARENA (or any similar platform), which can be significant, will be passed on to the user of a model such as the one developed in this research.

SIMULATION ADVANTAGES AND DISADVANTAGES

Prior to development of any successful simulation, the general simulation advantages and disadvantages encountered in previous simulation development efforts should be examined. Pegden et al. offer a concise discussion of simulation advantages and disadvantages. While the advantages and disadvantages (listed below) apply to simulation in general, they may be readily applied to transportation simulation applications. (A few transportation examples have been added.) Clearly this is not an exhaustive list of every possible advantage or disadvantage of simulation, but it does provide a meaningful overview of what one should expect.

The advantages of simulation described by Pegden include:

- New policies, operating procedures, and decision rules can be tested without disrupting existing activities. (i.e. testing new signal timing coordination plans, adaptive control strategies, etc.)
- New hardware design, and physical layouts may be tested before committing resources. (i.e. additional turn bays)
- Time can be controlled, sped up or slowed down.
- An understanding of the effect of specific variables and variable interaction on the system may be gained. (i.e. interaction between demand and green times)
- Bottlenecks may be identified.
- Models may be manipulated to aid in gaining an understanding of potential future or rare events, i.e. the ability to explore "what if" questions.

The disadvantages in simulation cited by Pegden et al. include:

- Specialized training is required for model builder.
- Simulation analysis may be expensive and time-consuming.
- Simulation results are difficult to interpret, as it is often difficult to determine if a model has captured a significant relationship in the system or the effects of randomness built into the model.

To alleviate some simulation disadvantages, especially extensive training requirements and excessive development time and costs, the practicing transportation community primarily relies on simulation packages. Extensive resources (both in time and funds) have been expended in developing transportation simulation packages. With the inclusion of graphical user interfaces many models are approaching "plug and play" capabilities, with a user being able to very quickly and cheaply learn how to build and run a real model. Unfortunately, the use of these simulation packages has disadvantages of its own, most notably the "black box" phenomenon and reduced modeling flexibility.

The "black box" problem is widely recognized. A modeler can enter data and receive results with little understanding of how the simulation operates and limited knowledge of the inherent assumptions. Even if a modeler desires to learn these details it can be a nearly impossible task. Detailed, up-to-date documentation is often difficult (if not impossible) to obtain and difficult to comprehend. Also, the user is allowed to skip the fundamental learning that is required in "ground up" simulation development, which results in a murkier understanding of the weaknesses and strengths of simulation.

As mentioned previously, simulation packages also greatly reduce overall modeling flexibility. A user is bound by the methods and assumptions of the given simulation package. It is virtually impossible for an end user to conceptualize, design and develop a simulation beyond the bounds set by the simulation package developer. Users are essentially unable to change or set up the model in any way not allowed by the developed user interface. If one desires to address a weakness or limit in the model he is required to obtain the source code, which can be proprietary and therefore impossible to access, or if accessible is often poorly documented and extremely difficult with which to work. Attempting to alter the underlying code of a simulation package can be disastrous. Even seemingly simple changes can become so daunting that they are never undertaken.

Mystkowski and Khan (1999) provide an excellent example of the difficulties of working with existing simulation packages. Mystkowski et al. compared queue length estimations from CORSIM, TRANSYT-7f, SYNCHRO3, PASSER-II-90 and SIGNAL94 with field data measurements from seven different locations. 0, taken from Mystkowski et al, provides a summary of the queue length estimation study.

Table 2.1 Estimating Queue Length Using SIGNAL94, SYNCHRO3, TRANSYT-7f, PASSER II-90, and CORSIM

		ANALYSIS PACKAGE					
INTERSECTION	Westbound Approach v/c	Compared to Observed Average Maximum Queue Length				Compared to Maximum Queue Length Seen During Observation Period	
		PASSER II-90	CORSIM	SYNCHRO3 (50%)	TRANSYT-7F	SIGNAL94	SYNCHRO3 (95%)
Speer/Colfax	0.740	⊕↓	●	⊕↑	●	●	●
County Line Road/ Quebec Street (through lanes)	>1	⊕↓	●	●	●	◒	◒
County Line Road/ Quebec Street (left lanes)	>1	⊕↓	●	●	●	●	◒
County Line Road/ Parkway Drive	0.526	◒	●	⊕↑	⊕↑	⊕↓	●
Blake Street/ 17th Street	0.494	⊕↑	●	⊕↑	●	◓	⊕↑
Blake Street/ 19th Street	0.349	●	◓	◓	●	●	◓
Blake Street/ 18th Street	0.374	◓	⊕↑	◓	⊕↓	⊕↓	◒

● Accurate Estimation (within two vehicles)

◓ Slight Overestimation (within three vehicles)

◒ Slight Underestimation (within three vehicles)

⊕↑ Overestimation by four or more vehicles

⊕↓ Underestimation by four or more vehicles

Source: Mystkowski, C. and Sarosh K. Estimating Queue Lengths by Using SIGNAL94, SYNCHRO3, TRANSYT-7f, PASSER II-90, and CORSIM, Transportation Research Record 1683, Washington D.C. 1999, Table 6, p. 115.

For this discussion the crucial point is not how well each model performs under each circumstance but that every one of these models did poorly in at least one instance. These are not new or poorly supported models, many have been under development for years, yet poor performance is still seen in some instances. This is not unexpected as it would be nearly impossible for any program developer to anticipate and accurately capture every possible situation. The difficulty arises in that the end users have no reasonable method to address a deficiency. What is lacking in these packages is the flexibility to address those situations not well handled by the simulation package "as is".

PROPOSED SIMULATION MODELING APPROACH GOALS

The simulation model developed in this research has two primary goals. First, the model lays the foundation for a transportation simulation approach that minimizes the "black box" problem and increases modeling flexibility while still providing an easy to use package in which typical, simple models may be quickly and accurately built. For example, at the present time a user who desires to study a real-world but non-typical situation is left with two stark choices: either work as well as possible within the confines of the existing simulation software packages or completely develop a new simulation that addresses the problem at hand. A more desirable approach to simulation would allow for the use of constructs from other models where applicable, while still retaining the flexibility to incorporate new basic modeling constructs. First, the simulation must provide a platform upon which to investigate adaptive control strategies. Adaptive control is on the leading edge of traffic signal control. Therefore, it is critical that the simulation be able to reliably and efficiently incorporate adaptive control.

In the detailed discussion of the proposed simulation model in the remainder of this chapter, and the remainder of this report, it will be seen how the hierarchical, object-oriented approach leads to such a simulation. For clarity, from this point forward the simulation modeling approach developed in this research will be referred to as the Open-TS3, Open architecture Transportation Simulation, version year 2003.

OPEN-TS3 PLATFORM - SIMAN AND ARENA BACKGROUND

Manufacturing Simulation as a Tool in Transportation

In an effort to solve transportation problems, transportation researchers and practitioners have at times sought tools and insights from other disciplines. The manufacturing industry has proven to be one area that has strong similarities to transportation systems, and thus has high potential for overlap in tools and ideas. At the engineering / operations level, manufacturing and transportation systems are both primarily concerned with competition for limited resources (machines, workers, material-handling / vehicles per lanes, available green time, transit capacity) and the formulation of a model that describes the resources and their interactions (Pegden, Shannon,and Sadowski, 1995). Thus, it is natural to investigate manufacturing simulation approaches for potential crossover to transportation simulation.

Pegden et al. discuss three distinct uses for simulation in manufacturing: 1) a design aid for determining factory layout, equipment decisions, and operating policies 2) A scheduling tool which allows for the exploration of plan changes from current conditions including how to reach optimal operation from current conditions, and 3) testing of real-time on-line system control. There are strong parallels for these categories within transportation. For example, 1) design aid – how many turning lanes, effect of number of toll booths on plaza operations, impact of HOV lanes, 2) scheduling tool – testing new signal control settings, analyzing bus headway changes, testing of signal timings, 3) real-time control – prediction of demand over some time horizon allowing for real-time updates to control plans, impacts of real-time information being given to system users.

Based on these clear similarities between manufacturing and transportation and the disadvantages of transportation simulation packages discussed earlier, it was decided to use SIMAN, a general-purpose simulation language primarily used in manufacturing, and ARENA, a hierarchical simulation platform that automates the creation of SIMAN code, in the development of the Open-TS3. As both SIMAN and ARENA are reviewed in the remainder of this chapter, it will become clear that they have excellent potential to allow for the creation of a transportation simulation (i.e. Open-TS3) that meets the previously stated goals.

SIMAN (Pegden et al, 1995)

As stated above, the Open-TS3 Platform developed for this research is built upon SIMAN and ARENA. Thus the ultimate capabilities of any Open-TS3 models are dependent upon the capabilities of the underlying SIMAN language and ARENA Platform. To gain a thorough understanding of the Open-TS3 platform, one must obtain at least an introductory understanding of the constructs and abilities of these two products. The next few sections of the report provide this background. This discussion is limited to the simulation of discrete systems for this research effort, as this is the primary area of interest. The reader is directed to Pegden et al. for additional information on discrete or continuous system modeling with SIMAN.

SIMAN Overview

SIMAN is a "powerful, general-purpose simulation language for modeling discrete, continuous, or combined systems." (Pegden, Shannon, and Sadowski, 1995) SIMAN is based on theoretical system concepts (Zeigler, 1976) in which the simulation is separated into a "model" component and an "experiment" component. In this framework the model describes the physical elements of the system (number of lanes, lane capacity, bus stop locations, and vehicle characteristics) and the experiment describes the experimental conditions (initial traffic conditions, traffic demands, O-D patterns, statistics to be gathered, and run time). Once defined, the SIMAN compiler links the model and experiment and performs the simulation. Although it is a general-purpose simulation, much of the original development and available literature regarding SIMAN (and ARENA) are slanted towards the manufacturing industry.

Entities, Attributes and Processes

In SIMAN a system is considered in terms of entities, attributes and processes. An entity is any object that moves through a system, changing the system state. As entities enter and leave, the number of entities within a system may change over time. Entities may reflect real world objects (i.e. autos, buses, pedestrians, etc.) or imaginary objects (entities that do not exist in the real-world system but are included for simulation convenience). For example, in Open-TS3 the entities that represent vehicles reflect real world objects while the entities used in the traffic signal control logic have no real world equivalent.

Each different entity type is characterized by a unique set of attributes. For example an auto's attributes may include acceleration / deceleration parameters, gas mileage, and emissions, while a transit vehicle's attributes may include maximum occupancy, acceleration / deceleration parameters, emissions, and route. A process in SIMAN is a set or sequence of activities involving entities. Two examples of processes in a transportation system include traffic signals (a vehicle must pass through an intersection, potentially incurring some delay) and transit stops (loading and unloading of passengers).

Model Blocks

A SIMAN simulation is built by utilizing a standard set of model and experimental blocks to create a block diagram of the process under study. SIMAN contains 10 basic model block types: hold, transfer, operation, queue, branch, pickq, qpick, match, select, and station (0).

Of these model block types the first three are multi-function and the last seven are single function. Each multi-function model block symbolizes a set of similar modeling functions. Hold blocks represent the detaining of an entity based on the current state of the system, transfer blocks represent movement of an entity from one location to another, and operation blocks represent modeling functions where entities perform a function at the block. When utilized in a model a multi-function block is given a block name according to the specific block function. For example three of the most commonly encountered SIMAN operations blocks are the DELAY block which delays entities by specific time, the CREATE block which creates entities and the ASSIGN block which assigns attribute values to entities. Throughout the text specific block names will be referenced in all capital letters.

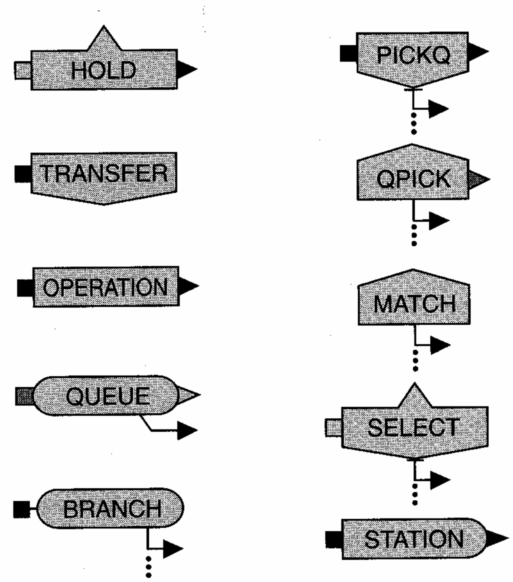


Figure 2.1 Basic SIMAN Model Blocks
Source: Pegden, C. D, Shannon, E. R, Randall, S. P., Introduction to Simulation using SIMAN, Second Edition, McGraw-Hill, 1995, Figure 3-1, p. 62

Each block has different operands depending on the purpose of that particular block. The block's operand values specify the exact block function. For example a basic DELAY block contains two operands, duration (how long an entity will be delayed) and storage ID (area where entities reside during delay).

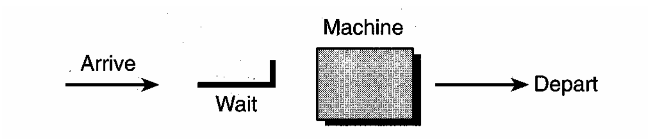
Experimental Blocks

As mentioned, a SIMAN simulation contains both model and experimental blocks. The experimental blocks specify the experimental conditions of the simulation, such as the simulation run length, the number of replicate runs, time to begin statistics collection, and total number of entities that may be created. Also included are blocks for defining the characteristics of resources and queues. For example, a model QUEUE block includes a queue ID, a capacity, and a balk label (overflow point if an entity arrives and the queue is at capacity) while an experimental QUEUES block contains the names of (and/or numbers), and ranking criterion for all queues in the model. The experimental QUEUES and RESOURCES blocks may be thought of as similar to variable declarations used at the beginning of FORTRAN, or C.

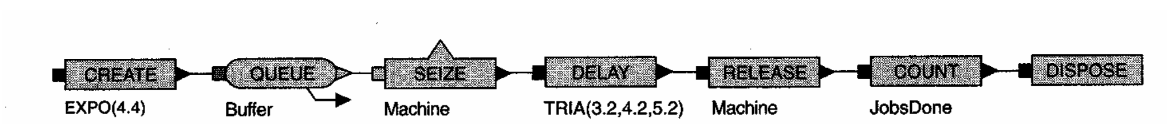
Block Diagrams

A block diagram is similar to a flow chart, where the processes through which an entity moves are mapped in their logical order. Block diagrams may reflect a real-world flow process, i.e. a vehicle (the entity) moving from intersection to intersection (the blocks), or be used to model some logical construct, as will be seen in Open-TS3 signal control logic. Once completed the block diagram is converted into equivalent SIMAN syntax statements (code) for computer execution. One of the advantages of ARENA (and Open-TS3) is that the process of converting block diagrams into equivalent code is automated.

0(a) shows the schematic for a simple workstation. Entities enter the system, arrive at the workstation, wait to be processed, are processed and finally exit the system.



(a)



(b)

Figure 2.2 (a) Schematic for Simple Workstation,
(b) SIMAN Model Block Representation of Simple Workstation
Source: Pegden, C. D, Shannon, E. R, Randall, S. P., Introduction to Simulation using SIMAN, Second Edition, McGraw-Hill, 1995, Figure 3-1, p. 62 and Figure 3-8, p. 84.

0(b) shows the equivalent SIMAN model block diagram representation of the simple workstation. In this model an entity is created in the CREATE block and then immediately proceeds to the QUEUE block. If the server is free and there is no queue present in the QUEUE block, the entity proceeds to the SEIZE block where it seizes the server, otherwise the entity joins the QUEUE block queue. Once the entity seizes the server it proceeds to the DELAY block where it is delayed by the service time. After this delay the entity proceeds to the RELEASE block and releases the server, then to the COUNT block where desired statistics are collected, and finally to the DISPOSE block, where the entity leaves the system.

At each block the exact functioning is controlled by the block's operand values - the values describing the block characteristics. In object oriented programming parlance a block diagram's operands may be thought of in a similar manner to that of an object's properties. For example, the CREATE block's operands are batch size (how many entities are created at once), offset time (the time from the start of the simulation to the creation of the first batch), interval time (the time between batch creations) and maximum batches (the maximum number of batches that may be created during the simulation). When an operand requires a time interval (or other numeric quantities) it is possible to either select a deterministic value or allow for a random number (chosen from a selected distribution) to be drawn for each entity upon arrival to the specific block. One or two operand values are typically shown for each block within a block diagram, avoiding the clutter and confusion that would be created if all operand values were listed. All operands must be accounted for when a block diagram is converted into SIMAN syntax.

Not shown in 0(b) but also required for any SIMAN simulation model are the experimental blocks. The experimental blocks do not graphically connect to the model block diagram but act separately to store model wide, general information. A set of experimental blocks that might be used with the example in 0(b) includes:

PROJECT - lists title, analyst's name, date, and summary report name
DISCRETE - sets the maximum number of entities that may be created
QUEUES - defines the queue name, number, and ranking criterion
RESOURCES -defines resource name, number, and capacity
COUNTERS - defines counter name, number, limit, and output file
REPLICATE - lists number of replications, begin time, replication length, and warm-up period

SIMAN Syntax

Typically a developer would design a model using block diagrams, then convert the block diagrams into SIMAN syntax for compiling and execution. As ARENA automates this conversion of block diagrams into syntax, developers are in many ways sheltered from the actual process of writing "code". If one can develop the logical flow structure through block diagrams, the code is written for them. However, in the development of simulation models, knowledge of the code structure can be very useful in the more advanced and efficient simulation model debugging tools and techniques.

As an understanding of SIMAN syntax is not critical to an understanding of ARENA or Open-TS3 no further discussions of syntax will be offered in this report. For completeness 0 gives the syntax for the discussed example. The interested reader is referred to Pegden et al. for further explanation and information.

```
BEGIN;
        ;   SAMPLE PROBLEM 3.1
        PROJECT,      Sample Problem 3.1, SM;
        DISCRETE,     100;
        QUEUES:       Buffer;
        RESOURCES:             Machine;
        COUNTERS:  JobsDone;
        REPLICATE:  1, 0, 480;
END;
```

(a) Experiment Source File

```
BEGIN;
        CREATE:       EXPO(4.4);               Enter the system
        QUEUE,        Buffer;                  Wait for the machine
        SEIZE:        Machine;                 Seize the machine
        DELAY:        TRIA(3.2,4.2,5.2);       Delay for process
        RELEASE:      Machine;                 Release the machine
        COUNT:        JobsDone;                Count completed jobs
        DISPOSE;                               Depart the system
END;
```

(b) Model Source File

Figure 2.3 (a) Experiment Source File, (b) Model Source File
Source: Pegden, C. D, Shannon, E. R, Randall, S. P., Introduction to Simulation using SIMAN, Second Edition, McGraw-Hill, 1995, Figure 3-9, p. 85 and Figure 3-10, p. 93

SIMAN Execution

In general, discrete simulations built with SIMAN are event based. During the running of a simulation, SIMAN will maintain an event stack (also referred to as an event calendar). Events, such as the creation of an entity, and an entity departing a block are scheduled in the event stack according to the time at which they are scheduled to occur. The simulation progresses in a sequential manner from event to event in the event stack. If more than one event is scheduled at the same clock time all actions are taken before the simulation time is advanced. As each event is completed, the event stack is updated to reflect the completion of this event and to include any new events resulting from the completed event. For example, if the last completed event was “entity number 2 enters the DELAY block” at time t this action would then be removed from the event stack and the event “entity 2 departs DELAY block” added to the event stack at time t plus the delay duration. 0 provides an example of the beginning steps in the execution of the example given in 0(b). Shown in 0 is the step-by-step manner in which a compiled SIMAN model would execute. The “Just Finished Event” shows the last action to be completed and the “Update Event Stack” reflects the event stack as it would appear immediately after the completion of the corresponding just finished event. Included with 0 are the values for the model and experimental block operands necessary to interpret the sample run.

Model Blocks
CREATE: Batch Size - 1, Offset Time - 5sec, Interval - 5 sec
QUEUE: Queue ID - queue_server, Capacity - unlimited
SEIZE: Resource ID - example_server, Number of Units - 1
DELAY: Duration - 10sec
RELEASE: Resource ID - example_server, Quantity to Release - 1
COUNT: Counter ID - example_count

Experiment Blocks
RESOURCES: Name - example_server, Capacity - 1
QUEUES: Name - queue_server, Ranking Criterion – FIFO (First In First Out)
COUNTERS: Name – example count

Simulation Trace

Row	Just Finished Event			Updated Event Stack		
	Time	Entity	Action	Time	Entity	Action
1	t = 0	-	Initialize Simulation	t = 5 t = 900	1 -	Create Entity End Simulation
2	t = 5	1	CREATE - Create Entity 1, set create next entity at t + 5	t = 5 t = 10 t = 900	1 2 -	Exit CREATE, Enter QUEUE Create Entity End Simulation
3	t = 5	1	Exit CREATE, Enter QUEUE	t = 5 t = 10 t = 900	1 2 -	Exit QUEUE, Enter SEIZE Create Entity End Simulation
4	t = 5	1	Exit Queue, Enter SEIZE - seize resource example_server	t = 5 t = 10 t = 900	1 2 -	Exit SEIZE, Enter DELAY Create Entity End Simulation
5	t = 5	1	Exit SEIZE, Enter DELAY – set exit DELAY at t + 5	t = 10 t = 10 t = 900	2 1 -	Create Entity Exit DELAY End Simulation
6	t = 10	2	CREATE – Create Entity 2, set create entity at t + 5	t = 10 t = 10 t = 15 t = 900	2 1 3 -	Exit CREATE, Enter QUEUE Exit DELAY Create Entity End Simulation
7	t = 10	2	Exit CREATE, Enter QUEUE server busy, enter queue_server	t = 10 t = 15 t = 900	1 3 -	Exit DELAY Create Entity End Simulation
8	t = 10	1	Exit DELAY, Enter RELEASE - release resource example_server	t = 10 t = 15 t = 900	1 3 -	Exit RELEASE, Enter COUNT Create Entity End Simulation
9	t = 10	1	Exit RELEASE, Enter COUNT - update example_count statistics	t = 10 t = 10 t = 15 t = 900	1 2 3 -	Exit COUNT, Enter DISPOSE Exit QUEUE, ENTER SEIZE Create Entity End Simulation
10	t = 10	1	Exit COUNT, Enter DISPOSE – entity 1 leaves the system	t = 10 t = 15 t = 900	2 3 -	Exit QUEUE, ENTER SEIZE Create Entity End Simulation
11	t = 10	2	Exit Queue, Enter SEIZE - seize resource example_server	t = 10 t = 15 t = 900	2 3 -	Exit SEIZE, Enter DELAY Create Entity End Simulation
12	t = 10	2	Exit SEIZE, Enter DELAY – set exit DELAY at t + 5	t = 15 t = 15 t = 900	3 2 -	Create Entity Exit DELAY End Simulation
13	t = 15	3	CREATE – Create Entity 3, set create entity at t + 5	T = 15 t = 15 t = 20 t = 900	3 2 4 -	Exit CREATE Exit DELAY Create Entity End Simulation
…	…	…	…	…	…	…

Figure 2.4 Example Execution, First Fifteen Seconds of Model in 0.

When designing a simulation with SIMAN it is important to be aware of a few processing rules regarding the order in which SIMAN executes events that are scheduled to occur at the same calendar time.

1) SIMAN operates in a First-In Last-Out manner. For example, if five events are scheduled at time t the last event to be scheduled (placed in the event stack) is the first event to be executed.
2) Once processing of an entity has begun that entity will continue to be processed until no further action can be taken at that calendar time. This rule is a direct result of the first rule.
3) After the completion of the processing of an entity, actions may then be taken by other entities, at that calendar time. If these actions result in the scheduling of events at the current calendar time for the original entity these new events will be completed in accordance with rules 1 and 2.
4) Entities that enter a QUEUE (or other Hold block) may not have their next event immediately scheduled due to stochastic model properties. In order to continue the progression of these entities through the simulation, SIMAN reviews the status of each entity in a hold block after each event execution. If, upon this check, an entity's next action may now be scheduled, the event is entered into the event stack and executed in accordance with the event time and above rules.

0 provides the processing of the event stack for the first fifteen seconds of the example in 0. The listed rules may be seen in this figure. For example, in row 5, the current time is t = 5 and the event stack is: at t = 10 Create Entity 2, at t = 10 Entity 1 Exit DELAY, and at t =900 End Simulation. Thus the next action taken (row 6) is the creation of entity 2. Now in accordance with the stated rules the updated event stack is: at t = 10 Entity 2 Exit CREATE and Enter QUEUE, at t = 10 Entity 1 Exit DELAY, at t = 15 Create Entity 3, and at t = 900 End Simulation. Thus entity 2 leaving the CREATE block has entered at the top of the event stack and is now the next event to be executed, not the exiting of entity 1 from the delay block. Entity 1 does not exit the DELAY block until row 8, after entity 2 has been processed as far as possible (rules 1 and 2). Also, note that in row 7 entity 2 enters the queue in the QUEUE block and no next event for entity 2 is registered in the event stack. It is not until row 9, when entity 1 releases the example server resource that entity 2 may be scheduled to leave the queue and seize the resource (rule 4). Finally, notice that entity 2 is initially processed as far as possible (rows 6 and 7) but, after the processing of entity 1 (rows 8, 9, and 10) that additional events may then be, and are, processed for entity 2 at the same calendar time (rule 3).

ARENA

At its most basic, ARENA is a SIMAN development platform, a Microsoft Windows environment in which SIMAN based simulations are created in "drag and drop" manner. ARENA contains a menu of graphical icons that represent the basic modeling blocks in 0. These icons are connected and edited to created a SIMAN model block diagram. ARENA then automates the process of generating the SIMAN syntax represented by the code.

Although, ARENA can be utilized as much more than a SIMAN implementation tool, it is intended to "combine the ease of use found in high-level simulators with the flexibility of simulation languages, even all the way down to general purpose procedural languages like the

Microsoft Visual Basic programming system, FORTRAN, or C…" (Kelton et al). ARENA achieves much of this flexibility through its hierarchical nature, 0. At the lowest level are the SIMAN, C/C++, Visual Basic, and FORTRAN languages. These alone may be used to create a simulation model. The strength of the hierarchical approach is derived from the next levels. In 0 these are the ARENA Templates; Support, Transfer and Common panels. (In later versions of ARENA these templates are revised and referred to as Basic Process, Advance Process, and Advanced Transfer panels.) These panels contain modeling blocks (modules) created form the basic SIMAN blocks found in 0. These blocks allow for an efficient reuse of SIMAN code and more rapid model development. For example a workstation block could be created that consists of the SIMAN block model in 0. Then for a user to place a workstation in a model he would need only place the higher level workstation block, not the seven block as seen in 0(b). Currently these higher-level ARENA panels contain modules designed around manufacturing process.

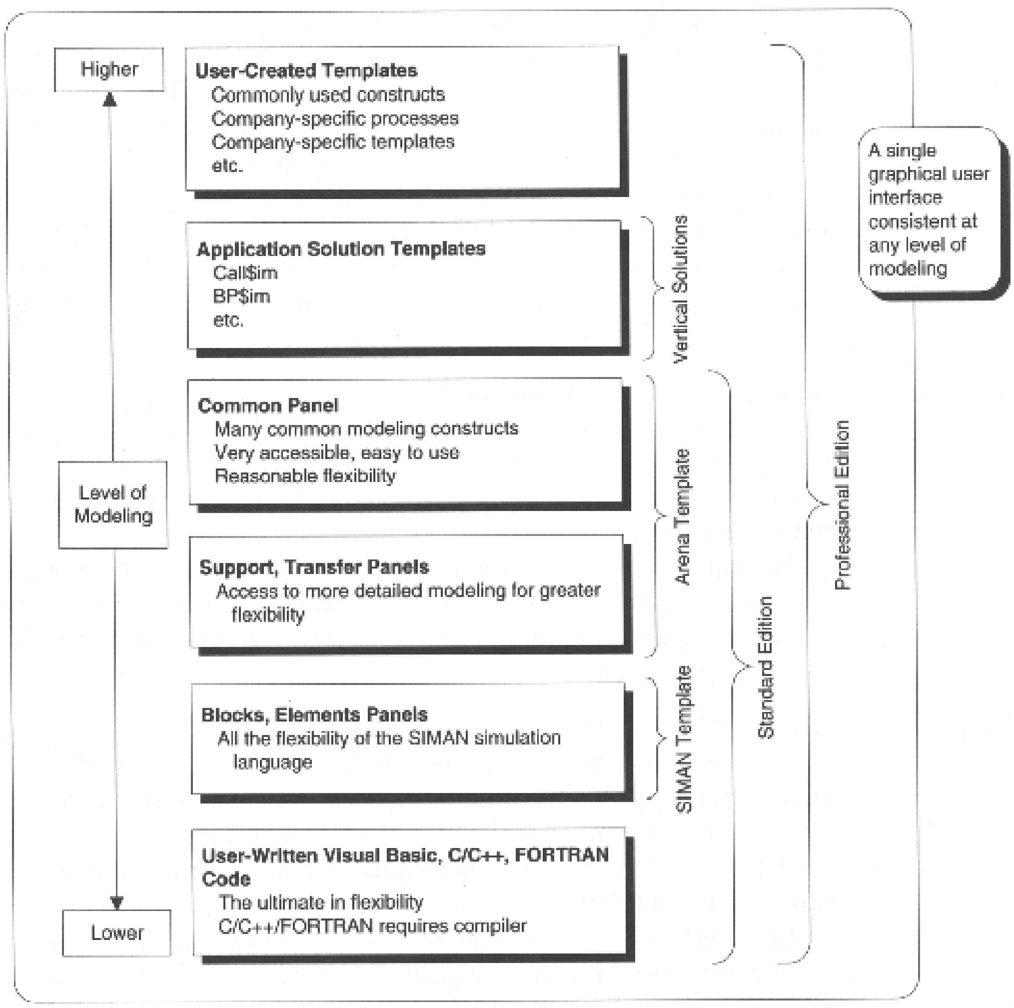


Figure 2.5 ARENA's Hierarchical Structure
Source: Kelton, W. D., Sadowski, P.S., Sadowski, A.S, Simulation with ARENA, McGraw-Hill, 1998, Figure 1-2, p. 12.

The highest level of the hierarchy may contain basic simulations in which a user need only change problem specific information to perform the desired analysis. A common example of this in ARENA literature is the study of call centers, where a basic call center model is developed and an end user needs to only adjust the basic model to their specific situation; number of operators, average call time, operator scheduling, etc.

As discussed in chapter three, Open-TS3 is designed to fill the mid-level hierarchy of 0, replacing the current ARENA templates for manufacturing simulations with templates containing modules designed for transportation simulations, i.e. enter, exit, intersection, signal light, approach lane, etc. Thus a transportation simulation modeler will have the basic building blocks required to quickly construct a general transportation model while still being able to access basic SIMAN / other programming constructs to address more unique simulation problems.

ARENA also contains dynamic animation capabilities. It is possible to animate the movement of entities from block to block in the model and to also create separation animation consisting of stations and routes. These animation capabilities are useful in model development and debugging and particularly desirable in transportation simulation applications. Also, in addition to entity movement animation there are numerous methods by which output statistics may be analyzed.

A final note should be made about ARENA. The fundamental approach to simulation taken by an ARENA / SIMAN model is very different from that of traditional transportation simulations; the simulation development is object-oriented. It is the object-oriented nature that allows for the modeling flexibility, reusability of components, and creation of more intuitive simulations. It is this same object-oriented nature that Open-TS3 capitalizes on in simulation development.

CONCLUSION

This chapter began by providing the user with a general simulation review. It was seen that simulation has both advantages and disadvantages and acknowledged that simulation has been and will continue to be a tool utilized by transportation professionals. It was shown that the decision to build a transportation simulation leads to a stark choice set, that of utilization of "black box" simulation vs. extensive model development. A user must either select a transportation simulation that in many ways is a "black box" they do not understand or start from scratch to develop their own simulation.

The second half of this chapter reviewed the tools, SIMAN and ARENA, utilized to develop a transportation simulation model, Open-TS3, which bridges the gap between "black box" simulation and extensive model development. By utilizing the hierarchical and object-oriented nature of ARENA / SIMAN it is possible to develop a modeling approach in which a user may quickly build a transportation simulation model utilizing existing transportation modules while having available to them a set of basic concepts that may be used to address specific, unique problems. The remainder of this report is devoted to the development and validation of this simulation model (Open-TS3)

CHAPTER 3 DEVELOPMENT OF A TRANSPORTATION HIERARCHICAL EVENT-BASED OBJECT-ORIENTED STOCHASTIC SIMULATION

OPEN-TS3 OVERVIEW

Description

The following categories summarize the main features of Open-TS3: Transportation, Hierarchical, Event-Based, Object-Oriented, Stochastic, and Simulation.

Transportation – Currently Open-TS3 simulates vehicular traffic. Individual intersections, arterials and networks may be analyzed with Open-TS3. Natural future extensions of Open-TS3 include other vehicle facilities such as, freeways, toll plazas and HOV/HOT lanes. Although, future extensions should not be constrained to traffic simulation, as Open-TS3 concepts may be expanded to include Air, Rail, Water and other modes and aspects of the transportation system.

Hierarchical – Open-TS3 consists of three tiers of blocks (objects). The top tier blocks may be combined to create an intersection, arterial, or network simulation model. Currently these blocks are ENTER, EXIT, PRETIMED, PRETIMED8P, ACTUATED8P, SIGNAL, QUEUECHANGE, TURNBAY, LANEADD, and LANEDROP. A SIGNAL block that interacts with adaptive control algorithms also has been developed. Each of these tier 1 blocks is constructed from hierarchy tier 2 and tier 3 blocks. Tier 3 blocks (the lowest tier) are the basic building blocks provided by ARENA, such as ASSIGN, BRANCH, COUNT, ROUTE, etc. Tier 2 blocks (APPROACH, SET TRAVEL TIME, UNBLOCK#/#, BLOCK#/#, PHASE#_PT, PHASE#_AC, and MAX/GAP) are intermediary blocks, constructed from tier 3 blocks with the intent of simplifying the construction and complexity of tier 1 blocks. 0 presents an overview of the Open-TS3 hierarchy. 0 and 0 are object class diagrams which show the hierarchical make up of the PRETIMED, QUEUECHANGE, ENTER, EXIT and TURNBAY blocks.

A user may build a simulation model in a "plug and chug" manner using only the blocks in tier 1, much like a user may use the many existing simulations without ever considering their underlying code. Although, one of the strengths of Open-TS3 lies in the ability of a user to meet specific unique simulation needs by examining and adjusting the underlying tier 1 block logic, constructed from tier 2 and 3 blocks, or create entirely new tier 1 and 2 blocks, without adversely impacting the operation of other tier 1 blocks. For example, if the current stop bar queuing model is not accurately capturing that being simulated, only the queuing model in the APPROACH block must be adjusted. The logic of tier 1 blocks that do not contain an APPROACH block require no adjustment and their operation will not be affected by the APPROACH block changes. Tier 1 and 2 blocks are discussed in detail in section 0. Tier 3 blocks (i.e. ARENA) are presented in Chapter two.

TIER 1
Used in the construction of Intersection, Arterial, and Network models.

Includes blocks: ENTER, EXIT, PRETIMED, PRETIMED8P, ACTUATED8P, QUEUECHANGE, TURNBAY, LANEADD, and LANEDROP.

TIER 2
Used as a building block in 1 Tier blocks to simplify construction and complexity of blocks.

Includes Blocks: SET TRAVEL TIME, APPROACH, UNBLOCK #/#, BLOCK #/#, PHASE#_PT, PHASE#_AC, and MAX/GAP

TIER 3
Basic ARENA logical building blocks.

Includes Blocks: ACTIVATE, ALTER, ASSIGN, BEGIN, BLOCK, BRANCH, CLOSE, COMBINE, COPY, COUNT, CREATE, DELAY, DISPOSE, DUPLICATE, ELSE, ELSE IF, END IF, ENDWHILE, GROUP, HALT, IF, INSERT, MOVE, PICKQ, PROCEED, QPICK, QUEUE, READ, ROUTE, RELEASE, SEARCH, SEIZE, SELECT,

Figure 3.1 Open-TS3 Hierarchy

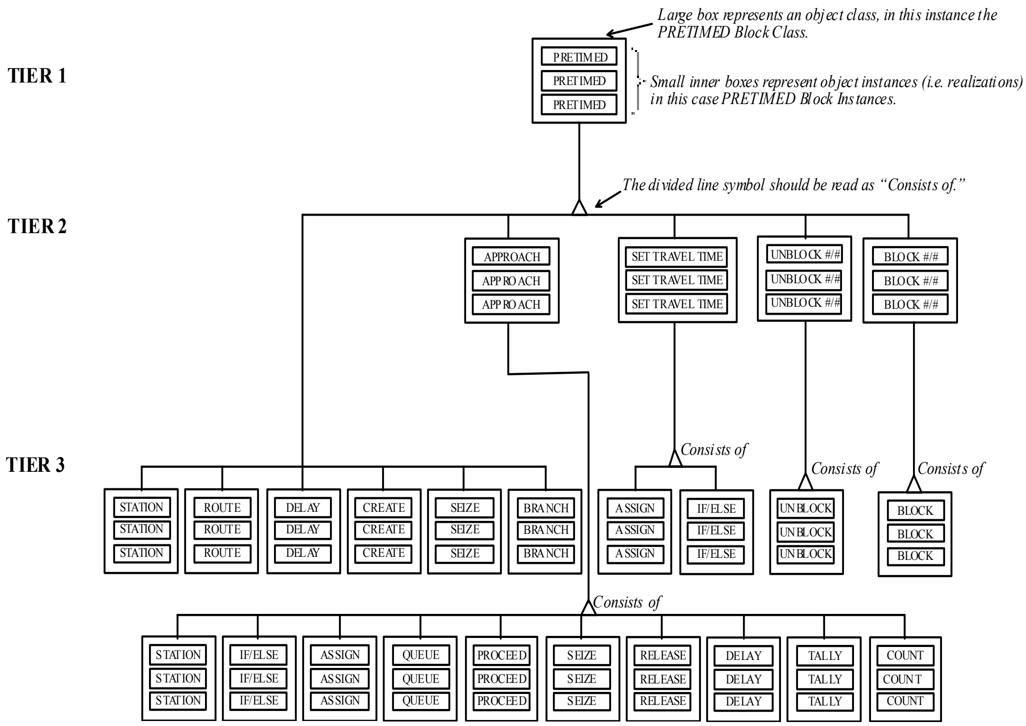


Figure 3.2 PRETIMED Object Class Hierarchy*

* Explanation of How to Read an Object Class Figure.

At the top of 0 is the PRETIMED Class (tier 1) from which PRETIMED objects are instanced. Each PRETIMED object (i.e. each PRETIMED block placed in a model) consists of objects from the APPROACH, SET TRAVEL TIME, UNBLOCK #/#, and BLOCK#/# classes (tier 2) and the STATION, ROUTE, DELAY, CREATE, SEIZE and BRANCH classes (tier 3). In turn each APPROACH object consists of objects from the STATION, IF/ELSE, ASSIGN, …., and COUNT classes (tier 3), each SET TRAVEL TIME object consists of objects from the ASSIGN and IF/ELSE classes (tier 3), etc.

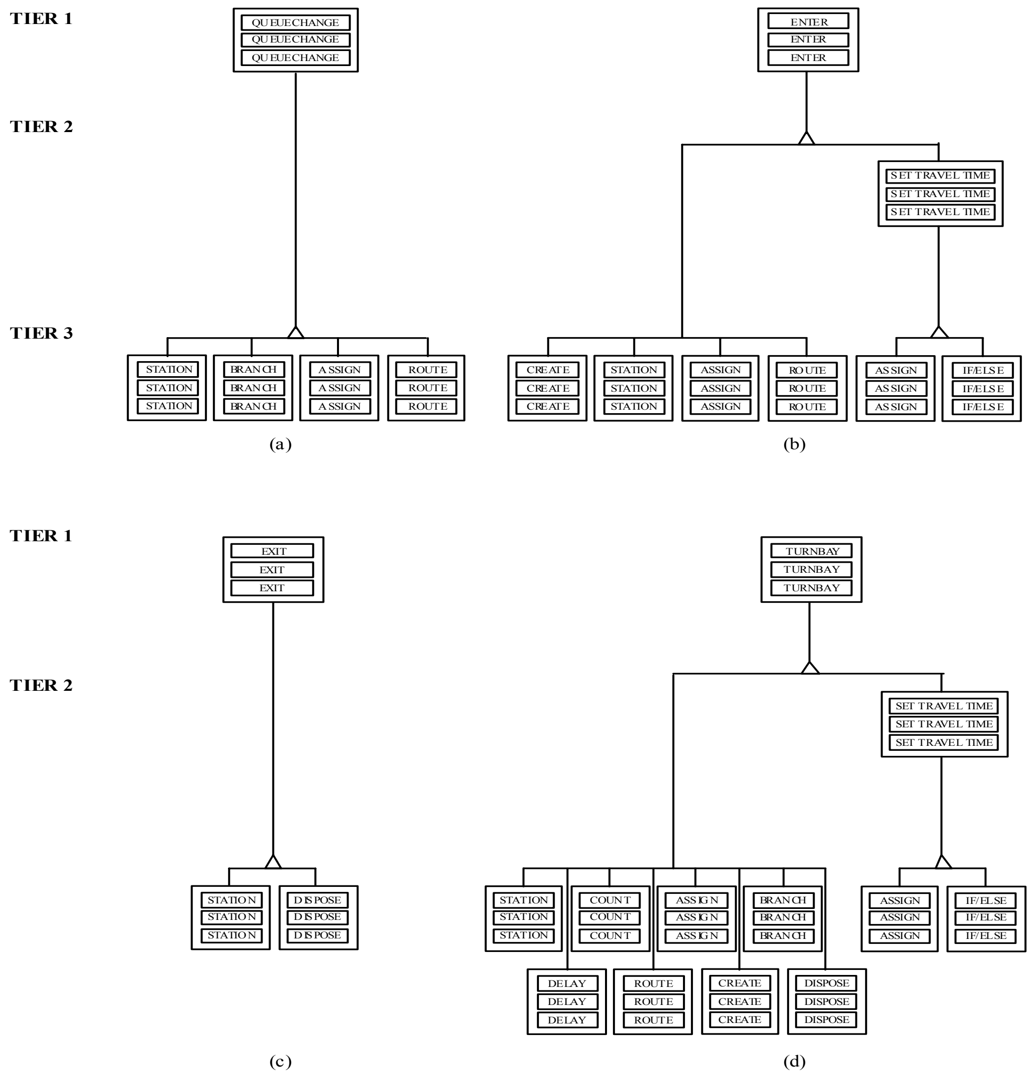


Figure 3.3 Object Class Hierarchy (a) QUEUECHANGE, *(b) ENTER, (c) EXIT, and (d) TURNBAY.*

(* See Object Class Hierarchy explanation in 0)

Event-Based – As discussed in Chapter two ARENA, and thus Open-TS3, is event based. Simply stated time progresses from event to event, not continuously or through pre-selected time steps. Once all actions have been completed at a simulation time the simulation clock is advanced to the next scheduled event, whether this is a time step of .05 or 5 seconds. For a complete discussion of the time based nature of Open-TS3 see section 2.5.2.7.

Object-Oriented – Roughly stated, object oriented programming is a programming approach where one first considers the software in terms of objects and how those objects interact with each other, rather than coding structures or program implementation. Object oriented programming has been utilized in many software applications, including simulations, but concepts such as encapsulation, polymorphism, classes, and object interfaces have not been applied to a computer representation of the real world physical elements of the transportation system. Object oriented programming in transportation simulation today is primarily limited to graphical user interfaces (GUIs) and data management, not the actual representation of the transportation system. By utilizing a simulation approach where the transportation system is seen as a collection of interacting objects it becomes a simpler and more straightforward task to create an open simulation architecture, where a wide array of developers and users may alter and contribute to the simulation.

Stochastic – While a Open-TS3 model may be constructed to be completely deterministic, it is also possible for a user to introduce randomness. Aspects of the simulation that may include randomness include, the creation (i.e. vehicle enter) interval, aggressiveness factor (which effects speeds, headways, and intersection start-up lost times) and turning movements. The stochasticity of the creation interval and aggressiveness factor may be set to follow numerous distributions, including; Exponential, Normal, Triangular, Uniform, Erlang, Lognormal, Beta, Poisson, Gamma, Weibull, Johnson, Continuous, and Discrete. Depending on the desired distribution, different parameters may be required. For example, to use Poisson the user must enter a mean, Normal requires a mean and standard deviation, and Gamma requires beta and alpha. Appendix A provides a discussion of random sampling in Open-TS3.

Simulation – Finally, Open-TS3 is a simulation, i.e. a computer representation of a real world system (currently existing or planned) that attempts to mimic the behavior of the real world system. The final quality of any results is a function of the accurateness of the Open-TS3 logic and the calibration of that logic to the system under consideration. As with all simulations, judgment must be exercised in the use of Open-TS3 and any conclusion drawn from results.

As a result of it's approach to simulation Open-TS3 has many advantages. While currently limited to intersection/arterial/network analysis it is readily expandable to other aspects of the transportation system. Much of this expansion potential is a result of Open-TS3's hierarchical, object-oriented structure. When a user wishes to model a transportation system feature other than those directly accounted for within current Open-TS3 blocks, such as a toll plaza, such development may be done in-house or by third-party developers. All other blocks, ENTER, EXIT, QUEUECHANGE, etc, may be used with the new toll plaza block(s). This "open architecture" approach frees a user from a dependence on Open-TS3 developers.

The hierarchical nature of Open-TS3 also allows for a minimal learning curve to initial model construction. One may quickly become efficient with tier 1 blocks, learning as little or as much as desired about the underlying logic (tiers 2 and 3) and still be able to construct a realistic, usable model. As users desires expand beyond the default tier 1 blocks they can learn and experiment with tier 2 and 3 blocks, performing more unique analyzes.

Finally the object-oriented approach to modeling represents a more "common sense" based approach to simulation. From an individual's earliest experiences one typically views the world in terms of objects and how they interact with each other; from a toaster's interaction with bread, to a key's interaction with a lock, to a car's interaction with a traffic signal. Utilizing peoples existing natural mechanisms for viewing their surroundings increases the likelihood of creating a more intuitive, understandable, efficient, and accurate simulation software package. (For additional information on the object oriented world view and it's advantages to software development see Chapter two of this text or D. Brown) In summary, while Open-TS3 will never be able to completely capture the infinite complexity of the real world it does provide the necessary tools and building blocks to achieve reasonable analysis of real world transportation systems.

Presentation of Open-TS3 Blocks

As stated the top tier of the Open-TS3 hierarchy currently consists of ten blocks; ENTER, EXIT, QUEUECHANGE, TURNBAY, LANEADD, LANEDROP, and four Intersection Blocks (PRETIMED, PRETIMED8P, ACTUATED8P, SIGNAL), and seven tier two blocks; SET TRAVEL TIME, APPROACH, UNBLOCK #/#, BLOCK #/#, PHASE#_PT, PHASE#_AC, and MAX/GAP. The purpose of each block is briefly discussed in this section. 0 (to be discussed) shows a sample single intersection simulation model utilizing most of these blocks.

ENTER – All vehicles are instanced (created) in an ENTER block, at which time all vehicle attributes are set to initial values. In no other block may a vehicle be created or enter a model. The ENTER block logic is given in section 0, and vehicle objects are discussed in section 0.

*EXI*T – All vehicles leave a model through an EXIT block. Final vehicle statistics are collected in this block. The EXIT Block logic is presented in section 0.

PRETIMED – This is one of the four intersection blocks. All intersection blocks tie together tier 2 APPROACH Blocks (to be discussed) with a signal control logic. PRETIMED utilizes a two or three phase pre-timed signal logic. All intersection blocks may model three-way and four-way signalized intersections and protected only left turn phasing. A single simulation model may utilize any combination of intersection blocks. The PRETIMED block logic is presented in section 0.

PRETIMED8P – Another intersection block the PRETIMED8P incorporates an eight-phase dual-ring pre-timed signal logic. This logic is implemented with the tier 2 PHASE#_PT blocks. The PRETIMED8P block logic is discussed in section 0.

ACTUATED8P – This intersection block models an eight-phase dual-ring actuated signal logic. This logic is implemented with tier 2 PHASE#_AC blocks. The ACTUATED8P block logic is discussed in section 0.

SIGNAL – This last intersection block allows for the simulation of adaptive signal control logic. This is accomplished through the use of an adaptive signal control Dynamic Link Library (.DLL). The SIGNAL block logic and DLL are discussed in Chapter 6.

QUEUECHANGE – The QUEUECHANGE block contains the logic by which a vehicle may choose to change lanes to enter a shorter queue, provided the queue serves the same movement. This block is not required for a simulation to execute. The logic for this block may be found in section 0.

TURNBAY – This block controls the logic by which vehicles are assigned to a turn movement at an intersection approach. Currently vehicle turning movement assignment is random, based on user input turn percentages. This block is not required in a single lane intersection model but necessary in most multi-intersection models where turning movements are desired. The TURNBAY block logic is discussed in section 0.

LANEADD – This block may be used to increase the number of lanes on a link. Utilizing logic similarly to the TURNBAY logic, vehicles are randomly assigned to the new lane according to user input lane percentages. This block is not required for a simulation to execute. LANEADD logic may be found in section 0.

LANEDROP – This block is used to reduce the number of lanes on a link, this is also not a required block. This logic may be found in section 0.

APPROACH – This is a second tier block that models the vehicle queue and stop bar departure on a single approach lane. An APPROACH block must be associated with another block that contains signal logic, such as the Intersection Blocks. The APPROACH block logic is discussed in section 0.

SET TRAVEL TIME – This is another second tier block. It is used to set the travel time on a link where randomness in travel times is introduced through the use of an aggressiveness factor. This block is included in all tier 1 blocks where a vehicle incurs a travel time to the next downstream block. The SET TRAVEL TIME logic is presented in section 0.

UNBLOCK#/# and BLOCK#/# - These are additional Tier 2 blocks. These blocks are utilized in the changing of signal indications in the PRETIMED and SIGNAL blocks. The primary purpose of these blocks is to simplify the tier 1 block logic. The UNBLOCK#/# and BLOCK#/# logic may be found in section 0.

PHASE#_PT, PHASE#_AC and MAX/GAP - These Tier 2 blocks are utilized in the PRETIMED8P and ACTUATED8P blocks signal switching logic. The primary purpose of these blocks is to simplify the tier 1 block logic. The *PHASE#_PT, PHASE#_AC and MAX/GAP* logic may be found in Hunter and Machemehl (2003).

As mentioned 0 shows an example of a single intersection model, comprising ENTER, EXIT, PRETIMED, QUEUECHANGE, and TURNBAY blocks. This example has three lane East Bound and West Bound approaches that flare out at the intersection to include single lane left and right turn bays. The North Bound and South Bound directions are two lanes each, one thru and one right turn. As seen all approaches have an upstream ENTER block and a downstream EXIT block. The East Bound and West Bound directions also contain LANECHANE and QUEUCHANGE blocks. The LANECHANGE block assigns movement designations (left, thru, right) to the vehicles that arrive from the upstream ENTER blocks. The QUEUECHANGE blocks are utilized on the intersection approach thru lanes, allowing vehicles the opportunity to switch to an adjacent thru lane if a shorter queue exists.

At the center of this example lies a PRETIMED Block which ties together 14 APPROACH blocks, one on each intersection approach lane; 5 EB, 5 WB, 2 NB, 2 SB. Although not indicated in the figure, these approaches are controlled by a three-phase cycle, leading EB/WB lefts, EB/WB thrus, NB/SB thrus. Also seen in 0 are the vehicle paths between blocks. This example contains the maximum number of lanes currently allowed within a PRETIMED block; five lanes on the major street (including turn bays, maximum of one bay for each turn movement) and two lanes on the side street. Also, a maximum of three phases is allowed, with a requirement of protected only left turn phasing. If additional intersections were included each would be controlled by a separate PRETIMED block.

0 shows the hierarchical breakdown of blocks in this example. This table shows the quantity of each type of block in each tier. It is readily seen how utilizing tier 1 blocks greatly reduces the complexity of model construction. 0 requires 21 tier one blocks to build a working, single intersection model. If this same model was constructed directly from tier 3 blocks (i.e. constructed with only available ARENA basic building blocks) nearly 900 blocks would be required. Clearly this would be an ominous task, with little likelihood of a user possessing the willingness to repeat the process for different geometric configurations. By utilizing Open-TS3 the task is greatly streamlined, allowing for efficient model development.

0 shows an example Open-TS3 model run. As seen while a simulation is running vehicle movements, queues and signal light indications are shown graphically. If desired, batch runs may be completed in which graphics are not utilized, decreasing the demanded on computer resources, significantly decreasing a model run time. Later in this chapter results from this example simulation run will be presented and discussed, including comparison with other simulations.

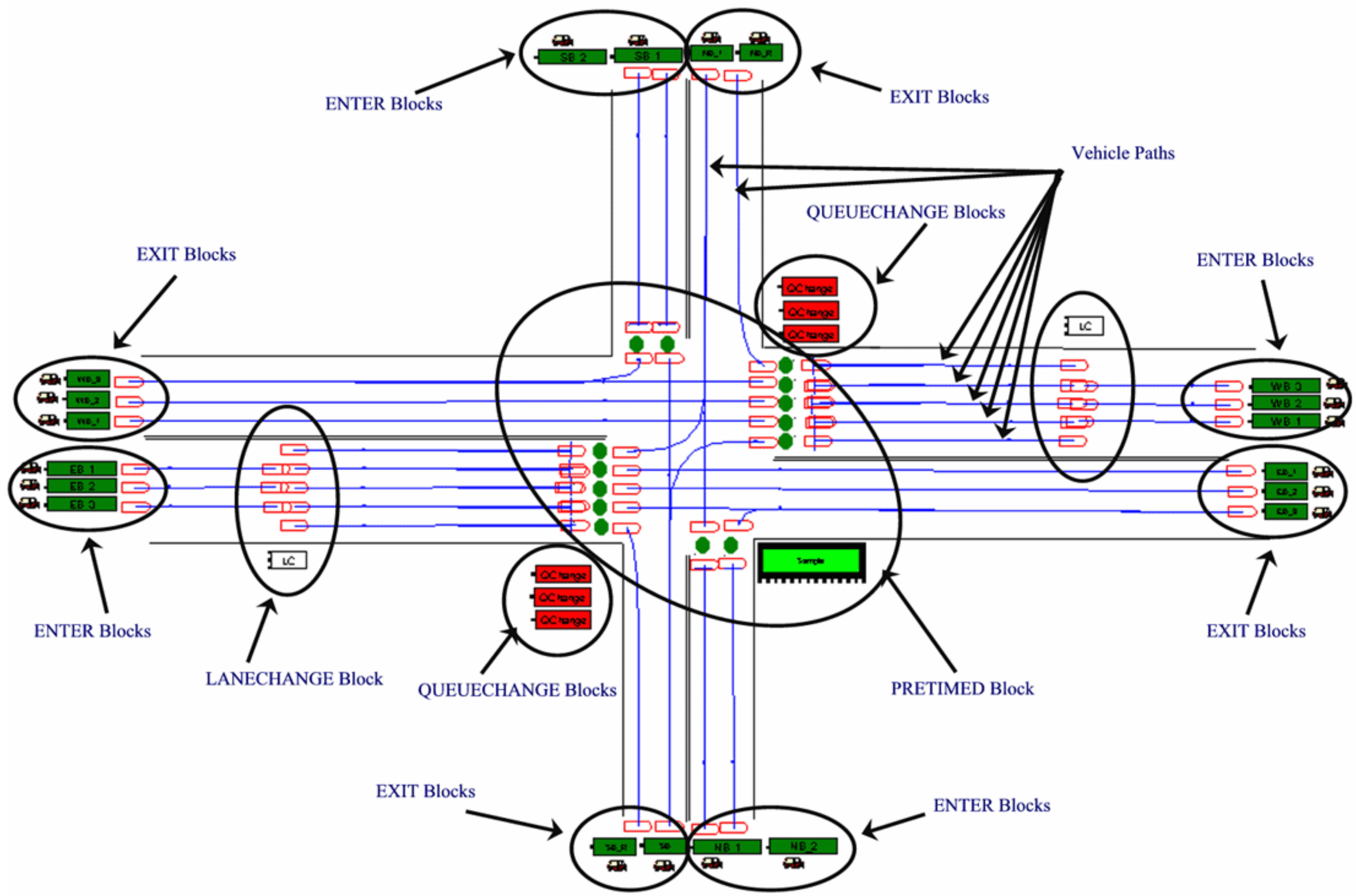


Figure 3.4 Example Single Intersection Open-TS3 Model

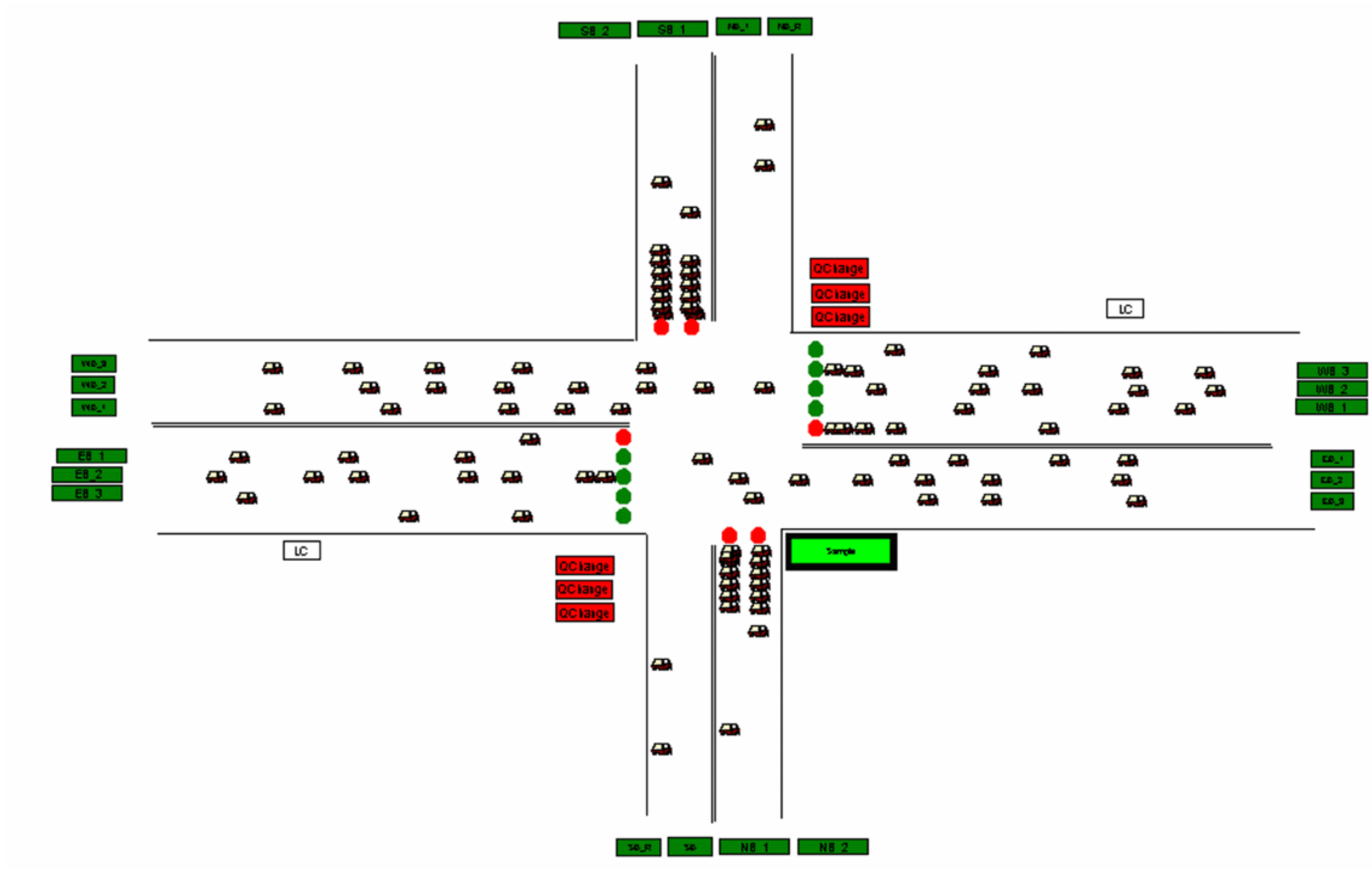


Figure 3.5 Example Single Intersection Open-TS3 Model During Execution

Table 3.1 Example Open-TS3 Listing of all Blocks

TIER 1 (8) ENTER	(8) EXIT	(2) TURNBAY	(2) QC	(1) PRETIMED *Total Tier 1 Blocks = 21*
TIER 2 (8) STT		(10) STT		(14) STT (14) APP (6) UnBlock (6) Block *Total Tier 2 Blocks = 58*
TIER 3 (1) CREATE (8) STATION (2) ASSIGN (2) ROUTE (24) ASSIGN (8) IF (8) ELSE (8) ENDIF	(8) STATION (8) DISPOSE	(16) STATION (16) COUNT (14) ASSIGN (10) BRANCH (18) DELAY (10) ROUTE (2) CREATE (2) DISPOSE (30) ASSIGN (10) IF (10) ELSE (10) ENDIF	(6) STATION (14) BRANCH (4) ASSIGN (4) ROUTES	(28) STATION (14) ROUTE (12) DELAY (1) CREATE (1) SEIZE (1) BRANCH (42) ASSIGN (14) IF (14) ELSE (14) ENDIF (28) STATION (56) IF (42) ELSE IF (28) ELSE (56) END IF (126) ASSIGN (14) QUEUE (14) PROCEED (14) SEIZE (14) RELEASE (14) DELAY (14) TALLY (28) COUNT (28) UNBLOCK (28) BLOCKS *Total Tier 1 Blocks = 889*

Note: QC = QUEUE CHANGE, APP = APPROACH, STT = SET TRAVEL TIME

VEHICLE OBJECT –ADD VEHICLE TRAVEL TIME

Overview

Prior to in-depth presentation of Open-TS3 blocks (section 0), one should acquire an understanding of the vehicle object, which is the computer-simulated representation of a real world vehicle. The vehicle object differs from the other block objects in that it moves through the model, whereas all the blocks are stationary. The vehicle object directly interacts (is processed, delayed, assigned to turn movement, etc.) with the model blocks. Vehicle objects are instanced only in the ENTER block (further discussion of this is provided in the ENTER block section, 0). A vehicle's state (moving, queued, turning, etc.) is a function of the state of the block in which the vehicle currently resides and the vehicle's attributes. For example, the travel time of a vehicle from an upstream to a downstream block, is a function of the user input average travel time which is set in the upstream block, the vehicle's aggressiveness factor which is an attribute, and the time to overtake the prior vehicle which is recorded in the SET TRAVEL TIME block.

Vehicle Attributes

Vehicle attributes capture both the characteristics of the vehicle/driver, such as vehicle aggressiveness, and simulation tracking and statistical data, such as Vehicle Id and Current Total Delay. The model block will often utilize vehicle/driver attributes in a vehicle's processing, for example a vehicle aggressiveness factor will effect travel time set in the SET TRAVEL TIME block. Attributes currently defined for each vehicle instance include; Aggressiveness Factor, Model Enter Time, Model Exit Time, Travel Time, Current Block Enter Time, Current Block Exit Time, Vehicle Id, and Current Total Delay.

> *Aggressiveness Factor* - Vehicle objects are randomly assigned an aggressiveness factor. Currently this factor is a uniformly generated random variate between 0.9 and 1.1. Minimum headway, start-up lost-time, intersection discharge headway and speed are all affected by the aggressiveness factor. For example, where the average vehicle speed in a model block is set to 35 mph a vehicle with a 1.1 aggressiveness factor will have a desired speed of 35 mph multiplied by 1.1, that is, 38.5 mph. If simulations with reduced stochasticity are desired the aggressiveness factor may be set to one for all vehicles.
>
> *Model Enter Time* – This is the time at which the vehicle is instanced in the ENTER block. Unlike many other simulations, vehicles are not stacked at entrance points waiting to be released, instead a vehicle is not instanced until it is to be released into the model. This has the advantage of reducing demand on computer resources but the disadvantage of potentially reducing control over the randomness between models in an alternative scenarios analysis. This will be further discussed when the sources of randomness in Open-TS3 are presented.
>
> *Model Exit Time* – This time, set in the EXIT block, is the time that the vehicle exits the model. A vehicle's Model Exit Time, subtracted from the Model Enter Time, yields the amount of time that the vehicle spent in the system.

Travel Time – This attribute is the travel time from the block in which the vehicle currently resides to the next downstream block. The attribute is set in the current (upstream) block once processing of the vehicle is complete and the vehicle is set to be released from the block. Where constant travel times are employed (i.e. all vehicle's Aggressiveness Factors equal one) all vehicles progress to the downstream destination in the user input travel time. Where field data on travel between an upstream and downstream block is in the form of speed it must be converted into travel time for model input.

Where an aggressiveness factor is utilized, it becomes possible for a vehicle to have a desired travel time that would lead to overtaking of an earlier released vehicle. Currently the Open-TS3 does not provide for passing. Thus, when this situation arises the vehicle's travel time to the downstream destination block is adjusted to equal that of the downstream vehicle plus a minimum headway. The logic for determining the travel time when an Aggressiveness Factor is utilized is contained in the SET TRAVEL TIME block and discussed in section 0. Currently no adjustment is made to the travel times accounting for current volumes on the link. Mid-link vehicle passing and volume related travel time adjustments may be incorporated into future versions of Open-TS3.

Current Block Enter Time and *Current Block Exit Time*– Upon vehicle entry into a model block the Current Block Enter Time is set to the current simulation time. The Current Block Exit time is similarly set when a vehicle has completed processing through the model block. The different model blocks utilize these attributes to aid in vehicle processing and statistics calculation. For example, the Current Block Enter Time is used in the delay calculation in the intersection blocks and the Current Block Exit Time is used in the downstream block arrival time calculation where a vehicle incurs a travel time to the next downstream block.

Vehicle Id - Every vehicle is assigned a unique Vehicle Id. This Id is utilized throughout the model in the tracking and processing of vehicles.

Block Delay – This is a delay that may be assigned to a vehicle while being processed by a block. This delay will be served in the block and is always reset to zero when a vehicle enters a new block. For example, the APPROACH block stores Start-Up Lost-Time delay in this attribute.

Current Total Delay – As a vehicle proceeds through a model it is likely that it will incur some delay. Each time a vehicle incurs delay it is included within the vehicle's total delay. Total vehicle delay statistics may then be collected as vehicles exit the model. This should not be confused with average intersection delay statistics that are maintained by the intersection blocks.

Finally, in order to allow for consistent interfaces and reduced complexity within the model all attributes that may be used throughout the model are included at the time of creation of each vehicle object. While the value of a vehicle attribute may change during a simulation no new attributes may be defined after the vehicle has been instanced. This is not an ARENA limitation but a modeling rule enforced in Open-TS3 to allow for uniformity in model block interfaces both within a particular model and among different models and users. The remainder of this section provides a discussion of the different attributes currently assigned to vehicle objects.

PRESENTATION OF OPEN-TS3 BLOCKS

This section provides a detailed presentation of each of the current Open-TS3 tier 1 and tier 2 blocks; ENTER (section 0), EXIT (section 0), APPROACH (section 0), Intersection Blocks - PRETIMED, PRETIMED8P, and ACTUATED8P - (section 0), QUEUECHANGE (section 0), TURNBAY (section 0), LANEADD (section 0), LANEDROP (section 0), SET TRAVEL TIME (section 0), UNBLOCK#/# (section0), BLOCK#/# (section 0) and, PHASE#_PT / PHASE#_AC / MAX/GAP (section 0). The presentation of each block will follow the same basic format. First will be a block overview, followed by required user inputs, and finally, discussion of other objects with which the block interacts. The heart of each block presentation then follows with a detailed description of the block's logic. This includes Open-TS3 lower tier logical construction and the logical equivalent in a traditional algorithmic form. Finally, items specific to that Open-TS3 block are discussed.

It should be realized that the traditional algorithmic logical form listed in these discussions is not required in the construction of a Open-TS3 model, at any tier, although it can be helpful when one initially begins to familiarize oneself with Open-TS3 and ARENA. Once a user becomes comfortable interpreting basic ARENA block logical construction the traditional algorithmic form will typically be deemed unnecessary. Also, not included in these discussions are the ARENA Operands that are required with each block logic construction. These operands may be thought of as the equivalent of variable defining and dimensioning in traditional programming languages. They are not discussed as little is gained in the understanding of the block's logic and there is a high potential for additional confusion, especially for the users new to ARENA.

ENTER Block

ENTER Block Overview

Figure 3.6 Open-TS3 ENTER Block Model Icon.

All vehicles enter through an ENTER Block, which feeds one lane. Thus, a three-lane entrance link would require three ENTER blocks. The downstream destination of vehicles leaving an ENTER block may be any block that accepts vehicles.

As mentioned in the Vehicle Object discussion all vehicles are instanced (created) in an ENTER Block. Vehicle instancing occurs at the rate set by the user. This rate may be set to be constant or random, with numerous available distributions. The mean entrance rate should be set to generate the desired hourly volume. Upon instancing, the attributes of each vehicle are set to their initial values, as discussed in the Section 0 (Vehicle Object), and the vehicle is immediately released to proceed to the downstream destination. In addition to the entrance rate, a user may set the allowable maximum number of arriving vehicles over the simulation time period. This maximum does not affect the entrance rate; it simply ceases the creation of arriving vehicles once the maximum has been reached.

Vehicles are not allowed to exit the ENTER block at a rate greater than the minimum acceptable headway. Where a vehicle creation rate smaller than the minimum headway is selected by the user, vehicles will be queued in the ENTER block and released in a First-In First-Out manner, at the minimum headway rate. Each individual vehicle travel time to the next destination is set in the SET TRAVEL TIME block, discussed in section 0.

ENTER Input Dialogue Box

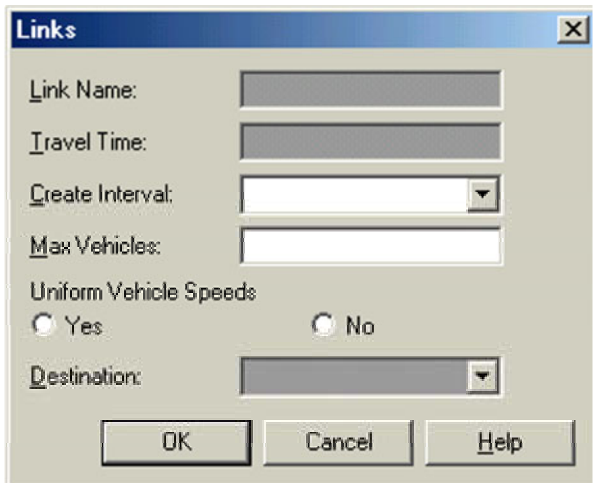


Figure 3.7 ENTER Block User Input Dialogue Box

The modeler is required to enter several pieces of data into the ENTER block at the time of model construction. 0 shows the user input dialogue box. The following defines each input.

Link Name This is the name given to the ENTER Block and must be unique among all other ENTER block names. This name will be utilized in other blocks where the ENTER block must be referenced.

Travel Time This is the average desired travel time from the ENTER block to the downstream block. Where raw data is in the form of speeds the speed must be converted into the required time to traverse the distance to the downstream block. Time is entered in seconds.

Create Interval When a value in seconds is entered it is assumed that arrivals are fixed at that interval. Where raw data is in hourly volumes it must be converted into a mean arrival rate before being entered into the model. It is also possible to designate the arrival to follow numerous distributions, including; Exponential, Normal, Triangular, Uniform, Erlang, Lognormal, Beta, Poisson, Gamma, Weibull, Johnson, Continuous, and Discrete. The different distributions are provided in a pull down list in the Create

	Interval input box. Depending on the desired distribution different parameters may be required, for example, to use Poisson the user must enter a mean, Normal requires a mean and standard deviation, and Gamma requires beta and alpha.
Max Vehicles	The maximum number of vehicles allowed to enter the model through the ENTER block. As stated earlier the maximum number of vehicles does not affect the arrival rate set in the Create Interval input box, it simple ceases the creation of arriving vehicles once the maximum number of vehicles has been reached. If no value is input than no limit is set.
Aggressiveness Factor	If "Yes" is selected, then the vehicle aggressiveness factor as described in the Vehicle Object section is utilized, otherwise "No" is selected and all aggressiveness factors will be set to one at the time of vehicle instancing.
Destination	This is the unique block name of the downstream vehicle destination.

Interaction with other Objects

The ENTER block must be associated with a downstream destination block, vehicle objects, and model wide variables.

Immediately upon instancing, the ENTER block releases vehicles to the downstream block. The downstream destination block may be any block that accepts vehicles, including; EXIT, APPROACH, QUEUECHANGE, TURNBAY, LANEADD, LANEDROP, and intersection blocks. The ENTER block and the downstream block should be on the same simulation hierarchical tier, which typically will be the top tier.

The ENTER block is the only block in the model where a vehicle object may be instanced. Once instanced the ENTER block interacts with the vehicle through the vehicle object interface. The primary interaction of the ENTER block with a vehicle instance is the setting of all attribute values to their initial values. The ENTER block also interfaces with model wide variables. Primarily to track the current simulation clock time, which is needed to ensure that vehicles are created and released in accordance with the user create interval. Fortunately a modeler need not be concerned with the form of the ENTER block / Open-TS3 Model or ENTER block / Vehicle Object interface at the computer language level as these interfaces are automatically maintained by ARENA. The modeler need only utilize the block level interaction as discussed in Chapter two.

ENTER Logic

Before presenting the ENTER block algorithm the notion should be explained. Algorithm steps are stated in the form; OBJECT(UID, Variable). OBJECT refers to a block or model object, i.e. ENT refers to an ENTER block and VEH to a vehicle instance. UID is the unique identifier of the referenced OBJECT. Every object in a model has a different unique identifier. Unique identifiers are further grouped by object classes, thus EUID is an ENTER unique identifier, VUID is a Vehicle unique identifier, SUID is a SET TRAVEL TIME unique identifier, etc. The value of OBJECT(UID, Variable) always refers to Variable, for the OBJECT instance with the specific UID. For Example, Veh(VUID, Enter_Time) = 12 may be read; vehicle VUID arrived at the current block at simulation time 12; and Set ENT(EUID, Arr_t) = t may be read; the next vehicle arrival time at ENTER block EUID is set equal to t. All Variables used in an algorithm are listed and follow the algorithm.

The ENTER block logical construction is shown in 0 and an algorithmic version is given in this section. Brief explanations (in *italics*) are included to assist in interpreting the ENTER algorithm.

ENTER Block Algorithm

STEP 0) ENT(EUID, Arr_Ct) = 0, ∀ EUID
ENT(EUID, Ent_Dest) = User Input Destination, ∀ EUID
ENT(EUID, Ent_DestTime) = User Input Travel Time, ∀ EUID
ENT(EUID, Ent_Int) = User Input Create Interval, ∀ EUID
ENT(EUID, Ent_Max) = User Input Max Vehicles, ∀ EUID
ENT(EUID, Ent_AggFac) = User Input inclusion of Aggressiveness Factor, ∀ EUID
ENT(EUID, Arr_Time) = 0 ∀ EUID
ENT(EUID, Arr_Int) = 0 ∀ EUID

Step 0 is an initialization step which occurs at the start of the simulation, t = 0. All blocks have an initialization step. For each ENTER block (all EUID) the current number of arrivals (Arr_Ct) and the time of the first arrival (Arr_Time) are set to Zero and the User Input Destination, Create Interval, Maximum number of vehicles, and inclusion of Aggressiveness Factor (Yes or No) are read.

Steps 1 through 11/11a give the algorithm by which each ENTER block, ENT(EUID), instances and processes a vehicle, Veh(VUID).

STEP 1) If t = ENT(EUID, Arr_Time) then Veh(VUID) Instanced in block ENT(EUID)

In step 1 a vehicle is instanced in the ENTER block whenever the simulation clock time equals a vehicle arrival time. The first vehicle is instanced at simulation clock time t = 0. Subsequent vehicles are instanced according to the user input create interval, as will be shown in steps 2 and 3.

STEP 2) Set ENT(EUID, Arr_Int) = Realization of ENT(EUID, Ent_Int)

In step 2 the time interval until the next vehicle instancing is set. Where the user has entered a fixed value in the Create Interval input that value is utilized. Where the user has entered a distribution in the Create Interval input a realization of that distribution is used. Each vehicle may utilize a different realization from that distribution. The method by which ARENA, and subsequently Open-TS3, determine realizations from distribution are discussed in Appendix A.

STEP 3) Set ENT(EUID, Arr_Time) = t + ENT(EUID, Arr_Int)

The time of the next vehicle instancing is set.

STEP 4) Set ENT(EUID, Arr_Ct) = ENT(EUID, Arr_Ct) + 1

Increase vehicle entrance count by 1.

STEP 5) If ENT(EUID, Ent_AggFac) = No Then
GoTo to Step 6
Else
GoTo Step 6a
End If

Step 6 checks if the user has selected the utilization of an aggressiveness factor. Steps 6 through 11 list the algorithm when an aggressiveness factor is not utilized, steps 6a through 11a list when an aggressiveness factor is utilized. The implementation of the IF ELSE GOTO logic is implemented in Open-TS3 (0) through ARENA switches. In this situation the switches enable one of the two possible logics (Steps 6 through 11 or steps 6a through 11a) to be included in the model when it is initialized. A detailed explanation of switches is given in the TURNBAY block discussion.

STEP 6) Set VEH(VUID, Enter_Time) = t
STEP 7) Set VEH(VUID, Agg_Fac) = 1
STEP 8) Set VEH(VUID, T_Time) = ENT(EUID, Ent_DestTime)

In steps 6 through 8 several vehicle attribute values are set. The time at which the vehicle entered the model is set to t, the travel time to the next block is set to the user input average travel time, and the vehicle Aggressiveness Factor is set to 1.

STEP 9) Wait VEH(VUID, T_Time)
STEP 10) Release VEH(VUID) to ENT(EUID, Ent_Dest)

In step 9 the travel time from the current ENTER block to the downstream destination block is imposed on the current vehicle. In step 10 the vehicle then enters the destination block, which may imply immediate processing or entering into a queue, depending on the downstream block.

STEP 11) If ENT(EUID, Arr_Ct) = ENT (EUID,Ent_Max) then END ENTER block

If the maximum number of vehicles is reached then no additional vehicles may be instanced.

STEP 6a) Set VEH(VUID, Enter_Time) = t
STEP 7a) Set VEH(VUID, Agg_Fac) = Realization of Uniform(0.9,1.1)
STEP 8a) Set VEH(VUID, T_Time) = See section 3.3.1.5, SET TRAVEL TIME
STEP 9a) Wait VEH(VUID, T_Time)
STEP 10a) Release VEH(VUID) to ENT(EUID, Ent_Dest)
STEP 11a) If ENT(EUID, Arr_Ct) = ENT (EUID, Ent_Max) then END ENTER block

Steps 6a through 11a operate in a manner similar to that of steps 6 through 11. One difference is in step 7a where the Aggressiveness Factor is now a realization from a uniform(0.9,1.1) distribution. Also, once Aggressiveness Factors are utilized the travel time in step 8 becomes variable. A detailed explanation of the logic used to determine the travel time is given in section 0.

Where:

ENTER Block (object) Variables – ENT(EUID, variable)

EUID =	ENTER block instance unique identifier
Arr_Ct =	The number of vehicles that have arrived by current simulation time
Arr_Int =	The time interval, in seconds, until the next vehicle is instanced
Arr_Time =	The simulation clock time at which the next vehicle is to be instanced.
Ent_Dest =	User Input destination of vehicle instance upon exiting enter block.
Ent_Int =	The user input Create Interval, this may be a specific value or a distribution that will be sampled.
Ent_Max =	User input maximum number of vehicles that may enter at ENTER block EUID.
Ent_AggFac =	User input showing if aggressiveness factor is to be utilized. If “Yes” then varying aggressiveness factors will be set on vehicles otherwise all aggressiveness factors will be set to 1.
Ent_DestTime=	This is the user input average travel time to the downstream destination block.

Vehicle Object Variables - VEH(VUID, variable)

VUID =	Vehicle Instance Unique Identifier
Enter_Time =	Model Enter Time (i.e., time at which vehicle is instanced)
T_Time =	Travel Time, from the current block to the downstream destination block
Agg_Fac =	Vehicle Aggressiveness Factor

Model Variables

t =	Current Simulation Clock Time

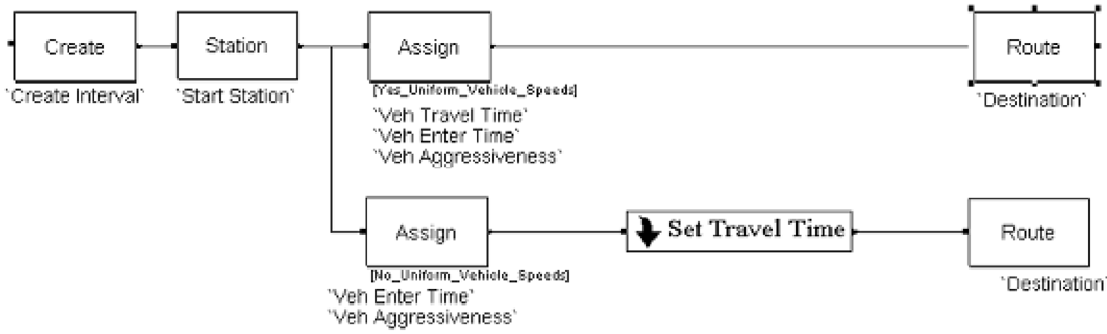


Figure 3.8 ENTER Block ARENA Basic Block Logical Construction

SET TRAVEL TIME Block

All blocks involving a travel time to the next downstream block incorporate a tier 2 SET TRAVEL TIME Block in their logic whenever an Aggressiveness Factor is utilized. Aggressiveness Factors result in vehicles having varying travel times between blocks. In general the Aggressiveness Factor affects the travel time linearly, that is, the average travel time is multiplied by a vehicles Aggressiveness Factor, yielding that vehicles travel time. The difficulty in this simple approach lies in the possibly of a situation arising where a more aggressive vehicle (higher aggressiveness factor) may have a travel time which results in an overtaking of a previously released vehicle with a lower aggressiveness factor. Currently Open-TS3 does not allow for vehicle passing between links so departing vehicle travel times must be checked, and adjusted where necessary, to avoid a passing situation. It is assumed in the SET TRAVEL TIME Block that the upstream vehicle desiring a shorter travel time (the vehicle that would wish to pass) is slowed by the slower vehicle and arrives at the downstream block after the slower vehicle. The logic to set these travel times is contained in the SET TRAVEL TIME Block.

The SET TRAVEL TIME Block is shown in 0 and an algorithmic version is given below. The SET TRAVEL TIME Block logic is active during the entire simulation run. Brief explanations (in *italics*) are included to assist in interpreting the Set Travel Time module algorithm.

SET TRAVEL TIME Algorithm

STEP 0) STT(SUID, Lead_Arr) = 0, $\forall$ EUID

Step 0 is the initialization step which occurs at the start of the simulation, t = 0. The SET TRAVEL TIME Block only maintains one variable, the lead-vehicle arrival time. This is the time that the last vehicle to exit the SET TRAVEL TIME Block is set to arrive at the next downstream block.

Steps 1 through 7 give the algorithm by which each SET TRAVEL TIME Block, STT(SUID), determines the downstream arrival time for vehicle, VEH(VUID), in ENTER block (EUID). Where STT is placed in other Tier 1 Blocks variables referring to the ENTER block are changed to the other tier 1 block.

STEP 1) VEH(VUID) enters STT(SUID) at time t

STEP 2) If STT(SUID, Lead_Arr) + STT(SUID, Min_HW) < t + VEH(VUID, Agg_Fac) * ENT(EUID, Ent_DestTime) THEN

GOTO Step 2
Else
GOTO Step 4
End If

Step 2 checks if the vehicle currently in the SET TRAVEL TIME Block will pass the last vehicle to exit the block if the average travel time is adjusted by the vehicles aggressiveness factor. If the current vehicle would arrive at the downstream block no sooner than the previous vehicle plus a minimum headway the averge travel time adjusted by the aggressiveness factor may be used, as in step two, else the travel time must be set as in step 4.

STEP 3) Set VEH(VUID, T_Time) = ENT(EUID, Ent_DestTime) * VEH(VUID, Agg_Fac)
STEP 4) Next STEP 5

In Step 3 the Travel Time is set to the average travel time adjusted by the aggressiveness factor. In Step 4 the algorithm is then sent to Step 5, where the downstream arrival time will be set.

STEP 5) Set VEH(VUID, T_Time) = STT(SUID, Lead_Arr) + STT(SUID, Min_HW) – t

Step 5 is the case where the vehicle travel time must be adjusted to avoid overtaking of the previously released vehicle. This travel time is set such that the current vehicle will arrive at the downstream block a minimum headway later than the previous vehicle.

STEP 6) Set STT(SUID, Lead_Arr) = t + VEH(VUID, T_Time)

Step 6 resets the lead vehicle arrival time to the current vehicles arrive time. This value will be used in the checking of the arrival time of the next vehicle to enter the SET TRAVEL TIME Block.

Step 7) Release VEH(VUID)

The current vehicle is released from the SET TRAVEL TIME Block.

Where:

Travel Time module (object) Variables – STT(AUID, variable)
SUID = Set Travel Block module unique indentifer

Lead_Arr = Lead Vehicles Arrival Time
Min_HW = Minimum acceptable headway between arriving vehicles

ENTER Block Variable – ENT(EUID, variable)
EUID = ENTER block instance unique identifier
Ent_DestTime= This is the user input average travel time to the downstream destination block.

Vehicle Object Variables - VEH(VUID, variable)
VUID = Vehicle instance unique identifier
T_Time = Travel Time, from the current block to the downstream destination block
Agg_Fac = Vehicle Aggressiveness Factor

Model Variables
t = Current Simulation Clock time

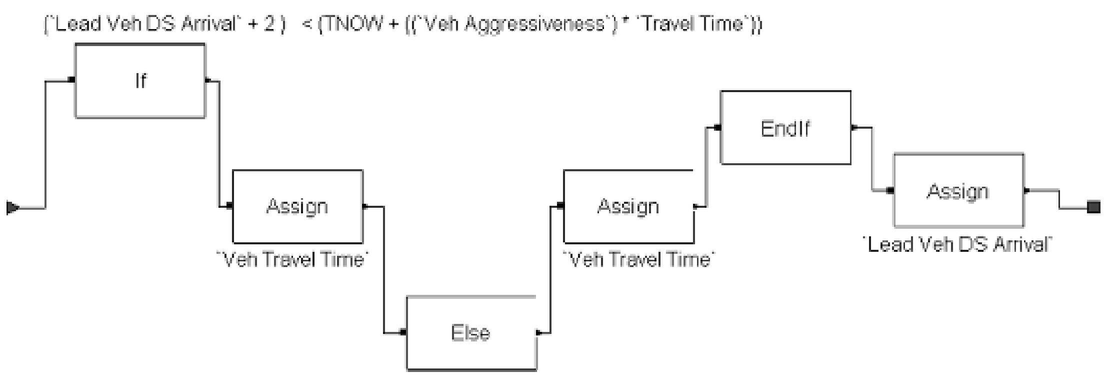


Figure 3.9 SET TRAVEL TIME ARENA Basic Block Logical Construction

EXIT Block

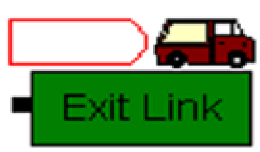


Figure 3.10 Open-TS3 EXIT Block Model Icon

All vehicles exit Open-TS3 model through an EXIT Block. Once a vehicle enters the EXIT block, final vehicle statistics are collected and the vehicle object is removed from the simulation (i.e. the vehicle instance no longer exists or consumes any computer memory).

Statistics are currently collected on vehicle delay and the total time a vehicle spent in the system. For each statistic, average, standard deviation, minimum, and maximum values are recorded.

Any number of lanes may be fed into an EXIT block but for modeling consistency and statistics collection it is recommended that each EXIT block be feed by one lane only. The upstream origin of vehicles entering an EXIT block may be any block that releases vehicles. As seen if 0 only one input is require for the Exit block, a unique EXIT Name. This name will be utilized as the destination name from upstream block. 0 shows the block diagram for the EXIT block. The algorithm is not listed as it is simply the vehicle enters the EXIT block, final statistics are collected and the vehicle object is deleted.

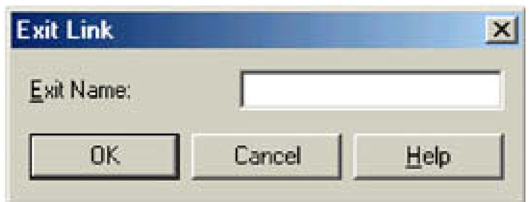


Figure 3.11 EXIT Block User Input Dialogue Box

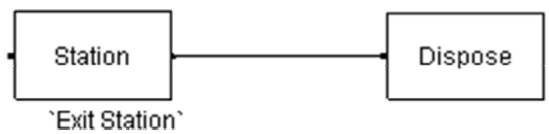


Figure 3.12 EXIT Block ARENA Basic Block Logical Construction

APPROACH Block

Approach Block Overview

Figure 3.13 Open-TS3 APPROACH Block Model Icon

The APPROACH block captures the process of vehicle entry to, queuing at, and departure from a signalized intersection approach lane. The APPROACH block may be utilized on the top tier of a simulation although it is more commonly employed as a major component (i.e. second hierarchical tier) of other top tier blocks. 0 shows the modeling icon used to represent the APPROACH block icon in a Open-TS3 model.

The APPROACH block is a major component of the intersection blocks. For example a multi-lane PRETIMED block intersection contains an APPROACH block for each lane, i.e. an approach with three thru lanes and a left turn bay is modeled with four APPROACH blocks, one for each thru lane and one for the left turn.

A vehicle enters the APPPROACH block from an upstream vehicle source, having already incurred the uninterrupted travel time from the upstream source to the intersection stop bar. Where no queue exists and the signal is red the vehicle will begin a new queue; where no queue exists and the signal is green the vehicle will proceed through the APPROACH block to the downstream vehicle destination. If a queue exists, regardless of signal indication, the vehicle will join the back of the queue.

The release of a vehicle from an APPROACH block is a function of queuing mechanics, traffic signal indications, and the travel time to the downstream intersection. That is, for a vehicle to depart an APPROACH block it must first reach the head of the queue (First In First Out queuing dynamics) and receive a green signal, then start-up lost-time, intersection passage time, and travel time to the downstream intersection must be served. The rate at which vehicles depart a queue is equal to the user input minimum startup headway and any assigned start-up lost-time. Once released from an APPROACH block a vehicle is assumed to have traveled through the intersection and to the downstream destination block.

The recorded vehicle delay consists of the time in queue plus the start-up lost time. The APPROACH Block continually updates average vehicle delay statistics for that block. Currently the APPROACH block utilizes a vertical queue and thus spillback effects are not captured.

Interfaces to Other Objects

An APPROACH block must be associated with an upstream vehicle source block, a downstream destination block, a traffic signal, vehicles, and model wide variables.

The upstream vehicle source of an APPROACH block placed in the top tier of the simulation hierarchy may be any block that releases vehicles, such as, ENTER, APPROACH, TURNBAY, and intersection blocks. Similarly, the downstream block may be any block that accepts vehicles, such as, EXIT, APPROACH, TURNBAY, and intersection blocks. Although, the APPROACH block will most commonly be utilized in the second hierarchical tier, as a major component of TIER 1 intersection blocks. When found in the second hierarchical tier the vehicles enter the tier 1 block and are processed by an APPROACH block as part of the tier 1 logic. Discussion of the APPROACH block in tier 1 intersection block's logic is presented in section 0.

The APPROACH block interacts with vehicles through the vehicle object interface. The APPROACH block sets certain vehicle attributes, including block enter time, travel time, current total delay, and current block exit time. There are also vehicle attributes that the APPROACH block may only read, such as the vehicle's unique identifier and aggressiveness factor. In the APPROACH block algorithm (section 0) and the APPROACH block tier 3 basic block logic (0) how this information is utilized is presented.

The APPROACH block must be interfaced with a signal switching object (section 0). The signal switching object contains the methodology by which the signal changes are controlled (pre-timed, actuated, or adaptive), currently found in the intersection blocks. The APPROACH block will receive a call from the signal switching object whenever the traffic signal indications change. The signal indication may only be set in the signal switching object, but other model objects, such as the APPROACH block, may only read the current state of the signal.

The APPROACH block allows other blocks to check (although not change, i.e. read only) the current queue length and average delay. The queue length (vehicle entering, exiting, tracking logic) and delay calculations logic are encapsulated within the APPROACH block. The SIGNAL and QUEUECHANGE blocks are examples of blocks that use of the queue length and delay information.

APPROACH Block User Input

The modeler is required to enter some data into the APPROACH block at the time of model construction. When the APPROACH block is utilized in the second hierarchical tier this data is entered through its corresponding top tier block. 0 shows the user input dialogue box. The following defines each input.

Int_Appr_N	The intersection name. This name must be the same name used in the traffic object associated with the APPROACH block.
NEMA Code	This is a user assigned number. APPROACH blocks with the same *Int_Appr_N* must have unique *NEMA Code* numbers. For consistency it is recommended that the NEMA (National Electrical Manufacturing Association) numbering system, or some similar system, be used. When a direction has multiple thru lanes a combination of the NEMA numbering system and lane number should be utilized. For example, if there are three eastbound thru lanes they could be number 2_1, 2_2 and, 2_3.
Headway	Minimum startup headway, in seconds, between vehicles. This along with lost time, controls the rate at which vehicles may depart a queue.
Dest	This is the downstream destination block of departing vehicles.
Time to Dest	The travel time to the destination block.

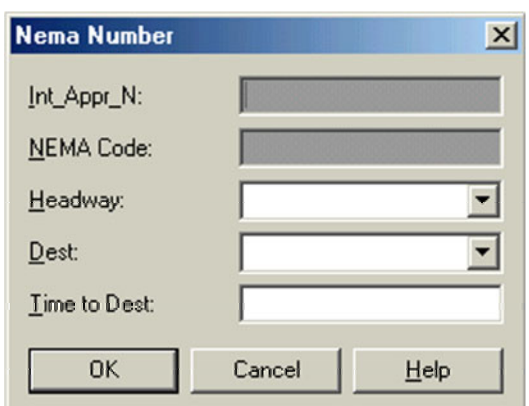


Figure 3.14 APPROACH Block Input Dialogue Box

APPROACH Logic

The APPROACH block logic is shown in 0 and an algorithmic version is given below. The APPROACH block logic is active whenever a vehicle is present on the approach. To assist in interpreting the algorithm, explanations (in *italics*) are provided for the various steps.

APPROACH Algorithm

STEP 0) APP(AUID, App_Res) = 0, ∀ AUID
APP(AUID, App_Q) = 0, ∀ AUID
APP(AUID, App_D) = 0, ∀ AUID
APP(AUID) Interfaced with SIG(SUID, phase)

Step 0 is an initialization step which occurs at the start of the simulation, t = 0. For each APPROACH block (all AUID) the stop bar is set to empty, the queue to zero, and the average delay to zero. The APPROACH block is also interfaced (set by the modeler at model set up for tier 1 APPROACH blocks, or set by tier 1 blocks for tier 2 APPROACH blocks) with a specific Signal object's phase. Signal objects are discussed with the PRETIMED block in Section 3.3.4.

Steps 1 through 17 give the algorithm by which each APPROACH block, APP(AUID), processes a vehicle, VEH(VUID), past a traffic signal, SIG(SUID).

STEP 1) VEH(VUID) Enters APP(AUID) at time t
STEP 2) Set VEH(VUID, Arr_t) = t
STEP 3) APP(AUID, App_Q) = APP(AUID, App_Q) +1

In steps 1 through 3 a vehicle (VUID) enters the APPROACH block from an upstream source. The vehicle's arrival time is recorded and the approach queue is increase by 1.

STEP 4) Wait until APP(AUID, App_Res) = 0

Step 4 is a waiting step, i.e. a step in which simulation clock time may pass before the next algorithmic step may be processed. If at the time of vehicle VUID's arrival, Veh(VUID, Arr_t), the stop bar is free (which also implies there is no queue) then step 5 is immediately executed (no simulated time passes). Otherwise, processing of the vehicle ceases until all other queued vehicles with an earlier arrival time are processed.

STEP 5) APP(AUID, App_Q) = APP(AUID, App_Q) – 1
STEP 6) If SIG(SUID, SigInd(phase)) = 0 then APP(AUID, App_HV) = VEH(VUID)
STEP 7) Set APP(AUID, App_Res) = 1

In steps 5 through 7 the vehicle moves into the free stop bar position. If the signal is red the vehicles unique identification number, VUID, is recorded as this vehicle is the lead queue vehicle when the light changes to green. (The ability to identify the lead *vehicles is required in a later step). Finally, the APPROACH stop bar is recorded as occupied.*

STEP 8) Wait until SIG(SUID, SigInd(phase)) = 1

Step 8 is another waiting step. If the signal light is red the vehicle is held at the stop bar until the light changes to green. When the light is green the processing of the vehicle through the intersection continues with steps 9 through 17.

STEP 9) If VEH(VUID) = APP(AUID, App_HV) then APP(AUID, App_LTC) = 0
STEP 10) APP(AUID, App_LTC) = APP(AUID, App_LTC) + 1
STEP 11) Set VEH(VUID, StUpD)

Steps 9 through 11 count the vehicles as the leave the stop bar during a green phase. The first three queued vehicles to leave the stop bar are assigned a start-up lost-time, according to Webster,(May, 1990).

STEP 12) Wait Veh(VUID, StUpD) * VEH(VUID, Agg_Fac)

Step 12 is another waiting step. If a vehicle has been assigned any start-up lost-time in steps 9 through 11 it is held for that time in step 12. As seen the default Webster average startup lost times are adjusted by an individual vehicle's aggressiveness factor.

STEP 13) Set Veh(VUID, D_App) = t - Veh(VUID, Arr_t)
STEP 14) Update App(AUID, App_D)

In steps 13 and 14 the vehicle completes its processing through the intersection. The Departure time and time spent (delay) at the intersection are recorded. The average approach delay is recalculated to include the current vehicle.

STEP 15) Wait Veh(VUID, Hdway)

Step 15 imposes a user input minimum departure headway, adjusted by a vehicle's aggressiveness factor. If there is a vehicle in the queue (i.e. APP(AUID, App_Q) > 0) it may not be released (steps 5 through 7) until this time is served.

STEP 16) Set APP(AUID, App_Res)
STEP 17) Release Vehicle to APP(AUID, App_Dest)

In steps 16 and 17 the stop bar is set to free and the current vehicle is released to It's downstream destination, allowing the next vehicle in the queue (step 4) to begin processing through the intersection (step 5).

Where:

APPROACH Block (object) Variables – APP(AUID, variable)

AUID =	APPROACH block instance unique identifier
App_Res =	Lane resource utilization, 0 = lead position empty (no queue), 1 = lead position taken
App_Q =	Approach Queue length at time t
App_D =	Average Vehicle Delay on Approach form time = 0 to time = t
App_Dest =	Destination of a vehicle instance after being processed by the APPROACH block
App_HV =	Unique identifier of vehicle at head of the queue during red phase
App_LTC =	Vehicle count, starting with the lead vehicle in the queue when the light turns green.

Vehicle Object Variables - VEH(VUID, variable)

VUID = Vehicle instance unique identifier

Arr_t = Current Block ENTER time

StUpD = Start-Up Lost-time delay, stored in vehicle's block delay attribute
Start-Up Lost-time is assigned according to Webster's delay, as follows

App_LTC = 1 then StUpD = 1.5
App_LTC = 2 then StUpD = 0.5
App_LTC = 3 then StUpD = 0.4
App_LTC > 3 then StUpD = 0

Hdway = departure headway between consecutive vehicles

D_App = delay experienced on approach

Agg_Fac = Vehicles Aggressiveness Factor

Traffic Signal Object Variables – SIG(SUID, variable)

SUID= Traffic Signal Block Instance Unique Identifier

phase = Intersection Movement, numbered according to NEMA numbering system

	1/5	Major Street Lefts
where phase =	2/6	Major Street Thru/Right
	4/8	Minor Street Thru/Right

SigInd(phase) = Signal Indication of phase at time t; where 0 if Red, 1 if Green

Green(phase) = Length of Green time for phase #, Where # same as in SigInd_#

Intph = Time remaining to be served of phase a t = 0

AllRed = Length of All Red Delay

Offset = Offset to beginning of East / West Green phase (i.e. phase = 2/6)

Model Variables

t = Current Simulation Clock Time

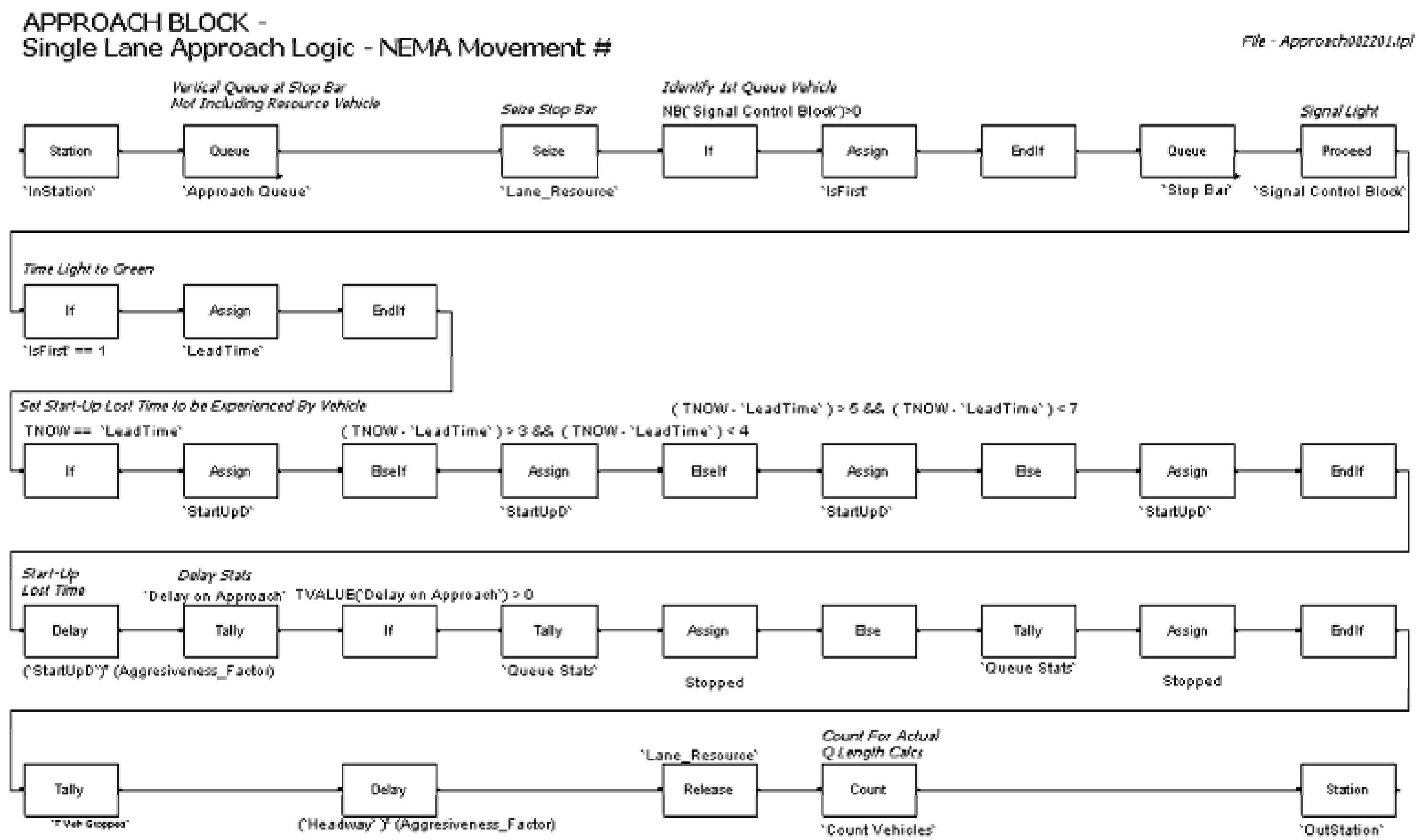


Figure 3.15 APPROACH Block Tier 3 Block Logical Construction

Intersection Blocks

An Intersection Block models an individual intersection in a Open-TS3 model. There are currently four different intersection blocks: PRETIMED, PRETIMED8P, ACTUATED8P, and SIGNAL. The PRETIMED block models two or three phase single-ring pre-timed control; the PRETIMED8P block handles pre-timed, eight phase dual ring control; the ACTUATED8P Block captures actuated eight-phase dual-ring control; and the SIGNAL block models adaptive control. This section will present a detailed description of the PRETIMED Block. Only a brief synopsis of PRETIMED8P and ACTUATED8P is presented, with the input dialogue boxes and block logic constructs given in Hunter and Machemehl, 2003. The SIGNAL block is discussed in chapter 6.

PRETIMED Block Overview

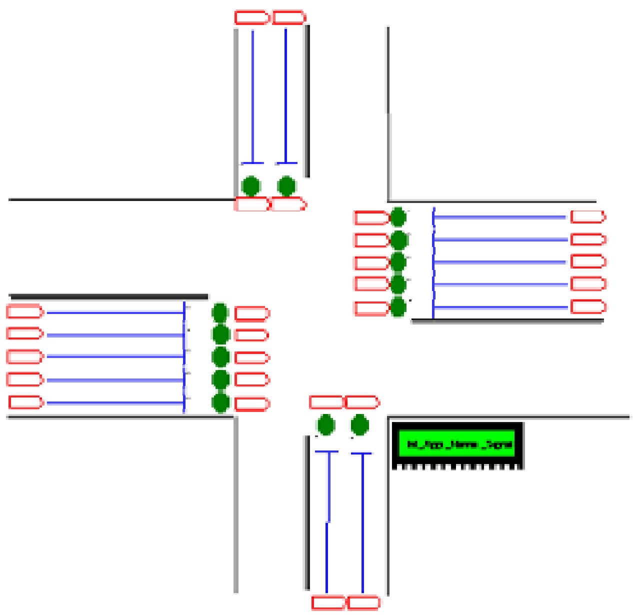


Figure 3.16 Open-TS3 Intersection Block Model Icon

As stated a PRETIMED block models an individual intersection in Open-TS3 model. Two or three phase pre-timed operation may be utilized and an intersection may have up to 4 total thru/right lanes (maximum 1 right) and 1 protected only left turn lane on the main arterial approach, and two thru/right lanes (maximum 1 right) on the minor street. 0 shows the model icon used to represent a Traffic Signal Block in Open-TS3 model.

Unlike other blocks which have a single logic structure, The PRETIMED Block consists of two distinct logic structures, a Signal Switching Logic object and an Intersection Movement object. The Signal Switching Logic object contains the signal light switching logic, which is based upon pre-timed control. The Intersection Movement object associates APPROACH blocks (which represent intersection lanes) and the Signal Switching Logic object together to form a complete intersection. The PRETIMED Signal Switching Logic object and Intersection Movement object are explained in the following sections.

Input

The modeler is required to enter data into the PRETIMED block at the time of model construction. This input is used by both the Signal Switching Logic object and the Intersection Movement object. The following defines each input.

Int_Appr_Name	The intersection name. This is the name that will be referenced by all APPROACH blocks associated with the given PRETIMED block.
1 and 5 Phase	The length of Phase 1/5 (Major Street - Left), in seconds. The phase time includes a 2 second all-red.
2 and 6 Phase	The length of phase 2/6 Phase (Major Street - thru/right), in seconds. The phase time includes a 2 second all-red.
4 and 8 Phase	The length of phase 4/8 (Minor Street – thru right), in seconds. The phase time includes a 2 second all-red.
Offset	This cycle offset, in seconds, set to the end of Phase 2 and 6 green. This must range between 0 and the sum of the three phases (1/5 + 2/6 + 4/8).
DestinationX_Y	This is the downstream destination for vehicles in movement X, lane Y. The lanes are numbered consecutively from the center most lane to the outside curb lane (Y = 1, 2, or 3), right lanes are labeled with Y = R. As left turns all currently occur from single lane bays during a protected movement phase, no Y value is given. For example destination 2_3 is the main street destination (NEMA movement 2) for vehicles in the third thru lane from the centerline.
Time to DestinationX	This is the minimum travel time from the intersection exit to the next downstream block for vehicles from movement X. Currently all lanes from the same movement are assigned the same time to destination.
Headways	The Headways button opens the departure headways input box, 0. These headways are the intersection departure headways that will be utilized by vehicles, as discussed in the APPROACH block (section 3.3.3).

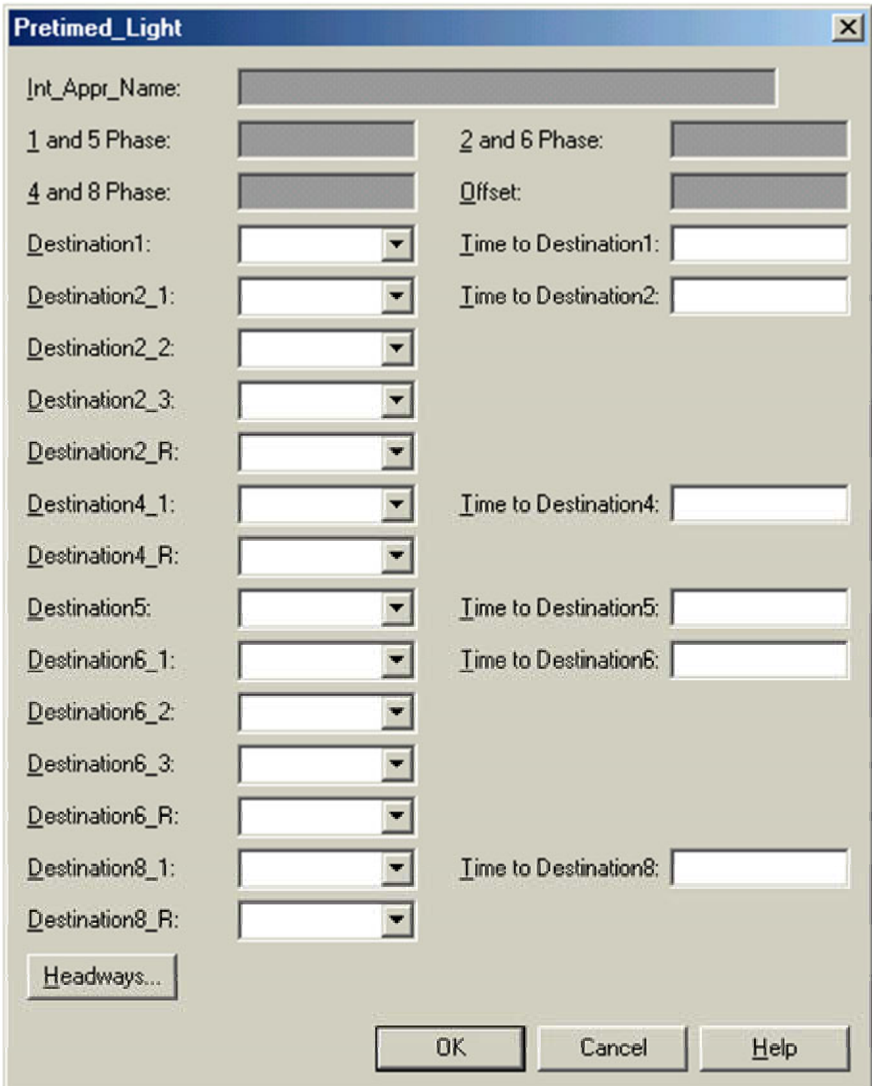


Figure 3.17 PRETIMED Block User Input Dialogue Box

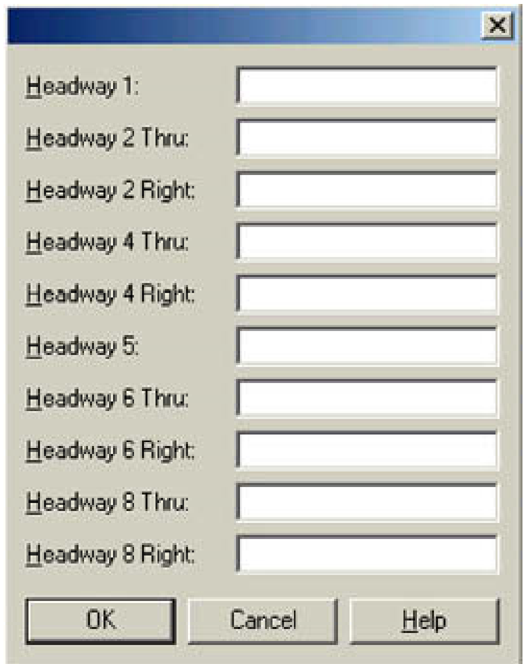


Figure 3.18 PRETIMED Block Departing Headways User Input Dialogue Box

Interfaces to other objects

A PRETIMED block must be associated with an upstream vehicle source block for each approach lane, a downstream destination block for each lane, vehicles, and model wide variables.

The upstream vehicle source for a PRETIMED block may be any tier 1 block which releases vehicles, including ENTER, QUEUECHANGE, PRETIMED, TURNBAY, LANEADD, LANEDROP, or another intersection block. Each lane of a PRETIMED block must have an upstream vehicle source, thus different block types may provide vehicles to a single PRETIMED block. For example, three different block types feed the PRETIMED block in the example model in 0. The upstream sources for the mainline thru vehicles are QUEUECHANGE Blocks, the sources for mainline right and left turns are the LANECHANGE blocks, and the minor street is fed directly by the ENTER blocks. The downstream source may be any tier 1 block that accepts vehicles. This includes, EXIT, QUEUECHANGE, other intersection blocks, TURNBAY, LANEADD and LANEDROP.

The PRETIMED block interacts with vehicles through the vehicle object interface. In the processing of a vehicle, a PRETIMED block will read and alter numerous vehicle attributes. Interaction with vehicles occurs primarily as a result of the APPROACH blocks contained within the Intersection Movement object. Also, the PRETIMED block accesses model wide variables, primarily the simulation clock time.

Most interactions between the PRETIMED block and the other simulation objects are a result of the Intersection Movement object. The Signal Switching Logic object is isolated within the PRETIMED block, maintaining the cycle pattern for that block, requiring no interaction with other objects, except for the simulation clock time.

The PRETIMED block allows other blocks to check attributes of the PRETIMED block. This includes all attributes made available by APPROACH blocks (section 0), the current status of the signal lights, and any intersection measures of effectiveness, such as current total delay.

Intersection Movement Object

The Intersection Movement object block logic is shown in 6.20 and its algorithm is listed in this section. The purpose of the Intersection Movement object is to associate the Signal Switching Logic object (i.e. a signal light) and the APPROACH blocks to form a complete intersection. In 0 there are fourteen pairs of entry (\`InStationNX_Y\` blocks) and exit (ROUTE blocks) points. Each set of entry and exit points represents an approach lane of the intersection and is labeled according to its associated NEMA movement number and lane. As seen, the major street left turn movements (NEMA movements 1 and 5, respectively) are single lane, the major street thru/right movements (NEMA movements 2 and 6, respectively) may contain as many as four lanes, and the minor street thru/right (NEMA movements 4 and 8, respectively) may be up to two lanes. Where the user does not utilize the maximum number of lanes, unassigned entry/exit pairs are ignored when the model is compiled.

Each vehicle approaching the intersection from an upstream block is assigned to one entry (lane) point. This assignment is performed by the user in the setting of routes between blocks (see Chapter two). Each time a vehicle entity enters, the Intersection Movement object algorithm is executed for that entry point.

PRETIMED Block Intersection Movement Object Algorithm

STEP 0) Read IMO(IUID, IMO_Dest(Lane)), $\forall$ Lane, $\forall$ IUID
Read IMO(IUID, IMO_DestTime(Lane)), $\forall$ Lane, $\forall$ IUID

Step 0 is the initialization step which occurs at the start of the simulation, t = 0. For each Intersection Movement object (all IUID) the downstream block and unimpeded time to that block are read for each intersection lane.

Steps 1 through 5 give the algorithm by which each Lane of an Intersection Movement object, IMO(IUID, Lane) processes a vehicle, VEH(VUID).

STEP 1) VEH(VUID) Enters IMO(IUID, Lane) at time = t
In Step 1 a vehicle enters a lane of the intersection.

STEP 2) Process through APP(AUID) Block
*An APPROACH block processes the veh*icle, see section 3.3.3.

STEP 3) Set VEH(VUID, T_Time) = See section 3.3.1.5, SET TRAVEL TIME
STEP 4) Wait VEH(VUID, T_Time)
STEP 5) Release VEH(VUID) to IMO(IUID, ILO_Dest(Lane))

In steps 3, 4, and 5 the travel time from the current block to the next downstream block is determined (see SET TRAVEL TIME, section 0), served, and the vehicle departs the PRETIMED Block.

Where:

Intersection Movement Object Variables – IMO(IUID, variable)

IUID =	Intersection Movement object unique identifier.
Lane =	Intersection Lane, labeled according to NEMA numbering and lane number, i.e. Lane 2_3 is NEMA movement 2 (eastbound thru) third lane from centerline.
IMO_Dest(x) =	User Input Downstream destination for vehicles departing from Lane x.
IMO_DestTime(x) =	User Input Average Travel Time to Downstream Destination block from Lane x.

Vehicle Object Variables - VEH(VUID, variable)

VUID =	Vehicle instance unique identifier
T_Time =	Travel Time, from current block to downstream destination block

Model Variables

t =	Current Simulation Clock Time

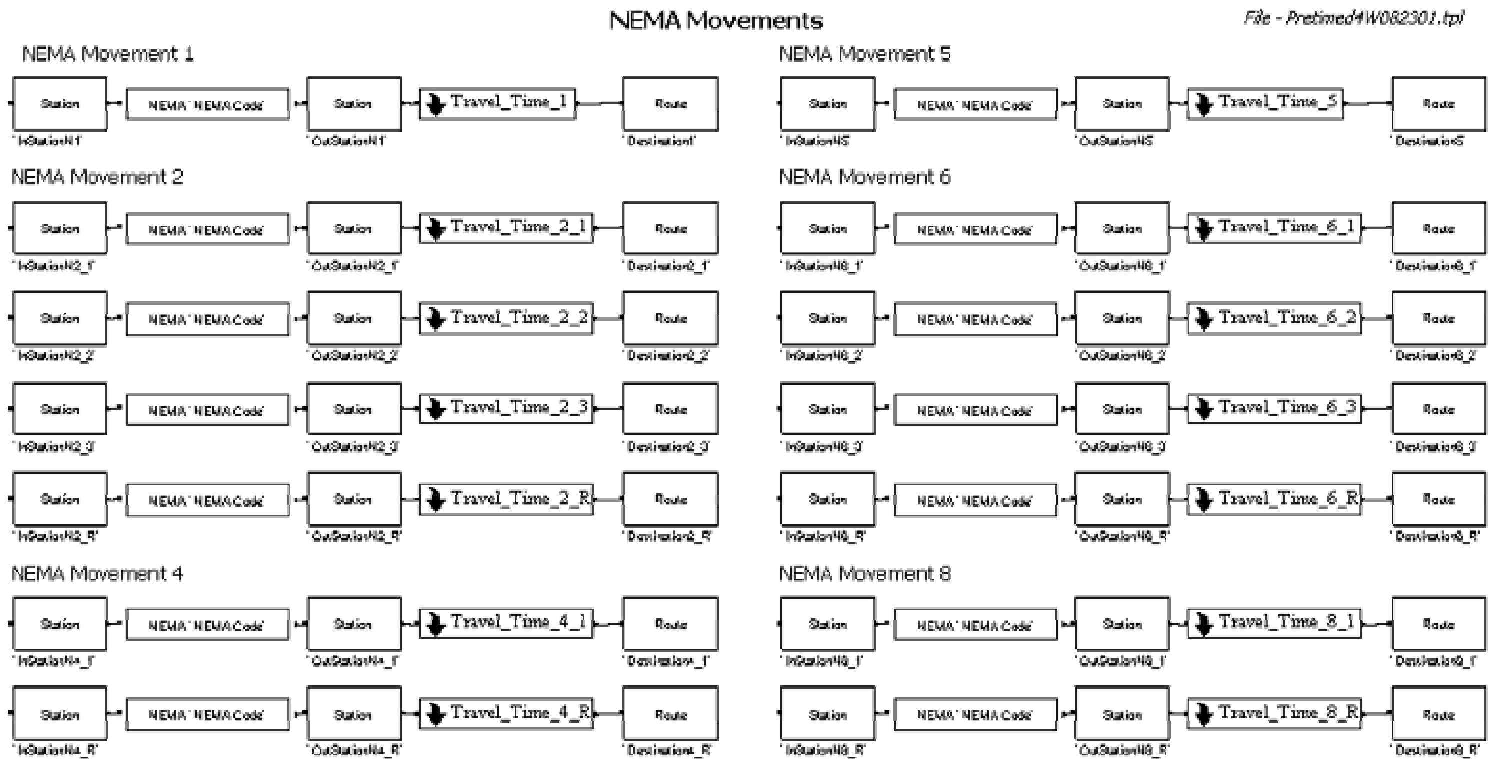


Figure 3.19 PRETIMED Block Intersection Movement Object Basic Block Logical Construction

PRETIMED Signal Switching Logic Object Block Logic and Algorithm

The PRETIMED Traffic Signal Object Block logic is given in 0 and an algorithmic version is given below. The physical equivalent to this object in the real world is the signal controller program. The Signal Switching Logic object captures the operations of the traffic signal, essentially acting as a clock, maintaining the intersection pre-timed cycle pattern. At any time t the state of signal (i.e. red or green) is read by the Intersection Movement object that processes any present vehicles accordingly.

This block logic consists primarily of two parts, the first determines the current point in the cycle at time t = 0 (the start of the simulation) based on the user input offset and serves the remainder of the cycle. The second part maintains the cycle throughout the remainder of the simulation. This first item is accomplished in algorithm steps 1 through 4 (the leftmost UNBLOCK#/# and DELAY blocks in 0). Steps 5 through 17 (the right side of 0) accomplish the task of maintaining the cycle pattern.

The Signal Switching Logic object is different from all the other blocks in that it contains an "imaginary" object, as discussed in Chapter two. In all other blocks the firing of the block logic is initiated either by the movement of a vehicle object through the block or a scheduled event. The Signal Switching Logic object instead initiates its logic by creating an object at the start of the simulation and continuously cycling that object through the logic, as a marker for where the PRETIMED block is in the signal cycle at the current clock time. This object has no real world physical equivalent; it is simply a logical construct. This is very similar to the way logic is processed in Petri Nets (List and TroutBeck, 1999), Section 2.2.2.

PRETIMED Block Traffic Signal Logic Algorithm

STEP 0) Read SIG(SUID, Offset), ∀ SUID
Read SIG(SUID, Green(phase)), ∀ phase, ∀ SUID
Set SIG(SUID, AllRed) = 2, ∀ SUID
Set SIG(SUID, SigInd(phase)) = 0, ∀ phase, ∀ SUID

Step 0 is an initialization step which occurs at the start of the simulation, t = 0. For each PRETIMED Block Traffic Signal Object (for all SUID) the offset and green phase lengths are read from the user inputs. All all-red times are currently defaulted to 2 seconds.

Steps 1 through 17 give the algorithm by which PRETIMED SIGNAL Control is modeled in a PRETIMED Block.

STEP 1) Set SIG(SUID, phase) to initial green phase, ∀ SUID
SET SIG(SUID, Intph), ∀ SUID

The initial green phase and amount of time remaining to be served for that phase are determined for the given offset and phase lengths. That is, if the offset and phasing pattern result in 20 seconds of phase 2 remaining to be served then SIG(SUID, Phase) is set equal to 2 and SIG(SUID, Intph) = 20.

STEP 2) Set SIG(SUID, SigInd(phase)) = 1
STEP 3) Wait SIG(SUID, Intph)

In steps 2 and 3 the initial phase is set to green and the remaining green time for that phase is served, i.e. when step 3 is simulation time t = 0 (time at simulation start) + Intph = Intph.

STEP 4) If SIG(SUID, phase) = 1/5 then GoTo Step 7
ElseIf SIG(SUID, phase) = 2/6 GoTo Step 11
Else GoTo Step 15

Step 4 switches the signal to the next appropriate green phase. Currently a three phase cycle is utilized; 1/5 (leading Major Street lefts) to 2/6 (Major Street thru/right) to 4/8 (Minor thru/right). Thus, if the initial green phase was 1/5 then the next green phase would be 2/6 (Step 7), if the initial green phase was 2/6 than the next green phase is 4/8 (Step 11), and finally if the initial green phase was 4/8 the next green phase is 1/5 (Step 15).

STEP 5) Set SIG(SUID, SigInd(1/5)) = 1
STEP 6) Wait SIG(SUID, Green(1/5))

This step is interpreted as; do not proceed to step 7 until the simulation clock time equals the current simulation clock time(t) plus the length of phase 1/5. Similar interpretations may be made for the Waits in steps 8 through 17.

STEP 7) Set SIG(SUID, SigInd(1/5)) = 0
STEP 8) Wait SIG(SUID, AllRed))
STEP 9) Set SIG(SUID, SigInd(2/6)) = 1
STEP 10) Wait SIG(SUID, Green(2/6))
STEP 11) Set SIG(SUID, SigInd(2/6)) = 0
STEP 12) Wait SIG(SUID, AllRed))
STEP 13) Set SIG(SUID, SigInd(4/8)) = 1
STEP 14) Wait SIG(SUID, Green(4/8))
STEP 15) Set SIG(SUID, SigInd(4/8)) = 0
STEP 16) Wait SIG(SUID, AllRed))
STEP 17) Go To Step 5

Steps 5 thru 17 are cycled through until the simulation is ended. These steps control the phasing pattern as discussed in Step 4. A two second all-red is included at the end of each green phase.

Where:

Traffic Signal Object Variables – SIG(SUID, variable)
SUID= Traffic Signal Block Instance Unique Identifier

phase =	Intersection Movement, numbered according to NEMA numbering system 1/5 Major Street Bound Lefts where phase = 2/6 Major Street Thru/Right 4/8 Minor Street Thru/Right
SigInd(x) =	Signal Indication of phase x at time t; where 0 if Red, 1 if Green
Green(x) =	Length of Green time for phase x
Intph =	Time remaining to be served of phase a t = 0
AllRed =	Length of All Red Delay
Offset =	Offset to beginning of East / West Green phase (i.e. phase = 2/6)

Model Variables

t =	Current Simulation Clock Time

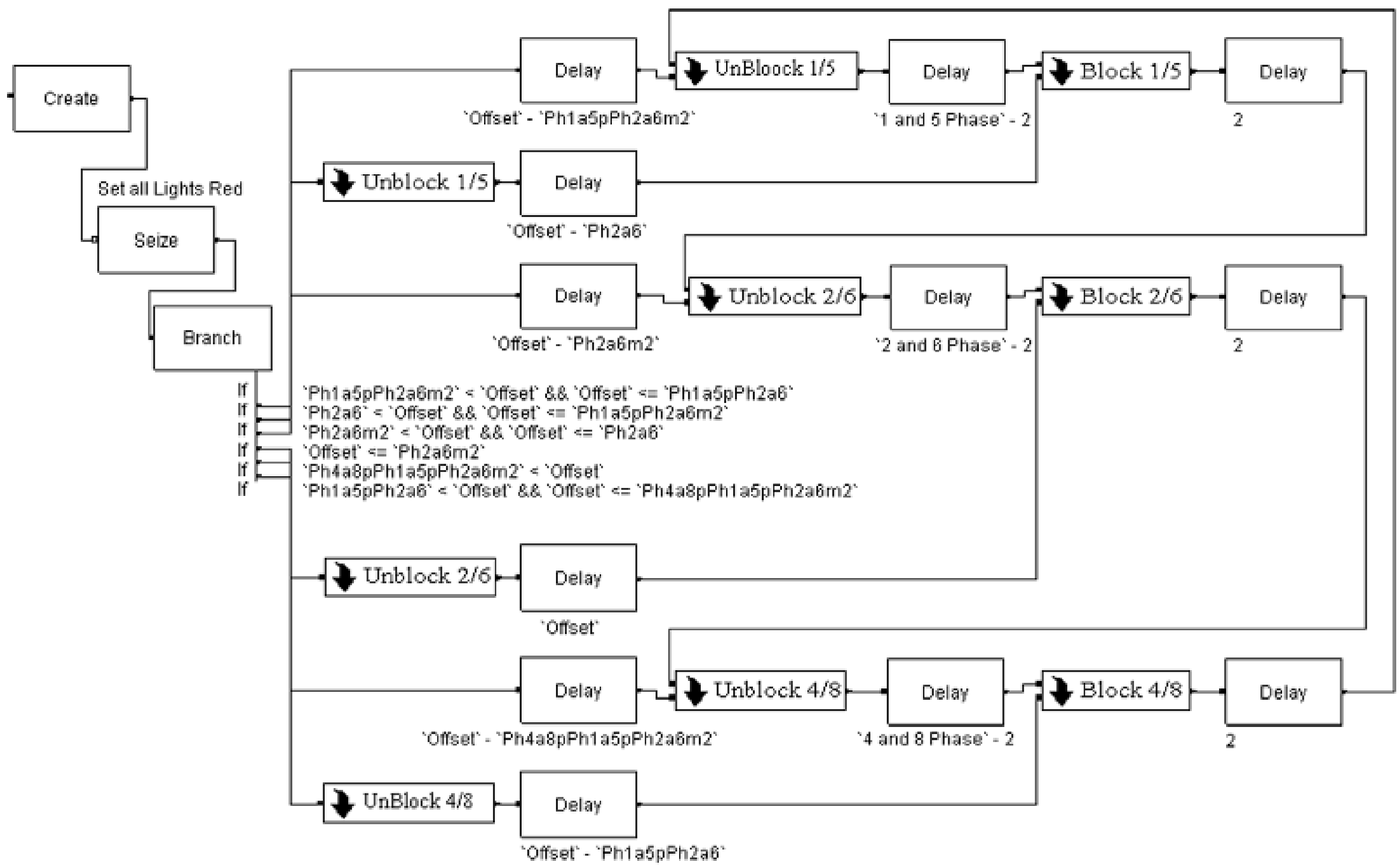


Figure 3.20 PRETIMED Signal Switching Logic Object Block Logic

PRETIMED8P and ACTUATED8P

Given the extensive nature of these block's logical constructs it is not useful to present detailed algorithmic forms as with the PRETIMED and other blocks discussed within this chapter. This section presents a brief synopsis of these blocks, with Hunter and Machemehl (2003) containing the object class hierarchies, input dialogue requirements, and logical constructs (PHASE#_PT, PHASE#_AC, and MAX/GAP blocks).

As stated in the beginning of section 0 the PRETIMED8P and ACTUTATED8P are intersection blocks. These blocks model more robust signal control. Each block models eight-phase dual-ring intersection signal control, with PRETIMED8P modeling pre-timed control and ACTUATED8P capturing actuated control. As with the PRETIMED block the PRETIMED8P and ACTUATED8P consist of an Intersection Movement object and a Signal Switching Object. Each block's Intersection Movement object logic follows the same general format as in the PRETIMED block while the Signal Switching logic is significantly more complicated. The PRETIMED8P Signal Switching Logic consists of eight objects (blocks PHASE1_PT to PHASE8_PT), each capturing the logic of one of the eight phases. The ACTUATED8P signal logic contains nine separate objects, one object to capture the logic of each phase (blocks PHASE1_AC to PHASE8_AC) and a object to capture the maxout / gapout logic (MAX/GAP). The logical constructs and input requirements for all of these objects may be found in Hunter and Machemehl, 2003.

QUEUECHANGE Block

Overview

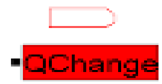


Figure 3.21 Open-TS3 QUEUECHANGE Block Model Icon

When a vehicle joins a multi-lane queue it is allowed to switch from its current lane to an immediately adjacent lane with a shorter queue, where both queues service the same movement. The QUEUECHANGE block has been constructed to facilitate this queue change. A QUEUECHANGE block may be used in Open-TS3 simulation anywhere multiple lanes serve a common movement, for example, multiple thru lanes on an intersection approach or multiple queuing lanes at a toll plaza.

The queue switching decision process begins when the vehicle arrives at the back of the queue and is completed before the advancement of the simulation clock time. The probability of a vehicle switching queues is a user calibrated step function based on the difference between the current and adjacent queue lengths. Different probabilities of switching may be assigned to queue differences 0, 1, 2, 3, and 4+. A vehicle only considers changing queues upon arrival to the back of a queue. Also, if the queues on both sides (driver and passenger) of the current queue serve the same movement only switching to the shorter queue will be considered.

Any block that releases vehicles can be upstream of a QUEUECHANGE block and any block that accepts vehicles and requires queuing, typically an intersection block will be downstream. As the QUEUECHANGE blocks function is to facilitate queue switching at the downstream block it is located with the downstream block, no additional travel time or delay is assigned as a result of a vehicle passing through the QUEUECHANGE block. Also, each lane of an intersection in which one desires to allow vehicles to change lanes within a queue requires a QUEUECHANGE block. The example model in 0 is seen to have six QUEUECHANGE blocks, one for each major street thru lane.

User Input

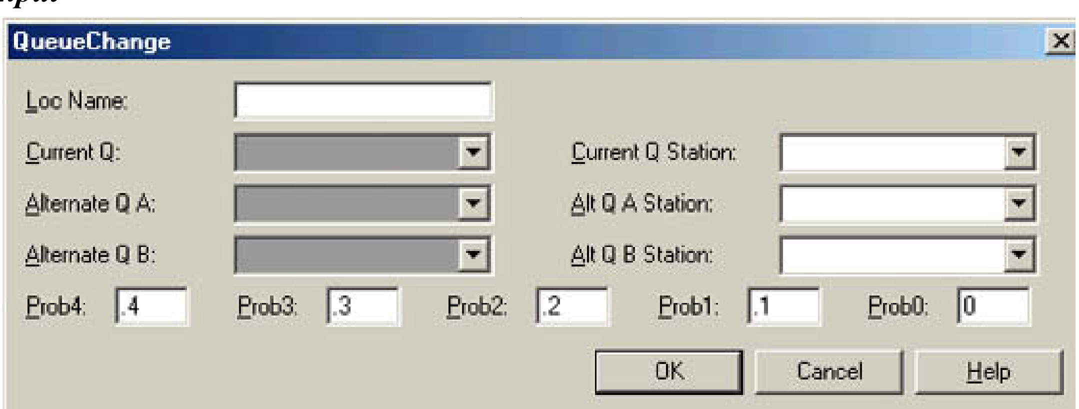


Figure 3.22 QUEUECHANGE Block User Input Dialog Box

0 is the QUEUECHANGE block input dialog box. The following describes each input.

Loc Name	A unique location name for the QUEUECHANGE block. This is the name other blocks will use to refer to the QUEUECHANGE block.
Current Q	This is the name of the QUEUE block (tier 3) that the arriving vehicle would enter if it does not change queue destinations. The QUEUE block name is defined in another block, typically an APPROACH block, and only referenced in the QUEUECHANGE block. A pull-down list of all QUEUE blocks currently defined in the model is provided. Additional discussion on selecting the correct QUEUE blocks is given in the Interfaces to Other Objects Section, 0.
Alternate Q A	This is the name of the driver side adjacent QUEUE block, where both QUEUE blocks (*Current Q* and *Alternate Q A*) service the same movement. If a driver side adjacent queue does not exist then no input is required.
Alternate Q A	This is the name of the passenger side adjacent QUEUE block, where both QUEUE blocks service the same movement. If a passenger side queue does not exist then no input is required.
Current Q Station	This is the name of the STATION block (tier 3) with which the current QUEUE block (*Current Q*) is associated. The STATION may be thought of as the simulated equivalent of the real world stop bar at the head of the queue. The STATION block name is defined in another block, typically the APPROACH block, and only referred to here. A pull-down list of all STATION blocks currently defined in the model is provided. Additional discussion on selecting the correct STATION block is given in the interfaces to other objects section 0.
Alt Q A Station	This is the name of the STATION block associated with *Alternate Q A*.
Alt Q B Station	This is the name of the STATION block associated with *Alternate Q B*.
Prob#	This is the user input probably associated with a vehicle changing lanes, where the difference between the current queue length and the alternate queue length is measured in vehicles.

Interfaces to other Objects

A QUEUECHANGE block must be associated with an upstream vehicle source block, a downstream block that accepts vehicles and requires queuing, vehicles, model wide variables and downstream QUEUE and STATION blocks.

The upstream vehicle source for a QUEUECHANGE block may be any block that releases vehicles, including ENTER, PRETIMED, TURNBAY, LANEADD and LANEDROP. The travel time listed in the upstream block should be the travel time from the upstream block to the block downstream of the QUEUECHANGE block. To understand this, it is important to recall that Open-TS3 utilizes a vertical queuing model. Thus, the QUEUECHANGE block is acting at the stop bar of the downstream block, and no additional free flow travel time is experienced from the QUEUECHANGE block to the downstream block.

The downstream block must be a block that captures vehicle queuing. Currently only Open-TS3 intersection blocks model queues although as noted in the input dialogue box, instead of referencing the tier 1 block, the tier 3 QUEUE and STATION blocks are directly referenced. It is necessary to directly reference these tier 3 blocks as just referencing the tier 1 block in which they reside may not be sufficient for unique identification. For example, the PRETIMED block in 0 contains 14 different QUEUE blocks, one for each lane. It is necessary to specifically designate which QUEUE block each QUEUECHANGE block is to be associated with. To allow for clarity in identifying the correct QUEUE and STATION blocks for user input, PRETIMED blocks automatically name these blocks “PRETIMED Block Name_NEMA Movement_Lane Number”_S for STATION blocks, and _Q for QUEUE blocks.

The QUEUECHANGE block also interfaces with vehicles and model-wide variables, as all the other tier 1 blocks that have been discussed. The QUEUECHANGE blocks offer no other attributes that other blocks can read.

QUEUECHANGE Logic

The algorithmic form of QUEUECHANGE logic is given in this section and the Open-TS3 QUEUECHANGE block tier 3 basic block logic is given in 0.

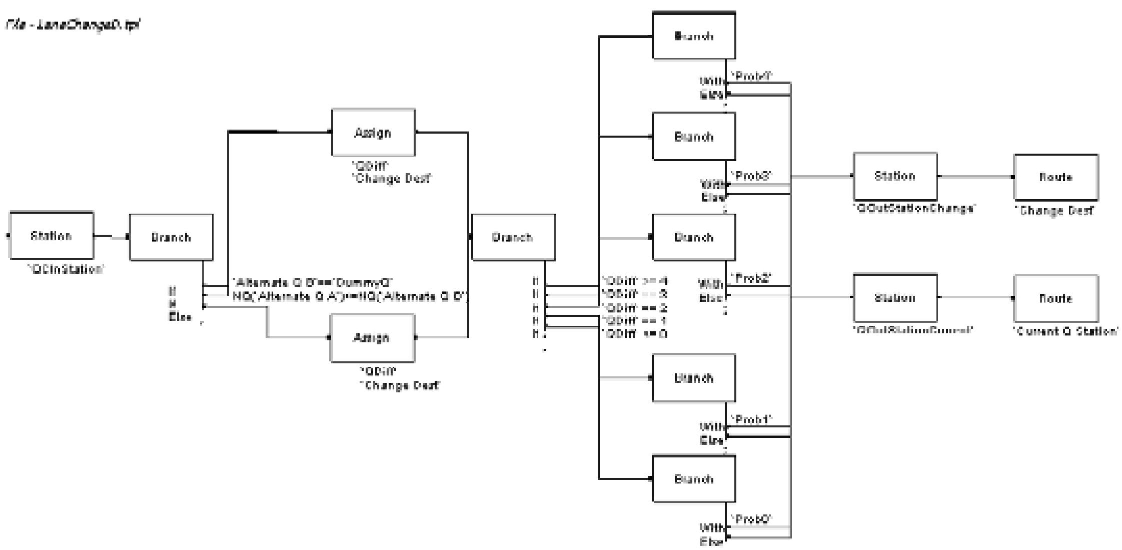


Figure 3.23 QUEUECHANGE Block ARENA Basic Block Logic

Queue Change Algorithm

STEP 0) QUE(QUID, Pr_QDiff) = user input queue changing probability, ∀ QUID, ∀ QDiff

QUE(QUID, DS_Dest) = Driver Side Queue APPROACH Block Id, ∀ QUID
QUE(QUID, PS_Dest) = Passenger Side Queue APPROACH Block Id, ∀ QUID
QUE(QUID, Ct_Dest) = Current Queue APPROACH Block Id, ∀ QUID

Step 0 is the initialization step which occurs at the start of the simulation, t=0. At initialization the user input queue changing probabilities are read and the current and adjacent driver side and passenger side queues are identified.

Steps 1 through step 7 give the algorithm by which each QUEUECHANGE block determines if a vehicle remains in the current queue or switches to a shorter adjacent queue serving the same movement.

STEP 1) VEH(VUID) Enters QUE(QUID) at time t
STEP 2) QUE(QUID, Q_CT) = APP(QUE(QUID, CT_Dest), App_Q)
QUE(QUID, Q_DS) = APP(QUE(QUID, DS_Dest), App_Q)
QUE(QUID, Q_PS) = APP(QUE(QUID, PS_Dest), App_Q)

In Steps 1 and 2 a vehicle enters a QUEUECHANGE block at simulation time t and the current, driver side and passenger side queue lengths are recorded.

STEP 3) If min(QUE(QUID, Q_DS), QUE(QUID, Q_PS)) = QUE(QUID, DS_Dest)
Set QUE(QUID, SW_Dest) = QUE(QUID, DS_Dest)
Else Set QUE(QUID, SW_Dest) = QUE(QUID, PS_Dest)

Currently, a QUEUECHANGE Block only allows a vehicle to consider its current queue and one adjacent queue. Therefore the shorter adjacent queue (driver or passenger side) is determined and set as the potential destination queue to be considered in steps 4 through 7.

STEP 4) Set QUE(QUID, Qdiff) = QUE(QUID, Q_CT) – QUE(QUID, Q_SW)

In step 4 the difference between vehicle VEH(VUID)'s current queue destination and the adjacent queue destination is calculated.

STEP 5) Set QUE(QUID, S_Pr) = U(0,1) realization
STEP 6) if QUE(QUID, S_Pr) <= QUE(QUID, Pr_QDiff)
Set QUE(QUID, QUE_Dest) = QUE(QUID, SW_Dest)
Else Set QUE(QUID, QUE_Dest) = QUE(QUID, CT_Dest)

In steps 5 and 6 a uniform distribution is sampled. If the realization is less than or equal to the user input probability of changing for the given queue difference then vehicle VEH(VUID) changes queues, otherwise it remains in the current queue.

STEP 7) Release Vehicle to QUE(QUID, QUE_Dest)

Processing of vehicle VEH(VUID) through QUEUECHANGE (QUID) is complete and the vehicle enters the selected queue, i.e. Q_Dest.

Where:

QUEUECHANGE Block (object) Variables - QUE(QUID, variable)

QUID =	QUEUECHANGE block instance unique identifier
Q_DS =	Driver side queue length
Q_PS =	Passenger side queue length
Q_CT =	Current lane queue length
QDiff =	Difference between current and shortest adjacent queue where: QDiff = 0, Q_CT-min(Q_DS,Q_PS) <= 0 i,, Q_CT-min(Q_DS,Q_PS) = i, i = 1,2,3 4, Q_CT-min(Q_DS,Q_PS) >= 4
Pr_QDiff =	User input probability of switching for QDIFF
DS_Dest =	Destination queue ID if vehicle switches to driver side queue
PS_Dest =	Destination queue ID if vehicle switches to passenger side queue
SW_Dest =	Vehicle destination queue ID if vehicle switches queues
CT_Dest =	Vehicle destination queue ID if vehicle remains in current queue
QUE_Dest =	Destination of vehicle instance after being processed by QUEUECHANGE block
S_PR =	Realization from U(0,1)

APPROACH Block (object) Variables – APP(AUID, variable)

AUID =	APPROACH block instance unique identifier
App_Q =	Approach Queue length at time t

Vehicle Object Variables - VEH(VUID, variable)

VUID =	Vehicle instance unique identifier

Model Variables

t =	Current Simulation Clock Time

TURNBAY Block

TURNBAY Block Overview

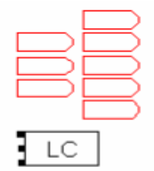


Figure 3.26 Open-TS3 TURNBAY Block Model Icon

The TURNBAY block assigns each vehicle to a movement (left, through, right) when the vehicle approaches an intersection. The assignment of vehicles to a turn movement is random, based on user input turn percentages. Where there exists more than one upstream lane the user may further define the percentage of turning vehicles that are to arrive from each upstream lane. The process of vehicle assignment to turning movements is discussed in detail in section 0.

Currently this block allows for one lane for each turning movement (left and right) and up to three thru lanes. Also, it is required that the number of upstream approach lanes equal the number of intersection thru lanes. Where this is not the case a LANEDROP or LANEADD block must first be used. Lane changing is taken to physically occur where the turn bay begins. As a vertical queuing model is utilized it is assumed that vehicles are always able to enter their desired lane, i.e. turn bay blocking due to adjacent queue backup is not captured. Also, the weaving process is not directly modeled, although user input has been provided to allow for some user calibration through the assignment of delays to vehicles changing lanes.

In the simulation block-diagram layout any block which releases vehicles, including ENTER, LANEADD, LANEDROP, and intersection blocks, may be placed upstream of a TURNBAY block and any block that accepts vehicles, including EXIT, LANEADD and LANEDROP, and intersection blocks may be placed downstream.

User Input

As seen in 0 the user is required to provide numerous pieces of data for the TURNBAY block. The requested data is as follows:

Lane Change Location A unique, user assigned, name that is used by other blocks when referencing the TURNBAY block

Number of Lanes The number of lanes upstream of the TURNBAY block.

Left Percent The percentage of total approach traffic to be assigned to the left turn movement.

Thru Percent The percentage of total approach traffic to be assigned to the thru movement.

Right Percent The percentages of total approach traffic to be assigned to the right turn movement.

% Lefts From Ln # The percentage of left turning vehicles from the lane number #. The sum of percentages from lanes 1, 2, and 3 must equal 100.

% Rights From Ln # The percentage of right turning vehicles from the lane number #. The sum of percentages from lanes 1,2, and 3 must equal 100.

Left Stop Bar	The STATION block that represents the left turning bay stop bar. See discussion in QUEUECHANGE (section 0) for a discussion on referencing STATION Blocks.
Lane # Stop Bar	The STATION block that represents lane number # stop bar.
Right Stop Bar	The STATION block that represents the left turning bay stop bar.
Time to Left Stop Bar	Minimum travel time to the left turn bay stop bar.
Time to Ln # Stop Bar	Minimum travel time to lane number # stop bar.
Time to Rt Stop Bar	Minimum travel time to the right turn stop bar.

Figure 3.27 TURNBAY Block User Input Dialogue Box

TURNBAY Logic Switches

A few notes of explanation are necessary to assist in understanding the TURNBAY logic in 0. This logic incorporates several switches (Boolean expressions), an ARENA logic tool, tied to different logic blocks. When a switch is false the blocks associated with that switch are not included in the compilation of the SIMAN code. Switches allow for the construction of a more robust logic, one that can be applied to different scenarios. In an ARENA block-diagram (0) text in square brackets, [ ], indicates a switch.

For example, without the use of switches a separate TURNBAY block would have to be developed for each possible number of thru lanes; 1, 2, or 3. Each of these possibilities is incorporated into the TURNBAY block through the use of four switches: One Lane, Two Lanes, Three Lanes, and Two or Three Lanes. One-Lane tests if there is exactly one thru lane, Two-Lane tests if there are exactly two thru lanes, Three-Lane tests if there are exactly three thru lanes, and Two- or Three-Lane tests if there are two or three thru lanes. To demonstrate the effect of switches, 0 shows the logic from 0 that would be compiled at a TURNBAY block location with only two thru lanes. In such an instance all blocks in 0 with a [Three-Lanes] or a [One-Lane] switch would not be incorporated in the compiled code.

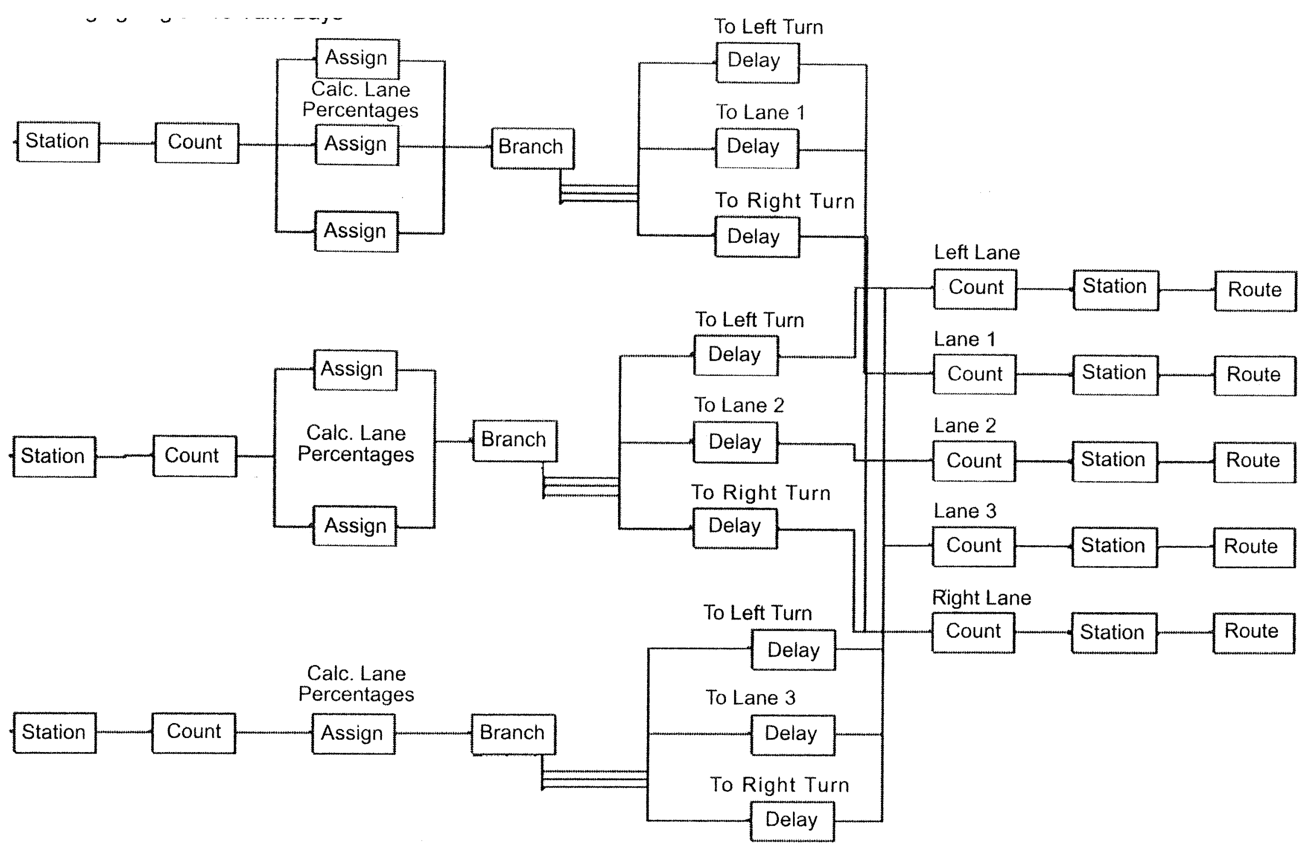


Figure 3.28 TURNBAY Block ARENA Basic Block Logic Construction

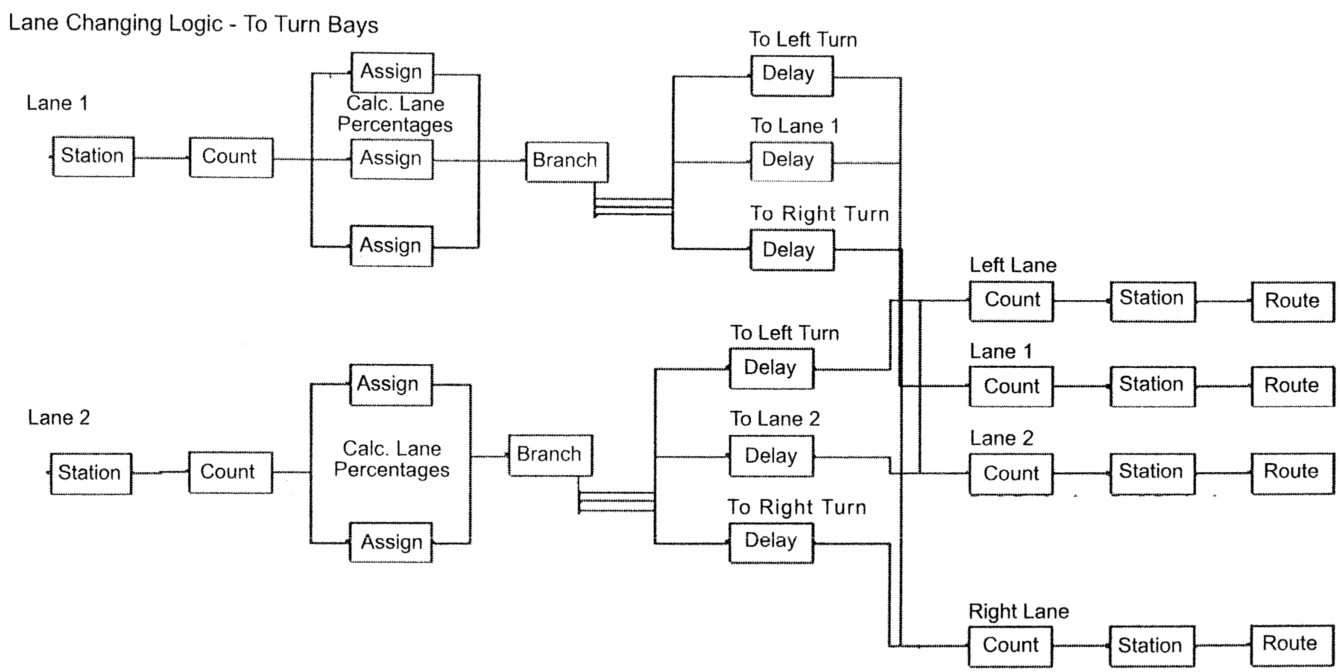


Figure 3.29 TURNBAY Block ARENA Basic Block Logic Construction for Two Upstream Lanes

TURNBAY Logic Algorithm

The section contains the lane changing logic modeled in 0 in algorithmic form. The TURNBAY logic first assigns vehicles to a movement (left, thru or right) and then to the appropriate approach lane (turning bay or thru lane).

TURNBAY Algorithm

STEP 0) TRN(TUID, SW_1L) = Value of One_Lane Switch ∀ TUID
TRN(TUID, SW_2L) = Value of Two_Lanes Switch ∀ TUID
TRN(TUID, SW_3L) = Value of Three_Lanes Switch ∀ TUID
TRN(TUID, SW_23L) =Value of Two_Or_Three_Lane Switch ∀ TUID
TRN(TUID, Pt_Lt) = User Input Approach Left Turn Percentage ∀ TUID
TRN(TUID, Pt_Rt) = User Input Approach Right Turn Percentage ∀ TUID
TRN(TUID, Lt(EnLn)) = User Input LT Percentage from Lane EnLn ∀ EnLn, ∀ TUID
TRN(TUID, Rt(EnLn)) = User Input RT Percentage from Lane EnLn ∀ EnLn, ∀ TUID
TRN(TUID, TRN_Dest(ExLn)) = User Input Destination ∀ ExLn, ∀ TUID
TRN(TUID, D_EnLn(ExLn)) = User Input Delay ∀ EnLn ∀ TUID
TRN(TUID, TRN_DestTime) = User Input Travel Time∀ TUID
Set TRN(TUID) block logic based on switches, SW_1L, SW_2L, SW_3L, and SW_23L

Step 0 is the initialization step which occurs at the start of the simulation. In this step the values of the ARENA Switches are determined and the logic is set (see section 3.3.6.3). Also the user inputs for overall approach left and right turns, individual lane shares of left and right turns, downstream block destination and minimum downstream travel time are read.

Step 1 through Step 11 give the algorithm by which a vehicle, VEH(VUID), is processed through a TURNBAY Block, TRN(TUID).

STEP 1) VEH(VUID) Enters TRN(TUID, EnLn) at time t
STEP 2) TRN(TUID, En_Ct(EnLn)) = TRN(TUID, En_Ct(EnLn)) +1

In steps 1 and 2 vehicle VEH(VUID) enters the TURNBAY block TRN(TUID) at Lane EnLn and that lane's counter is incremented by 1.

STEP 3) $$\text{TRN(TUID,TVP(EnLn))} = \frac{\text{TRN(TUID, En\_Ct(EnLn))}}{\sum_{x=\text{Approach Lanes}} \text{TRN(TUID, En\_Ct(x))}} \quad \forall \text{ ExLn}$$

In Step 3 the Percentage of the total entering approach traffic on each lane is updated.

STEP 4) Update TRN(TUID, Pr_Lt(EnLn)), TRN(TUID, Pr_Rt(EnLn))

Step 4 updates the probability that vehicle VEH(VUID), which enters TURNBAY block TRN(TUID) on lane EnLn, will turn left or right. This calculation is described in section 3.3.6.5.

STEP 5) Set TRN(TUID, S_Pr) = U(0,1) realization
STEP 6) If
TRN(TUID, S_Pr) <= TRN(TUID, Pr_Lt) then TRN(TUID, ExLn) = Left
Else if
TRN(TUID, Pr_Lt) < TRN(TUID, S_PR) <= TRN(TUID, Pr_Rt) + TRN(TUID, Pr_Lt)
then TRN(TUID, ExLn) = Right
Else
TRN(TUID, ExLn) = TRN(TUID, EnLn)

In steps 6 and 7 a realization from a uniform(0,1) distribution is taken and VEH(VUID) is assigned to a turning movement based on the sample and turn movement probabilities calculated in step 4.

STEP 7) Wait TRN(TUID, D_EnLn_ExLn)
STEP 8) TRN(TUID, Ex_Ct(ExLn)) = TRN(TUID, Ex_Ct(ExLn)) + 1

In steps 8 and 9 vehicle VEH(VUID) is delayed the user input delay for lane change and the exit lane vehicle count is incremented by 1.

STEP 9) Set VEH(VUID, T_Time) = See section 3.3.1.5, SET TRAVEL TIME
STEP 10) Wait VEH(VUID, T_Time)
STEP 11) Release VEH(VUID) to TRN(IUID, TRN_Dest(ExLn))

In steps 9, 10, and 11 the travel time from the current block to the next downstream block is determined (see SET TRAVEL TIME, section 3.3.1.5), served, and the vehicle departs the TURNBAY Block.

Where:

TURNBAY Block (object) Variables – TRN(TUID, variable)

TUID =	TURNBAY block instance unique identifier
SW_1L =	Value of One_Lane switch; true if one lane, false if two or three lanes
SW_2L =	Value of Two_Lanes switch; true if two lanes, false if one or three lanes
SW_3L =	Value of Three_Lanes switch; true if three lanes, false if one or two lanes
SW_23L =	Value of Two_Or_Three lanes switch; true if two or three lanes, false otherwise
EnLn =	Lane in which vehicle enters, where EnLn = 1,2, or 3

ExLn = Lane in which vehicle exits, where ExLn = Left, 1, 2, 3, or Right
En_Ct(x) = Enter lane vehicle count for EnLn x
Ex_Ct(x) = Exit lane vehicle count for ExLn x
Pt_Lt = Left turn percentage of total volume
Pt_Rt = Right turn percentage of total volume
Lt(x) = Percentage of total left turn from lane EnLn x
Rt(x) = Percentage of total right turns from lane EnLn x
TRN_Dest(x) = Destination of a vehicle instance after being processed by the TURNBAY Block
TRN_DestTime = User Input Minimum time to downstream destination block
Pr_Lt = Probability vehicle enters left turn bay
Pr_Rt = Probability vehicle enters right turn bay
TVP(x) = Percent of total approach traffic to enter model by current time t on lane x.
D_EnLn_ExLn = Delay Experienced from Switching from EnLn to ExLn
S_PR = Realization from U(0,1)

Vehicle Object Variables - VEH(VUID, variable)
VUID = Vehicle instance unique identifier
T_Time = Travel time from the current block to the downstream destination block

Model Variables
t = Current Simulation time Clock

Lane Specific Turn Movement Probabilities

If all approach vehicles were considered to have the same turning probability, regardless of lane, then the user input turning movement percentages could be used directly as the turning probabilities, without the need for additional calculations. Although, the TURNBAY block allows for additional calibration by allowing the user to specify overall approach level turning movement percentages and the turn movement percentage from each upstream lane. The complications in determining a vehicles likelihood of turning in the TURNBAY logic are due to the allowance of upstream, lane specific turning movement probabilities, and the inherent uncertainly in the volume distribution over these lanes.

While in a non-congested system reasonable aprori estimates of overall approach volumes may be available, the likelihood of reasonable estimations of the volume distribution over the upstream approach lanes is poor. 0 demonstrates how the lane specific probabilities of a vehicle turning left changes for approaches with the same overall upstream volumes, overall left turn percentage, and desired turn movement percentages from each upstream lane but with the traffic distributed differently over the upstream lanes. As seen, there can be a significant difference in the lane specific turning probabilities due to the different lane distributions.

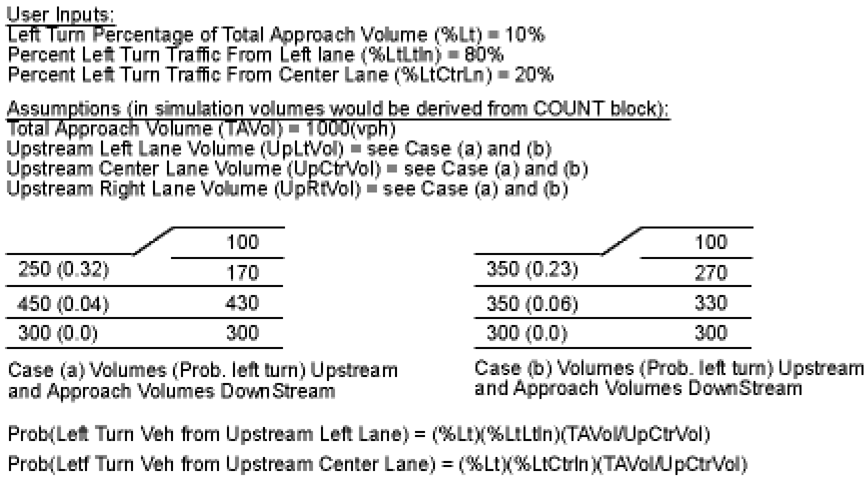


Figure 3.30 Example Determination of Left Turn Turning Probabilities by Lane (volumes are vph)

In the TURNBAY logic a cumulative count is maintained of the number of vehicles that have passed on each lane from the beginning of the simulation until the current time. These counts are converted into hourly volumes and used to determine lane specific left turn probabilities according to the methodology in 0. If a user does not set total turn movement percentages from each upstream lane it is assumed that all left turning vehicles are to be taken from the leftmost lane and all right-turning vehicles are to be taken from the rightmost lane.

LANEADD Block

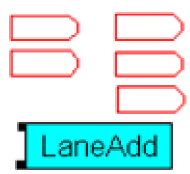


Figure 3.31 Open-TS3 LANEADD Block Model Icon

Overview

The LANEADD block is used to add an additional lane to a link. Where multiple additional lanes are required, an additional LANEADD block is required for each additional lane. A LANEADD block may be used anywhere in a Open-TS3 model, provided there is an upstream vehicle source and a downstream vehicle destination. There are no limitations such as mid-block lane changing.

Vehicles switch into the added lane at the LANEADD block. The user must specify the percentage of total traffic on all upstream lanes that is to switch to the new lane and that percentage is utilized as the probability of any individual vehicle switching. Currently, vehicle weaving resulting from a vehicle switching lanes is not directly modeled. It is assumed that if a vehicle desires to change to the new lane it may, and that it is not delayed as a result of a weaving maneuver. Although, minimum headways are maintained, thus if two vehicles attempt to switch to the new lane, over a time period less than the minimum headway, the later vehicle will be delayed a sufficient time to provide a minimum headway between the vehicles.

As currently designed the LANEADD block utilizes a relatively simply logic in which all vehicles have the same likelihood of switching lanes and no delay is incurred as a result of the lane change. It should be noted that if a user desires a more robust lane add model that a TURNBAY block may be utilized. This is accomplished by assigning the additional lane as a turn movement in the user input data and setting the thru lanes and turn lane to a common downstream approach. While weaving is still not directly modeled, the TURNBAY block does offer some additional abilities for the user to fine tune and calibrate the model.

User Input

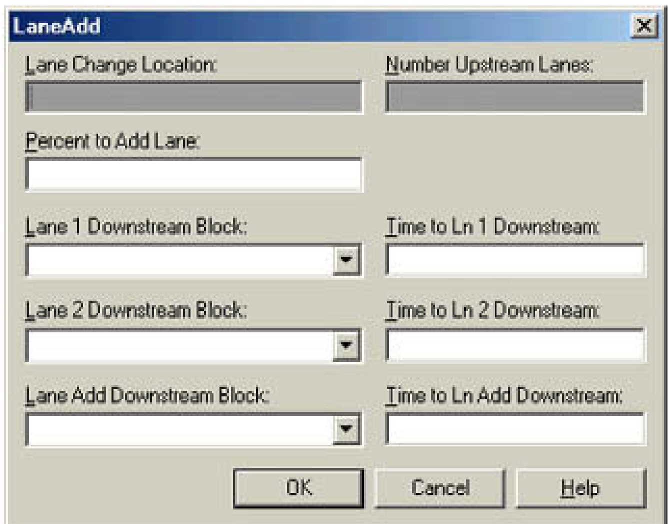


Figure 3.32 LANEADD Block User Input Dialog Box

0 is the LANEADD block input dialogue box. The following describes each input.

Lane Change Location A unique location name for the LANEADD block. This is the name other blocks will use to reference the LANEADD block.

Number Upstream Lanes This is the number of lanes immediately upstream of the LANEADD block.

Percent to Add Lane The percentage of all upstream vehicles that desire to switch to the additional lane. This percentage is utilized as the probability that each vehicle will switch to the new lane.

Lane1 DS Station The downstream STATION block (tier 3) to which vehicles remaining in Lane 1 proceed. The STATION block will be defined in another tier 1 block, see QUEUECHANGE block discussion.

Time to Ln 1 DS Station The minimum travel time to the Lane 1 downstream STATION block.

Lane 2 DS Station The downstream STATION block to which vehicles remaining in Lane 2 proceed.

Time to Ln 2 DS	The minimum travel time to the Lane 2 downstream STATION block.
Lane Add DS Station	The downstream STATION to which vehicles that switched to the additional lane proceed.
Time to Ln Add DS	The minimum travel time to the additional lanes downstream STATION block.

Interfaces to other Objects

A LANEADD block must be associated with an upstream vehicle source block, a downstream block that accepts vehicles, model wide variables and downstream STATION blocks. The travel time listed in the upstream block should be the travel time from the upstream block to the merge point, i.e. the location of the LANEADD block. Likewise, the travel time in the LANEADD block is the minimum travel time from the merge point to the downstream destination.

Similar to the QUEUECHANGE it is necessary to specifically designate which downstream STATION block with which each exiting lane is to be associated. As discussed in the QUEUECHANGE block the input dialogue contains a pull-down list of all stations defined in the model.

Finally, the LANEADD block interfaces with vehicles and mode wide variables, as all the other tier 1 blocks that have been discussed. The LANEADD block also maintains counts on each lane that may be included in model output. These counts are not required for the LANEADD block logic to function but are included as a tool for verifying model results.

LANEADD Logic Algorithm

This section contains the LANEADD logic modeled in 0 in algorithmic form. This logic acts to randomly assign vehicles from the upstream lanes to the addition lanes, according to user input probabilities.

LANEADD Algorithm

STEP 0) LAD(LUID, Pr_Switch) = User input Prob. of switch to additional lane, ∀ LUID
LAD(LUID, LAD_Dest(ExLn)) = User input minimum travel time to downstream destination, ∀ LUID, ∀ ExLn

Step 0 is the initialization step which occurs at the start of the simulation, t=0. At initialization the user input probability of a vehicle switching to the new lane and the travel times to the downstream destinations are read.

Step 1 through Step 6 give the algorithm by which each LANEADD block adds a new lane to a link and distributes vehicles to that lane.

STEP 1) VEH(VUID) Enters LAD(LUID, EnLn) at time t

In Step 1 a vehicle VEH(VUID) enters the block LANEADD(LUID) on Lane EnLn.

STEP 2) Set LAD(LUID, S_PR) = U(0,1) Realization
STEP 3) If LAD(LUID, S_PR) <= LAD(LUID, Pr_Switch) then

LAD(LUID, ExLn) = LAD(LUID, AddLn)
Else
LAD(LUID, ExLn) = LAD(LUID, EnLn)

In Steps 2 and 3 vehicles are randomly assigned to the new lane. Vehicle VEH(VUID) is assigned a realization from a U(0,1). If that probability is less than or equal to the user input probability, VEH(VUID) switches to the new lane, otherwise it remains in its current lane.

STEP 4) Set VEH(VUID, T_Time), *See section 3.3.1.5 SET TRAVEL TIME*
STEP 5) Wait VEH(VUID, T_TIME)
STEP 6) Release VEH(VUID) to LAD(LUID, LAD_Dest(Exln))

In Steps 4, 5 and 6 the travel time to the downstream destination STATION is set and served, and the VEH(VUID) is released to the downstream STATION. Step 4 insures that minimum headways are maintained.

Where:

LANEADD Block (object) Variables - LAD(LUID, variable)

LUID =	LANEADD block instance unique identifier
EnLn =	Lane in which vehicle enters, where EnLn = 1 or 2
ExLn =	Lane in which vehicle exits, where ExLn = 1, 2, or Add
Pr_Switch =	User input probability that a vehicle switches to the new lane
LAD_Dest =	Destination of vehicle instance after being processed by LANEADD block
S_PR =	Realization from U(0,1)

Vehicle Object Variables - VEH(VUID, variable)

VUID =	Vehicle instance unique identifier

Model Variables

t =	Current Simulation Clock Time

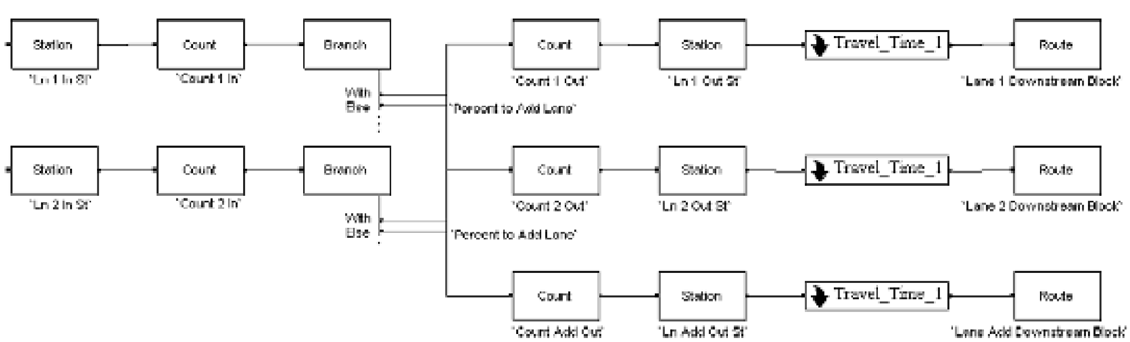


Figure 3.33 LANEADD Block ARENA Basic Block Logic

LANEDROP Block

Figure 3.34 Open-TS3 LANEDROP Block Model Icon

Overview

The LANEDROP block is used to drop a lane from a link. Where it is desired to drop multiple lanes a LANEDROP block is required for each lane to be dropped. As with a LANEADD block a LANEDROP block may be used anywhere in a Open-TS3 model provided there is an upstream vehicle source and a downstream vehicle destination.

Again, similar to the LANEADD block, vehicle switching is taken to occur at the location of the LANEDROP block. Vehicles in the lane to be dropped may switch to any of the remaining lanes, according to user input probabilities. Thus a user must be careful to insure that the sum of probabilities of a vehicle switching to the remaining lanes equals 1. For example if a three lane link is to drop lane 3 (curb side) then the user must assign the Lane 3 traffic to the remaining lanes, 1 and 2. The amount of traffic assigned to each lane, for example 40% to lane 1 and 60% to lane 2, is an input that must be calibrated by the user.

As with the LANEADD block the LANEDROP block utilizes a relatively simple lane switching model where vehicle weaving is not directly modeled. It is again assumed that a vehicle will be able to switch lanes and that it is not delayed as a result of a weaving maneuver. Only the need to maintain minimum headways will cause a vehicle to be delayed. Thus, similar again to the LANEADD block, if two vehicles attempt to switch to the same lane, over a time period less than the minimum headway, the later vehicle will be delayed a sufficient time to provide a minimum headway between the vehicles.

User Input

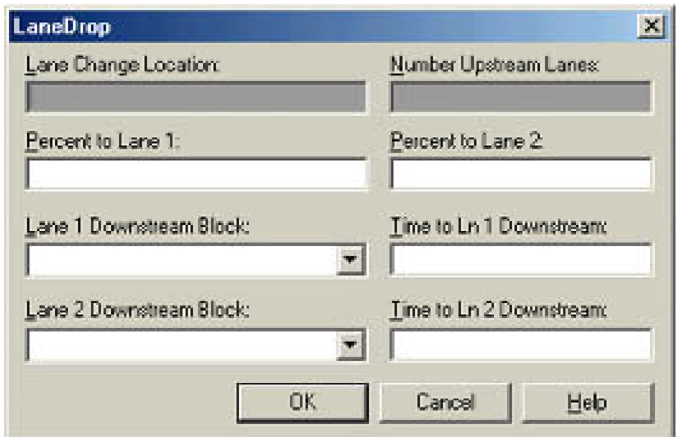


Figure 3.35 LANEDROP User Input Dialogue Box

0 is the LANEBLOCK block input dialogue box. The following describes each input.

Lane Change Location A unique location name for the LANEDROP block. This is the name other blocks will use to reference the LANEDROP block.

Number Upstream Lanes This is the number of lanes immediately upstream of the LANEADD block.

Percent to Lane 1 The percentage of vehicles that will switch from the lane being dropped to Lane 1. This percentage is the probability utilized for vehicles switching from the dropped lane to Lane 1. The sum *Percent to Lane 1* and *Percent to Lane 2* must equal 100.

Percent to Lane 2 The percentage of vehicles that will switch from the lane being dropped to lane 2. This percentage is the probability utilized for vehicles switching from the dropped lane to Lane 2. The sum *Percent to Lane 1* and *Percent to Lane 2* must equal 100.

Lane1 DS Station The downstream STATION block (tier 3) to which vehicles in Lane 1 proceed. The STATION block will be defined in another tier 1 block, see QUEUECHANGE block discussion.

Time to Ln 1 DS Station The minimum travel time to the Lane 1 downstream STATION block.

Lane 2 DS Station The downstream STATION block to which vehicles in Lane 2 proceed.

Time to Ln 2 DS The minimum travel time to the Lane 2 downstream STATION block.

Interfaces to other Objects

The interfaces to other objects for the LANEDROP block are the same as that of the LANEADD block. If desired the user is referred to section 0 in the LANEADD section to review these interfaces.

LANEDROP Logic Algorithm

The section contains the LANEDROP logic modeled in 0 in algorithmic form. The LANEDROP logic acts to randomly assign vehicles according to user input probabilities from the lane to be dropped to the remaining lanes.

LANEDROP Algorithm

STEP 0) LDR(DUID, Pr_Lane1) = User input Prob. of Switch to Lane 1, ∀ DUID
LDR(DUID, Pr_Lane2) = User input Prob. of Switch to Lane 2, ∀ DUID
LDR(DUID, LDR_Dest(ExLn)) = User input minimum travel time to downstream destination, ∀ DUID, ∀ ExLn

Step 0 is the initialization step which occurs at the start of the simulation, t=0. At initialization the user input probabilities of a vehicle on the lane to be dropped switching to lane 1 or lane 2 are read. The sum of these probabilities must be one. Also, the travel times to the downstream destinations are read.

Step 1 through Step 8 give the algorithm by which each LANEDROP block drops a lane and distributes vehicles to the remaining lanes.

STEP 1) VEH(VUID) Enters LDR(DUID, EnLn) at time t
STEP 2) Set LDR(DUID, S_PR) = U(0,1) Realization

In step 1 a vehicle VEH(VUID) enters the block LANEADD(LUID) on Lane EnLn and a U(0,1) realization is recorded for use in the alogorithm.

STEP 3) If LDR(DUID, EnLn = 1) Then LDR(DUID, ExLn) = 1
STEP 4) If LDR(DUID, EnLn = 2) Then LDR(DUID, ExLn) = 2
STEP 5) If (LDR(DUID, EnLn = Drop) And (LDR(DUID, S_PR) <= LDR(DUID, Pr_Lane1))) then
LDR(DUID, ExLn) = 1
Else
LDR(DUID, ExLn) = 2

In Steps 3 through 5 the exit lane is determined for vehicle VEH(VUID). If the vehicle entered on lane one or two it will exit on that same lane. If a vehicle entered on the lane to be dropped it will switch to either lane one or two, according to the user input probabilities.

STEP 6) Set VEH(VUID, T_Time), *See section 3.3.1.5 SET TRAVEL TIME*
STEP 7) Wait VEH(VUID, T_TIME)

STEP 8) Release VEH(VUID) to LDR(LUID, LDR_Dest(Exln))

In Steps 6, 7 and 8 the travel time to the downstream destination STATION is set and served, and the VEH(VUID) is released to the downstream STATION.

Where:

LANEDROP Block (object) Variables - LAD(LUID, variable)

DUID =	LANEDROP block instance unique identifier
EnLn =	Lane in which vehicle enters, where EnLn = 1 or 2
ExLn =	Lane in which vehicle exits, where ExLn = 1, 2, or Add
Pr_Lane1 =	User input probability that a vehicle from the dropping lane switches Lane 1
Pr_Lane2 =	User input probability that a vehicle from the dropping lane switches Lane 2
LDR_Dest =	Destination of vehicle instance after being processed by LANEDROP block
S_PR =	Realization from U(0,1)

Vehicle Object Variables - VEH(VUID, variable)

VUID =	Vehicle instance unique identifier

Model Variables

t =	Current Simulation Clock Time

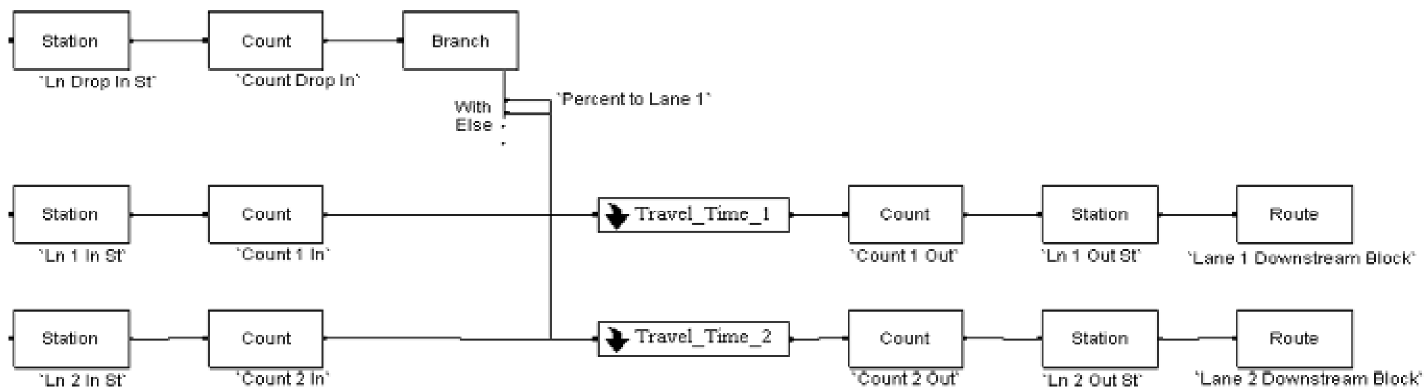


Figure 3.36 LANEDROP Block ARENA Basic Block Logic

CONCLUSION

This chapter has been dedicated to the description of Open-TS3 modeling approach. It was seen that Open-TS3 provides an intuitive and flexible simulation modeling architecture through the use of a hierarchical object-oriented simulation approach. A user may achieve any desired level of understanding of the inner working of the simulation model. One may quickly develop a simulation by using Open-TS3 blocks (ENTER, EXIT, TURNBAY, etc.) or one may develop specially tailored blocks, utilizing the core SIMAN modeling constructs. Importantly, one may also access the spectrum between these two diverging simulation options, developing one's own blocks from basic constructs while having the ability to utilize Open-TS3, or other developed, modeling blocks in a ready and simple manner.

In the next chapter it will be seen that the Open-TS3 is reliable as well as flexible. Validation against another simulation and against real-world data will lead to an acceptable level of confidence in Open-TS3's operation. Finally, in chapter six the Open-TS3's flexibility will be further demonstrated by the implementation of adaptive control into Open-TS3.

CHAPTER 4 OPEN-TS3 MODEL VERIFICATION AND VALIDATION

INTRODUCTION

In chapter 3 Open-TS3 simulation modeling approach was detailed. Two major tasks in the development of a simulation not discussed in chapter 3 where verification and validation. This chapter briefly discusses the verification process and then presents a rather detailed validation effort. This validation effort will be seen to build user confidence in the modeling approach reliability.

VERIFICATION

Verification refers to the process by which the simulation (computer code, file I/O, etc.) is checked to ensure that the model logic has been correctly implemented and is executing as intended. Model verification has been conducted throughout the entire Open-TS3 development and validation process. It is not extensively discussed in this report except to state that repeated and in-depth verification was conducted. Each individual model block (and block sub-sections) was tested to guarantee that operation was as intended. Also, numerous small models, each consisting of only a few block elements, were created to test logic implementation.

VALIDATION

Model validation tests how well the model logic reflects the real world. Thus, the ideal validation study incorporates real world data. Unfortunately, the collection of real world data can be a time consuming and costly process, and capturing all desired volume / geometry combinations and field measurements of delay, emissions, etc. in the real world can be difficult and inaccurate. Thus, a validation procedure that overcomes the time, cost, data collection, and measurement problems of real world studies is often sought. Typically this validation relies on comparison of the model to be validated against other models.

The resources available for this initial Open-TS3 validation effort excluded the possibility of a solely real world data based study. Instead, a combination of the two preceding validation approaches was undertaken. First a comparison of Open-TS3 against CORSIM, arguably the most widely accepted and utilized simulation model, is presented. While not ideal, this validation allowed for a clear control of the experimental conditions and a broad stroke of volume and signal timing conditions. One clear drawback to this validation approach is that CORSIM logical errors are now introduced into the validation process. Thus the Open-TS3 validation effort is further supplemented with a comparison of Open-TS3 performance to data collected from a network in downtown Chicago, Illinois.

Open-TS3 vs. CORSIM Validation

Two different geometric scenarios were studied for initial model validation, an isolated intersection (Figure 3.4) and a three intersection arterial (0). Characteristics similar to both validation scenarios include the following.

Phasing - Three phase signal timing (leading E-W lefts) is utilized on all intersections. Phase, cycle lengths, and offsets are the signal parameters that are adjusted in the validation process.

Turn Movements – All left turn movements are protected and right turns on red are not allowed. Vehicles are not allowed to turn left from the Northbound or Southbound approaches. Demand volumes are varied in the different validation scenarios.

Geometry - All left and right turn movements utilize turn bays. In all scenarios the E-W approaches contain single lane right and left turn bays. In the isolated intersection scenarios the E-W approaches include three thru lanes while in the arterial scenarios there are two thru lanes. In all scenarios the N-S approaches are comprised of a single right turn bay and a single through lane. No lanes in any of the scenarios are shared (i.e. contain both right and thru or left and thru vehicles.)

Start-Up Lost Time – Average Start-up lost time is set to 2.4 seconds in both CORSIM and Open-TS3. Ideally, start-up lost time is calibrated through field measurements. Although, since this validation is a comparison of two simulation models a reasonable estimate (Webster's, HCM 2000) of start-up lost time was selected and utilized in both models.

Departure Headways – The average intersection departure headway is set to 2.0 seconds for all movements, in both simulations.

Queue Changing – As seen in section 3.3.5 the probability of a vehicle switching to a shorter adjacent queue may be set in Open-TS3. The calibration of the queue changing probabilities was completed before the validation studies. Initial CORSIM runs where made to gauge CORSIM's lane change frequency. It was observed that CORSIM very aggressively switches vehicles to the lane with the most unoccupied space. Lane balance (i.e. similar length queues in lanes for the like movements) was consistently achieved in CORSIM. This was also verified in Wong (1990). Thus, vehicle queue changing probabilities in Open-TS3 were set to achieve a high frequency of balanced queue lengths. The probability of switching for queue differences of 0, 1, 2, 3, and 4+ were set to 0, 0.65, 0.75, 0.85, and 0.95.

Open-TS3 vs. CORSIM Validation – Isolated Intersection Overview

This section presents Open-TS3 and CORSIM simulation results for an isolated intersection. It will be seen that CORSIM and Open-TS3 were found to exhibit similar values and trends for several measures of effectiveness (volumes processed, average vehicle delay, average queues and average speed) in non-congested situations. In over-congested situations both models also highlighted operational problems in the intersection performance although the absolute difference between the measures of effectiveness values simulated by the two models could be significant.

Open-TS3 vs. CORSIM Validation – Isolated Intersection Cycle Length and Demand Volumes

Comparisons are made under low to over saturated traffic conditions, for different cycle lengths. Fifteen different volume / cycle length scenarios are tested. In addition five replicate runs are performed for each scenario, for a total of 75 runs of each simulation model. Table 4.1 shows the volume and cycle length scenarios tested along with the approximate theoretical capacity for each movement.

Table 4.1 Cycle Length / Demand Volume Scenarios

Cycle Length	Phase Lengths			Demand Volumes					Theoretical Capacity				
	EB Left	EB Thru/Rt	NB Thru/Rt	EB Left	EB Thru	EB Right	NB Thru	NB Right	EB Left	EB Thru	EB Right	NB Thru	NB Right
60	10	30	20	73	515	73	120	120	168	2304	768	468	468
60	10	30	20	145	1030	145	240	240	168	2304	768	468	468
60	10	30	20	218	1545	218	360	360	168	2304	768	468	468
60	10	30	20	290	2060	290	480	480	168	2304	768	468	468
60	10	30	20	363	2575	363	600	600	168	2304	768	468	468
90	15	45	30	73	515	73	120	120	212	2436	812	512	512
90	15	45	30	145	1030	145	240	240	212	2436	812	512	512
90	15	45	30	218	1545	218	360	360	212	2436	812	512	512
90	15	45	30	290	2060	290	480	480	212	2436	812	512	512
90	15	45	30	363	2575	363	600	600	212	2436	812	512	512
120	20	60	45	73	515	73	120	120	234	2502	834	534	534
120	20	60	45	145	1030	145	240	240	234	2502	834	534	534
120	20	60	45	218	1545	218	360	360	234	2502	834	534	534
120	20	60	45	290	2060	290	480	480	234	2502	834	534	534
120	20	60	45	363	2575	363	600	600	234	2502	834	534	534

$$Theoretical\ Capacity = \left(\frac{Green\ Time\ Per\ Cycle - Lost\ Time}{Veh\ Departure\ Headway} \right) \times \left(\frac{3600\ \text{sec}/\,hr}{Cycle\ Length} \right)$$

In sections 0 through 0 several measures of effectiveness are considered; volume processed (vehicles), delay (sec/veh), average queue (vehs), maximum queue (vehs) and average speeds (mph).

Open-TS3 vs. CORSIM Validation - Isolated Intersections Volume

The first question addressed in the validation of Open-TS3 is whether under similar flow demands Open-TS3 and CORSIM process similar volumes. 0 through 0 present the average volumes processed by each simulation for the east bound and north bound movements.

East Bound Thru and Right Turn Movements - Both models consistently process similar volumes for all under capacity scenarios, with the difference between average volumes not exceeding three percent. In the overcapacity scenarios (2575 vph thru demand) CORSIM tends to process approximately 10% fewer vehicles. CORSIM consistently processes less than theoretical capacity and Open-TS3 slightly more than theoretical capacity.

In addition, except for the overcapacity scenarios, both simulations exhibit similar standard deviations. Only in the overcapacity scenarios does CORSIM consistently exhibit a higher standard deviation on the thru movement. These same similarities are also seen on the westbound thru and right turn movements.

North Bound Thru and Right Turn Movements - On the northbound thru and right turn movements for the low to moderate demand scenarios both models process similar numbers of vehicles. For the near to exceeding capacity scenarios it is seen that Open-TS3 has a tendency to process more vehicles than CORSIM. For example in the 480 veh demand / 90 sec cycle and 600 veh demand / 90 sec cycle scenarios Open-TS3 processes 480 and 532 vehicles and CORSIM 427 and 428 vehicles, for differences of 11% and 20%, respectively. The 480 vehicle demand scenario is slightly below theoretical capacity and the 600 vehicle demand scenario exceeds theoretical capacity by nearly 15%. Clearly the overcapacity scenario is an area of some discrepancy between the models and Open-TS3 should be further fine-tuned against real world data in future model development.

East Bound Left Turn Movement - For the low and moderate east bound left turn demand scenarios similar volumes are again processed by both models. Once the demand approaches or exceeds capacity CORSIM consistently processes more vehicles than Open-TS3. For example, in the 60 second cycle / 363 left turn vehicle demand scenario CORSIM processes 252 vehicles while Open-TS3 processes 197 vehicles, a difference of 22%. Both models process more than theoretical capacity (168 veh), with the CORSIM model processing substantially more vehicles. Observations of the CORSIM model show that on some cycles CORSIM processes as many as 5 left turn vehicles. An average headway of 1.6 sec/veh, excluding any lost time, 1.1sec headway if the 2.4 second lost time is taken into account. Open-TS3 model is not set to be this aggressive with the left turn movement. Again, overcapacity demands are an area where future studies utilizing real world data should be performed. Until these studies are completed overcapacity left-turn situation should be handled cautiously within both Open-TS3 and CORSIM.

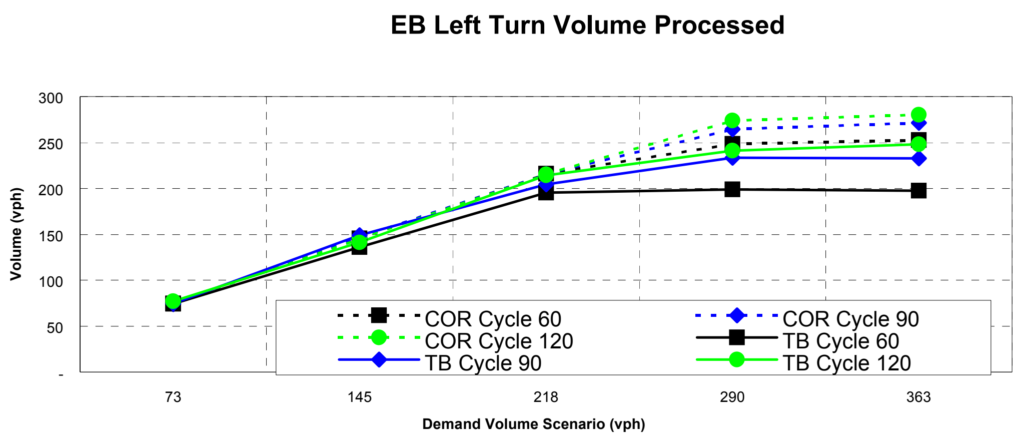


Figure 4.1 East Bound Left Turn Volumes Processed by CORSIM (COR) and Open-TS3 (TB) Under Different Demand Volume / Cycle Length Scenarios

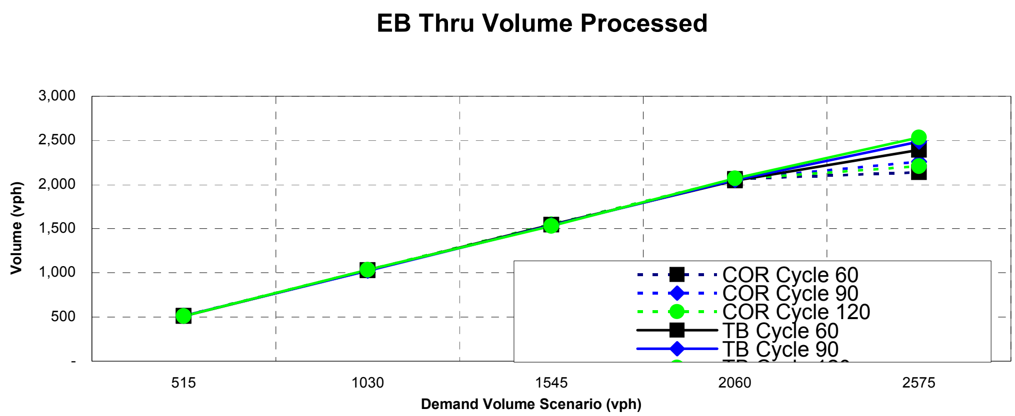


Figure 4.2 East Bound Thru Volumes Processed by CORSIM (COR) and Open-TS3 (TB) Under Different Demand Volume / Cycle Length Scenarios

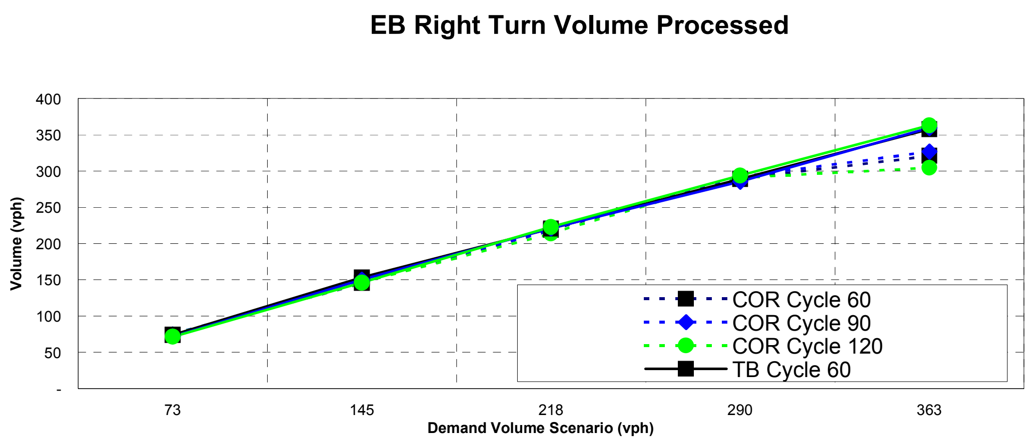


Figure 4.3 East Bound Right Turn Volumes Processed by CORSIM (COR) and Open-TS3 (TB) Under Different Demand Volume / Cycle Length Scenarios

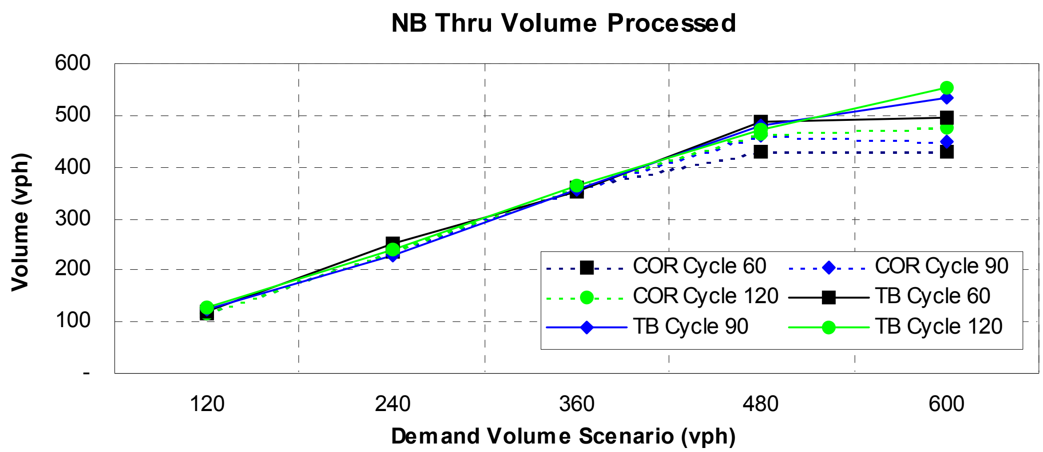


Figure 4.4 North Bound Thru Volumes Processed by CORSIM (COR) and Open-TS3 (TB) Under Different Demand Volume / Cycle Length Scenarios

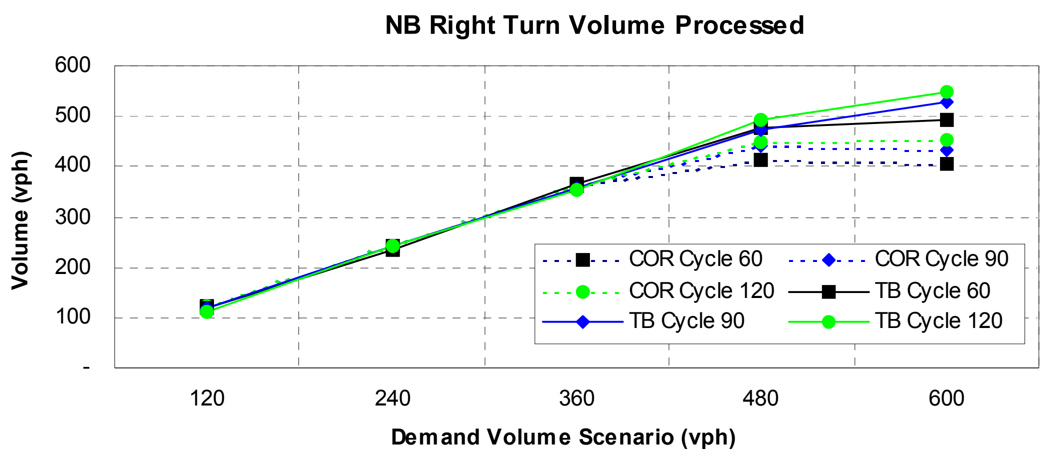


Figure 4.5 North Bound Right Turn Volumes Processed by CORSIM (COR) and Open-TS3 (TB) Under Different Demand Volume / Cycle Length Scenarios

Open-TS3 vs. CORSIM Validation - Isolated Intersections Delays

A commonly desired measure of effectiveness from any traffic simulation is average vehicle delay, defined to be the time for a vehicle to traverse an intersection less the desired (free flow) time. 0 through 0 present the average delays for the eastbound and northbound movements.

East Bound and North Bound Thru Movements - As seen in 0 and 0 the eastbound and northbound thru delays for all but the overcapacity scenarios are similar in both simulations. The difference in Open-TS3 and CORSIM simulated delays does not exceed a few seconds. In the overcapacity scenarios both simulations capture high delays although the values are significantly different. The exceedingly higher, Open-TS3 northbound delays, most likely result from the Open-TS3 vertical queuing model, which fails to limit the queue length by the approach link length. In CORSIM when the queue length reaches the link length vehicles may no longer enter the link, essentially lessening demand and queues, and thus delay. As for the near capacity scenarios additional study is necessary to determine the most realistic simulation of the real world. Although, even with the significant differences in the overcapacity and near capacity scenarios both simulations capture high delays and long queue lengths, which should be a warning to any engineer utilizing either simulation.

Eastbound Left Turn Movement Delays - The simulated left turn delays follow much the same pattern as the left turn volumes. In under-saturated conditions the two simulations provide similar delays while in near capacity and overcapacity scenarios the delays are quite different. As discussed in the volume section the CORSIM left turn model allows a higher maximum number of vehicles to be processed per cycle than the Open-TS3 model. Thus, vehicles begin to accrue higher delays at a lower demand in Open-TS3 than in CORSIM. Adjustments could be made to the Open-TS3 model to allow for a more accurate reflection of the CORSIM model at near and overcapacity left turn demand although additional calibration with real world traffic demand and resulting measures would provide superior Open-TS3 model calibration. At this time in the research it is realized that caution should be exercised with both the CORSIM and Open-TS3 model in overcapacity situations.

Eastbound and Northbound Right Turn Delays - The simulated right turn delays were also an area of some difference between the models. Typically the Open-TS3 simulated a lower average right turn delay than CORSIM. Where the thru demand was under capacity the CORSIM right turn delay was typically around five seconds higher than the Open-TS3 simulated delay. This difference leads to a recommend calibration in future Open-TS3 runs. In the current Open-TS3 simulation runs no delay was assigned to vehicles changing from the through lanes to the right turn lanes, although Open-TS3 is set to incorporate such a delay (see section 3.3.6). In future Open-TS3 runs it is recommended that a minimal delay, 5 seconds, should be assigned to this lane change maneuver. Future versions of Open-TS3 will be able to better address this situation through the use of a LINK block, to be discussed in the recommended future improvements to Open-TS3 section.

Where the thru movements are overcapacity and long thru queues exist the difference between the Open-TS3 and CORSIM right turn delays are significant. In the CORSIM model a right turn vehicle may be significantly delayed in the thru vehicle queue before it reaches the right turn bay, this does not occur in Open-TS3 due to the vertical queuing model. As a result Open-TS3 right turning vehicles experience significantly lower lane delays. As already stated the vertical queuing model will be revisited in future additions of Open-TS3 and the analysis of overcapacity situation should be treated cautiously.

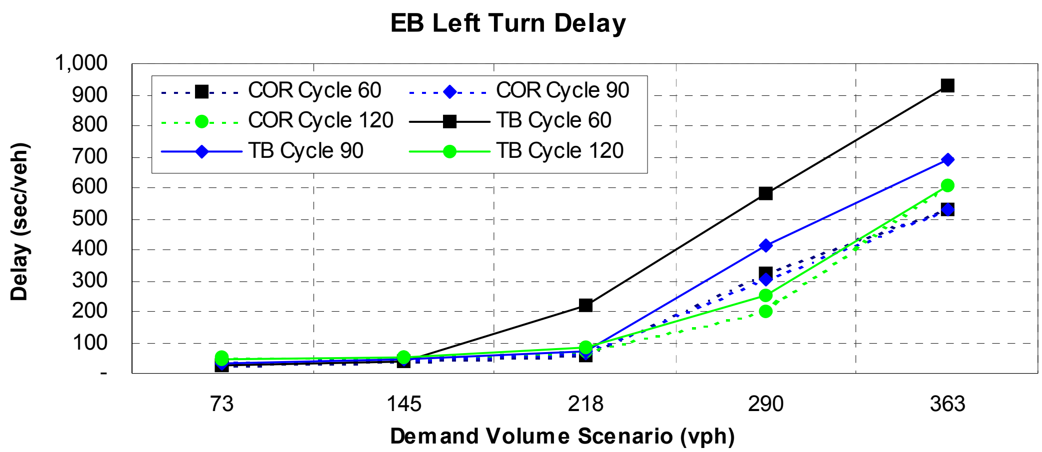


Figure 4.6 Eastbound Left Turn Delay for CORSIM (COR) and Open-TS3 (TB) Under Different Demand Volume / Cycle Length Scenarios

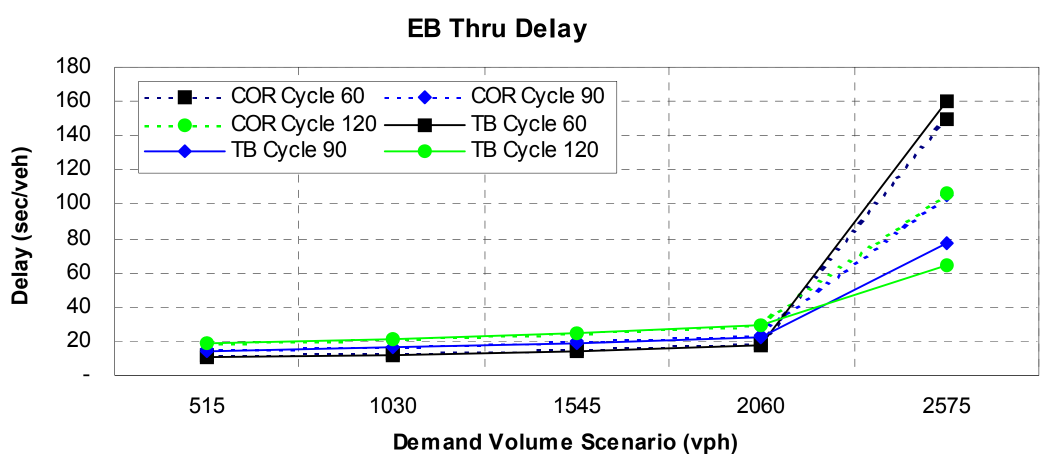


Figure 4.7 Eastbound Thru Delay by CORSIM (COR) and Open-TS3 (TB) Under Different Demand Volume / Cycle Length Scenarios

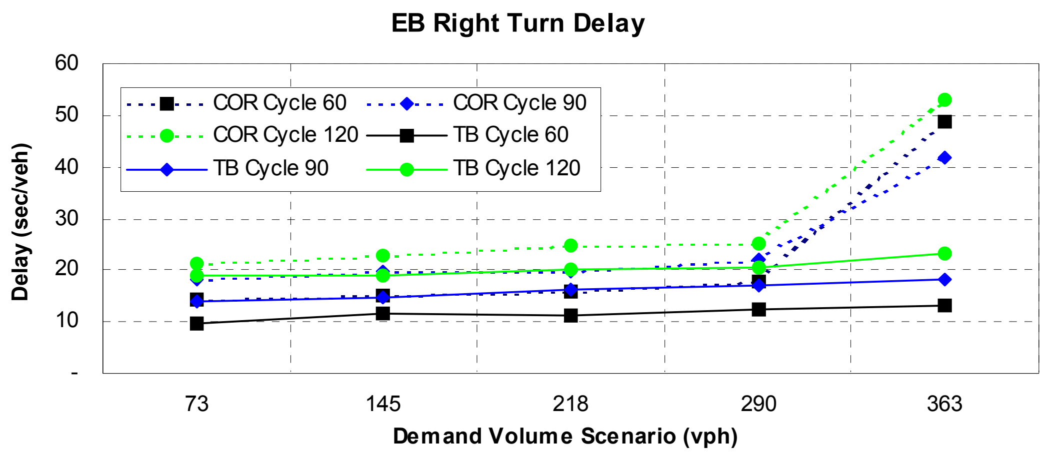


Figure 4.8 Eastbound Right Turn Delay for CORSIM (COR) and Open-TS3 (TB) Under Different Demand Volume / Cycle Length Scenarios

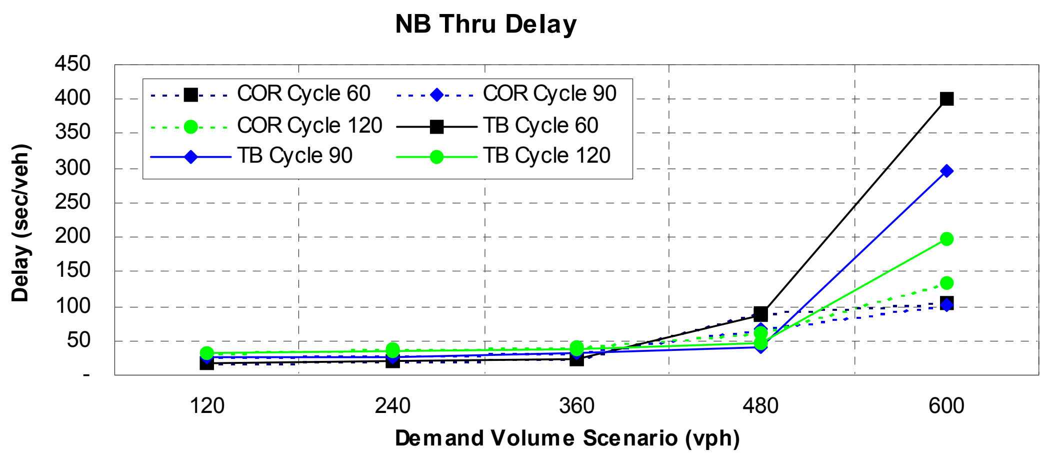


Figure 4.9 Northbound Thru Delay for CORSIM (COR) and Open-TS3 (TB) Under Different Demand Volume / Cycle Length Scenarios

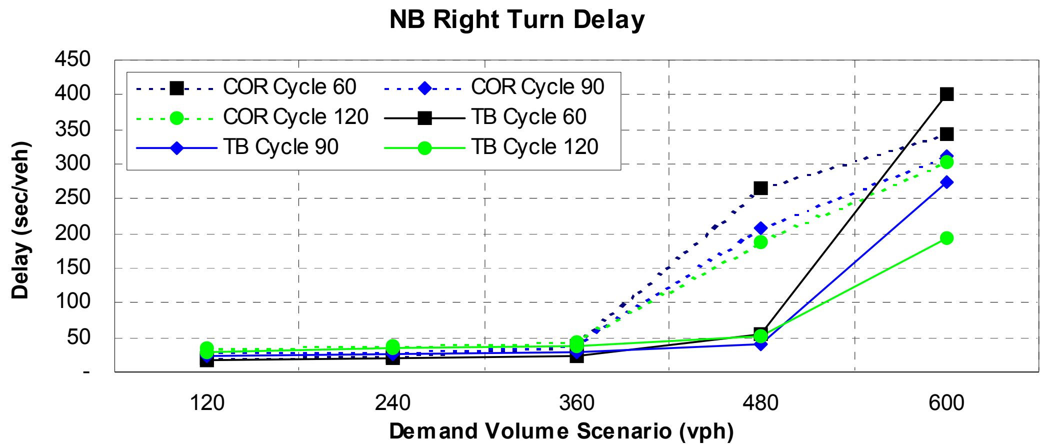


Figure 4.10 Northbound Right Turn Delay for CORSIM (COR) and Open-TS3 (TB) Under Different Demand Volume / Cycle Length Scenarios

Open-TS3 vs. CORSIM Validation – Isolated Intersection, Queues and Speed

Aside from a few general comments the queues and speed discussion will not be as in-depth as the delays and volumes discussions in an effort to avoid repetition and unnecessary additional text.

Queues - In general the queues follow much the same trends as the delays, reasonable agreement in the under-capacity demand scenarios with more significant differences in the overcapacity demand scenarios. In overcapacity scenarios the simulation with the higher delay also tends toward higher queues. This is most notable in the left turn movement where Open-TS3 produces significantly higher delays and queues when capacity is exceeded. Also, it is again noted that a major difference between models exists in that Open-TS3 does not have a theoretical limit to its queue length where CORSIM queue lengths are limited by the link length.

Speed - As with the other discussed measures of effectiveness the simulated speeds demonstrate reasonable agreement. The speed generated by Open-TS3 and CORSIM typically differ by only a few miles per hour, typically less than a one-mile per hour difference in the non-congested scenarios. Unlike the other measures discussed, similar speeds are also found in the overcapacity scenarios. Once a high level of delay is reached, average speeds become very low, less than 5 mph, with additional increases in delay resulting in relatively small changes in speed. Thus, it is only the near capacity scenarios, where one model simulates an approach as slightly overcapacity and the other model as slightly under capacity that any notable difference in simulated speeds may be observed.

Validation – Isolated Intersection, Statistical Analysis

In the comparison of the measures of effectiveness produced by the Open-TS3 and CORSIM testing for statistically significant differences should be considered. One applicable test is the two-sample t test. This test assumes that both populations are normal and the population variances are equal (population members are replicate run MOE values). The first assumption, that populations are normal, may be taken as reasonable based on the following reasoning. Each MOE average is based on five random samples (five replications). Each random sample (replication) is an average based on at least 70 vehicles (typically hundreds). While it is unlikely that the individual vehicle MOEs are normally distributed the replicate run average MOE, may be assumed to be normally distributed in accordance with the central limit theorem. The second assumption, that the population variances are the same, is more difficult to justify. With the given data the most reasonable assessment is that the population variances are at least of the same order, based on the similarity in sample standard deviations, and any differences in variance will not be large enough to significantly affect the power of the test.

Hypothesis testing was completed for all the delay scenarios. Many of the Open-TS3 vs CORSIM measures were shown to be significantly different while many were shown to not be significantly different, with no real pattern to which pairs of measures were and where not significantly different. The problem encountered is that of the difference between statistically significant difference and practically different. For example, consider the 2060 veh demand / 60 second cycle demand scenario where the Open-TS3 and CORSIM average thru delays are 17.05 and 18.30 seconds and standard deviations are 0.38 and 0.70 seconds, respectively. A Two-Sample t test with alpha of .05 will show these two delays to be statistically, significantly different. This is clearly due to the low variances. It must be quickly realized that in traffic simulation (and analysis and data collection) 17.05 and 18.30 second delays are not practically different. A few seconds difference in any traffic analysis should never be taken as a condemnation of a methodology. Comparison and reasonableness of results becomes an area where expert judgment and experience become crucial in identifying problem areas.

Open-TS3 vs. CORSIM Validation – Isolated Intersection, Summary

The Open-TS3 isolated intersection validation study utilizing CORSIM demonstrated that similar absolute values, variability and trends are simulated by the models for several measures of effectiveness (volumes processed, average vehicle delay, average queues and average speed) in less than capacity situations. In overcapacity situations both models also highlighted operational problems in the intersection performance although the absolute difference between the measures of effectiveness values simulated by the two models could be significant.

The thru and right turn movements provided the best agreement between the models with the left turn movements the least agreement. Clearly there is a difference in the left turn capacity modeled by the two simulations. Future improvements to Open-TS3 (and CORSIM) should include study of the modeling of overcapacity situations and at what demand level is capacity reached. Also, additional realism will be incorporated into Open-TS3 with the addition of a horizontal queuing model and direct modeling of vehicle interaction on the link between intersections.

Open-TS3 vs. CORSIM Validation – Arterial

Presented in the following sections are the Open-TS3 and CORSIM simulation results for an arterial. As with the isolated intersection discussion an overview of the arterial geometric, demand and signal parameters is given followed by a presentation of several different measures of effectiveness; volumes, delay, queues, and speed.

For the demand level studied (approximately 85% of capacity) that the two models demonstrate excellent agreement in the captured absolute MOE values and MOE trends as signal parameters changed. This leads to an increased confidence in the use of Open-TS3 for both the study of existing systems and the optimization of future systems.

Open-TS3 vs. CORSIM Validation – Arterial Geometric, Demand and Signal Parameters

As described in section 0 the simulated arterial (0) is a two-lane East-West arterial, consisting of three intersections, with all eastbound and westbound approaches consisting of two through lanes, a left turn bay, and a right turn bay. The distance between intersections (from stop bar to stop bar) is set at 1320 ft, i.e. the distanced traveled at 30mph (the average free flow speed) in 30 seconds.

The arterial validation study is primarily concerned with the operation of intersection approaches where there is a modeled upstream intersection, as other approaches will operate in a manner similar to that of the isolated intersection. Thus in the following discussion only the eastbound and westbound directions are considered. Northbound and southbound operations behave as in the previous isolated intersection discussion, and no new insight is gained by repeating that discussion.

It was seen in the isolated intersection discussion that the simulation results covered a range of cycle lengths and volumes. From this study it was concluded the Open-TS3 operated well in non-congested situations, under all cycle lengths considered. Based on these results a single cycle length and volume demand scenario was selected for the arterial study. A thru volume demand of approximately 85% of capacity was selected (0), as an overcapacity demand would suffer the same vertical queuing and other issues raised in the previous isolated intersection validation study and a demand well below capacity would offer little interaction between intersections. A common cycle of 90 seconds was chosen, although there is no particular advantage or disadvantage to this cycle length, other than it is a cycle length found in many signal systems.

The effect of the offset is the variable of most interest in the arterial study. Thus six different offset scenarios are modeled. Listed as (Intersection 1 offset in seconds, Intersection 2 offset in seconds, Intersection 3 offset in seconds) these six cases are (0,0,0), (0,15,30), (0,30,60), (0,45,0), (0,60,30), and (0,75,60). These offset scenarios essentially step the East to West offset in 15 second intervals (0,15,30,45,60, and 75). Five replicate runs were completed for each offset scenario.

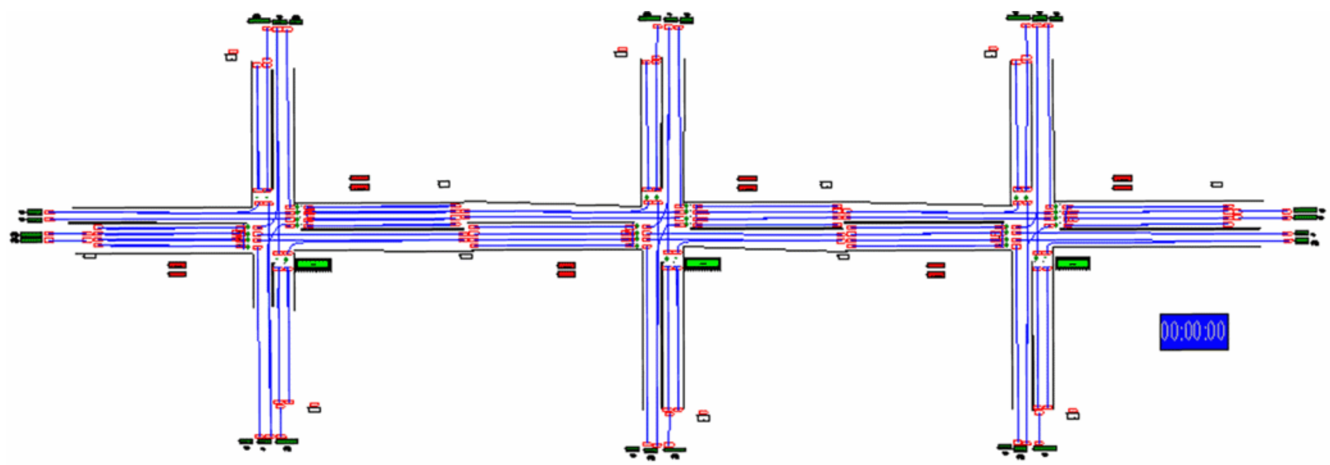


Figure 4.11 Open-TS3 Arterial Simulation Model

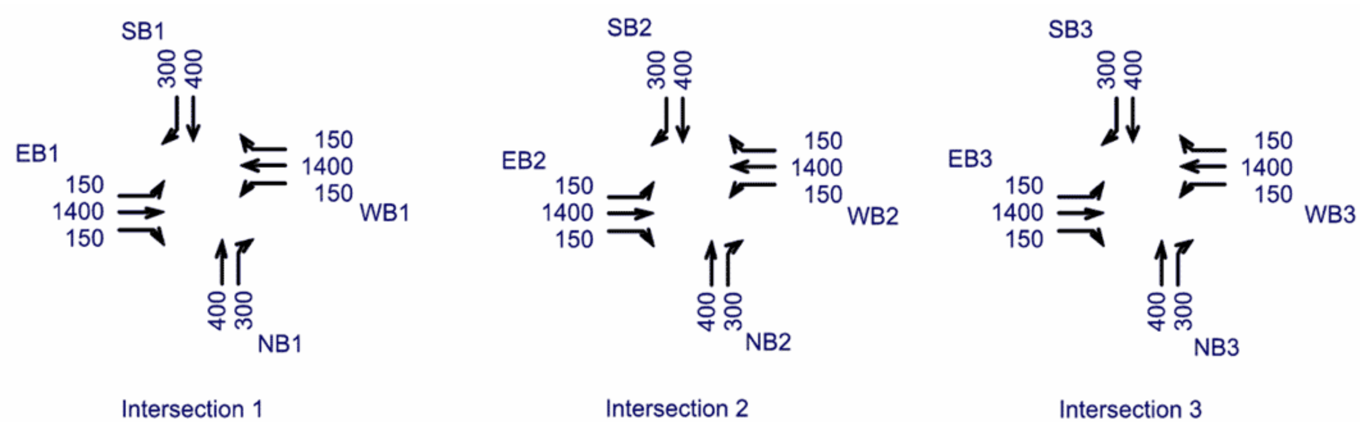


Figure 4.12 Arterial Demand Volumes and Intersection and Approach Names

Open-TS3 vs. CORSIM Validation – Arterial Volumes

In both the eastbound and westbound directions both simulations process similar through, left and right turn volumes. This is as expected as it was seen in the isolated intersection validation that both models were able to successfully process less than theoretical capacity demands. Both models also had similar standard deviations, with the turn movements standard deviation typically 10% or less of the turn volume and the thru movement standard deviations typically 3% or less of the thru volume.

Open-TS3 vs. CORSIM Validation – Arterial Delays

In the arterial operations the affect of vehicle platooning and platoon arrive times on an intersection approach delay can be significant. It is critical that a model accurately reflect the creation of platoon and platoon movements. A model's failure or success in capturing the interaction of vehicles with upstream intersections will directly impact the accurateness of any delays measured.

0 through 0 show the thru, left turn, and right turn delays for the East Bound and West Bound directions, under the different offset scenarios. For all three movements the calculated delays from both models are similar, typically within five or fewer seconds. The EB1 and WB3 approach delays are seen to be nearly the same for all offset scenarios. This is as expected since these are the approaches upon which vehicles enter the arterial, thus there are no platoon or upstream effects captured. The internal approaches, EB2, EB3, WB1, and WB2 delays vary according to the offset scenarios. It is important to note that both simulation models, Open-TS3 and CORSIM, exhibit similar trends in delays over the different offset scenarios. Thus, both models determine similar optimal delay (minimal delay) offsets. As a major purpose in the development of Open-TS3 is its future use as an optimization tool it is important that reliability is gained in both the absolute delay values and delay values relative to other scenarios (offset solutions).

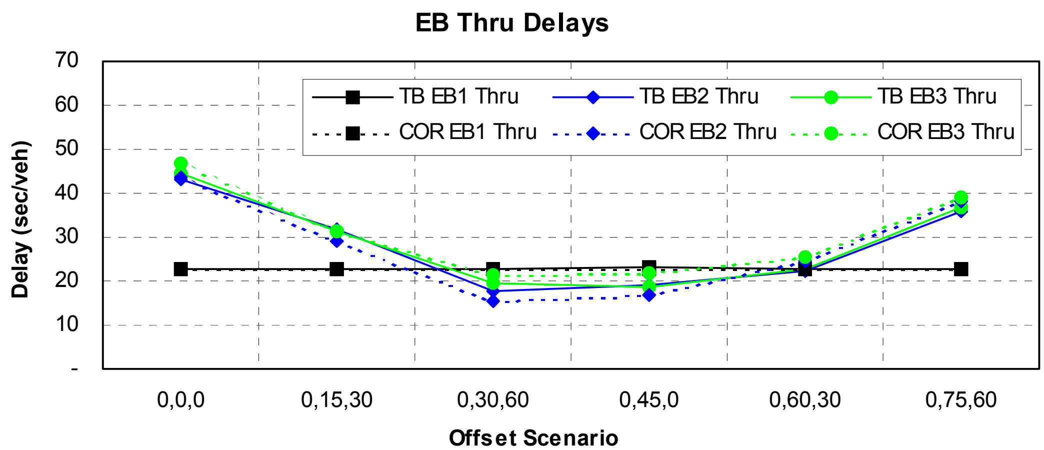


Figure 4.13 Eastbound Thru Arterial Delays for CORSIM (COR) and Open-TS3 (TB) under different offset scenarios.

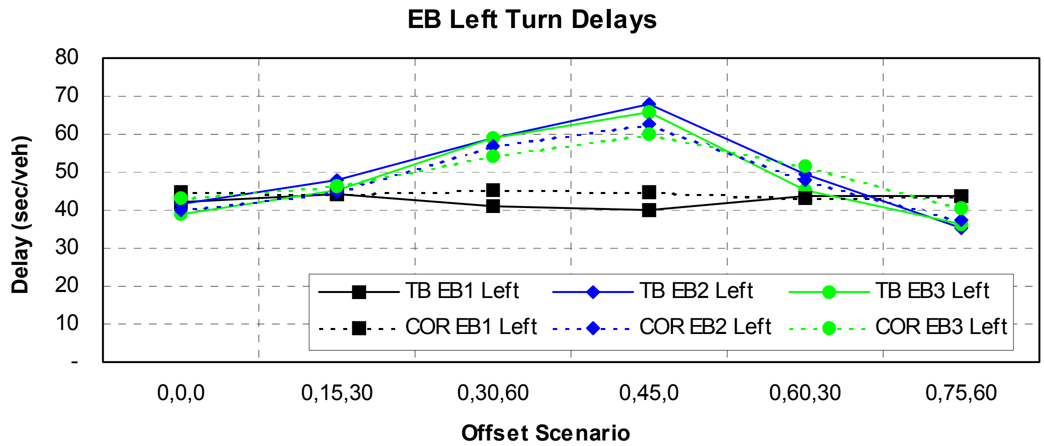


Figure 4.14 Eastbound Left Turn Arterial Delays for CORSIM (COR) and Open-TS3 (TB) under different offset scenarios.

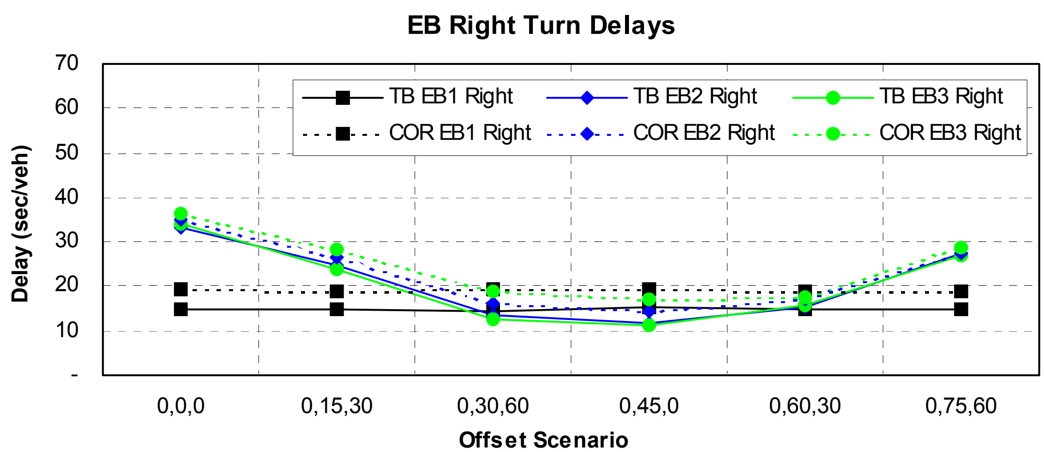


Figure 4.15 Eastbound Right Turn Arterial Delays for CORSIM (COR) and Open-TS3 (TB) under different offset scenarios.

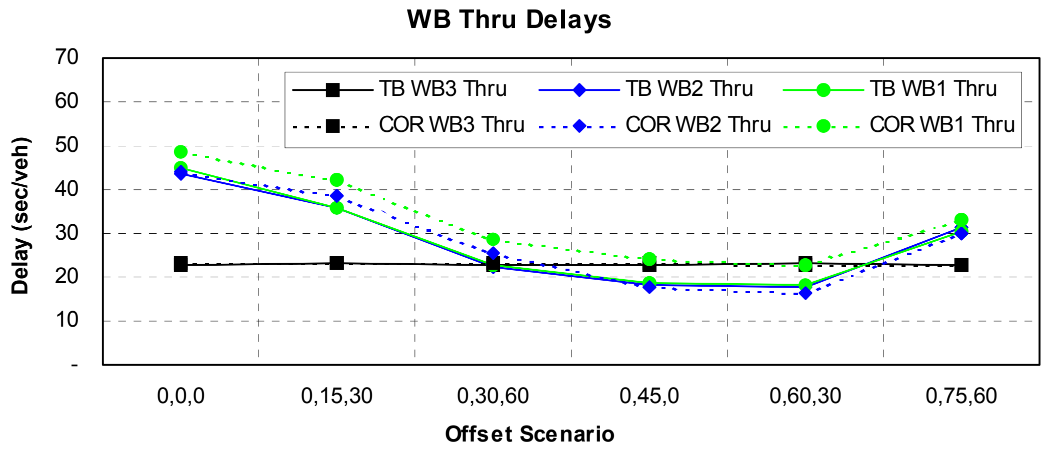


Figure 4.16 Westbound Thru Arterial Delays for CORSIM (COR) and Open-TS3 (TB) under different offset scenarios.

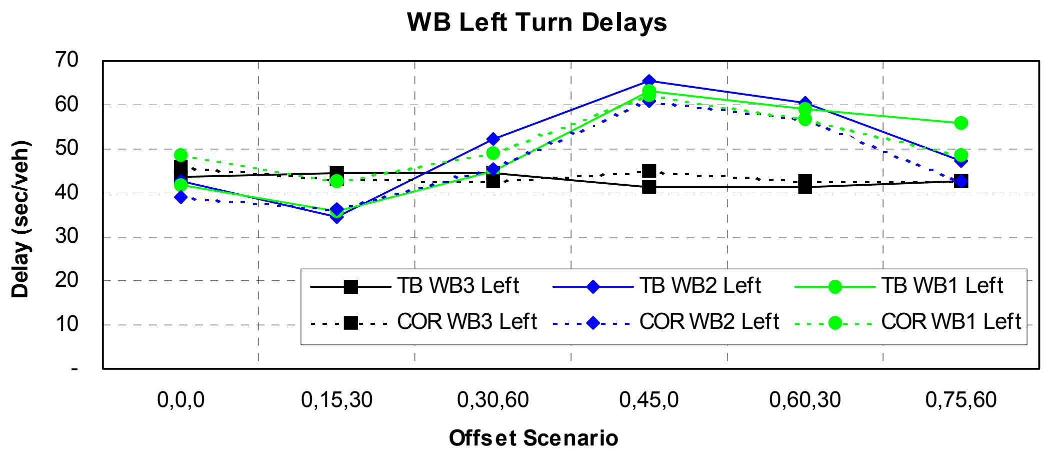


Figure 4.17 Westbound Left Turn Arterial Delays for CORSIM (COR) and Open-TS3 (TB) under different offset scenarios.

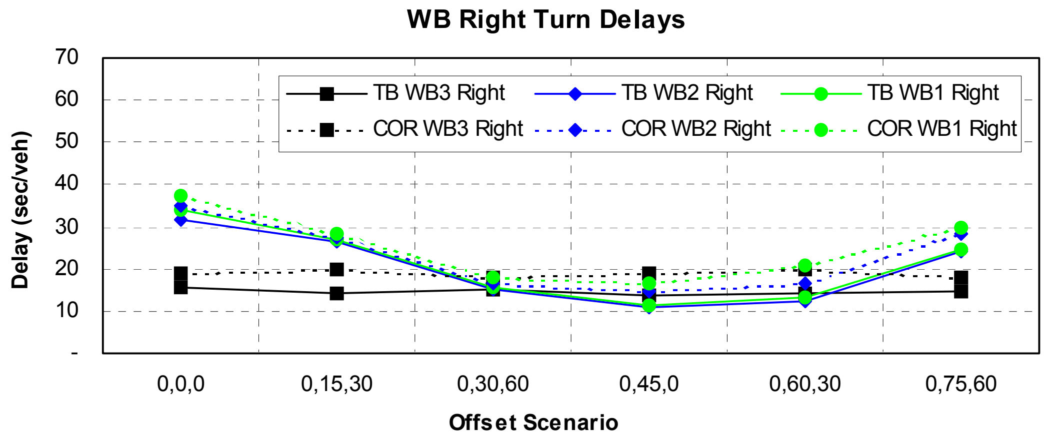


Figure 4.18 Westbound Right Arterial Delays for CORSIM (COR) and Open-TS3 (TB) under different offset scenarios.

Open-TS3 vs. CORSIM Validation – Arterials Queues and Speeds

As with the single intersection validation discussion, to avoid unnecessary repetition only a few general comments are offered on the arterial queues and speeds.

Queues – 0 through 0 graph the average queues for the Eastbound and Westbound thru, left turn and right turn movements. For the given offset scenarios there are no practically significant differences between the queues modeled by the two simulations. The average queues simulated by both models follow similar trends, in both the East Bound and West Bound direction, for the thru, left turn and right turn movements. The queue differences are always within 2.1 vehicles and typically within 1 vehicle or less of each other. It should again be recalled that all scenarios modeled have traffic demands less than capacity, overcapacity demand scenarios would not be expected to achieve such high agreement between predicted CORSIM and Open-TS3 queues.

Speeds – 0 through 0 graph the average speeds for the Eastbound and Westbound, thru, left turn and right turn movements. As with the delays and queues there is excellent agreement between Open-TS3 and CORSIM. Both the absolute speeds and the speed trends simulated by both models are similar. Since accurate speeds play a crucial role in the optimization of any network, this agreement greatly increases confidence in the use of Open-TS3 in optimization procedures.

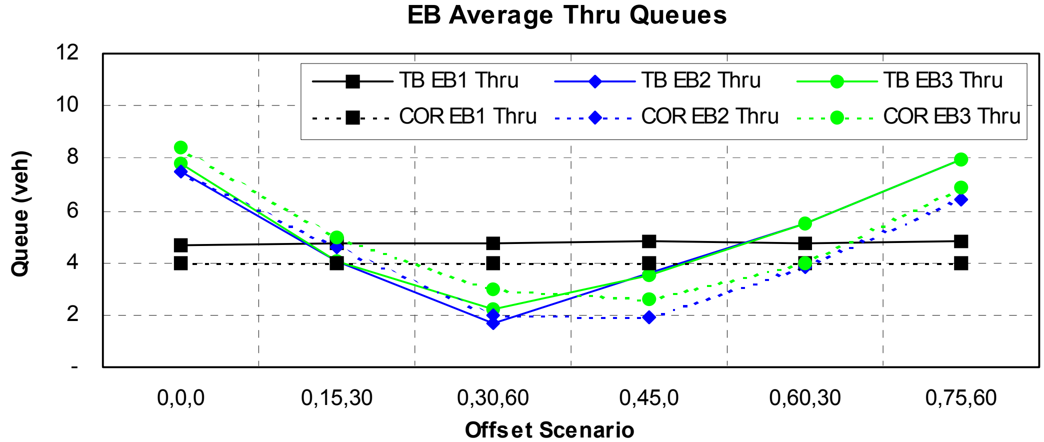


Figure 4.19 Eastbound Thru Arterial Queues for CORSIM (COR) and Open-TS3 (TB) under different offset scenarios

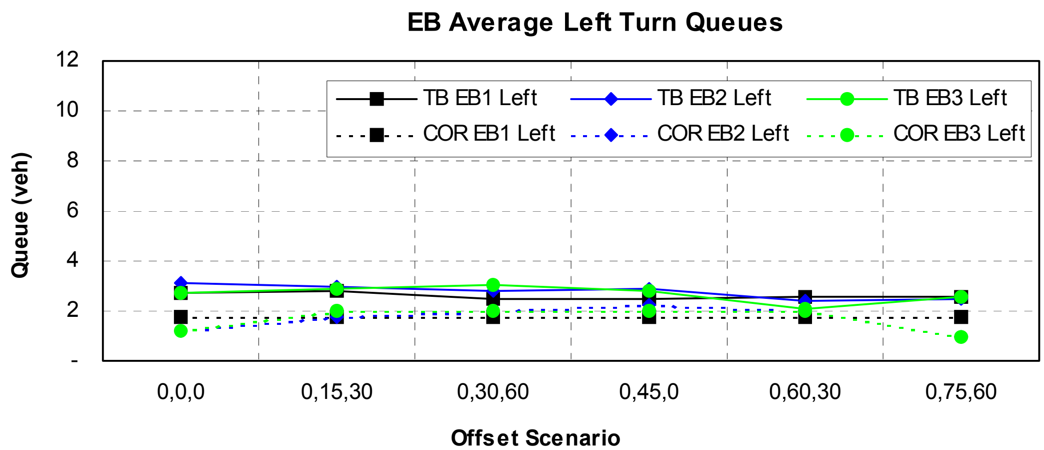


Figure 4.20 Eastbound Left Turn Arterial Queues for CORSIM (COR) and Open-TS3 (TB) under different offset scenarios

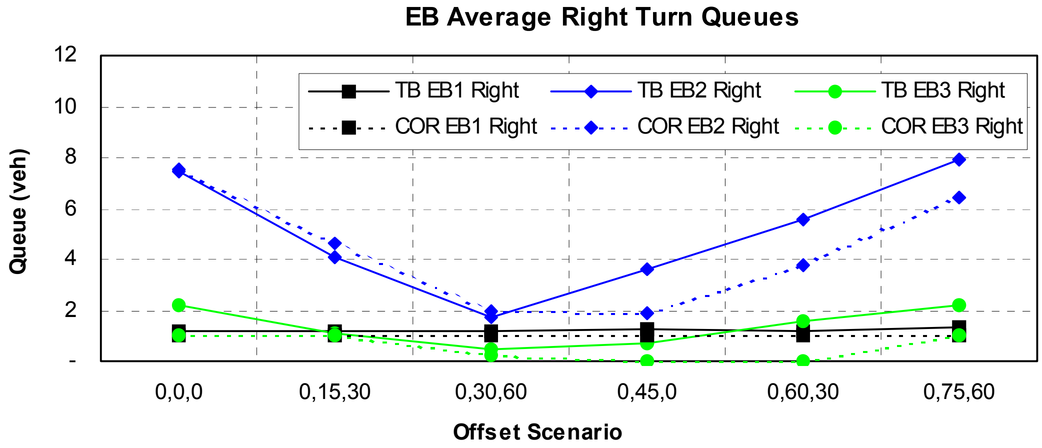


Figure 4.21 Eastbound Right Turn Arterial Queues for CORSIM (COR) and Open-TS3 (TB) under different offset scenarios

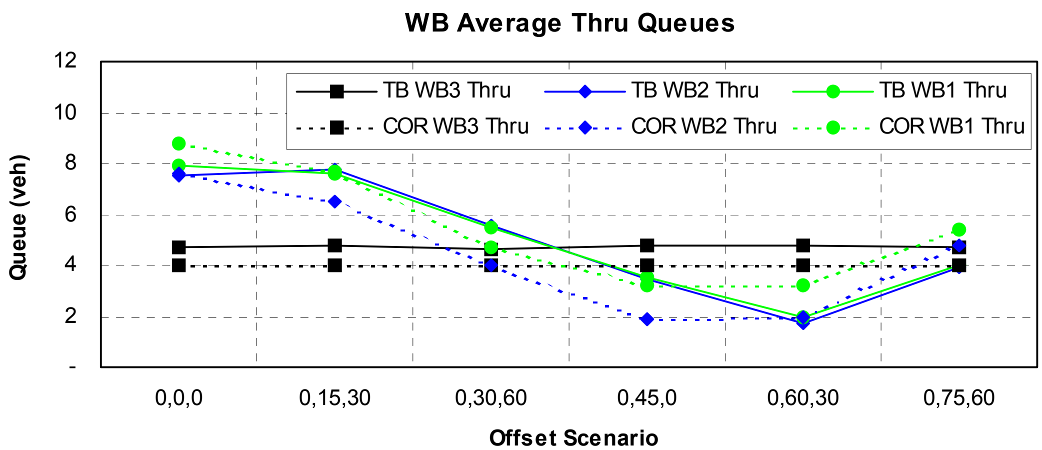


Figure 4.22 Westbound Thru Arterial Queues for CORSIM (COR) and Open-TS3 (TB) under different offset scenarios

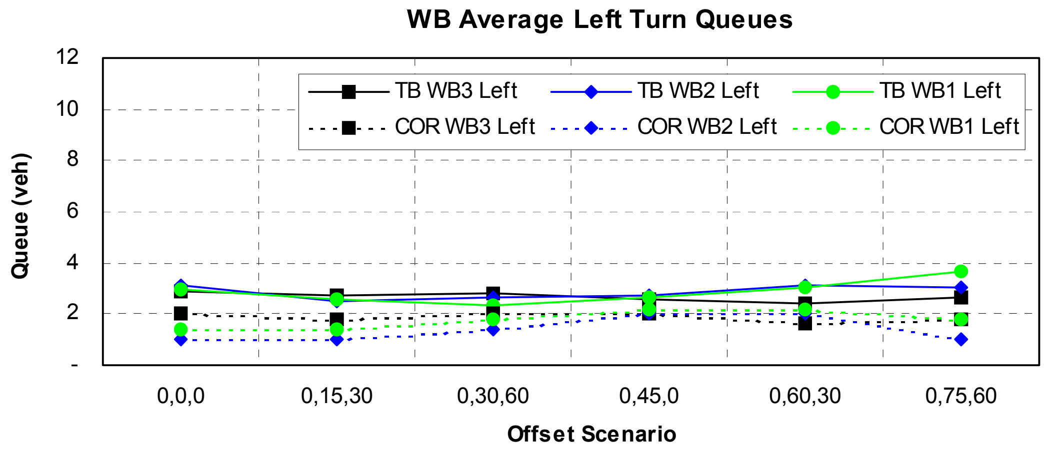


Figure 4.23 Westbound Left Turn Arterial Queues for CORSIM (COR) and Open-TS3 (TB) under different offset scenarios

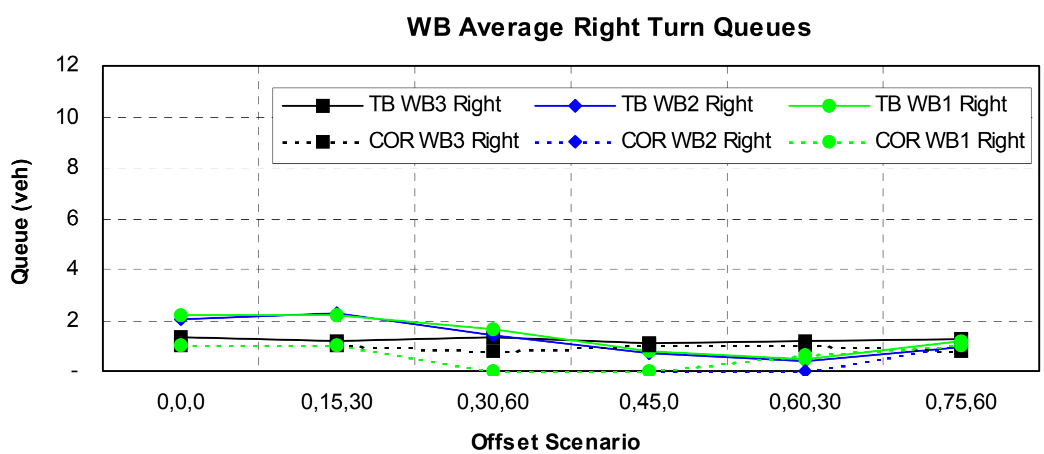


Figure 4.24 Westbound Right Turn Arterial Queues for CORSIM (COR) and Open-TS3 (TB) under different offset scenarios

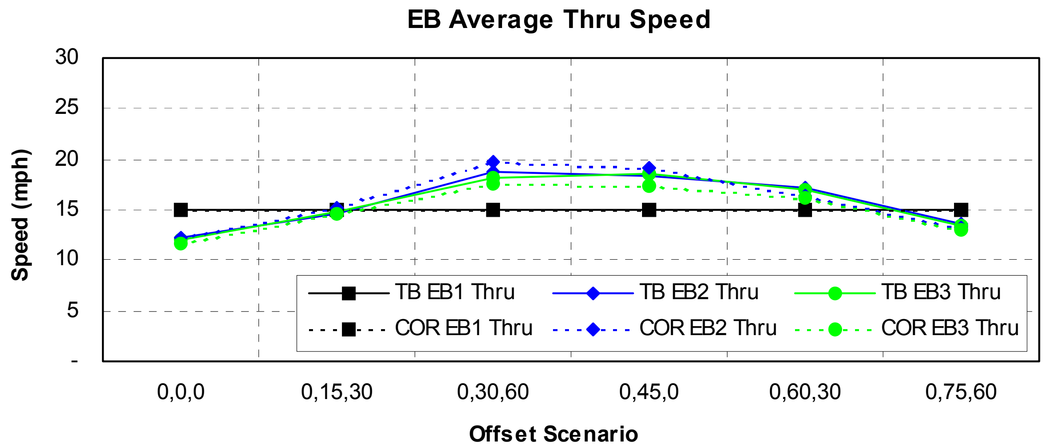


Figure 4.25 Eastbound Thru Arterial Speed for CORSIM (COR) and Open-TS3 (TB) under different offset scenarios

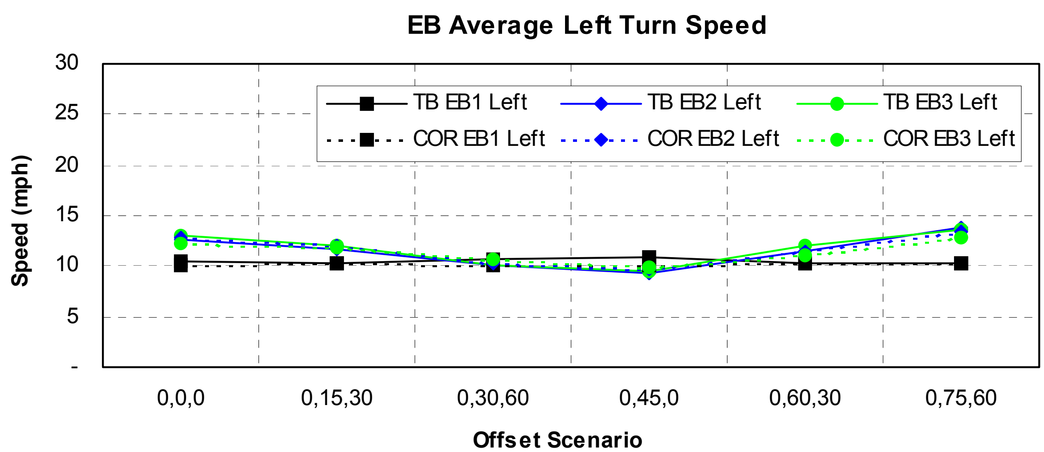


Figure 4.26 Eastbound Left Turn Arterial Speed for CORSIM (COR) and Open-TS3 (TB) under different offset scenarios

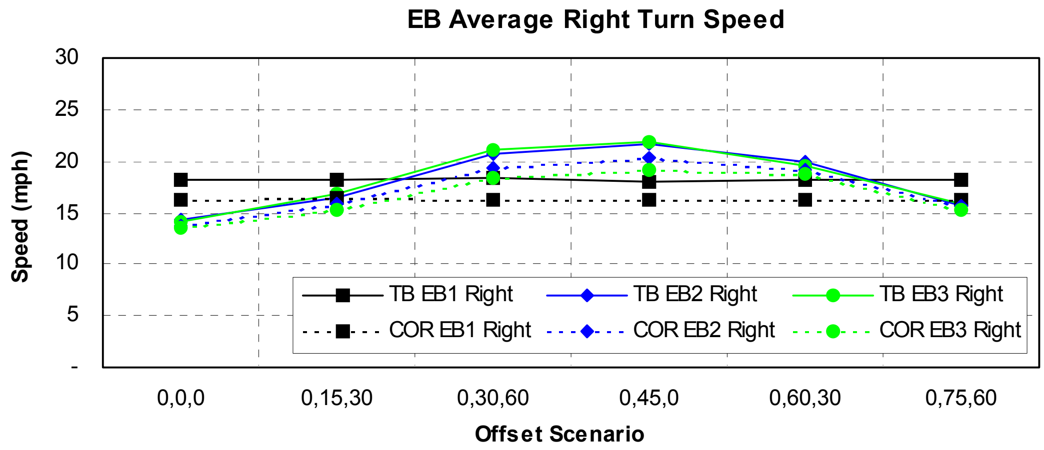


Figure 4.27 Eastbound Right Arterial Speed for CORSIM (COR) and Open-TS3 (TB) under different offset scenarios

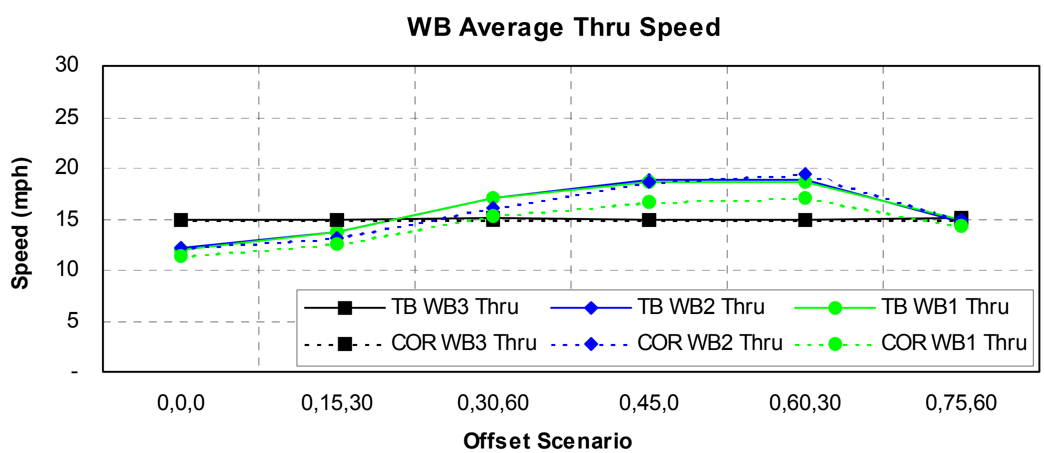


Figure 4.28 Westbound Thru Arterial Speed for CORSIM (COR) and Open-TS3 (TB) under different offset scenarios

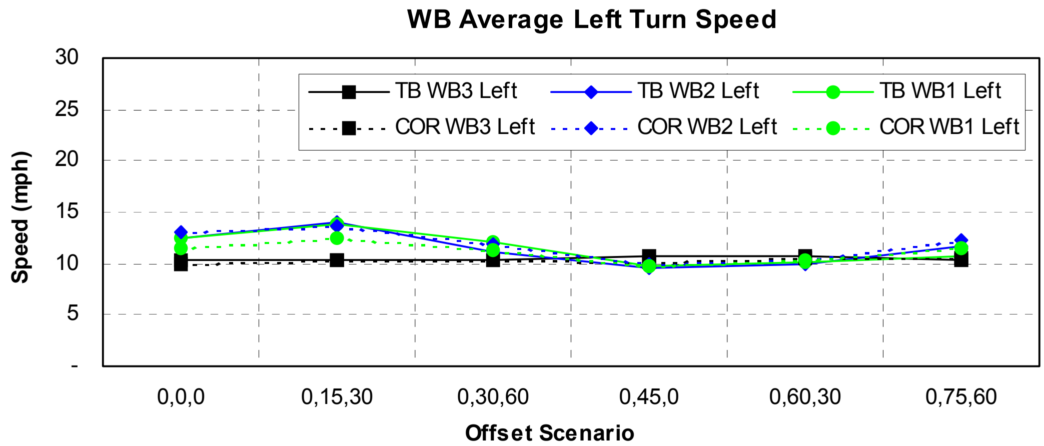


Figure 4.29 Westbound Left Turn Arterial Speed for CORSIM (COR) and Open-TS3 (TB) under different offset scenarios

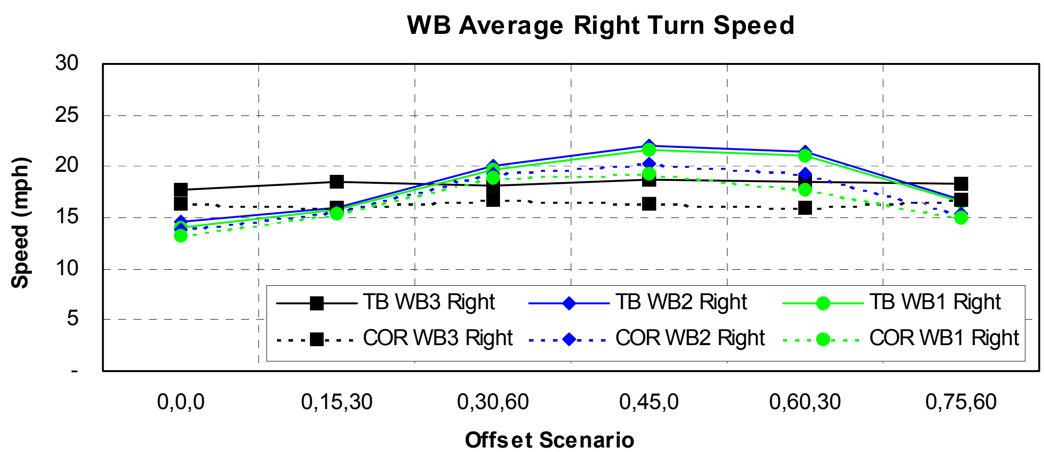


Figure 4.30 Westbound Right Turn Arterial Speed for CORSIM (COR) and Open-TS3 (TB) under different offset scenarios

Open-TS3 vs. CORSIM Validation – Arterial Summary

At the demand studied (approximately 85% of capacity), CORSIM and Open-TS3 demonstrate excellent agreement in the captured absolute MOE values and MOE trends, as signal parameters changed. The arterial volumes, delays, speeds and queues all showed agreement. Along with the single intersection validation results the arterial validation leads to an initial acceptable level of confidence in the utilization of Open-TS3 for both the study of existing systems and the optimization of future systems. The next section will further build that confidence through the comparison of Open-TS3 with real world data.

Open-TS3 Real-World Data Validation

In the preceding sections, significant confidence was gained in Open-TS3 approach through a comparison with CORSIM. The next step of the validation effort is a comparison to a real-world system, as the ultimate goal of a simulation is to reflect the behavior of real-world systems. The comparison of the simulation output with real world operations presented in this section will include a discussion of the required real world data, difficulties encountered in obtaining data, a presentation of the real world network for which data was obtained, and a comparison and discussion of the measured real world versus simulated operation.

Real-World Data Collection

Two categories of data are required for simulation model validation. The first category consists of the data required to utilize a transportation simulation package to analyze a system. (In this section "system" is defined as a generic term that may refer to a single intersection, an arterial, or a network.) This data will be referred to as operational data. The second category of data required captures the effectiveness of the real-world system; this data corresponds to the typical simulation outputs. This data will be referred to as descriptive data.

Operational Data – As stated, operational data refers to the required simulation model inputs. This data is grouped into geometric, traffic, and control data. (This grouping is intended for the validation needs of the proposed simulation. Data such as origin-destination pairs, trip chaining opportunities, mode splits, etc. are not discussed, although it is recognized that there are simulation models that utilize this type of data.)

Geometric Data – These data describe the system's physical characteristics. Included within these data are the number of lanes on each roadway section, movements allowed from/to an approach lane, and roadway distances (i.e. distance between intersections, length of turn bays, etc.). Additional geometric data may include lane widths, speed limits, shoulder widths, adjacent parking, and grades.

Traffic – In the context of simulation inputs, traffic data refers to the volumes being processed through an intersection and the average vehicle travel speeds. It is desirable for volume to be collected in the form of peak hour counts, conducted for each movement, on each intersection approach. Volume counts may include a categorization of vehicles, for example as cars, single axle trucks, multi-axle trucks, and buses. Vehicle travel speeds are taken to be the average speeds traveled between intersections, excluding speed reduction due to traffic signals. Speed limits often act as a substitute for this speed. Traffic data may also include saturation flow, start-up lost time measurements, and pedestrian volumes.

Control – Control data are the data required to simulate the signal control scheme operating in the field. Where applicable, control data includes: signal control type (pre-time, semi-actuated, fully-actuated, or adaptive), offsets, cycle lengths, phase lengths, min and max green times, phasing patterns, right-turn-on-red treatment, and pedestrian timings. Also included in control data, for semi- and full-actuated control, are detector locations and operation settings.

Descriptive Data – These data describe the effectiveness of the system. Typical measures include: delay, travel time, queue lengths, and stops. While many other measures may also be utilized, these four are typical and provide a reasonable representation of an arterial's performance. (One major form of descriptive data not discussed is environmental, which is not considered, as it is not included in the current proposed simulation.)

Delay – Delay is the "additional travel time experienced by a driver, passenger, or pedestrian." (Highway Capacity Manual, 2000) Delay is likely the most common measure of a system's performance. Delay is normally considered at the intersection level, either for the intersection as a whole, by approach, or per movement. In the assignment of a Level of Service to a signalized intersection/approach/movement, control delay is utilized, measured in seconds per vehicle (Highway Capacity Manual, 2000). Control delay is the delay incurred by a vehicle due to the presence of a control device (i.e. traffic signal). This delay is the difference in time between traversing a roadway section unimpeded versus traversing the roadway section with signal control. Total delay differs in that it is the difference between a vehicle's desired and actual travel time. Along with the control delay, total delay incorporates additional delay due to factors other than control devices, such as interactions between vehicles and friction from side street parking. As part of the validation effort it was determine that a data set that included delays was imperative as delay is arguably the most relied upon traffic analysis measure in the United States.

Travel Time – Simply stated, travel time is the time required for a vehicle to travel, including delay, over a section of roadway. Travel time is typically measured in seconds per vehicle or minutes per vehicle. Travel time is commonly used in measuring the effectiveness of signal timing improvements along an arterial. Interchangeable with travel time is average speed; the roadway section length divided by the travel time. Thus, this average speed accounts for the delay.

Stops – Another common descriptive measure is stops. Stops are usually considered in one of two forms, total number of stops (either as an absolute number or a percentage of total vehicles processed) or average number of stops per vehicle over some route/system.

Queue length – Queue length is nominally taken as the number of vehicles stopped (or moving slowly) at a particular location.

Acquisition of Real-World Validation Data

This section presents a brief synopsis of the effort to acquire suitable data for the simulation model validation. This should not be regarded as a formal, in-depth study into the issues of data collection; instead as a presentation of the Open-TS3 validation study data acquisition experience and some conclusions drawn from that experience.

As seen in section 0 two types of data are required for the proposed simulation validation study: descriptive and operational. The possibility of conducting a field study to collect data was deemed not reasonable due to both financial and time constraints. Therefore, all efforts were directed at locating existing data sets that provide the required operational and descriptive data. At the beginning of this search for data it was assumed that operational data sets would be plentiful. Countless networks throughout the United States and the world have been analyzed using simulation. The creation of these simulations required the collection of operational data. It was also assumed that descriptive data would be rare as the need (and costs) to collect such data is typically alleviated by the use of simulations.

Initially, developers or sponsors for other simulation models were contacted (by email, phone, or in person) under the initial assumption that these models would have undergone a similar validation process to that being undertaken for Open-TS3. The goal was to obtain the data sets used in the validation of the other models for use in the Open-TS3 validation effort. In addition, a search of literature and web resources was conducted. These efforts to find data proved surprisingly fruitless, with no useful data set initially obtained. Discussion with developers of several different models has led to the impression that the primary validation method is user feedback. Users are informally comparing model results with what they "see" in the field. Experience has led the developers and users to develop a high degree of confidence in many of these simulations. (As an aside it should be emphasized that this discussion is for signalized networks. There was a notable availability of freeway descriptive data.)

Aside from user feedback as the primary validation method, several other reasons did surface for the lack of data sets. The first is cost. Obtaining descriptive data, particularly delays, is an expensive undertaking. Such costs are difficult to justify when these measures are readily obtained from a simulation. To the end user one of the very benefits of simulations is avoidance of such costs. A second difficulty encountered was that such data could be considered proprietary. Two instances did occur where data sets were found that may have been useful, but difficulties were encountered because the data was deemed proprietary. In both instances private consulting firms collected the data but the contact at each firm stated that the data was the property of their client. In one case the firm was reluctant to pursue the possibility obtaining permission from their client to share the data. In the other instance the firm (Parsons Engineering, Dallas, Texas) did contact their client (The City of Dallas), successfully obtaining permission for the release of the data for validation purposes. Unfortunately the time required to obtain release of, gather, and deliver the data did not allow for its use in the current analysis. However, this data set is potentially very useful and will be used in future validation efforts.

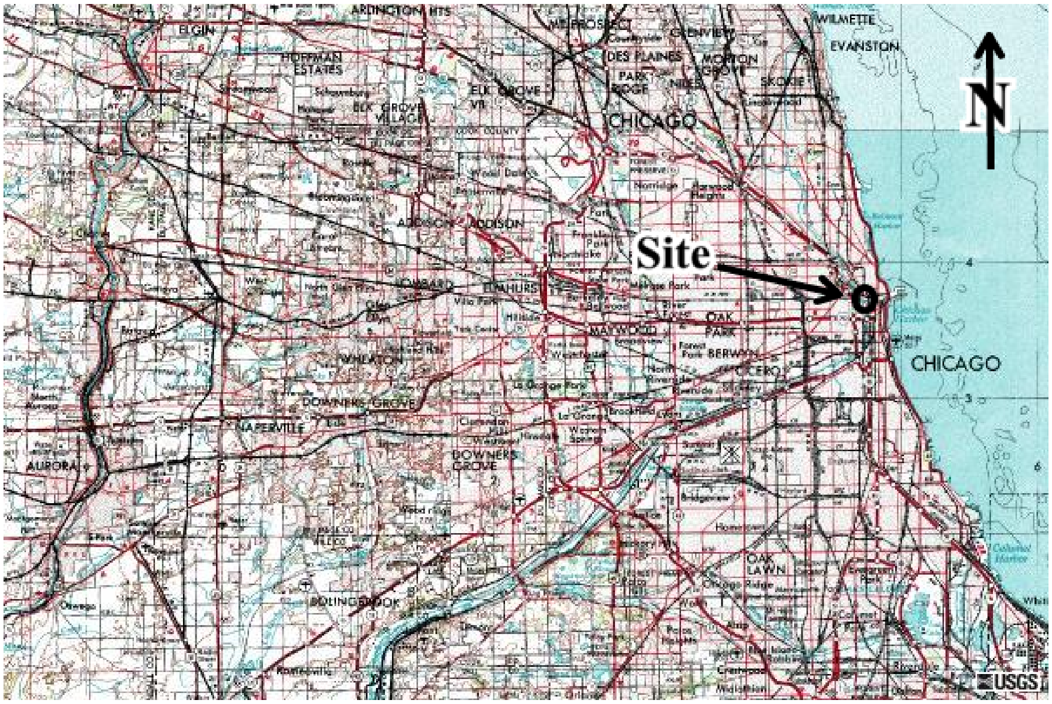
This view of lack of data has been recently confirmed. In Spring 2003 an invitation-only meeting for NGSIM (Next Generation Simulation) was held. This meeting consisted of approximately twenty leading simulation developers. An attendee to this meeting reported that one of the main issues recognized by the group was a lack of data. This group seemed to be placing great hope in the future of video-based data collection.

Description of Obtained Real World Data Set

Given the difficulties discussed, discovery of a useful data set was fortunate. The remainder of this chapter will describe this data set, the data collection methodology, the Open-TS3 simulation of the network, and finally compare the Open-TS3 results with the measured field results.

Data Set Source - The utilized data set was part of a RT-TRACS (Real-Time Traffic Adaptive Control System) field test. This test was a field evaluation of RTACL (Real-Time Adaptive Control Logic) on a twelve-intersection network just north of downtown Chicago (Owen, Stallard and Steiger, 2001). As part of this evaluation, before and after conditions were measured in the field. The before condition field measurements provided consist of both the operational and descriptive data; the same type of data sought for the Open-TS3 validation effort. The adaptive control field test involved numerous participants: FHWA, Chicago Department of Transportation, Chicago Bureau of Electricity, PB Farrradyne, and ITT Systems (Owen, Stallard and Steiger, 2001). ITT Systems, who was responsible for performing the field evaluations, was the primary contact for obtaining the field data utilized in this validation effort.

Site Description - AM and PM descriptive and operational measures were collected for a twelve intersection network just north of downtown Chicago, 0.



(Source: US Geological Survey, http://terraserver.microsoft.com/)

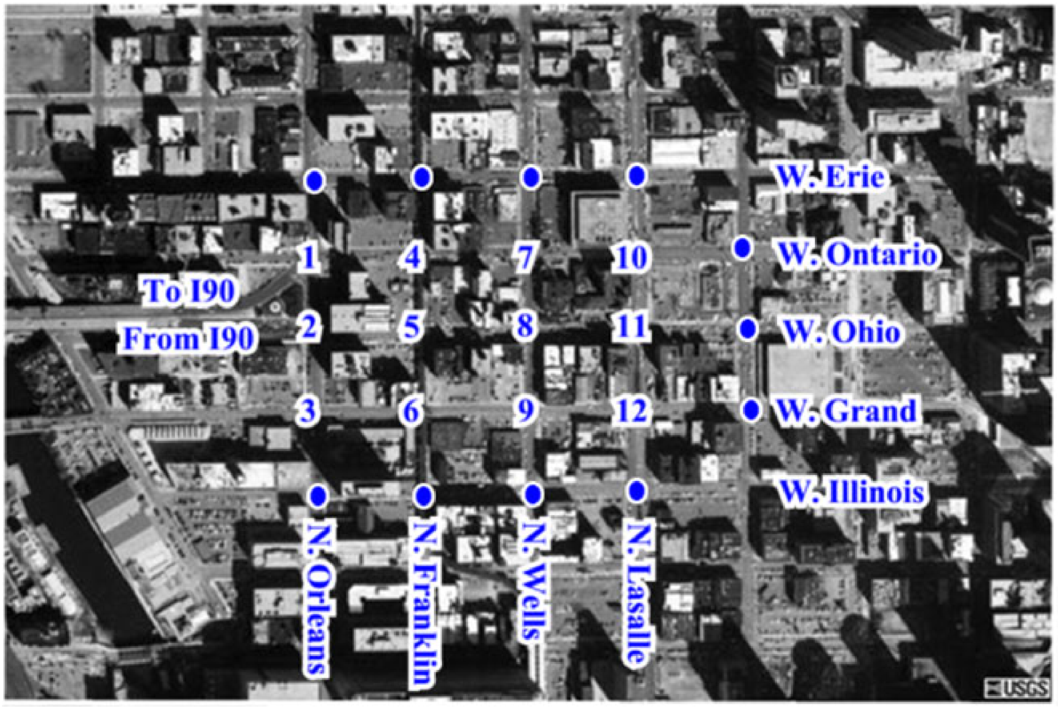
Figure 4.31 Chicago Site Location

0 provides an aerial overview of the twelve intersections (numbered 1 through 12) and the neighboring intersections. This twelve-intersection grid is bounded by West Ontario on the North, West Grand on the South, North LaSalle on the East, and North Orleans on the West. Typical block lengths are 300' to 400' feet with posted speed limits of 30 to 35 miles per hour.

The neighboring intersections on West Erie (North), West Illinois (South), and North Clark (East), are intersections where signal control data were also known. These intersections are indicated by the dots in 0. These intersections were included in the simulation model (for a total of 23 simulated intersections), allowing for nearly all approaches on the twelve primary intersections to have arrival patterns more consistent with those found in the field. Only the eastbound intersection approach arrivals on North Orleans do not account for the impact of upstream intersections.

The network consists of both one- and two-way streets. In the east-west direction West Ontario is one-way westbound (4 lanes), West Ohio is one-way eastbound (4 lanes), and West Grand is one-way eastbound from North Clark to North Wells (3 lanes) and is two-way from North Wells to North Kingsbury (2 lanes each direction). In the North-South direction North Orleans is one-way northbound (4 lanes) between West Ontario and West Ohio and two-way otherwise (1 or 2 lanes each direction), North Franklin is two-way (1 lane each direction), North Wells in one-way southbound (two lanes), and North LaSalle is two-way (3 lanes each direction).

Traffic volume data are limited, based on 15-minute counts performed at each intersection during 1999 and 2000 (Owen, Stallard and Steiger, 2001). While not ideal, this should still provide for a reasonable estimate of traffic demand. The signal timings are all pre-timed with a 75 second background cycle. Timing plans were two or three phase. As part of the adaptive control evaluation, the intersection offsets for the before conditions were optimized. (Owen, Stallard and Steiger, 2001)



(Source: US Geological Survey, http://terraserver.microsoft.com/)

Figure 4.32 Aerial Photo of Chicago Site Location

The primary source for the collection of descriptive data was travel time runs utilizing probe vehicles instrumented with Starlink GPS antennas and receivers. Five probe vehicles were utilized during the peak hours over a three-day period in late in October 2000. The probe vehicles followed specific routes, shown in 0 and 0. Routes involving I-90 access were determined to be critical and were therefore assigned two probe vehicles, all other routes were assigned a single probe vehicle (Owen, Stallard and Steiger, 2001). Over the peak periods, the probe vehicles were able to obtain 10 to 95 observations for each route, with most approaches receiving 30 to 60 observations. From this data, travel time and control delay information was determined for most of the twelve intersections. All reduction of raw data was performed by ITT Systems as part of the RT-TRACS field evaluation with the published results being utilized for this validation effort. For raw data the reader is referred to the RT-TRACS field study report (Owen, Stallard and Steiger, 2001). The most notable lack of data is for North Franklin, traveling southbound, which was not part of any of the probe vehicle routes. No queue length or stop data was obtained as part of the data collection efforts.

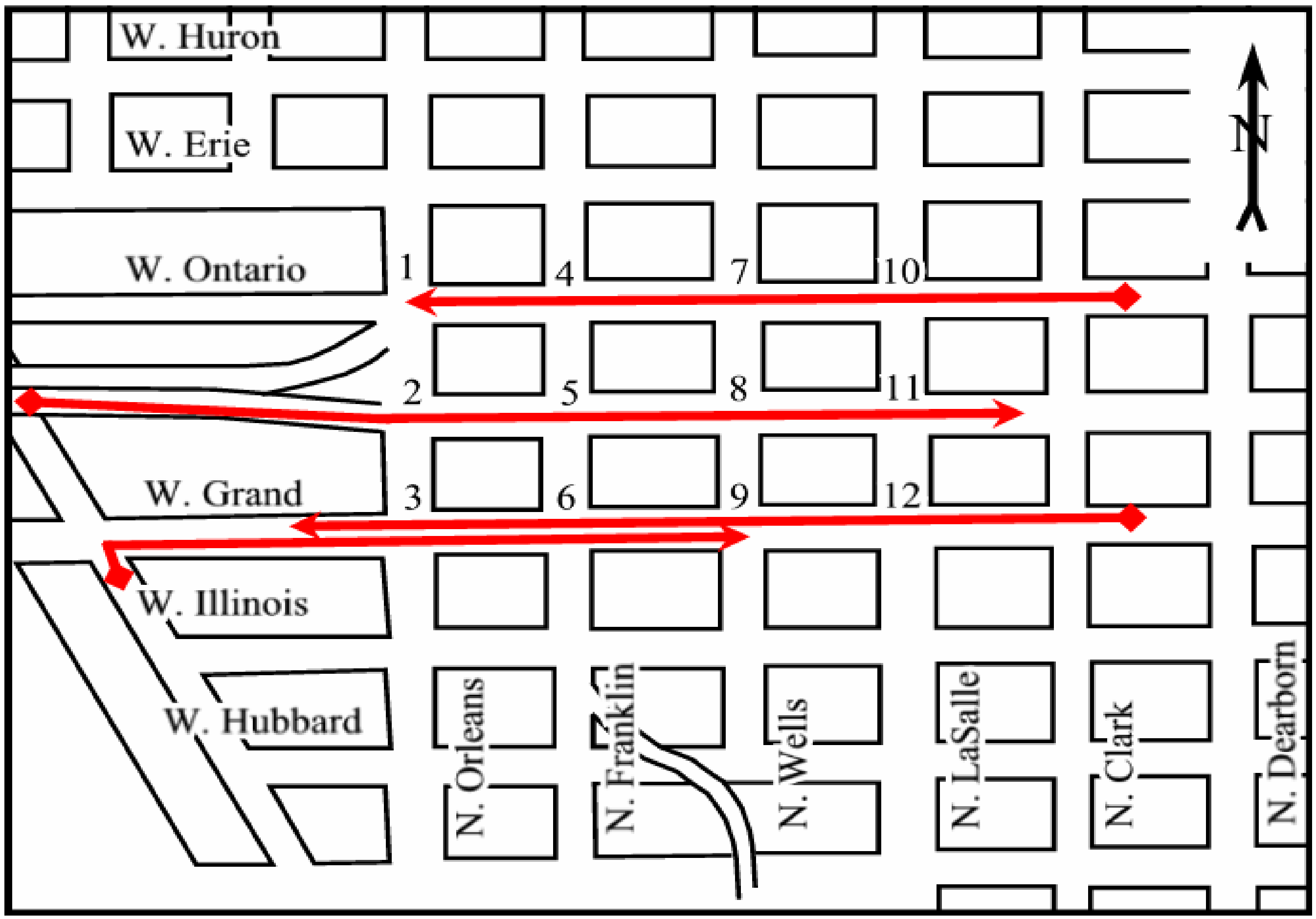


Figure 4.33 East – West Probe Vehicle Routes

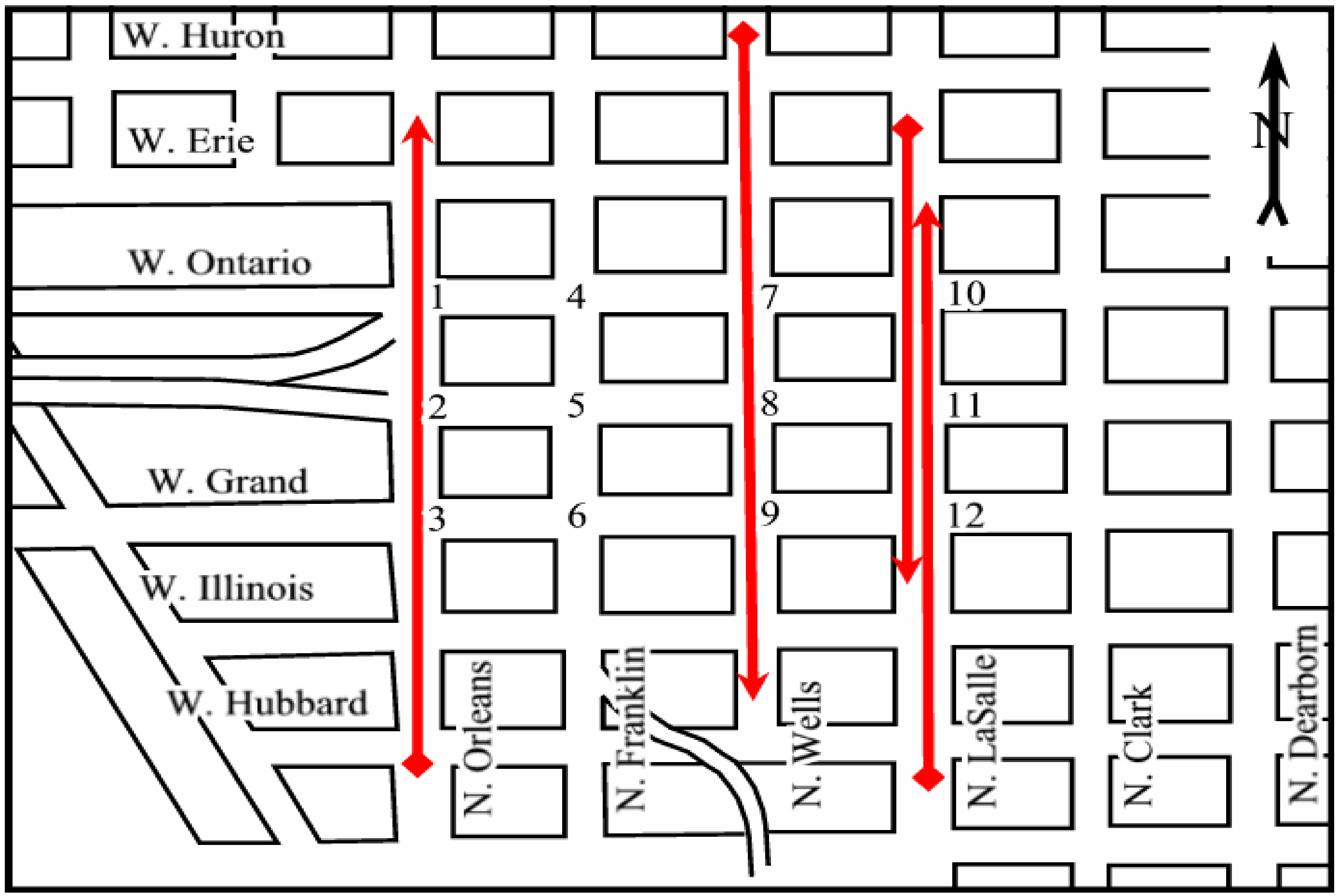


Figure 4.34 North – South Probe Vehicle Routes

Simulation Model of Real-World Site

Utilizing the operational data, a simulation model was constructed for the twenty-three intersection network for which data were collected. The developed simulation utilized the modeling blocks discussed in chapter three. Descriptive data were collected for the primary twelve intersections. 0 is a screen capture of the developed model.

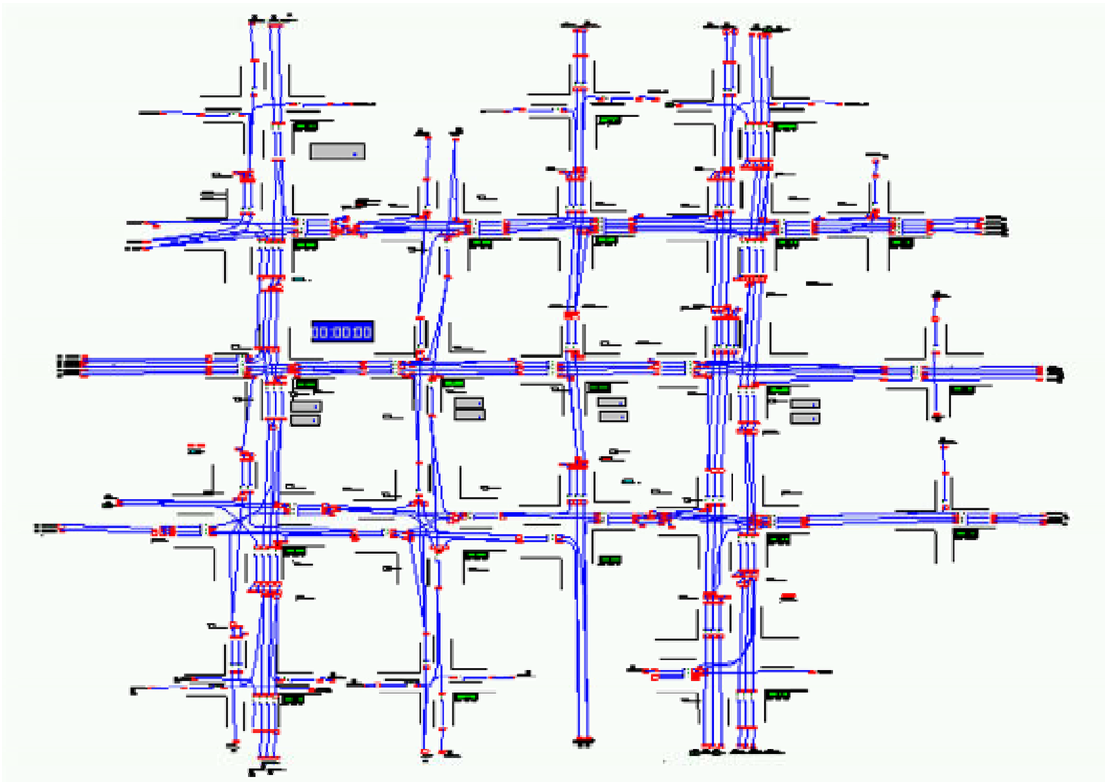


Figure 4.35 Chicago Site Open-TS3 Model

Real World Validation Results Presentation and Discussion

As stated, delay is arguably the most utilized measure of effectiveness and therefore must be the foundation of any validation effort. The following discussion will first compare the AM peak period results, then the PM peak period results, and finally travel time results for both peak periods. 0 and 0 contain the simulated and field measured control delays for the AM and PM peak periods, respectively. Each control delay value listed applies to each approach's through movement. As seen in 0 and 0 all probe vehicle routes travel straight through the network; no turning movements are made by the probe vehicles. Therefore, delays calculated from the probe vehicle data are for the through movements only. Thus, for a consistent comparison the simulated delays given in 0 and 0 are also for through movements only.

The question of key importance that must be answered is whether the simulation accurately reflects real-world conditions. The primary basis upon which this question is answered is difference between the simulated and measured delays (or other measures).

Deciding whether this difference is acceptable is not a trivial matter. In section 3.4.3.5 the importance of statistical difference versus practical difference in comparison of traffic measures was discussed. As shown simple statistical measures may lead to erroneous conclusions. Thus, the comparison of the field measures verses simulated measures stresses the importance of practical differences. This approach is also not without drawbacks. It relies on the expertise and judgment of the individual undertaking the comparison. It is difficult to determine the exact knowledge base, and potential biases, from which an individual arrives at conclusions. Different individuals may reach different conclusions from the same set of data. For example, the validation discussion relies on the traffic engineering expertise of the author, while all attempts are made to be unbiased it is clear that other readers may reasonably draw different conclusions.

Future research will be undertaken to address the issues raised in the preceding discussion. It is necessary to develop a methodology by which practical significance between models may be categorized, in a reproducible manner, which does not rely on the judgment of any single, or group of, individuals.

AM Peak Period Control Delay – Overall, the AM peak period simulated versus probe vehicle control delays demonstrate reasonable agreement. For example, the critical I-90 access route, West Ohio westbound, the simulated versus probe vehicle control delays are (all in sec/veh) 7.1 vs 3.2, 11.9 vs 13.0, and 16.0 vs 10.7 for the intersections with North Franklin, North Wells, and North LaSalle, respectively. Considering the uncertainty in the volume information, peak period fluctuations, and day-to-day fluctuations, these delays should not be argued as practically different. Both the probe vehicle and simulated delays indicate similar traffic conditions. A review of 0 leads to the same conclusion of similar probe vehicle and simulated traffic conditions for West Ontario westbound, North LaSalle northbound and southbound, and North Wells southbound. Reasonable agreement between control delays is seen on all of these routes, increasing the confidence in the simulation model.

Further review of 0 does however show that not all probe vehicle and simulated control delays indicate similar operating conditions. A prime example of disagreement is the northbound approach at the North Orleans and West Ontario intersection. The probe vehicle delay is 1.2 sec/veh while the simulated delay is 21.9 sec/veh. A 1.2 sec/veh control delay (probe vehicle measured delay) implies that nearly all of the probe vehicles passed through the intersection unimpeded by the traffic signal. The 21.9 sec/veh control delay (simulated delay) implies that at least a percentage of vehicles are hindered by the signal control. Considering just the control delays the probe vehicle and simulation would seem to indicate different operating conditions.

A review of the data collection methodology reveals how both the simulation and probe vehicles may be reflecting different aspects of the real-world operation. From 0 it is seen that the probe vehicle route that includes this approach from which the control delay is measured begins south of West Hubbard on North Orleans, traveling northbound on North Orleans thru West Hubbard, West Grand, West Ohio, and finally West Ontario. A review of the signal timing plans and offsets demonstrated that the signals have been optimized for this route. Platoons of vehicles traveling along this route fall within a green band; they should not be hindered by an intersection's signal control. Thus, the majority of probe vehicles would be expected to take advantage of this green band, incurring little delay, as measured in the field.

In contrast the simulated control delay is not calculated from a sampling of probe vehicles but from all vehicles that travel northbound through the intersection of North Orleans and West Ontario. In this instance, this is a significantly different vehicle population than that captured by the probe vehicles. The intersection of North Orleans and West Ohio provides access to the network from I-90. The North Orleans and West Ohio intersection west approach's left turn movement (i.e., traffic from I-90 turning northbound onto North Orleans) is significant, at approximately 950 veh/hr, nearly double the northbound traffic, from the south approach. Many of these vehicles from I-90 travel north on North Orleans through the West Ontario intersection. These vehicles are not within a green band and are hindered by the North Orleans and West Ontario signal control.

Thus, the simulation is capturing a major movement (from I-90 to northbound on North Orleans) not reflected in the probe vehicle measurements. The probe vehicle measurements dramatically fail to capture the overall performance of the approach, underestimating the through movement control delay. Wherever an upstream turning movement feeds a substantial portion of an approach's through movement, a system of probe vehicle routes such as those utilized is likely to fail to accurately reflect the approach's operation. The intersection of North Wells and West Grand southbound approach is another example of this effect. The right turn movement onto North Wells from West Ohio is a significant movement that is not captured by the probe vehicles. Again this leads to a significant skewing of the probe vehicle control delay.

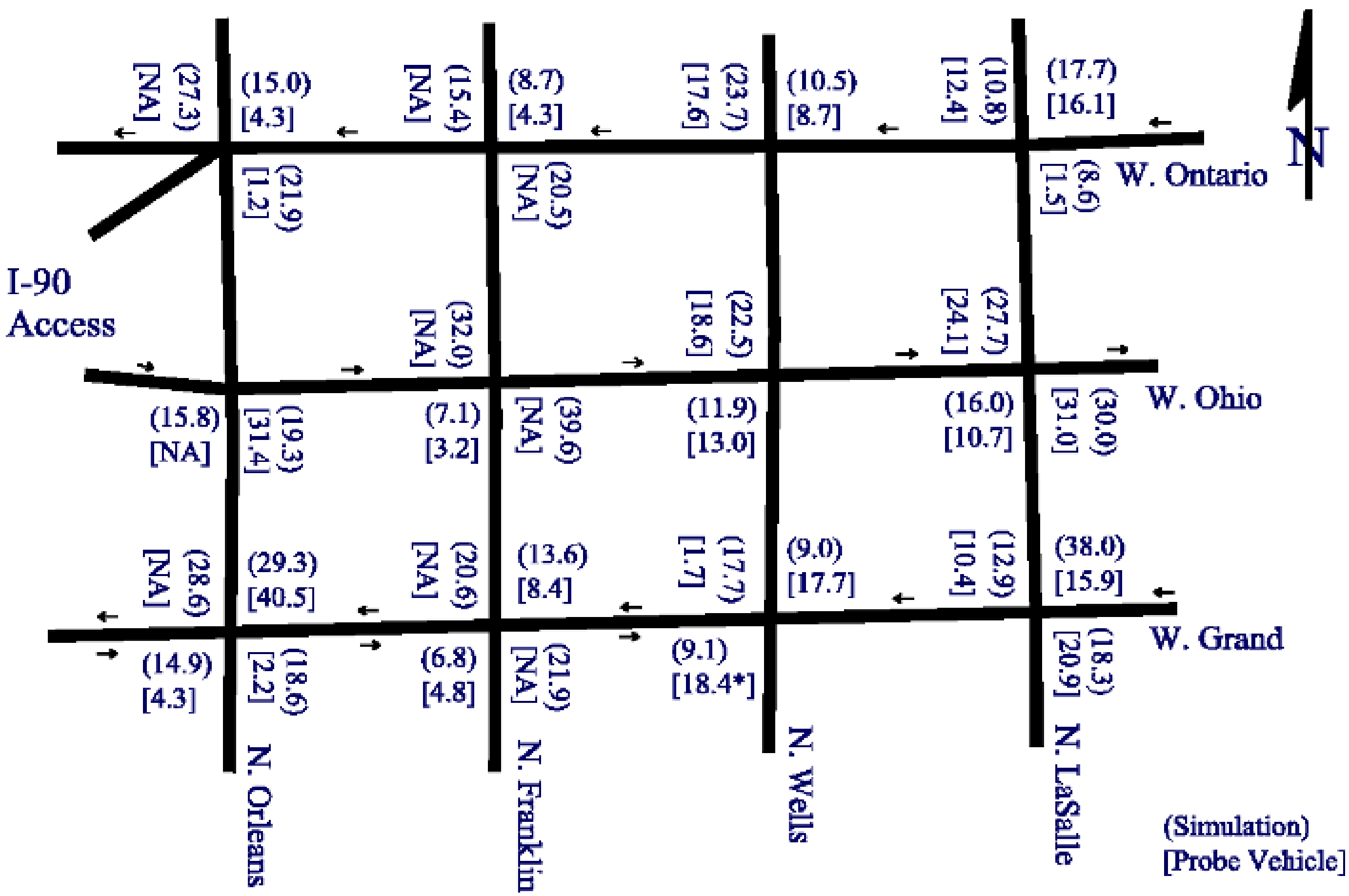


Figure 4.36 Chicago Simulation and Probe Vehicle Control Delay (sec/veh) – AM Peak

PM Peak Period Control Delay – The PM peak period simulated versus probe vehicle control delays may be found in 0. These delays do not demonstrate the same level of agreement as the AM peak period. While there are still examples of agreement, such as the West Ohio westbound traffic flow there are significant areas of disagreement. Areas of particular concern are the West Ontario and West Grand eastbound and the intersections of West Ontario and West Grand with North LaSalle.

Significant upstream turning movements do not readily explain these control delay differences. There are several possible explanations. The first is the likelihood of the simulated volumes being significantly different from the volumes during the probe vehicle study. The volumes were collected up to a year prior to the conducting of the probe vehicle runs. Also, during the probe vehicle study a bridge providing access out of downtown Chicago was closed, leading to a significant increase in the northbound traffic on LaSalle during the PM peak (Owen, Stallard and Steiger, 2001). This detour is not reflected in the volume counts, which are the basis of the simulated volumes. Observation during the probe study stated that the West Ontario and North Ontario intersection was severely congested. The modeled volumes do not lead to such congestion. The possibility clearly exists that the data given for the before conditions does not match the field conditions during the probe vehicle measurements.

A second possible explanation for the observed difference is that the signal timing plans given in the raw data were different from the actual field conditions. The before signal timings were developed as part of the adaptive control test with the intention of allowing for a comparison of a well timed "before" system against the "after" adaptive control. The before timings were developed utilizing the simulated volumes. The possibility exists that given the bridge closure, or for other reasons, these timings where not implemented.

A third possibility is that the simulation has accurate initial data and is not adequately reflecting the real-world operation. Unfortunately, with the known possible discrepancies in the input data it is not possible to clearly distinguish if the error lies in the input data or the simulation model, although it is possible to gain some additional insight. As part of the adaptive control study, ITT Systems developed CORSIM models of the before conditions. 0 includes the ITT Systems CORSIM model results for the Westbound traffic on West Ontario and West Grand, the two routes with significant discrepancies. The CORSIM results have a closer correlation to Open-TS3 results than the probe vehicles results. Thus, if the input data is correct then CORSIM also is incorrectly reflecting the system operation. While it is certainly possible that both Open-TS3 and CORSIM are incorrect it appears more reasonable that the discrepancies result from inaccurate input data.

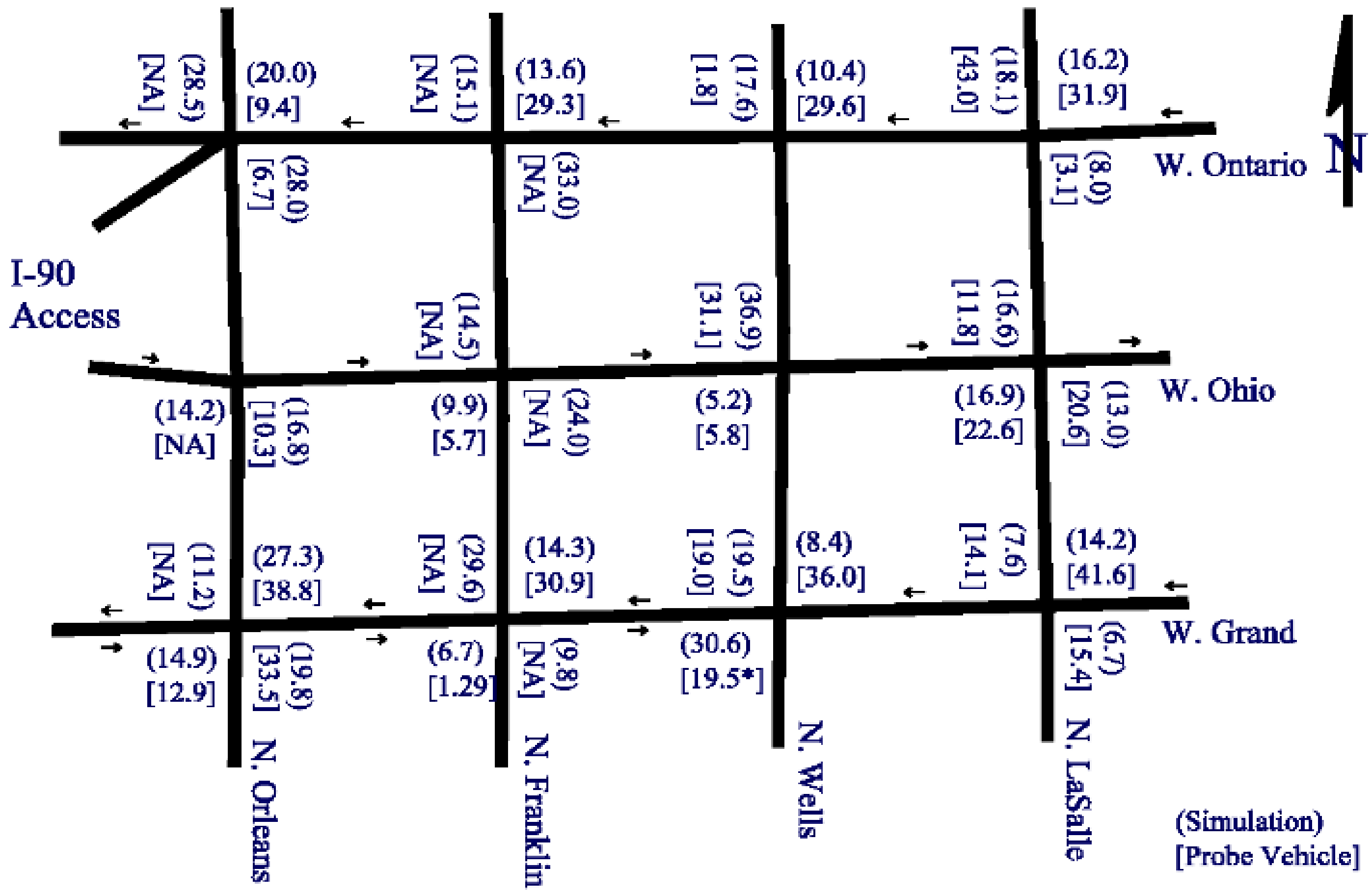


Figure 4.37 Chicago Simulation and Probe Vehicle Control Delay (sec/veh) – PM Peak

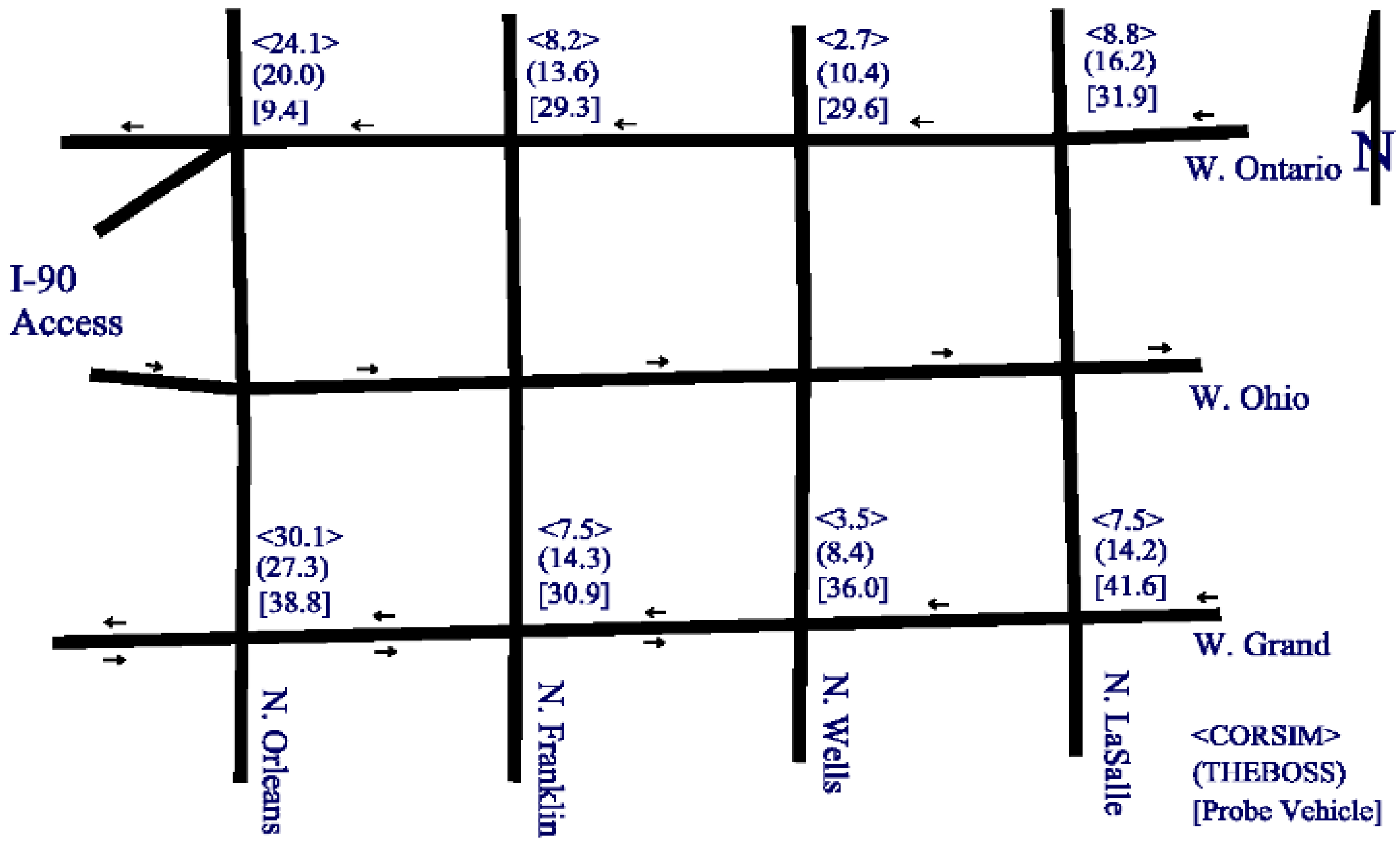


Figure 4.38 Chicago Open-TS3, CORSIM, and Probe Vehicle Control Delay (sec/veh) – PM Peak, Selected Routes

Real World Validation Summary

Conclusions may be drawn from this validation effort not only about the validity of the proposed model but also about the availability, quality, and use of real-world data.

The first conclusion is that complete, real-world data sets are extremely difficult to acquire. While initial expectations were that descriptive data would be rare, the extensive difficulty in acquiring data sets was quite unexpected. Even when a data set was obtained as seen in the above discussion, the quality of the data was not at a desirable level for an in-depth and reliable simulation validation effort.

The second conclusion is that one must be careful to understand the data collection methodology and data limitations. For example, the delay results presented in the obtained adaptive control study appear simply as approach control delays. It is critical to realize that all control delays are for through movements, derived from upstream through movements. This is a significant limitation to the data set that must be understood when utilizing the data, as seen in the control delay discussions. (As an aside, it is distinctly possible that the study from which the data was taken did not take these limitations into account. For example, the northbound approach of the West Ontario and North Orleans intersection was stated to have degraded performance measures under adaptive control, when compared to the before conditions. It is quite likely that the adaptive control was better serving the traffic that arrived from I-90, the dominant upstream movement, at some cost to the North Orleans traffic. Improved service to the I-90 vehicles would not be captured by the probe vehicles, leading to a potentially false impression that the approach operated better under the before conditions.)

The remaining conclusion revolves around the success of the simulation in the validation result. The AM results clearly lead to increased confidence in the modeling approach while the PM results are inconclusive. At a minimum, it may be reasonably stated that no red flags regarding the simulation methodology were raised as a result of this validation. It is also reasonable to state that given the results, combined with the earlier CORSIM results, a reasonable level of confidence in this modeling approach is warranted.

Possibly the most significant result of this validation is the clear realization that there is a need for a field study with the expressed goal of obtaining data sets for the validation of arterial analytical and simulation models. This would not only be useful for validation of Open-TS3 but also would provide a means for direct validation of the many other simulations that are used in practice today.

VALIDATION CONCLUSION

This chapter has concentrated on the validation of Open-TS3 simulation model. Utilizing the model blocks developed in the previous chapter Open-TS3 was compared against both the well-respected CORSIM model and real-world data. Through this validation effort Open-TS3 was not only seen to be flexible but also capable of accurate simulation arterial network conditions under non-congested conditions. The intersection and arterial validation studies (Open-TS3 vs. CORSIM) allowed the user to gain confidence in the volumes, delays, queues, and speeds simulated by Open-TS3. Excellent agreement was seen between Open-TS3 and CORSIM, except for two areas where some caution in the use of Open-TS3 (and CORSIM) was highlighted. First, in overcapacity demand scenarios, it was seen that CORSIM and Open-TS3 could simulate significantly different MOE values. Second, it was seen that left turn operations could also be an area of significant differences in terms of capacity thresholds and operations after capacity had been exceeded.

Next Open-TS3 was compared against the real-world conditions of a network in downtown Chicago. This effort led to several conclusions/understandings. The first being that caution must be exercised in the utilization of real-world data. One must be careful to understand the limitations and nature of the data. The second is that currently there exists a great dearth in usable, available data sets. Finally, the third, that given reasonable data Open-TS3 was able to reasonably capture real-world conditions. The real world validation effort furthered the initial confidence gained by the Open-TS3 versus CORSIM validation effort.

Thus, in summary, it has been seen that Open-TS3 provides a flexible modeling environment, in which users may have confidence in the results. In chapter six, Open-TS3 is even further expanded to include adaptive signal control.

CHAPTER 5 REVIEW OF TRAFFIC SIGNAL TIMING CONTROL AND DESIGN

INTRODUCTION

Over the last half century plus significant effort has been placed into developing signal control strategies that improve the operation of our transportation systems. This chapter investigates current and past signal control and coordination strategies, including a review of pre-timed, actuated and some of the many existing and proposed real-time adaptive control methodologies. This review is meant to be informative, but not to provide a complete in-depth description of all the literature to date on signal timing, as this effort in itself would fill many volumes.

This chapter first presents a discussion of the three basic signal control categories; pre-timed, (semi-) actuated, and adaptive. Included within this discussion is a review of off-line software typically utilized in the determination pre-timed and actuated signal timings and in some instances enhanced to work in a real-time adaptive manner. As adaptive control is focus for implementation of advanced traffic signal control into Open-TS3 this discussion includes a generalized discussion of features common to many adaptive control strategies and individual overviews of some of the currently existing and proposed adaptive control strategies. Finally, a discussion is offered that delves into past and present experiences in the real-world implementation of adaptive control strategies.

PRE–TIMED SIGNALS

Pre-timed signal control generally refers to signal operation in which timing plans (cycle length, phase duration, phase order, etc.) are determined off-line (typically based on historical data), remain constant over specific time periods, and are updated in the controller (either local or master) based on time of day, operator commands or volume data. Predominately pre-timed signal coordination strategies involve operating all signals at a common cycle length (or multiple of) based on the average demand at the critical intersection (the intersection requiring the longest cycle under isolated conditions) at pre-specified offsets. There exists no more intelligence in a pre-timed signal than that put into the off-line timing plan development. Pre-time control is incapable of treating any stochastic variations in traffic demand in any efficient, real-time, responsive manner.

Prior to the revolutionary advances in computer technology that have occurred over the last few decades signal timing and coordination schemes were primarily calculated by hand. The offsets, and phase splits were determined according to general rules based on single alternate, double alternate, etc. type coordination schemes. Other "by hand" methods were also utilized, such as the Kell method, which could develop schemes not as constrained as the alternate offset strategy (Gordon, Reiss, Haenel, Case, French, Mohaddes, and Wolcott, 1996). Improved and increasingly complicated coordination schemes began to be developed as advancements in computer computational abilities and software occurred. These advances in off-line software development are discussed after the actuated signal discussion.

Attempts to improve the pre-time signal plan development process have lead to substantial research effort and experimentation devoted to the arrival approximations that should be made at the critical intersections (stochastic vs. deterministic), whether methods for determining isolated intersection split lengths necessarily lead to optimal coordinated timings, the effects of half-cycling lower volume intersections, what is the "best" coordination objective (minimize delay, maximize bandwidth, minimize stops, etc.) etc. (Gordon, Reiss, Haenel, Case, French, Mohaddes, and Wolcott, 1996; Sripathi, Gartner, and Stamatiadis, 1995; Wallace and Courage, 1995; Skabardonis, 1991).

ACTUATED SIGNALS

Lin (Lin and Vijayakumar, 1988) provides a reasonable summation of actuated control when he states that traffic-actuated signal control essentially bases whether to terminate or extend the current green duration on a gap seeking logic, which attempts to determine when arriving vehicles no longer efficiently utilize the green. At an intersection under actuated control the presence of queued vehicles and gaps in the approaching traffic stream are detected through the utilization of inductive loops or other detection technology. This information is used in the actuated controller's logic to adjust the cycle and phase lengths in a real-time cycle-to-cycle traffic responsive manner. While unable to alter the preset phase order, an actuated controller has the capability to be set to "skip" a phase should there be no demand for it at the current time. At an isolated intersection actuated control typically offers vastly superior operation over pre-timed control, as a result of an actuated signal's ability to alter cycle lengths and splits, more efficiently handling real-time stochastic effects.

As with pre-timed controllers there is often an attempt to include actuated controllers in coordination schemes. Although, in a coordinated system an actuated controller operates in a semi-actuated manner, that is, not all phases may be gaped out or skipped and a background cycle length constraint is imposed. Improvements similar to those found at isolated intersections are not typically found between semi-actuated and pre-timed coordinated signals. While offering some potential efficiency improvements, coordinated semi-actuated signals are hindered by many of the constraints of pre-timed signals, such as common background cycles and fixed offsets. Under some circumstances semi-actuated control may even result in less efficient performance due to problems such as early release from green and downstream queue interference (Shoup and Bullock, 1999).

Prior to discussing adaptive control some discussion is offered on computer software programs designed to determine optimal signal timings and offsets off-line. It will be seen in a later discussions that some of these programs have been adapted for adaptive control strategies.

OFF-LINE SOFTWARE

As computer technology improved many of the "by hand" signal timing plan development methods were incorporated into various software packages. Along with the traditional methods the availability of increased computational power also allowed for the development of many new approaches to constructing signal timing plans (isolated and coordinated). Concepts such as mathematical programming and heuristic search procedures were, and still are, utilized as a result of possibilities made available by computer advances. Most Off-Line procedures tend towards one (or a combination of) of two goals: Delay Minimization or Bandwidth Optimization.

Delay Minimization - TRANSYT-7f

Probably the most utilized software program for arterial and network signal timing optimization in the United States is the TRANSYT-7f program (May, 1990). TRANSYT-7f is based on the TRANSYT software originally developed in 1967 by Dennis Robertson (1981) at the Transport and Road Research Laboratory (TRRL) in Great Britain. Since 1967 nine English versions of TRANSYT (TRANSYT1 through TRANSYT9) have been developed at the TRRL (May, 1990). Since 1981 the American version of TRANSYT (based on TRANSYT7) has been under continual development for FHWA at the University of Florida Transportation Research Center, with the latest version being TRANSYT-7f, version 8.1 (Park, Messer, and Urbank, 1999). It will be seen in the adaptive control discussion that TRANSYT forms the foundation for SCOOT, one of the more established adaptive control strategies.

TRANSYT-7f is a macroscopic, deterministic simulation and optimization model (169). The optimization model utilizes an iterative hill climbing (gradient search) technique. TRANSYT-7f optimizes cycle length, green splits and offsets, phase sequence is not optimized and is a required input. Several objective functions are available for optimization including, a disutility index (a function of delay, stops, fuel consumption and, if desired, queue spillover), progression-opportunities (a function of maximum bandwidth and short term progression opportunities), various forms of throughput maximization and combinations (linear, nonlinear, multi-staged) of these objectives.

A few features of TRANSYT-7f include that delay is based on Webster's basic model with some adjustments, the simulation attempts to account for platoon dispersion and queue blocking, a user may choose to optimize either in a linkwise or stepwise fashion, horizontal queues are modeled based on shock-wave dynamics, multi-cycle simulation is allowed and optimization may be performed for oversaturated conditions. TRANSYT-7f provides measures for delay, level of service, stops, maximum back of queue, queue spillback, throughput, fuel consumption and total operating cost.

Probably the two greatest recognized drawbacks of TRANSTY-7f are that is does not optimize phase sequences (in actuality it mainly optimizes offsets instead of green splits) and that it may converge to a local optima with final solutions dependent upon initial values of offsets and green splits (Park, Messer, and Urbank, 1999; Hadi and Wallace, 1998; 169). Improvements to TRANSYT-7f are continually being researched and implemented including, updating embedded parameters and models (Yu, 1997), methods to incorporate phase sequence optimization (Hadi and Wallace, 1994; Park, Messer, and Urbank, 1999), and improved queue representation (Park, Messer, and Urbank, 1999).

Bandwidth Optimization - MULTIBAND

One of the most well know bandwidth optimization programs is MULTIBAND (Gartner, Assmann, Lasaga, and Hou, 1991; Stamatiadis and Gartner, 1991; Sripathi, Gartner, and Stamatiadis, 1995), which is an extension of MAXBAND (Chaudhary, Pinnoi, and Messer, 1991; Cohen, 1983). The MULTIBAND approach differs significantly from the TRANSYT-7f in that while TRANSYT-7f typically desires to minimize some measure of delay MULTIBAND deals with bandwidth optimization. (PASSERII and PASSER IV are additional popular bandwidth optimization programs.) While the bandwidth approach has the advantage of including phase sequence optimization, optimization of bandwidth does not necessarily imply minimization of delays (Stamatiadis and Gartner, 1991).

Most bandwidth optimization programs consider the bandwidth to be uniform throughout the arterial, although MULTIBAND allows for different bandwidths in different arterial sections. This feature allows for the creation of signal plans and offsets that are not based on total directional traffic but instead account for turn-in and turn-out traffic, which often causes significant variation in traffic flows. This theoretically allows for signal timing plans and offsets which are more efficient than those determined by many of the other bandwidth optimization programs. In addition to link specific bandwidths, the common cycle length, link specific progression speed and phase sequences are optimizable features in MULTIBAND. Through simulation experiments Gartner (1991) demonstrated that MULTIBAND offered significant potential for improvement with reductions in average stopped delay and average delay on the order of ten to twenty-five percent over solutions generated under uniform bandwidth constraints.

Genetic algorithms

More recently Genetic algorithms (heuristic search procedures based loosely on the mechanics of natural selection and natural genetics) have been developed to determine (near-) optimal signal timings for a network. The basic components of genetic algorithms include a binary string representation of alternative solutions (population members), an object function to evaluate the alternative solution (gauge the "fitness" of population members) and Genetic operators that mimic the biological evolution process (evolution from generation to generation of solutions). (Hadi and Wallace, 1998; Karaboga and Pham, 1998) Genetic algorithm minimally utilized three basic "genetic" operators to manipulate members of a population over several generations in an attempt to improve their fitness. These operators are, reproduction (sometimes referred to as selection), which selects individuals with higher fitness, crossover, which creates members of the next general from the current generation, and mutation, which allows for exploration into unexplored areas of the solution space. Major factors in the successful application of any genetic algorithm include an efficient and complete translation of the problem into the required binary code, development of initial populations and calibration of the functioning of the genetic operators. (Karaboga and Pham, 1998)

The primary advantage typically offered by the algorithms is that they provide optimal phase sequences, cycle lengths, green, splits and offsets. Most of the other available algorithms do not optimize all four of these elements, requiring the use of several programs in conjunction in the development of optimal timings. These algorithms have also claimed to find some success in provided timings superior to that of other methods (TRANSYT-7f) for oversaturated conditions. (Park, Messer, and Urbank, 1999, Park, Messer, and Urbank, 2000) Some of the most recent work in the application of Genetic algorithms to signal systems have been performed by Park, Messer and UrBanik (1999; 2000), Hadi and Wallace (1998), and Pham and Karaboga (2000).

Other Off-line Software

While TRANSYT-7f is probably the most utilized model numerous other models have been developed, each with its own strengths and weaknesses, such as offering microscopic simulation, phasing plan optimization, incorporating stochastic arrivals, allowing multiple time periods, bandwidth optimization, improved user interfaces, etc. A few of the traffic simulation and/or optimization models for isolated and/or multiple intersections that have been developed are, TEXAS(Lee, Rioux, and Copeland, 1977; Lee, Grayson, and Copeland, 1977), CAPCAL(May, 1990; Gordon, Reiss, Haenel, Case, French, Mohaddes, and Wolcott, 1996), SOAP(May, 1990; Gordon, Reiss, Haenel, Case, French, Mohaddes, and Wolcott, 1996), CORSIM(Wang and Prevedourous, 1998; Mystkowski and Sarosh, 1999), MAXBAND(Chaudhary, Pinnoi, and Messer, 1991; Cohen, 1983), PASSER II(Malakapalli and Messer, 1993; Mystkowski and Sarosh, 1999), PASSER IV(Chaudhary and Messer, 1993; Liu, Chaudhary, Simeonidis, and Sirigiri, 1995), SIGOP(May, 1990; Gordon, Reiss, Haenel, Case, French, Mohaddes, and Wolcott, 1996), etc. Clearly, it can be seen that a great deal of effort has been spent in developing different off-line approaches to traffic modeling and signal optimization. Often users attempt to overcome the weakness of one model (such as lack of optimization of phase sequences in TRANSYT-7f) by using several models in conjunction to arrive at final signal timing plans (Rogness and Messer, 1983; Cohen, 1983).

ADAPTIVE CONTROL

There exists no Webster's dictionary type definition for the term adaptive control. Loosely speaking any signal timing strategy that attempts to account for real-time traffic fluctuations is usually categorized as adaptive. Notably this definition could be interpreted to include actuated control although, in general, actuated control is considered as a separate category, often one of the yard sticks by which adaptive control is measured. Lin (1988) attempts to clarify this issue by defining adaptive control as any strategy that accounts for traffic fluctuations and provides service superior to that of actuated control.

Categorization of Adaptive Control Strategies

Adaptive control strategies have been part of the traffic engineering toolbox for nearly forty years. The following discussion presents some of the categorizations (Binary / Sequencing, Centralized / Decentralized / Hierarchical, and Cyclic / Acyclic) of adaptive control that have been suggested and provides an excellent method by which some of the main features of adaptive control may be introduced.

Binary Choice vs. Sequencing Approach

One of the earliest adaptive control categorizations was that by Lin (1988) in which he divided strategies into Binary Choice and Sequencing Approaches. In a binary choice approach, time is divided into successive small intervals, on the order of two seconds, where within each interval a decision is made to extend or terminate the green. Binary approaches tend to consider small time horizons, up to 10 seconds. By limiting the time horizon these strategies attempt to avoid the difficulties of predicting future traffic. Lin presented several isolated intersection examples of this type of control logic, including, TOL by Bang, 1976, Millers algorithm by de la Breteque and Jezequel, MOVA logic by Vincent and Young, and SAST logic. SCOOT and SCATS are more recent network examples, although they incorporate certain network level constraints to account for the lack of a long planning horizon.

A adaptive control strategy that uses a sequencing approach attempts to determine the optimal signal switching pattern over a longer future period. This involves traffic demand predications not required by the binary approach. To account for the uncertainty in traffic predications most sequencing approach based strategies do not implement the determined timing for the entire horizon but instead for just the first few seconds. The optimization is then repeated for the next time horizon (the old shifted by the implemented seconds) with updated detector data and arrival predications. This implementation strategy is often referred to as a rolling horizon.

At the time of Lin's discussion OPAC was one of the only strategies to fall within this category, although many of the more recently developed strategies fall under the sequencing approach. As the scope of adaptive control has been expanded from local intersection control the network level many researchers have become convinced of the need for traffic predictions and signal timing optimization over longer time horizons. The longer time horizon, and thus increased system data, would seem to present the possibility for the selection of better, more pro-active, timing plans. Unfortunately the difficulties inherent in obtaining accurate predictions over extended time periods have lead in some situations to sub-optimal timings and progressively deteriorating system measures.

Centralized, Hierarchical and Decentralized Control

Another prominent feature utilized for adaptive control classification is whether the strategy follows a centralized, hierarchical or decentralized control architecture (Yagar and Dion, 1996).

In centralized adaptive control systems all timing calculations and decision making occurs at the central computer level. The tasks of the local controllers is to implement the timings from, and gather data to be sent back to, the central computer. The centralized control strategy attempts to optimize a global objective function (total delay, total stops, average queues, etc.). SCOOT is an excellent example of a centralized adaptive control strategy.

Probably the greatest problem encountered in most centralized strategies is that of dimensionality. As the network size grows the required computational effort increases rapidly, placing a serious limitation on the network size that may be handle on a real-time basis. Even on a relatively small system it is difficult to guarantee finding a globally optimal solution in real-time. To operate within a feasible time frame on even moderately sized networks centralized strategies are forced to incorporate heuristic methods with a goal obtaining a near-optimal solution.

In the distributed system control architecture all calculations are performed at the individual intersections. There are no network level coordination considerations. Many distributed control systems due incorporate nearest neighbor communication, allowing for the sharing of data, including predicted future phase changes, detector actuations, traffic flows, etc. One distinct advantage of a distributed system is that with the system divided into a set of subproblems (intersection level optimizations) the dimensionality drawback inherent in centralized systems is avoided. Although in gaining this advantage the strategy loses the ability to apply control schemes that consider network wide implications. OPAC, GASCAP and SPPORT are examples of distributed control systems.

Hierarchical control is a combination of central and distributed control. Signal timing optimization calculations occur at both the central and local level. Hierarchical control strategies try to incorporate the "best" features of the centralized and decentralized control architectures. By maintaining some level of central control network wide traffic operations may be taken into account but by allowing some optimization to be performed at the local level the dimensionality problems may be reduced. For example, at the central level offsets and phase orders may be determined but and at the local level green splits may be adjusted based on the most recent traffic data. SCATS and RHODES are two examples of hierarchical control systems.

Cyclic vs. Acylic

The final adaptive control feature to be discussed is cyclic versus acyclic optimization (Conrad, Dion, and Yager, 1998). Strategies that utilize cyclic optimization continue to use the descriptive constructs familiar to traditional signal timing; cycle lengths, splits and offsets. To achieve the desired objective (saturation flow level, bandwidth maximization, delay minimization, etc) these strategies allow for small, frequent adjustments, in real-time, to the signal timing parameters to account for fluctuations in current traffic conditions. Cyclic optimization strategies are often centralized strategies with a system wide objective function measure. SCOOT is an excellent example of the cyclic optimization approach.

Acyclic optimization approaches abandon the concepts of cycle lengths, splits and offsets. Phasing sequences are developed for some fixed time horizon, typically in terms of signal switching patterns, without considering cycle or split constraints. As with cyclic optimization the objective function is may be based on delay, saturation flows, etc. OPAC and GASCAP are excellent examples of acyclic optimization approaches.

Presentation of specific adaptive control strategies

The following provides a presentation of some of the existing and proposed adaptive control strategies. While a discussion is not offered on every strategy developed to date the cross-section of strategies presented encompasses most of the features and approaches found in the adaptive control field.

SCOOT (Bretherton, 1996; Rakha et. al. 1995; Gordon et. al. 1996; Robertson et. al., 1991)

One of the better know centralized adaptive control models in use is SCOOT (Split, Cycle, and Offset Optimization Technique), which was developed by the Transport and Road Research Laboratory (TRRL). SCOOT performs a real-time incremental optimization of signal settings utilizing a traffic simulator similar to that of TRANSYT. To perform the optimization SCOOT computes a cyclic traffic flow profile (CFP, a record of vehicle platoons) every four seconds, for every link, based on upstream detector measurements. SCOOT then utilizes the TRANSYT dispersion model to project these profiles downstream, making optimal decisions based on the chosen performance index (PI). During the optimization process user defined weights (preferences) may be given to specific links or routes. Typically the PI will include bandwidth, average queues (which may be translated into delay) or vehicle stops (Bretherton, 1996). Also, SCOOT may make adjustments in an attempt to keep the maximum saturation on the heaviest phase below ninety percent. (Traffic demand in SCOOT in not actually measured in vehicles, but rather, in Link Profile Units, LPU's, which is a hybrid measure of traffic flow and detector occupancy. On average 17 LPU's are equivalent to one vehicle but this may vary by as much as 50% between sites. [Hounsel, McLoed, and Burton, 1990])

The SCOOT signal optimizer essentially uses an elastic coordination plan that stretches and shrinks in an incremental fashion to reflect the current traffic conditions (Rakha and Van Aerde, 1995). Before each scheduled phase SCOOT determines if it is better to advance, retard or leave as is the scheduled phase change. Also, during each cycle SCOOT will decide if the offset should be advanced, retarded, or unaltered, based on the given PI (Gordon, Reiss, Haenel, Case, French, Mohaddes, and Wolcott, 1996). These changes lead to a traffic plan that evolves over time, as oppose to time based system that requires a sharp switching from one plan to another under pre-specified conditions. Earlier versions of SCOOT limited the incremental changes to 4 seconds but more recent versions are more flexible allowing system operators to select larger maximum change intervals (Bretherton, 1996).

Generally, SCOOT is expected to provide the most benefit in situations where vehicular flows are heavy, complex and vary unpredictably (Robertson and Bertherton, 1991). SCOOT has been shown to achieve reductions in network travel times in the range of 10 percent, when compared to conventional fixed time control (Rakha and Van Aerde, 1995). SCOOT is constantly be upgraded and enhanced, with a few of the improvements to more recent versions including incorporating logic for congested and saturated conditions, support for bus priority, an automatic SCOOT traffic information data base, an incident detection system, gating, vehicle emission estimates and bicycle logic (Bretherton and Bowen, 1990; 1996; Bretherton, 1996).

SCATS (Sim se. at., 1979; Luk, 1984; Nguyen, 1996)

The Sydney Co-ordinated Adaptive Traffic Control System was developed by the Roads and Traffic Authority (RTA) of New South Wales, Australia in the late 1970's. SCATS is a hierarchical, cyclic, traffic responsive signal control strategy. The system operators may choose between combinations of different goals including, minimize stops, minimize delay and maximize throughput. SCATS, similar to SCOOT, bases all optimization on current traffic information and does not utilize prediction over a long future time horizon. Instead SCATS (and SCOOT) rely on the implementation of frequent, small changes to the timing plans (cycle length, splits and offsets) in order to compensate for the fluctuations in real-time traffic (Luk, 1984). In the SCATS approach the overall network under control is divided into numerous subsystems with optimization decisions made at the network, subsystem and individual intersection level.

At the network level offsets between intersections in different subsystems are set and subsystems may be selected to "marry" or "divorce". Intersection membership in subsystems is an operator input. At the subsystem level each subsystem selects a critical intersection (based on a ratio of saturation flow to detected flow at the stop line) upon which the subsystem common cycle length is based. Also at the subsystem level offsets and phase sequences for the intersections within that subsystem are selected. At the individual intersections a vehicle actuation logic is utilized to make small phase split length adjustments on a cycle-by-cycle basis in response to the most recent stop line traffic flow information. In this optimization phase skipping is allowed.

Studies performed by the SCATS developers have shown improvement in delays and travel times (as high has 40% reduction in travel times when compared to optimized fixed times) although other tests have found SCATS to provide results comparable to that of optimized fixed timings (Luk, 1984). One simulation study performed by Wolshon et. al. (1999) undertook a detailed analysis of the impact of SCATS signal timing control. For the system tested the SCATS logic was set to equalize saturation on all approaches. Wolshon et. al. observed that SCATS more evenly distributed the delay over all approaches than fixed-time strategies although in some cases this resulted in an overall increase in total intersection delay. In these situations minor and left turn movements would experience decreased delay while major street movements would experience increased delay. Although, Wolshon does state that the simulation model utilized for the test may not have adequately accounted for platooning due to upstream intersections, so these results should be interpreted with some caution.

RHODES (Mirchandani et. al., 1998; Head et. al.; Dell 'Olmo et. al., 1991)

One of the more recent approaches to real-time traffic adaptive control for a network RHODES developed as part of the RT-TRACS effort (to be discussed). The RHODES strategy is a acyclic three level hierarchical approach, consisting of network, arterial and intersection levels. The RHODES system involves traffic flow predictions, at increasing levels of detail (platoons to individual vehicles), at each hierarchical level. (PREDICT algorithm (Head, 1998), TURN algorithm(Mirchandani and Head, 1998) and APRES-NET model(Dell 'Olmo and Mirchandani)) The optimization criteria can be any desired by the system operators as long as it may be expresses in traffic measures of effectiveness (average delays, stops, throughput, etc).

To date extensive development and implementation tests have occurred on the Arterial level logic (REALBAND) and intersection logic (COP).

REALBAND, developed by Dell'Olmo and Mirchandani, begins by identifying platoons and predicting their sizes, speeds, and arrival times at downstream intersections over a 200 to 300 second future time horizon. The signals are then set such that given the predicted platoon propagation over some time horizon the desired objective function may be optimized. A rolling horizon strategy used to allow for real-time updating of traffic information and subsequently the optimal signal timings.

One of the main objectives of REALBAND is to optimally resolve conflicts created by two platoons demanding conflicting movements at the same time at an intersection. This optimization is currently completed through the development of a decision node tree. Branches are created which account for the possibility of the different conflicting platoons receiving the green and also for the potential of splitting a platoon. The path on the decision tree that optimizes the given performance criterion is chosen as the initial solution. Optimization may then be performed at the intersection level as more detailed information is gathered, subject to the constraints from the REALBAND solution. This optimization uses the COP (Sen and Head, 1997) optimization procedure, which is a forward based dynamic optimization procedure. The COP optimization also uses a rolling horizon to predict arrivals and queues at each intersection although the horizon utilized is shorter than that of REALBAND, on the order or 20 to forty seconds. The optimization does not necessarily require a predetermined phase sequences but a sequence may be added as a constraint if desired by the system operators. (COP is implemented into Open-TS3 and described in detail in Chapter 6.)

The main advantage of RHODES approach claimed by Dell'Olmo and Mirchandani is that it uses real-time data (collected from existing surface street detectors) to explicitly identify platoons and predict and pro-actively plan their movements, with increased benefits are reaped by the multi-level optimization approach. Laboratory tests have shown significant reductions in both average vehicle delays and variance of delay. The authors do note that the current version of REALBAND is computationally intense in that it involves completely building each possible branch of the decision tree. Currently they are conducting research with an aim of creating an effective method to prune the decision tree, leading to a more efficient and faster model.

OPAC (Gartner et. al., 1991; Gartner et. al., 1995; Chen et. al., 1987)

OPAC (Optimization Policies for Adaptive Control) is one of the original signal-timing strategies to utilize a rolling horizon in an attempt to achieve an optimal solution in a manner that is not an extension of traditional strategies (i.e. pre-timed or semi-actuated control) but a new problem approach. Prior to OPAC most of the existing real-time signal system control strategies were creative implementations of off-line signal timing control concepts, with the "real time" adjustments for traffic fluctuations running several cycles or time intervals behind the actual fluctuation.

According to Gartner et. al. OPAC was developed to calculate, in real-time, near-optimal signal controls using on-line data that is typically readily available from upstream detectors. In the earliest papers on this topic Gartner foresaw the potential for OPAC to form a building block for demand-responsive decentralized network control. Gartner et. al has written that a currently under development OPAC networking module will be "capable of providing real-time, traffic adaptive control for signal networks that combines the advantages of distributed cycle-free optimization at the local level with system-wide coordination at the network level." This model, is referred to as VFC-OPAC (Virtual Fixed Cycle OPAC)(Gartner, Stamatiadis, and Tarnoff, 1995)

The OPAC logic is considered to be based on a pseudo-Dynamic Programming technique, thus, OPAC finds a near-optimal solution and not necessarily an optimal solution(Chen, Cohen, Gartner, and Liu, 1987). This pseudo-DP version was developed by Gartner to overcome difficulties with the originally developed DP version. In Simulation Study of OPAC: A Demand-Responsive strategy for Traffic Signal Control (Chen, Cohen, Gartner, and Liu, 1987) Gartner writes that this pseudo-DP version of OPAC has the following features:

5) The optimization process for a control period is divided into stages of T-seconds. The stage length should be in the range of 50-100 seconds (i.e. similar to a cycle for a fixed-time traffic signal) and consist of an integer number (k) on the basic time intervals, or steps (usually, T=5 seconds each).
6) During each stage, at least one signal change (switchover) is required and up to three switchovers are allowed. This is designed to provide sufficient flexibility for deriving an optimal demand-responsive policy, i.e. one that would cover all feasible situations.
7) For any given switching sequence (t_1, t_2, t_3) during a stage n, we calculate a performance index P_n on each approach consisting of the total delay during the stage (in vehicle steps);

$$P_n(t_1,t_2,t_3) = \mathrm{SUM}(Q_0 + A_i - D_i)$$

where Q_0 = initial queue
A_i = arrivals during step i
D_i = departures during step I

8) The optimal procedure used for obtaining the optimal switching sequence is an Optimal Sequential Constrained search method (OSCO for short). The objective function which is the sum of the performance indices on all approaches (total delay) is evaluated sequentially for all feasible switching sequences. At each iteration, the current objective function is compared with the previously stored value and, if lower, supersedes it. The corresponding switching points and the terminal queue-length are also stored. At the end of the search, the values in storage are the optimal solution.

The preceding is the basic optimization procedure used during a single stage of the OPAC procedure. A recognized drawback to this procedure is the need for an accurate prediction of traffic flow over the optimization period. To lessen the impact of this problem Gartner further supplemented this optimization procedure to allow for an updating of traffic data and a recalculation of the near-optimal timings at a much shorter time interval than that being optimized. This strategy, typically referred to as a rolling horizon strategy, will be presented in Chapter 6. Gartner was one of the first to utilize a rolling horizon approach, which has since found wide spread application in advanced transportation control.

SPPORT (Han et. al., 1994; Conrad et. al., 1998; Yagar and Dion, 1996)

Signal Priority Procedure for Optimization in Real Time (SPPORT) is a rule-based logic originally developed for control at an isolated intersection. A major goal of SPPORT is to accurately represent, and develop optimal signal timings for, discrete non-cyclical events, such as temporary lane blockage due to bus on/off loading. Through the incorporation of traffic-transit interaction the developers argue that SPPORT more accurately reflects real-time traffic flow and is able to minimize costs and delays for car and transit passengers.

As with many of the other strategies SPPORT utilizes a rolling horizon approach, updating the optimization analysis at specified time intervals (5 seconds in initial strategy development), continually implementing the best phasing. SPPORT is different though from most other strategies that utilize a rolling horizon in that instead of dividing time into a finite number of steps it models discrete time events directly.

In SPPORT an optimization rule is essentially a set of preferences given to different possible traffic events. The first member of the list receives the highest priority, the second the next highest, and so on. An example of an ordered list could be,

1) An emergency vehicle is present.
2) A queue is served at full saturation on all green approaches.
3) A queue is served at full saturation on one green approach.
4) A bus arrives or is leaving.
5) A queue is served at below saturation flow on all approaches.
6) A bus is leaving.

A phasing sequence is developed based on the current intersection traffic, a projection of events 60 to 90 second into the future, minimum and maximum phase length constraints and the rule set. That is, for the given example a phasing sequence is developed which places the highest priority on serving emergency vehicles (if present or predicted), then on serving queues at full saturation on all approaches, then on serving a queue at full saturation on one approach, etc.

SPPORT may consider many different rule sets, each developed off-line with some particular goal; maximum weight given to transit, transit vehicles ignored, priority service to queues, etc. SPPORT simulates the phasing sequences developed for each rule set and selects the timing strategy that minimizes the overall objective function (for example a function of cost and delay). In addition to the events in the above example prioritized events include, a queue exceeds maximum allowed queue, a streetcar arrives, a streetcar is loading, a street car finishes loading, a bus is loading, and a queue is requested.

SPPORT has enjoyed some success with isolated controllers, performing well in a simulation experiment of a transit-dominated area in Toronto. Based on these successes SPPORT is being expanded to a network version, utilizing a distributed control architecture. The network version will utilize information from neighboring intersections, improved queue modeling and consideration of spillback rules with an ultimate goal of developing timings that are optimal at the local intersections and preferred by neighboring intersections.

Other researchers have also worked towards developing adaptive control strategies that incorporate transit. Chang et. al. (Chang, Vasudevan, and Su, 1995) is one example that begins a study of integrating bus preemption and signal control functions in an adaptive control system. UTOPIA (Lin and Vijayakumar, 1988), is an early example of a system that incorporates transit preemption.

PRODYN (Farges, et. al., 1990)

PRODYN utilizes an acyclic, distributed control architecture and is designed to control either an intersection or a network. PROYDN also incorporates a rolling horizon scheme in which switching decisions are made in 5 second step intervals. During each step, based on currently recorded upstream detector actuations, vehicle arrivals at the stop line (vertical queue model) are predicted over a 75 second future time horizon. Optimal phase switches for the future time horizon are then developed based on these predicted arrivals, initial queues and any timing constraints (min/max greens, etc). The first step (5 seconds) is implemented and the algorithm then repeats the predictions and optimization over the new (5 second advancement) time horizon with updated data (i.e. rolling horizon implementation). In the network application the development of arrival predictions also takes into account predicted departures from neighboring intersections.

Optimization is based on a delay minimization over the time horizon, including a termination criterion that estimates delays associated to the state at the end of the horizon. The optimization is performed in a forward dynamic programming procedure, similar to the approach of the COP algorithm in RHODES.

Tests of the PRODYN system at isolated intersections have shown it to provide superior performance (delay and stops) under peak hour traffic conditions than traditional fixed time or actuated control timing strategies. Under some off-peak conditions PRODYN was found to not to be quite as efficient as fixed-time strategies. Under simulated network operation PROYDN provided superior performance over all time periods, with a network wide measured decrease in delay of 12%.

MOVA (Vincent et. al., 1986)

One of the earliest attempts at adaptive control <u>M</u>odernised <u>O</u>ptimised <u>V</u>ehicle <u>A</u>ctuation is an isolated intersection strategy developed at TRRL. This strategy employs two detectors per approach lane (placed at 40 and 100 meters) in an attempt to reduce the tendency of many actuated controllers to extend green time when traffic is flowing at a rate well below saturation flow. A lanes initial green will be that required to release vehicles between the stopline and 40m setback detector. After this initial green headways are examined to determine if a reduction in saturation flow has occurred. Based on the most recent flow levels and arrivals the MOVA logic determines whether it is more beneficial to continue extending the green or switch the green to a currently red approach. This is an extension to adaptive control work perform by Miller (as reference in Lin, 1988). MOVA exemplifies the tendencies of many of the earlier strategies to optimize over a very short time horizon, essentially trying to tweak the actuated control concept.

GASCAP (Owen et. al., 1999)

<u>G</u>eneralized <u>A</u>daptive <u>S</u>ignal <u>C</u>ontrol <u>A</u>lgorithm <u>P</u>roject (GASCAP) is a distributed, rule based adaptive control approach that utilizes queue estimates along with a set of rules to determine signal changes at each intersection. GASCAP incorporates only limited communication between nearest neighbor intersections. In this control scheme there is not an attempt to optimize any network-wide or intersection level objective functions.

The GASCAP Methodology utilizes a queue estimation algorithm that predicts the number of queued (stopped) vehicles on an approach, vehicles approaching the queue and gaps for permitted left turners. To determine a control strategy that best suits the predicted queue information GASCAP utilized one of two different strategies. The first is a set of rules for uncongested control and the second a near real-time development of fixed time plans for congested control.

In uncongested control five rule sets are applied to determine intersection signal switches. These rule sets are 1) demand rules, which corresponds to isolated intersection control, 2) progression rules, which consider arrival patterns due to signal plans of adjacent intersections, 3) urgency rules, which address intersection approaches that are nearing saturation 4) cooperative rules, which allow neighboring intersections to cooperate when nearing saturated conditions (i.e. handling spillback, etc.) and 5) safety rules, which account for minimum green, conflicting movements, etc. Each rule set submits a recommendation for the next movement to an event list. GASCAP then assign a priority to these movements with the highest priority typically being selected (safety rules are never violated, regardless of assigned priority).

Under congested conditions GASCAP utilizes information from upstream detectors to develop a fixed timing plan. The plan development is based on detector occupancy measurements over the previous fifteen minutes, converted into volumes. The basic approach is to select a fixed cycle length, adjusted every other cycle, where splits and offsets are based on previous volumes.

GASCAP performance was compared against that of RHODES, OPAC and a baseline condition on for three different scenarios. Reported simulation testing (Owen and Stallard, 1999) stated GASCAP control resulted in system delays comparable to that of RHODES and superior to OPAC and the baseline case.

GASCAP offers a very interesting and intelligent approach to signal control. The primary concern thought not address by GASCAP and other fully decentralized control architectures is potential for system wide degradation over a period of time. It would seem that without any long term system considerations it is possible for GASCAP to developed a progressive set of timings where the overall performance begins to deteriorate as was experience by many of the advanced systems in the 1970's (Pooran, Tarnoff, and Kalaputapu). While this was not experienced in the three tests it is not clear that this possibility is being guarded against.

Others have also worked with developing distributed, rule-based approaches to signal optimization. Findler and Stapp (1992) and SPPORT (Han and Yagar, 1994; Conrad, Dion, and Yagar, 1998) provide other examples of such an approach.

LOCAL

The LOCAL model (Memon and Bullen, 1996) was developed as part of the initial efforts in the RT-TRACS (Pooran, Tarnoff, and Kalaputapu) development. In the development of the LOCAL model Memon and Bullen compare utilizing a Quasi-Newton Methodology with a Genetic Algorithm approach to real-time operation. The genetic algorithm approach is similar to the off-line approaches previously discussed (Hadi and Wallace, 1998; Park, Messer, and Urbank, 1999; Park, Messer, and Urbank, 2000). Memon and Bullen state that their selection of a genetic algorithmic approach to real-time control was based on genetic algorithms ability to find an optimal solution in an efficient manner, the robustness of the approach, that genetic algorithms can handle the multivariate nature of the real-time traffic control problems, that genetic algorithms may be easily hybridized and that genetic algorithms are domain and problem independent during the search procedure. Memon and Bullen found that the results obtained from the genetic algorithm approach where comparable to the much more computational intense Quasi-Newton approach. The genetic algorithm required less computation time, fewer lines of code and was more easily implemented. While neither optimization technique tested clearly achieved better results than the other it was felt that the computational aspects of the genetic approach lead to a conclusion that is seemed to be a feasible option for real-time control.

Neural Networks

Another potentially promising control method may be found in Neural Network Models. Loosely speaking a neural network model creates artificial neurons that emulate the mechanism of biological neurons. The process of generating signal controls is then split into two processes, model training and model optimization (Nakatsuji and Kaku, 1991). Nakatsuji and Kaku (1991) begin the development of a multi-layered neural network model, which has the ability to self organize a traffic control system. Nakatsuji and Kaku state that neural computers have the potential to compute in parallel, learn from past experience and avoid entrapment in local minimums and thus should serve well in the control of traffic systems. While rather complicated the model developed by Nakatsuji and Kaku is only concerned with the optimization of signal phase splits. In this work many simplifying assumptions are still being made including pre-determined common cycle length and no offsets between intersections. At this time efforts in Neural Network control is in the early stages and it is difficult to gauge the overall promise of such strategies.

Other

A few other adaptive control strategies are briefly discussed as they offer a different approach or twist not seen in the previously discussed strategies.

Kwon and Stephanedes (1998) developed an isolated intersection control method based on a link congestion index that quantifies a link-wide level of congestion. The level of congestion on a link is quantified on a scale from 0 to 1 based on detectors throughout the link. The basic phase switching rule (subject to minimum, maximums and fixed phase sequence) is to switch to the next phase when the next phase's congestion index is equal or greater than the current phase's congestion index. In simulation tests this strategy did provide performance (reduction in total vehicle hours) superior to that of pretimed and actuated control.

Elahi, Radwan and Goul (1991, 1992) also worked on the development of an adaptive control system for isolated intersections. Their basic strategy was to develop a knowledge-based control system, that is, a system that continually updates its knowledge base and adapts to variations in traffic flow through the use of an inference engine. Basically, the proposed strategy is that at the end of each cycle a level of service is determined for each approach and the intersection (based on Highway Capacity Manual (137) equations), if these meet or exceed user defined thresholds the existing timing is continued, if they fall below the thresholds the traffic for the next cycle is forecasted based on current traffic and a volume database (knowledge base). A new set of signal timings is then developed, possibly including switching control types, i.e. actuated to pretimed. At the end of each cycle the volume database is updated with any new trends now reflected in the database. In a limited number of simulation tests the strategy lead to decreases queues and delays when compared to pretimed and actuated control. Others have also considered knowledge-based systems, for example PHAST, developed by Linkenheld et. al. (1992)

Bullock and Shoup (1999) and Bullock and Catarella (1998) develop an off-line procedure whereby they utilize link travel times to set offsets for a coordinated arterial signal system. The main goals of this research were to alleviate the early return to green and downstream queue problems that often affect the coordination of (semi-) actuated signals.

Essentially the authors develop a strategy whereby semi-actuated controller offsets may be set to account for signal cycles where actuated approaches typically do not utilize the entire maximum green time resulting in this time defaulting to the main-line coordinated approach. The authors believe that this work lays the foundation for the development of an on-line adaptive offset tuning methodology based on real-time travel data.

Abu-Lebdeh and Benekohal (1997) developed a procedure for signal control on oversatured arterials that incorporated Genetic Algorithms. Their basic premise is that of dynamic signal coordination based on queue management and efficient green time utilization. The problem is formulated to achieve through-put maximization and solved utilizing Genetic Algorithms. While still in the earlier stages of development the authors stated that based on initial results the approached seemed to have good potential for on-line real-time implementation.

Relijic (1996) attempts to solve for network signal timings based on a multicriteria problem statement, that is, there is an attempted to simultaneously minimize several criteria, such as average delay, average number of stops, average fuel consumption, etc. Relijic solves this problem using a multicriteria objective dynamic programming method that obtains a set of Pareto optimal solutions. Relijic demonstrates that a multicriteria problem statement may be developed which is separable and non-linear may be constructed and solved utilizing a multicriteria objective programming approach. The solution to such a multicriteria problem is also shown to not be unique, leading to the concept of Pareto minimum. Thus, in this approach not one but many dynamic programming problems must be solved. While Relijic was successful in the development of a solution for small network it was recognized that that as the network size increase the solution method quickly suffers from the problem of dimensionality. That is, the computational effort increases so rapidly with the increase in network size that the solution method becomes unrealistic, to the extend the Relijic states that future endeavors will center on developing alternate approaches to the development of the Pareto optimal solution set.

Finally, Hadi and Wallace (1994) incorporated Cauchy Simulated Annealing into a TRANSYT-7f optimization in order to include phase sequence into the overall optimization they noticed that although through different methods (PASSER II, Simulated Annealing with varying parameter) they were able to develop optimized solutions with similar bandwidth and progression opportunities values (the studied objective function) with phase sequences, cycles and offsets that differed significantly. Their observations suggest that at least multiple progression-based solutions close to the optimal solution can exist.

Signal Control Implementation Issues - RT-TRACS Control Strategy

As seen signal control, from pre-timed to real-time, has been an active research area. Unfortunately the record of success in actual implementation of advanced control strategies has been mixed to poor. One discussion on this may be found in *RT-TRACS: Development of the Real-Time Control Logic* (Pooran, Tarnoff, and Kalaputapu). RT-TRACS (<u>R</u>eal-<u>T</u>imed <u>T</u>raffic <u>A</u>daptive <u>C</u>ontrol <u>S</u>ystem) is a real-timed traffic control system that is being designed for use in an Intelligent Transportation System (ITS). RC-TRACS does not utilize a single control strategy but instead draws from a suite of strategies, from fixed timed plans to adaptive control, in an attempt to overcome the drawbacks of previous "advanced " control strategies. Tarnoff and Gartner refer to a system planned to be implemented by 1997 but to date no published findings from such a trial have been found. As part of the RT-TRACS development FHWA funded efforts by various researchers and consultants to develop real-time traffic adaptive signal control strategies, with those strategies deemed to meet the functional specification requirements to be incorporated into RT-TRACS.

To understand the implementation difficulties of advanced system control that have occurred to date one should first become familiar with the different levels of system control. As part of the development of signal system control the U.S. Department of Transportation created the UTCS (Urban Traffic Control System) research project in which levels of system control were defined. UTCS control strategies are categorized as first, second or third generation. First generation control (1-GC) utilizes signal timing plans calculated off-line (i.e. with TRANSYT-7f, etc.), based on historical traffic data, which are then pre-stored in the controllers. First generation plans do not involve real-time traffic predication. The plan in use is updated typically no more than every fifteen minutes based on time-of-day, traffic responsive or manual selection. While cycle length is fixed through a section, some fine tuning of splits may occur at the intersections.

Second generation control strategies (2-GC) typically perform an on-line optimization where traffic predications are historically based while possibly accounting for current trends. Signal timings are updated as often as every five minutes. As with the first generation strategies, a common background cycle is used throughout a section, although the grouping of signals is a variable parameter based on prevailing traffic conditions. Additional fine tuning of splits and offsets may also occur during each cycle.

Third generation models (3-GC) include strategies that are fully responsive, providing on-line signal control. A third generation system may update signal timings every three to five minutes with cycle lengths variable in both time and space, i.e. the cycle may vary both among different intersections and at the same intersection during a control period. Traffic predications may also be historically based accounting for current trends through the utilization of some type of smoothed values.

While the UTCS project was begun in 1967 the potential of the advanced control strategies found in the second and third generation systems (adaptive control falling into the third generation) has not been realized. Many of the systems that have been implemented to date fall into the first generation (or 1 ½ generation). An impression exists among many transportation professional that a first generation system will provide better operations than a second or third generation system. A number of potential reasons for the current lack of acceptance of adaptive control strategies by U.S. engineering community are discussed by Tarnoff and Gartner (1995), These include funding constraints, lack of up to date information by those who implement the systems, lack of evidence that these systems actually work and justify additional costs, difficulties in using these systems, failure of systems under congested situations and inadequate understanding of limitations of real-time systems developed in the past.

Much of the Current belief in the superiority in first generation systems may be traced to the failure of second and third generation systems developed and tested primarily in the 1970's. Gartner et al. (1995) argues that this poor performance of real-time signal strategies should not be interpreted as a failure of real-time control. Instead it is a result of the inadequacies in some of the real-time methodologies previously attempted and a lack of understanding and flexibility in implementing the control strategies. While a more advanced system may potentially offer a considerable improvement it may also severely degrade the system. Failure to recognize both the strengths and weaknesses of the different strategies, and in which traffic flow regimes a strategy works well, and in which it does not, has led to the avoidance of such systems.

Before, and as part of the RT-TRACS effort, adaptive control strategies, such as OPAC, PRODYN, and UTOPIA, have been (and are still being) developed. Detailed evaluation and observation have led to the realization that no one strategy is more effective than all the other under all traffic conditions, Different strategies may work better in different geographic areas, at different times of day, under various traffic demands (under saturated, saturated and congested), even on different parts of a single system at the same time. Different conditions lead to a varying performance by the different operating strategies.

As mentioned the RT-TRACS concept is to develop a suite of control strategies and logic to determine which strategies are best for a particular traffic condition, allowing different control strategies to work in parallel, providing optimal control on different sections of a network. RT-TRACS not only incorporates real-time control strategies but also includes and acknowledges that in some situations fixed time control or a centralized architecture similar to that of SCOOT may best serve traffic. Traffic predication and strategy selections are to be based on a Case Based Reasoning (CBR) system being developed for RT-TRACS, Essentially, the CBR will aid in solving new coordination problems by finding and adapting solved problems similar to the current conditions. RT-TRACS will be able to utilize stored plans, perform on-line updates to store plans, utilize real-time control with fixed cycle lengths, incorporate fully responsive real-time control and implement congestion control strategies.

CONCLUSION

This chapter covered current and past signal control and coordination strategies, including a review of pre-timed, actuated and some of the many existing and proposed real-time adaptive control methodologies. In the previous chapters it was seen that Open-TS3 is capable of modeling pre-timed and actuated control. In Chapter 7 it will be shown that Open-TS3 is robust and flexible enough to incorporate adaptive signal control.

CHAPTER 6 IMPLEMENTATION OF REAL-TIME ADAPTIVE SIGNAL CONTROL INTO OPEN-TS3

INTRODUCTION

Thus far in this report, simulation and traffic control have been reviewed and the development of Open-TS3. In this chapter an initial real-time adaptive control study is undertaken. The adaptive control strategy presented within this chapter is an extension of COP (Controlled Optimization of Phases at an Intersection, Sen and Head, 1997), which was developed as part of the RHODES effort (Sen and Head, 1997).

Presented in this chapter is a brief discussion of dynamic programming, followed by a presentation of the rolling horizon concept. After this background, COP will be presented along with a developed extension to COP signal control. The implementation of COP adaptive control into Open-TS3 is then discussed along with results for a few sample networks where the adaptive control strategies were implemented.

DYNAMIC PROGRAMMING

Dynamic programming (DP) forms the foundation of the COP effort. Thus, prior to presentation of the COP based real-time control strategy, a brief synopsis of dynamic programming is offered. For a more in-depth discussion of dynamic programming the reader is referred to Dreyfus and Law, The Art and Theory of Dynamic Programming (191).

Features common to all dynamic programming procedures are effectively summarized by Dreyfus and Law (191) as:

> …. first, the recognition that a given "whole problem" can be solved if the values of the best solutions of certain subproblems can be determined (the principle of optimality); and secondly, the realization that if one starts at or near the end of the "whole problem," the subproblems are so simple as to have trivial solutions.

These concepts may be readily understood in the context of a shortest path problem. Consider an optimal path from point A to point D where point B is on the path between A and D. From the principle of optimality it is known that regardless of the path from A to B, the path from B to D must be the shortest possible path between points B and D. Furthermore, say point C is on the path from B to D. In a similar manner, following the principle of optimality, regardless of the path from B to C the path from C to D must be the shortest path between points C and D. This principle is crucial in that it allows the "whole problem", i.e. the shortest path from A to D, to be sub-divided into a series of smaller subproblems from which the overall solution may be drawn. The primary task in utilizing dynamic programming to determine an optimal solution is splitting the problem into subproblems, and relating these subproblems to the original problem solution. The rule that assigns values to the subproblems is commonly referred to as the optimal value function and the rules that relate the subproblems are commonly referred to as the optimal policy function (rule by which the optimal solution may be retrieved from the subproblem solutions) and recurrence relations (rule by which the principle of optimality relates one subproblem to the next). Dreyfus and Law are clear in stating that while there are certainly many special "tricks" for specific types of problems, the principles and primary insights into dynamic programming are summarized in the preceding discussion. The "art of dynamic programming" is the ability to determine the appropriate optimal value function and thus optimal policy function and recurrence relation. An overly simplified optimal value function will lead to incorrect results, while an overly complicated function may lead to an optimal solution but the computational effort may be much greater that that required from a well formulated problem.

Utilizing DP may achieve significant computational efficiencies, when compared to more brute force methods such as enumeration of all possible solutions, then selecting the optimal. For example, Dreyfus and Law present a shortest-path problem from one corner of a grid layout to the opposing corner. Where there are 6 stages (in this case 6 arcs in a path) the dynamic approach requires roughly one-fourth the computational effort as full enumeration. When the number of possible arcs in a path increases to 20, dynamic programming requires roughly 0.1% as much computational effort as full enumeration. This trend of increasing comparative efficiency as problem size increases is one of the significant benefits of DP. Unfortunately, as constraints are added to a problem, the efficiency of a DP approach can be significantly decreased. The addition of a single constraint may easily double the computational effort.

ROLLING HORIZON

Prior to the development of COP, there were other notable attempts at utilizing DP in real-time traffic signal control. Possibly one of the most well known is OPAC (Optimized Policies for Adaptive Control Strategy), developed by Gartner (Gartner, Tarnoff, and Andrews, 1991). Gartner's elimination of traditional concepts (i.e. cycle length, splits, etc.) and use of DP in the OPAC logic made it an exceptional contribution to the science of traffic signal control. Indeed, in many ways COP builds upon Gartner's initial efforts.

A recognized drawback in the OPAC strategy, and the dynamic programming approach in general, is the need for an accurate prediction of arrivals on an approach over the optimization time period. Accuracy in traffic predication rapidly decreases as the optimization horizon increases. To lessen the impact of this problem Gartner further supplemented his optimization procedure to allow for updating of traffic data and a recalculation of the timings at a much shorter time interval than that being optimized, utilizing a strategy typically referred to as a rolling horizon. OPAC was one of the original signal-timing control strategies to utilize a rolling horizon. Since OPAC, rolling horizons have been utilized in many traffic control and prediction research efforts, with the Open-TS3 implementation of COP being no exception. The following is a brief presentation of the rolling horizon strategy, based of the original work of Gartner (Gartner, Tarnoff, and Andrews, 1991)

In a rolling horizon strategy (0), the stage length (or projection horizon) is split into two sections, a head and tail. The head is the amount of time that the strategy will shift or "roll" between stages. Therefore, upon each "roll" the optimal policy calculated for the head will be implemented while the tail will become part of the next stage to be optimized.

The head typically is on the order of ten to twenty seconds. The key to a successful rolling horizon implementation is for arrivals predicted during the head period to reasonably reflect what is actually realized. The primary factor in the determination of the head length is the placement of upstream traffic detectors. Once vehicles have crossed a detector, their arrivals may be predicted with a higher level of accuracy. Therefore, the travel time from the detector to the intersection is a sensible choice for the head time.

The tail length is the length of the optimization period minus the head length. Different projection horizons (optimization periods) have been utilized in different research efforts (Gartner, Tarnoff, and Andrews, 1991; Sen and Head, 1997; Han and Yagar, 1994; Farges, Khoudor, and Lesort, 1990). The choice of the length of the projection horizon is not an exact science, but is instead a compromise between differing attributes of the optimization procedure. For example, the projection horizon is limited by the complexity of the optimization algorithms and available computer processing power. A fundamental requirement of adaptive control is that it run in real-time. A longer projection horizon requires more computing power, thus computing power is a constraint. Also, the adaptive control algorithms that utilize a rolling horizon utilize arrival predictions during the tail. As the projection horizon increases, predications near the end of the tail become increasing less reliable, reaching a point of diminishing returns, where little is gained from the additional effort. While no hard rules exist, projection horizons on intersection approach level analysis typically do not exceed a few minutes.

0 and the basic steps that follow are taken from Gartner et. al.(1987) and provide a clear illustration into the fundamental concepts of the rolling horizon strategy.

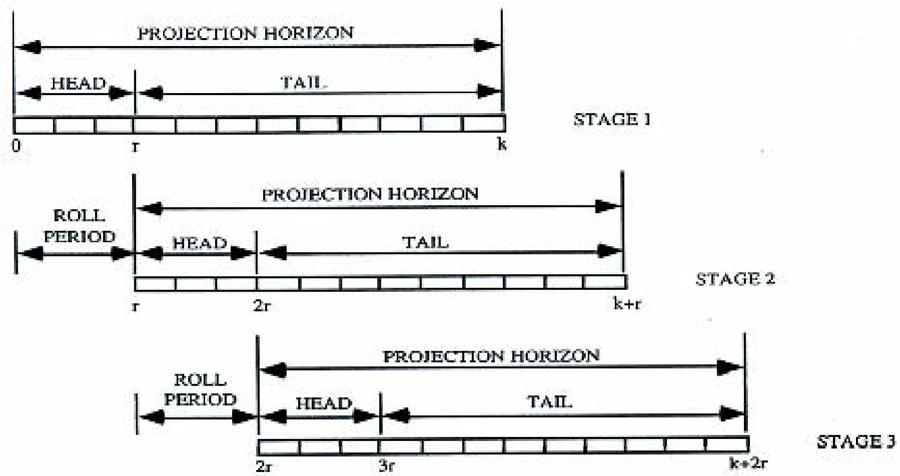


Figure 6.1 Illustration of the Rolling Horizon Strategy

The basic steps are as follows:

Step 0. Determine projection horizon and roll period r.

Step 1. Obtain flow data for the first r steps (head) from the detectors and calculate flow data for the next k-r steps (tail) from an appropriate tail model and detector data from previous projection horizons.

Step 2. Calculate the optimal switching policy for the entire stage by OSCO (Optimal Sequential Constrained search method).

Step 3. Implement the switching policy only for the roll period (head).

Step 4. Roll forward the projection horizon by r units to create a new stage. Repeat steps (1) – (4)

In step 0 Gartner determines the projection horizon and roll length, which is also the head length. Step 1 consists of predicting the arrivals over the projection period. Gartner considered several different arrival patterns (fixed, static, and dynamic) for the tail period. In Open-TS3 implemented adaptive control, it will be seen that Open-TS3 is utilized for traffic prediction. In step 2 the optimal signal timing plan for the entire projection horizon is determined. Step 3 then implements this timing for the roll period. Step 4 starts the process over (return to step 1) with the time now advanced by the roll period.

The previous two sections have provided a background for dynamic programming and rolling horizons. The next sections will present the dynamic programming and rolling horizon approach utilized in the Open-TS3 adaptive control implementation.

CONTROLLED OPTIMIZATION OF PHASES AT AN INTERSECTION

As stated, the dynamic programming algorithm utilized for the demonstration of adaptive signal control in Open-TS3 is Controlled Optimization of Phases at an Intersection (COP), developed by Sen and Head. The primary goal of this (and any) adaptive signal control strategy is to provide an optimal signal-timing pattern given real-time vehicle arrivals. COP allows the optimal timings to be a function of one of several performance measures: stops, queues, and delays. In this Open-TS3 effort, delay minimization is chosen as the optimization goal.

COP utilizes a forward dynamic programming approach where a value function for any stage represents "an accumulated measure of effectiveness for the current and all previous stages". In this DP formulation the recursion is based on using phases as stages. In the Dreyfus and Law terminology this equates a subproblem as the determination of the phase length that optimizes the objective function. This is distinctly different from OPAC and many other approaches where stage lengths are fixed and the subproblem is the determination of the signal switches during the stage that optimize the objective function.

The remainder of this section is dedicated to a presentation of COP. This includes the notation, optimal value function, recurrence relation, boundary conditions, and the forward recursion and retrieval of optimal policy algorithms. (Reference clarification: The notation and equations in this section are taken from Sen and Head (Sen and Head, 1997) with only minor changes to 1) present the formulation in amanner similar to that utilized in Dreyfus and Law (191) and 2) to specify the equations for delay, where Sen and Head generalized for delay, queues, or stops. A significantly more detailed discussion of COP may be found in referenceSen and Head, 1997.)

Notation

P	$\equiv$	Set of Phases. The cardinality of this set will be denoted $\|P\|$. Individual phase are indexed by ϕ.
T	$\equiv$	Total number of discrete time-steps. Each period of length Δ is indexed by $t \in [1,T]$.
γ	$\equiv$	Minimum green time (integer number of time-steps).
r	$\equiv$	Effective clearance interval (integer number of time-steps), $r \leq \gamma$.
j	$\equiv$	Index for stages of the DP.
l	$\equiv$	Index in P that denotes the initial phases.
x_j	$\equiv$	Control variable denoting the amount of green-time allocated to stage j.
s_j	$\equiv$	State variables denoting the total number of time-steps that have been allocated after stage j has been completed.
$X_j(s_j)$	$\equiv$	Set of feasible control decisions, given state s_j

$f_j(s_j,x_j)$ ≡ Performance measure at stage j, given state s_j and control x_j. The performance measure in this implementation is delay, incurred from time $s_j - x_j$ to s_j.

$v_j(s_j)$ ≡ Value function (cumulative value of prior performance measures), given s_j.

Optimal value function - (Rule that assigns a value to each stage)

Min $v_j(s_j)$ = minimize the delay at the intersection after stage j has been completed, with s_j time steps having been allocated.

Recurrence relation - (Formula that relates various values of V(S) based on principle of optimality***)***

$$v(s_j) = \mathrm{Min}_{xj}\ \{v_{j-1}(s_{j-1}) + f_j(s_j,x_j)\ \mid x_j \in X_j(s_j)\} \tag{6.1}$$

Where:

$$X_j(s_j) = \begin{cases} \{0\} & \text{if } s_j - r \\ \{0, \gamma, \gamma+1, \ldots, s_j - r\} & \text{otherwise} \end{cases} \tag{6.2}$$

$$s_{j-1} = s_j - h_j(x_j) \tag{6.3}$$

given:

$$h_j(x_j) = \begin{cases} 0, & \text{if } \mathrm{x}_{\mathrm{j}} = 0 \\ x_j + r, & \text{otherwise} \end{cases} \tag{6.4}$$

The recurrence relation 6.1 is a simple delay minimization, with $v_{j-1}(s_{j-1})$ representing the optimal delay over time steps 0 to s_{j-1}, through phase $j-1$ and $f_j(s_j,x_j)$, the delay incurred at the intersection during phase j. This delay is the minimum value over all possible x_j, that is given s_j the set of all feasible phase j lengths: $X_j(s_j)$. The set of feasible lengths x_j for the given s_j is defined in equation 6.2, which allows for a phase length of zero and all phase lengths from the minimum green to the number of time steps under consideration (minus r to account for a clearance interval). Equations 6.3 and 6.4 provide the relationship between state variables sj and $sj-1$, ensuring that the clearance interval is accounted for.

Boundary Conditions

$v_o = 0$

Simply stated, at time step 0 the total delay is taken to be zero.

Forward Recursion and Retrieval of Optimal Policy

For each stage the minimum possible delay is determined. Thus the value of the optimal value function given the final stage (the final stage is determined by the stopping criteria below) is the minimum delay over the projection horizon. By recording the optimal $x_j(s_j)$ for each of the j stages during the forward recursion the optimal signal timing pattern may be retrieved once the final stage has been determined. Below is the forward recursion and retrieval of optimal policy algorithms, per Sen and Head.

Forward Recursion

0. Initialize $v_o \equiv 0$. $j = 1$
1. for $s_j = r, \ldots T$, {

   $v_j(s_j) = \text{Min}_{xj} \{ f_j(s_j,x_j) + v_{j-1}(s_{j-1}) \mid x_j \in X_j(s_j) \}$

   record $x_j^*(s_j)$, an optimal solution to the above problem

   }
2. if ($j < |P|$), $j \leftarrow j + 1$ and repeat from step 1.

   Else if ($v_{j-k}(T) = v_j(T)$) for all $k \leq |P| - 1$,

   STOP.

   else $j \leftarrow j + 1$ and repeat from step 1.

The stopping criterion in step 2 determines the final j, i.e. the minimum number of phases required to minimize the objective function. The principle behind this stopping criterion is rather intuitive. As the recursion algorithm proceeds, the optimal value function of each stage will be less than or equal to that of the previous stage. If all phases are checked and the optimal value function remains constant than there can be no benefit to continuing the optimization, thus the minimum has been reached. In other words once the recursion cycles through all possible phases and no improvement in the optimal value is achieved then continuing to cycle through the phases cannot further improve the optimal value.

Retrieval of Optimal Policy

0. $s^*_{j-(|p|-1)} = T$
1. for $j = J - (|P| - 1), J - (|P| - 2), \ldots, 1$ {

   read $x_j^*(s_j^*)$ from Table computed in forward recursion

   if (j>1) $s_{j-1}^* = s_j^* - h_j(x_j^*(s_j^*))$

   }

The retrieval is relatively straightforward. By recording the optimal control variable, $x_j^*(s_j)$, for each subproblem, the optimal phase pattern is obtained by reading back through the table of optimal control variables starting at stage j – (|P| -1). (Note: j-(|P|-1) is the first stage to exhibit the minimum delay in the stopping criterion in the forward recursion.)

CONSTRAINED COP

Based on the COP algorithm, the adaptive control algorithm Constrained COP was developed for this research effort. The underlying goal of this strategy is to limit the effect of prediction error, which is the cause of one of the primary complaints leveled against adaptive control strategies: inconsistent performance. Where the predicted arrivals differ substantially from those realized, adaptive control may result in timings that perform significantly worse than what would have been achieved with an offline solution.

Constrained COP attempts to improve the consistency of the adaptive control approach through the incorporation of offset windows. In this adaptive strategy a background cycle is maintained, with the offset to each intersection measured relative to the background cycle. (That is, the time difference between the background cycle start and the mainline green start for an intersection.) The desired offsets are determined offline, prior to the adaptive control analysis. To maintain the benefits of flexibility in timing plans, an offset window is utilized instead of a strict offset. Instead of requiring that the coordinate phase begin exactly at the offset it is allowed to begin within some range of the offset. For example at an intersection with an offset of fifty seconds the Constrained COP strategy, with a twenty second offset window, allows the coordinated green to begin anytime from forty to sixty seconds, relative to the background cycle. Constrained COP requires a switch to the coordinated phase during the offset window; otherwise the characteristics of the strategy are similar to that of COP.

The remainder of this section follows a similar format to that of the COP discussion. This includes the notation, optimal value function, recurrence relation, boundary conditions, and the forward recursion and retrieval of optimal policy algorithms.

Notation

The notation is the same as that in section 0, with the following additions:

cp	≡	Phase that is coordinated, $cp \in P$.
c_j	≡	Indicator for whether stage may satisfy coordination.
o_j	≡	Indicator for whether stage j begins during offset window.
off	≡	Offset for coordinate phase.
ohf	≡	Half length of offset window.

Optimal value function

This rule is the same as that for COP (Section 0), with the addition that coordination constraint must be meet.

Min $v_j(s_j, cp, off, ohf)$ = minimize the delay at the intersection after stage j has been completed, with s_j time steps having been allocated, and the coordinated phase (cp) beginning within the offset window [$off - ohf$, $off + ohf$].

Recurrence relation

$$v(s_j, cp, off, ohf) = \text{Min}_{xj} \{v_{j-1}(s_{j-1},cp,off,ohf)+ f_j(s_j,x_j) \mid x_j \in X_j(s_j)\} \qquad (6.5)$$

Where:

$$X_j(s_j) = \begin{cases} \{0\} & \text{if } s_j - r \\ \{0, \gamma, \gamma+1, \ldots, s_j - r\} & \text{otherwise} \end{cases} \qquad (6.6)$$

$$s_{j-1} = s_j - h_j(x_j) \qquad (6.7)$$

given:

$$h_j(x_j) = \begin{cases} 0, & \text{if } \mathrm{x_j} = 0 \\ x_j + r, & \text{otherwise} \end{cases} \qquad (6.8)$$

$$\sum_j c_j o_j \geq 1 \qquad (6.9)$$

given:

$$c_j = \begin{cases} 1, & \text{if } \mathrm{j \bmod |P| = cp} \\ 0, & \text{otherwise} \end{cases} \qquad (6.10)$$

$$o_j = \begin{cases} 1, & \text{if } (off - ohf) \leq T - x_j \leq (off + ohf) \\ 0, & \text{otherwise} \end{cases} \qquad (6.11)$$

This recurrence relation mirrors that of COP with the addition of the constraint represented by equations 6.9, 6.10, and 6.11. Equation 6.10 provides the mathematical definition for c_j, which indicates whether a stage may satisfy coordination. For example, assume a three phase intersection timing plan, with phase 2 coordinated, has a Constrained COP dynamic program solution with seven stages, {1,2,3,4,5,6,7}. Additionally, assume that in this execution of Constrained COP that stage 1 maps to Phase 2; then by definition stages {1,2,3,4,5,6,7} map to phases {2,3,1,2,3,1,2}. As phase 2 is the coordinated phase this implies that stage 1, 4, or 7 must begin within the offset window. Thus, the values of c_j for this example are {1,0,0,1,0,0,1}. Equation 6.11 simply states that if a stage *j* begins during an offset window then $o_j = 1$, and if stage j does not begin within the offset window then $o_j = 0$.

Now, given equations 6.10 and 6.11, the offset constraint, Equation 6.9, may be understood. This constraint insures that at least one instance of the coordinated phase begins during the offset window. It is important to note that this constraint does not imply that all instances of the coordinated phase begin within the offset window. That is, from the previous example only one stage of the set {1,4,7} must begin within the offset window. One or both of the other two stages may also begin in the offset window, may have a zero green (i.e. the phase is skipped), or the may begin outside of the offset window.

An example of the behavior that may be realized through Equation 6.9 may be seen in some of the more advanced pre-timed coordinated systems. Consider a system that includes an intersection with significantly lower volumes than other system intersections. A pre-timed coordination plan may half-cycle the low volume intersection, meaning only every other instance of the coordinated phase falls within the coordination scheme (i.e. green band). Constrained COP will allow the adaptive equivalent of half-cycling, without specifically designating that an intersection be half-cycled. As volumes fluctuate an intersection may behave as if half-cycled, while in other instances behave as if full cycled.

Boundary Conditions

$v_o = 0$

Simply stated, at time step 0 the total delay is taken to be zero.

Forward Recursion and Retrieval of Optimal Policy

The forward recursion for Constrained COP follows the same format as that for COP, with increased complexity required to minimize the value function due to the added constraint. Given stage j, the following combinations of value functions (v) for j-1, and performance measures (f) for j, satisfy the offset constraint:

$$f_j(s_j,x_j)_{cs} + v_{j-1}(s_{j-1}, cp, off, ohf)_{cns} \quad (6.12)$$
$$f_j(s_j,x_j)_{cns} + v_{j-1}(s_{j-1}, cp, off, ohf)_{cs} \quad (6.13)$$
$$f_j(s_j,x_j)_{cs} + v_{j-1}(s_{j-1}, cp, off, ohf)_{cs} \quad (6.14)$$

Where *cs* indicates that the offset constraint is satisfied and *cns* indicates that the offset constraint is not satisfied. Thus in equation 6.12 stage j satisfies the constraint, therefore the constraint need not be satisfied by the previous stages, {1,2,…,j-1}. In equation 6.13 stage j does not satisfy the constraint, therefore the constraint must be satisfied by at least one prior stage. Finally equation 6.14 represents the situation where stage j and at least one prior stage satisfy the constraint. Although, it is necessary to maintain equation 6.14 in the recursion since $v_{j-1}(s_{j-1}, cp, off, ohf)_{cs}$ is a constrained version of $v_{j-1}(s_{j-1}, cp, off, ohf)_{cns}$, therefore $v_{j-1}(s_{j-1}, cp, off, ohf)_{cns} \leq v_{j-1}(s_{j-1}, cp, off, ohf)_{cs}$, leading to Eq (1) ≤ Eq(3). Thus, the minimum value from the minimizations of equations 6.12 and 6.13 provides the minimum constrained value function for stage j, equation 6.5.

From equations 6.13 and 6.14 it is seen that the minimization for Constrained COP at stage j requires the constrained and unconstrained (original COP) solutions for stage j-1. Therefore as part of the recursion algorithm the value functions and optimal control variables at stage j must be recorded for both Constrained COP (*cs* subscript) and COP (*cns* subscript). Also, for the constrained solution it must record whether or not v_{j-1} satisfies the constraint (i.e. was equation 6.13 or 6.14 the minimum).

Forward Recursion

1. Initialize $v_{o,cs} \equiv 0$. $v_{o,cns} \equiv 0, j = 1$
   for $s_j = r, \ldots T,$ {

$$v_j(s_j, cp, off, ohf)_{cs} = \text{Min}_{xj} \begin{cases} f_j(s_j, x_j)_{cs} + v_{j-1}(s_j\text{-}1,,,)_{cns} \\ f_j(s_j, x_j)_{cns} + v_{j-1}(s_j\text{-}1,,,)_{cs} \\ f_j(s_j, x_j)_{cs} + v_{j-1}(s_j\text{-}1,,,)_{cs} \end{cases} \quad | \, x_j \in X_j(s_j)$$

record $x_j^*(s_j)_{cs}$, optimal solution to the above constrained problem, record if v_{j-1} constrained or unconstrained.

$v_j(s_j, cp, off, ohf)_{cns} = \text{Min}_{xj} \{ f_j(s_j, x_j)_{cns} + v_{j-1}(s_{j-1,,,})_{cns} \mid x_j \in X_j(s_j) \}$
record $x_j^*(s_j)_{cns}$, optimal solution to the above unconstrained problem

}

2. if ($j < |P|$, $j \leftarrow j + 1$ and repeat from step 1.
Else if ($v_{j-k}(T)_{cs} = v_j(T)_{cs}$) for all $k \leq |P| - 1$,
STOP.
else $j \leftarrow j + 1$ and repeat from step 1.

For each stage both the constrained and unconstrained minimum delay (value function) is determined. The constrained optimal value function given the final stage is the minimum delay over the projection horizon. By recording the optimal $x_j(s_j)_{cs}$ and $x_j(s_j)_{cns}$ for each of the j stages during the forward recursion the optimal constrained signal timing pattern may be retrieved once the final stage has been determined.

Retrieval of Optimal Policy

0. $s^*_{j-(|p|-1)} = T$
1. for $j = J - (|P| - 1), J - (|P| - 2), \ldots, 1$ {
   read if v_{j-1} constraint satisfied or constraint not satisfied
   If v_j constraint satisfied read $x_j^*(s_j^*)_{cs}$
   If v_j constraint not satisfied read $x_j^*(s_j^*)_{cns}$
   if (j>1) $s^*_{j-1} = s^*_j - h_j(x^*_j(s^*_j))$
   }

The retrieval is similar to that of COP. For each subproblem the optimal control variables, $x_j^*(s_j)_{cs}$ *and* $x_j^*(s_j)_{cns}$, were recorded. The optimal phase pattern is obtained by reading back through the table of optimal control variables for which the constraint is satisfied starting at stage $j - (|P| \text{ -}1)$. Once the stage that satisfied the constraint is reached, the remaining stages will be read from the unconstrained solution.

ADAPTIVE CONTROL IMPLEMENTATION IN OPEN-TS3

0 presents the framework for implementing adaptive signal control in Open-TS3. A broad outline of this framework follows: Beginning with a set of initial timings (developed off-line) Open-TS3 provides arrival predictions over the projection horizon to the adaptive signal control logic dynamic link library (.DLL). Utilizing the adaptive control logic, optimal signal timings are developed and fed back to Open-TS3 to be simulated. Open-TS3 is used to measure the performance of the system during the head period, as this reflects the real-world timing implementation, and to record the arrivals over the tail to be used as the arrival predictions in the optimization of the next projection horizon. Time is advanced by the head length and the arrival predications are fed to the adaptive control logic for the next optimization. This process continues forward over the analysis period via the rolling horizon. Performance measures collected by Open-TS3 reflect the quality of the signal control, allowing for comparisons to other control strategies.

The remainder of this section provides a detailed description of the framework presented in 0. This includes a discussion of the each step of the implementation, vehicle predication methodology, Open-TS3 SIGNAL Block (the SIGNAL block implements adaptive control), and the development of the adaptive DLL. Hunter and Machemehl contains the adaptive signal code and SIGNAL block SIMAN construction.

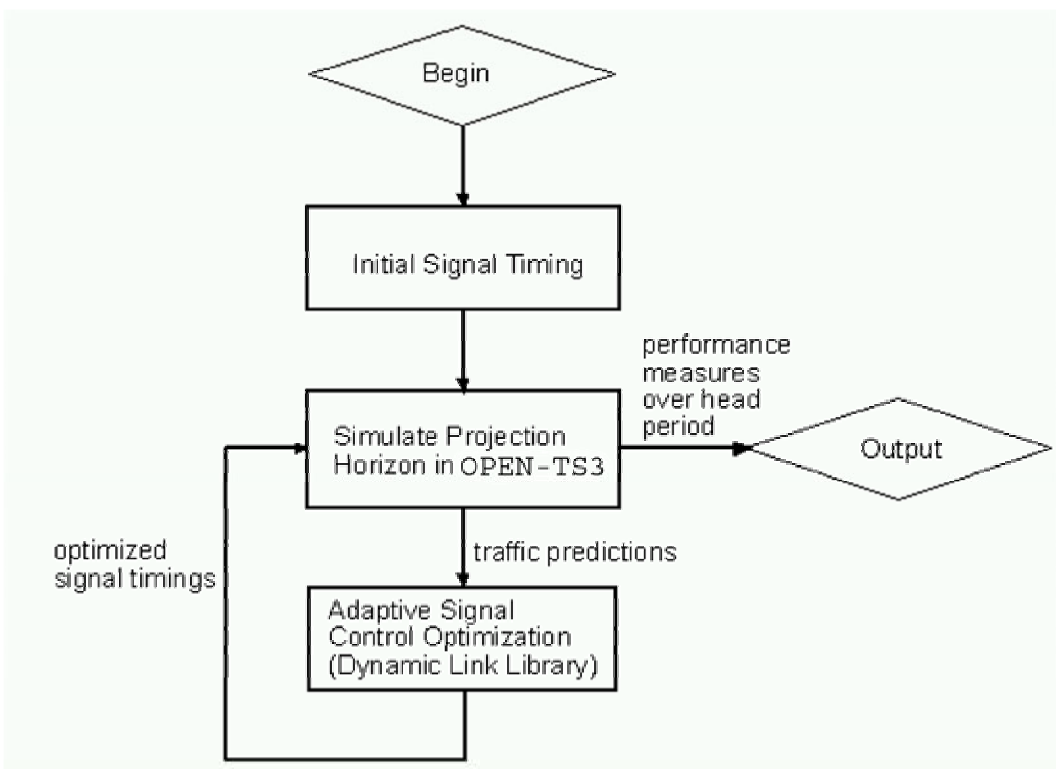


Figure 6.2 Adaptive Control Implementation in Open-TS3 Framework

Open-TS3 Adaptive Control Implementation Framework

This framework is based on the rolling horizon. Presented first are the implementation steps, followed by a discussion of each.

Implementation Steps

Step 0. Develop initial timings, determine projection horizon and roll period.

Step1. Simulate latest optimized timing plan in Open-TS3. Record performance measures for initial twenty seconds. Record vehicle arrivals and initial queue lengths for remaining one hundred seconds, the next projection horizon to be optimized. Pass arrival and queue information to adaptive control .DLL.

Step 2. Optimize signal timings utilizing adaptive control .DLL (contains COP and Constrained COP) based on predicted arrivals.

Step 3 Complete timing plan for end of tail for Open-TS3 simulation. Pass optimization to Open-TS3, Goto step 1

Discussion of steps

Step 0. Develop Initial Timings, determine projection horizon and roll period.

Signal timings for the first projection horizon are required as part of the initialization. The need for initial timings arises from the dual roles of Open-TS3: to simulate optimized timings (step 1) and to predict the arrivals utilized in the optimization (step 2). As the first set of optimized timings are developed in step 2 it is necessary to provide Open-TS3 with an initial set of timings for the first projection horizon. These timings may be developed utilizing any of the numerous methods for determining signal timings offline (see chapter 2). In this Open-TS3 effort TRANSYT-7f was used to develop the initial timings, thus the initial projection horizon utilized an optimal offline timing plan. Future research may undertake a study of the sensitivity of the Open-TS3 adaptive control implementation to the initial signal plan.

Also as part of the initialization, the entry link approaches (approaches with no modeled upstream intersection) in sample networks were assumed to have detectors placed twenty seconds in advance of the intersection, resulting in selection of the twenty second roll period. For approaches with an upstream intersection, that upstream intersection acted as the detector. A projection horizon of one-hundred seconds was chosen, as this is typical of the length in other similar research efforts (Gartner, Tarnoff, and Andrews, 1991; Sen and Head, 1997). The sensitivity of the optimization procedure to the projection horizon and the roll period lengths are clearly areas worthy of future study.

Step 1. *Simulate latest optimized timing plan in Open-TS3. Record performance measures for initial twenty seconds. Record vehicle arrivals and initial queue lengths for remaining one hundred seconds; the next projection horizon to be optimized. Pass arrival and queue information to adaptive control .DLL.*

As stated previously, Open-TS3 serves two primary purposes in this adaptive control study, the simulation of optimal timings to measure the performance of the control strategy and the development of arrival predications for use in the optimization. In step 1 Open-TS3 fulfills both functions. Open-TS3 simulates the latest set of timings developed in steps 2 and 3 (a one-hundred and twenty second timing plan, see step 3). The performance of the first twenty seconds provides a measure of the quality of the optimal solution over the previous head length for which the timings where optimized. At the completion of the analysis period this data may be used to compare signal control performance against other signal control strategies (i.e. pretimed and actuated) that utilize the same arrival streams as Open-TS3. The arrivals over the remaining one-hundred seconds and queue lengths at the beginning of this time are recorded and passed to the .DLL to be utilized in the next optimization (i.e. the next execution of step 2).

By utilizing Open-TS3 to generate arrival predictions, the real world equivalent assumed is that of communication between neighboring intersections, allowing detector actuations and output flow predictions from the upstream intersection to be used in the development of downstream intersection flow profiles. Also, by recording the queue lengths in Open-TS3 at the beginning of the optimization period it is assumed that queues can be measured in the field in real time.

The SIGNAL block has been developed for Open-TS3 to fulfill the roll of interfacing with the adaptive control .DLL. As with the PRETIMED8P and ACTUATED8P blocks it is not useful to continue to present detailed algorithmic forms as with the PRETIMED block and other blocks described in chapter 3. As with the PRETIMED8P and ACTUATED8P Hunter and Machemehl (2003) contains the object class hierarchies, input dialogue requirements, and logical constructs for the SIGNAL block. Although a brief synopsis of the block is provided.

The primary difference between the adaptive control SIGNAL block and the other signal blocks is the use of Visual Basic to fill block inputs, instead of user interfaces. Through the use of Visual Basic Open-TS3 is allowed to call the .DLL, export the arrivals, and import the optimized timings. Instead of hard coding timings into the block (such as with the PRETIMED block) it is possible to read timings optimized for that simulation by the adaptive control .DLL. In other respects the SIGNAL block is similar to the PRETIMED block. Both blocks utilize the same Intersection Movement logic. Also, even though timings are read into the Signal Switching object through Visual Basic both the SIGNAL and PRETIMED blocks use only up to three phases. Although, this SIGNAL block three phase constraint is dictated less by the block logic and more by the adaptive control .DLL which is currently set for only three phases.

Step 2. *Optimize signal timings utilizing adaptive control .DLL (contains COP and Constrained COP) based on predicted arrivals.*

As seen, the arrivals over the projection horizon are predicted in step 1 and passed to the adaptive control dynamic link library. The developed .DLL is capable of both COP and Constrained COP, presented in sections 0 and 0. Hunter and Machemehl, 2003, contains the pseudo code and program code (Visual Basic) for both COP and Constrained COP).

A few points about the use of the arrivals in the adaptive control should be noted. For the example networks (Section 0), vehicles were modeled to arrive on Open-TS3 entry links (approaches where an upstream link is not included in the model) according to a Poisson distribution. To better reflect data that would be available in the real world, the optimization algorithm replaces Open-TS3 recorded tail arrivals with a fixed tail scheme (i.e. the tail arrivals are assumed to be constant, based on the mean hourly arrival rate). Thus, in the optimization, entry link arrivals are known with certainty during the head (i.e. vehicles have crossed a detector), but tail arrival times are predictions that may vary from the actual times.

For the entire projection horizon, intersection approach arrivals on internal or non-entry links utilize those predicted by Open-TS3 in step 1. Thus, for any approach on an internal link, vehicles on that internal link during the previous projection horizon will have approach arrival times known with certainty in the current projection horizon optimization. Otherwise the arrival time of a vehicle on an approach may not be realized as the upstream signal timing upon which the approach arrival time is dependent (since the vehicle has yet to pass through the upstream intersection) may change during the current optimization. That is, if the optimal timing solution does not change after a "roll" then the previously assumed upstream intersection departures will not change, resulting in the arrival predictions being realized. Whereas, if the previously determined upstream timing pattern should change after a roll, then the predicted upstream departure pattern would not be realized, and the approach arrivals based on that prediction would be incorrect.

Given these arrivals, updated optimal signal timings are determined. The adaptive control strategies utilized are COP and Constrained COP. Open-TS3 calls the adaptive control optimization for each intersection independently. Thus, for the current projection horizon each intersection is optimized without consideration of its optimization's impact on the performance of neighboring intersections. Although interaction between intersections is inherent in the optimization procedure, as an intersection's arrivals are based on the neighboring intersection's departures, optimized during the previous projection horizon.

Step 3 Complete timing plan for end of tail for THBOSS Simulation. Pass optimized timings to Open-TS3.

In step 3 the optimized signal timings are passed to Open-TS3. In step 1 Open-TS3 will simulate the first twenty seconds (the head) collecting performance data for evaluation of the control and then implement the "roll" and simulation the next project horizon to be optimized. There is one notable problem: the optimal timing pattern from step 2 is one-hundred seconds but step 1 requires one-hundred and twenty seconds - the twenty seconds which is implemented (with performance measures gathered) and the next one-hundred second projection horizon. In order for Open-TS3 to simulate the next projection horizon it requires a timing pattern for the entire horizon. To provide the timing pattern needed for the final twenty seconds of the projection horizon it was assumed that the signal timings would follow the pattern of the initial timings developed in step 0. With this twenty seconds added to the optimal pattern in step three, the timings may now be passed to step one and the algorithm may proceed to step one.

SAMPLE ADAPTIVE CONTROL EVALUATIONS IN OPEN-TS3

This section presents some results for the adaptive control strategies COP and Constrained COP, tested on three different networks, under varying volume conditions. Utilizing these different networks and demand conditions the behavior of the adaptive signal control strategies for traffic systems of varying complexity is captured. Also, to allow for a comparison of the adaptive control performance to the state of the practice, a simulation utilizing optimal offline timings for each network / volume scenario is performed.

Each of these strategies was readily evaluated in Open-TS3, on the different networks and for the varying conditions. Since the control logic is incorporated into the intersection block logic (SIGNAL and PRETIMED) the layout of the network was not critical to the time required to develop each model. Utilizing the Open-TS3 tier one blocks networks are able to be quickly developed in a "drag and drop" manner. Also, the ability of Open-TS3 to isolate the logic at each entry point (ENTER Blocks) allowed for a separate random number stream to be assigned to each entering traffic stream. This insures that when comparing different strategies each is tested under identical "random" demand conditions.

Sample Networks

The three network layouts may be found in 0 through 0. Three basic network configurations are modeled: a four-intersection two-way arterial, two intersecting two-way four-intersection arterials, and a grid of nine intersections (all traffic two-way). For the adaptive and offline timing plans all intersections are assumed to operate with a maximum of three phases; north-south, east-west, and mainline dual lefts. While different demands are tested on the networks in all instances intersection volumes remain below capacity. The following provides information specific to each sample network.

Network 1 – two-way arterial. Network 1 provides one of the most commonly encountered traffic control design optimizations undertaken – arterial optimization. Three different demand scenarios are considered, 1A, 1B, and 1C (0). These scenarios are constructed to provide varying levels of complexity. Network 1A is the most amenable to optimization. The same demand is placed on each intersection and equal spacing exists between intersections. Network 1B varies the demand on each intersection, with total intersection demand increasing eastbound, from intersection 1 to 3, demand then decreases on intersection 4. The directional split (east to west demand versus west to east demand) is relatively balanced. Similar to network 1B, network 1C also has varying intersection demands and spacings but incorporates a directional split with a notably higher west to east travel demand.

Network 2 – intersecting two-way arterials. Network 2 increases the system complexity compared to network 1 by including an intersecting four-intersection two-way arterial. Two different scenarios are examined for network 2. The first, 2A (0), is the least complex of the two. All intersections on the east-west arterial (intersections 1, 2, 3, and 4) experience the same demand and the remaining three intersections on the north-south arterial (intersections 5, 6, and 7) experience similar demands. Also, the distance between all intersections is the same. Network 2B (0) increases the complexity by varying the intersection demands and spacings.

Network 3 – nine intersection grid. Network 3 is one of the most difficult signal control networks, a grid of intersecting two-way arterials. As with network 2, two different scenarios are considered for network 3. For network 3A (0), all intersections receive the same traffic demand and are evenly spaced. Network 3B (0) varies the demand across all intersections, with the center east-west arterial (intersections 1, 2, and 3) experiencing the highest demand. The distance between intersections is also varied in network 3B.

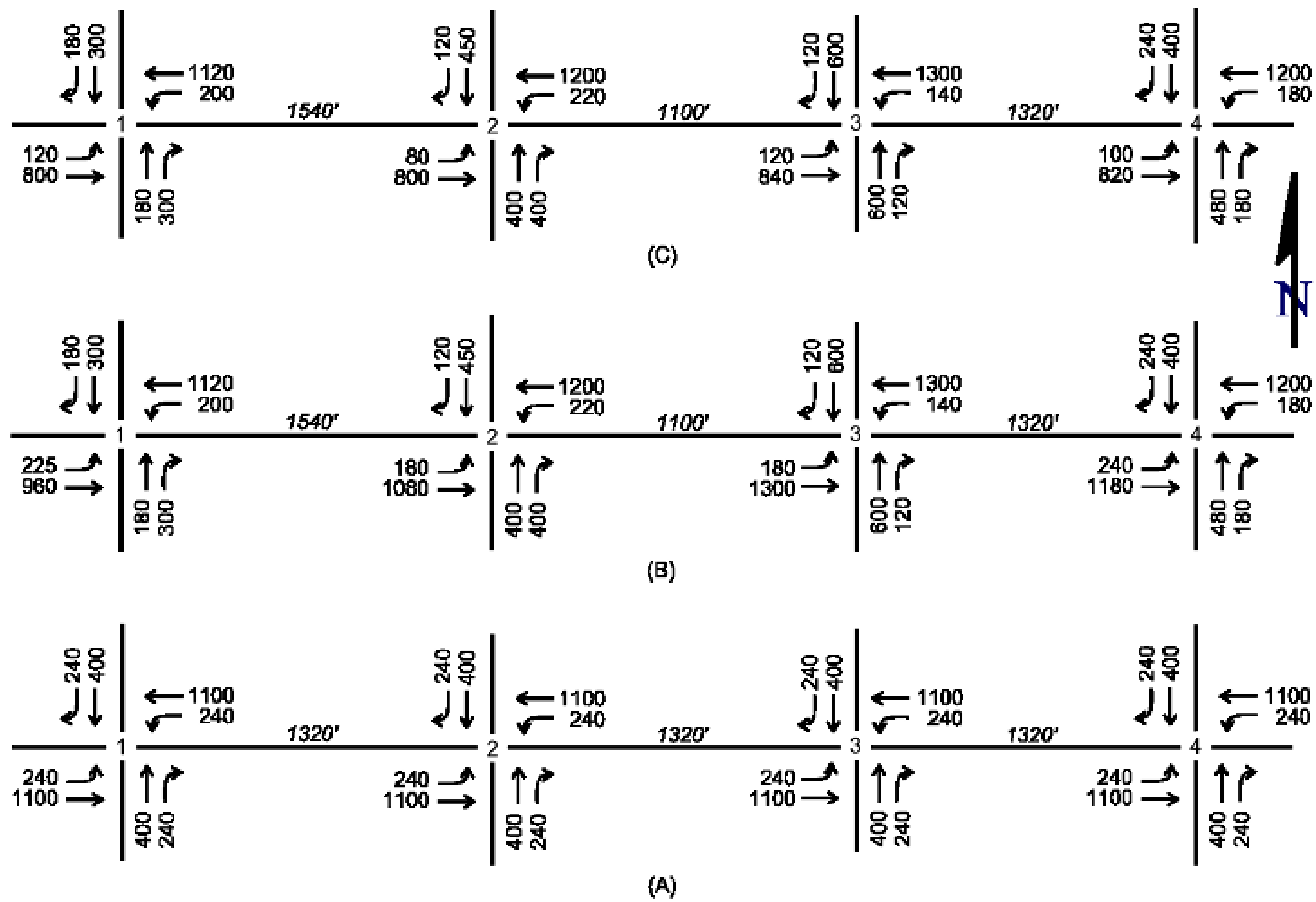


Figure 6.3 Network 1 – Demand Volumes and Intersection Spacing

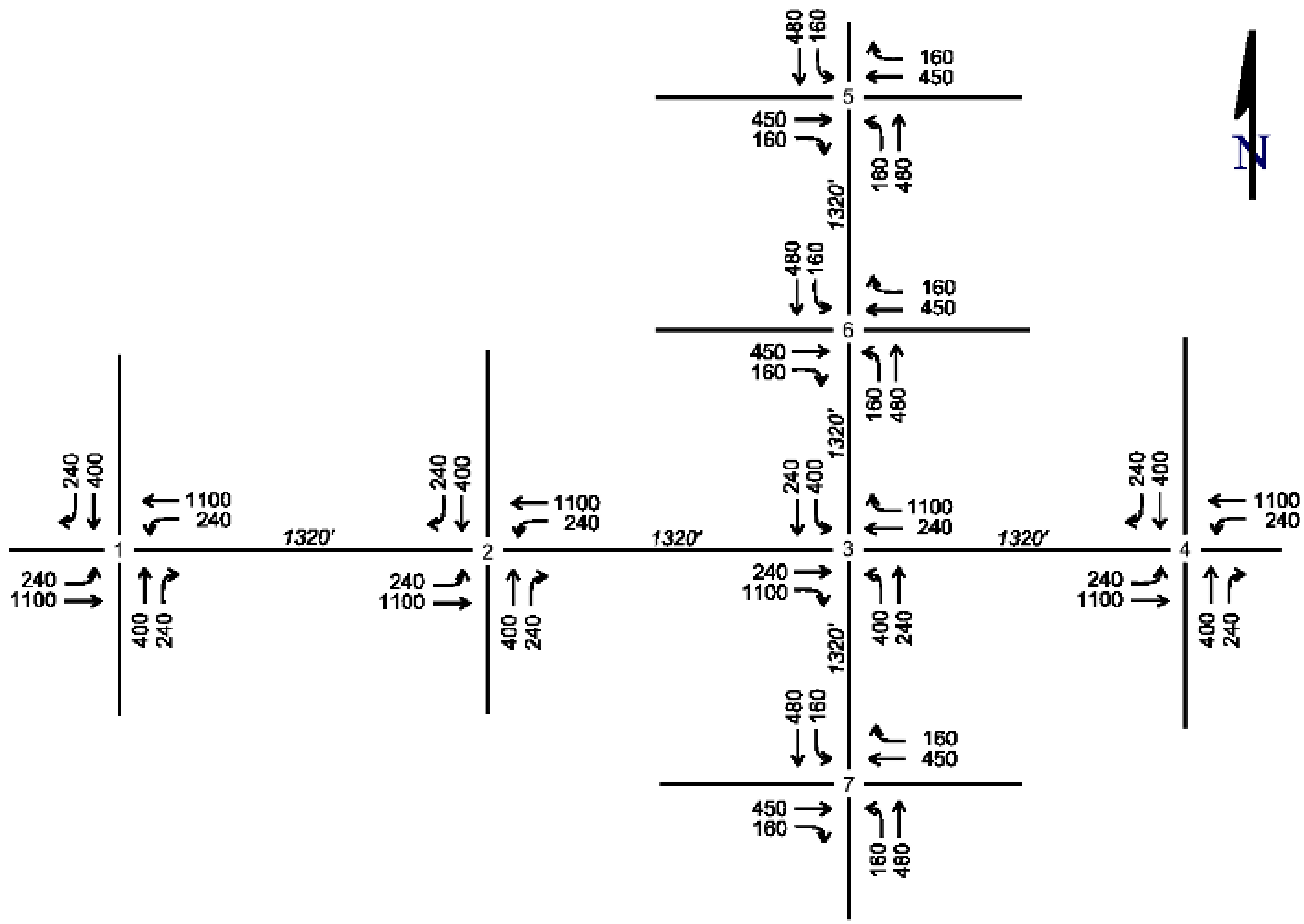


Figure 6.4 Network 2A – Demand Volumes and Intersection Spacing

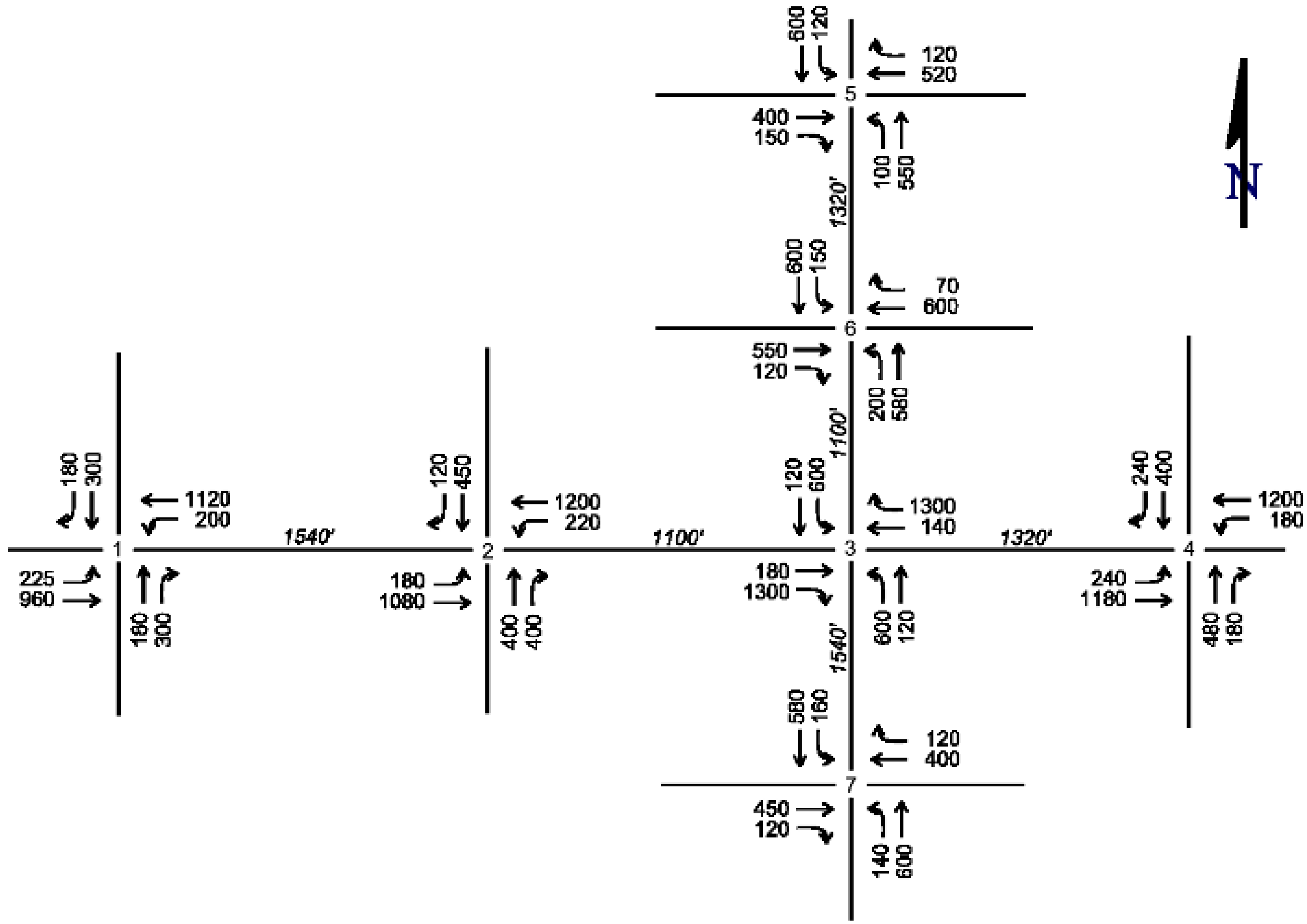


Figure 6.5 Network 2B – Demand Volumes and Intersection Spacing

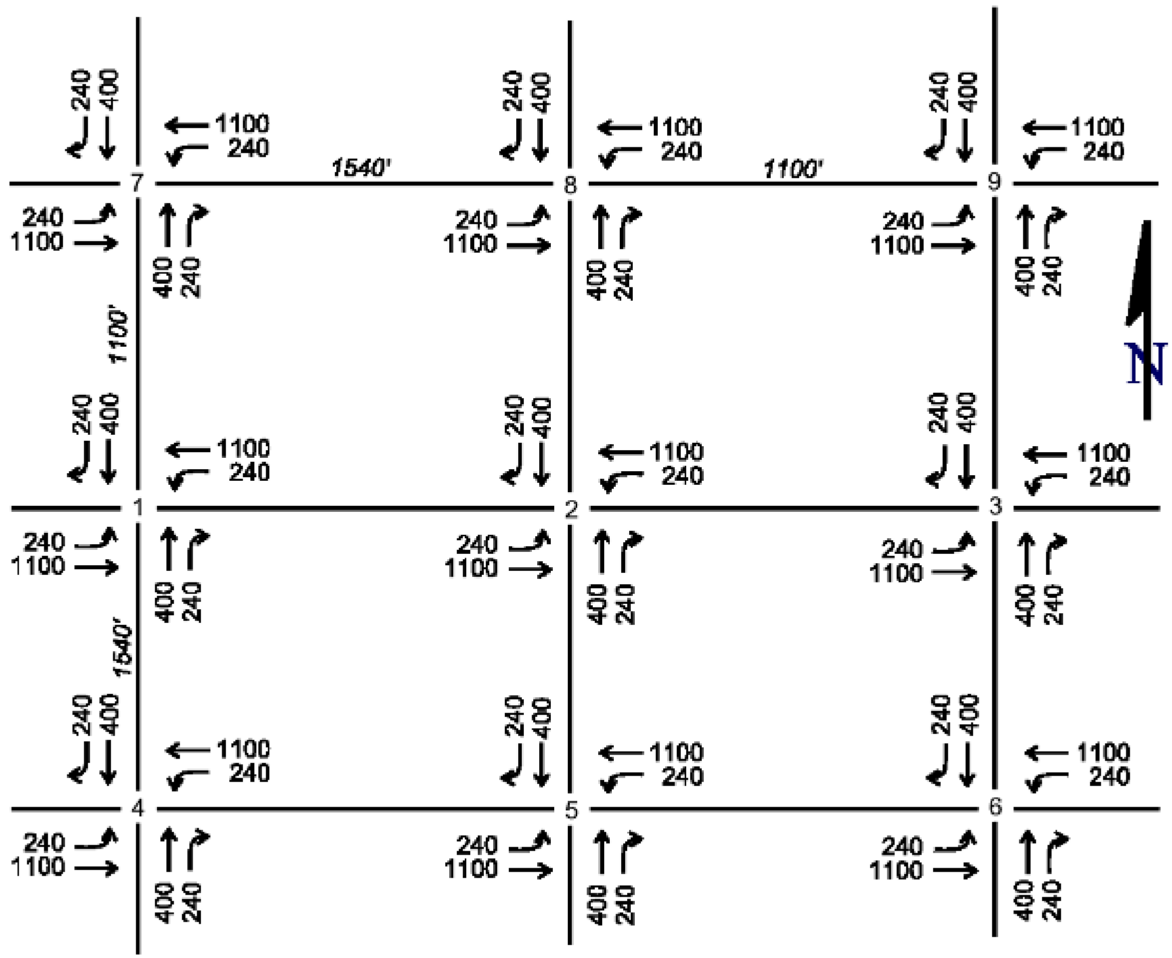


Figure 6.6 Network 3A – Demand Volumes and Intersection Spacing

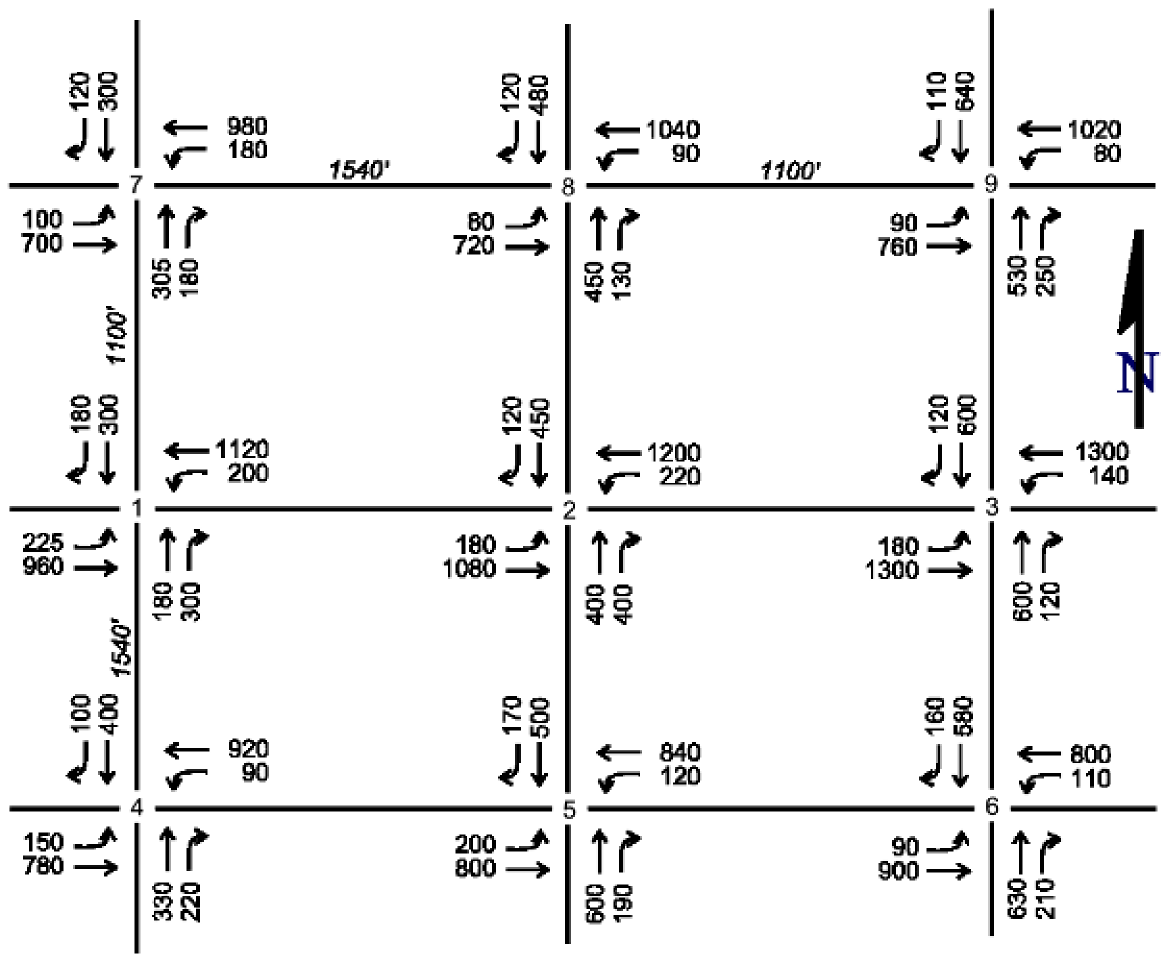


Figure 6.7 Network 3B – Demand Volumes and Intersection Spacing

Results and Discussion

As average delay is the most commonly referenced performance measure it will be the primary focus of this results discussion. Although the interested reader may refer to Appendix B, where for each replicate run (five replicates were conducted for each volume scenario) average delay, maximum delay, average queue, and maximum queue may be found per intersection / per movement / per lane. As stated earlier three different control types are tested for each network demand scenario: Offline (optimized), COP, and Constrained COP. The COP and Constrained COP timings were determined as discussed in the prior sections of this chapter while the offline timings were determined using TRANSYT-7F.

0 through 0 present the average delay for each intersection, for each of the network scenarios described. Each figure contains two diagrams. Diagram A displays the average delay for each intersection and its associated 95% confidence interval, while diagram B displays the average delay for each replicate run. Diagram A also contains the overall average network delay for each control strategy.

Network 1 Results – 0 gives the average network and intersection delays for network 1A, where each intersection is under the same demand condition. It is seen that COP provides the best overall service with an average network delay of 19.2 sec/veh, followed by the offline solution with a delay of 24.2 sec/veh, and finally Constrained COP with a delay of 28.8 sec/veh. As expected the most flexible strategy, COP, provided the best average delay. Unfortunately Constrained COP did not perform as well as hoped, providing worse service than both COP and the offline solution. Although, possibly more interesting than the average delay is the replication delay variability captured by Open-TS3. From 0 it is seen that the Constrained COP and COP replication delays at each intersection encompass a significantly wider range than the offline solution. This is a demonstration of one of the primary strengths of offline signal timing approaches. While the offline strategy may not provide the lowest average delay it does provide the most predictable, with average intersection delays for each replication falling within a range of just a few seconds. These results are reflective of prior real world experiences with adaptive control. Open-TS3 has successfully captured the behavior commonly associated with adaptive control – good average performance but a wide variability in actual realizations.

The network 1B and 1C results also show the lowest average network delays under COP, followed by the offline solution, and finally Constrained COP. Although, in these networks the difference between the lowest average network delay (COP) and highest (Constrained COP) is only slightly more than two seconds, in both 1B and 1C. This is a barely distinguishable difference. At the intersection level variability seems to mask the small differences among the control schemes and there is no overwhelmingly notable performance trend. It is again important to note that the Constrained COP and COP intersection replication delays have significantly wider ranges than the offline control, reflecting again the experiences of other researchers and practitioners that adaptive control may on average provide better overall service but lacks consistency leading to on overall less desirable solution that the offline "less intelligent" solutions.

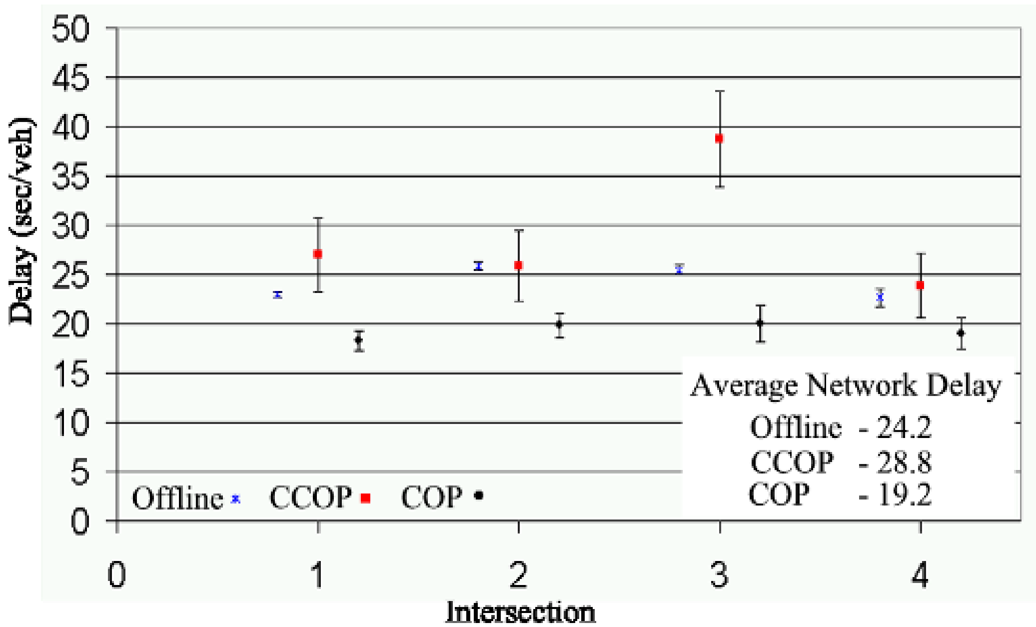
Network 2 Results- Network 2 increases the complexity of the system. In both 2A and 2B (0 and 0, respectively) demand scenarios COP again provides the best average performance. Although unlike the network 1 scenarios Constrained COP now provides the next best average network delay, followed by the offline solution. This is as one would expect since as a system becomes more complicated it becomes increasing difficult to provide highly desirable offline solutions. Although, as with network 1B and 1C these delay differences, particularly between the offline and Constrained COP strategies are relatively minor.

The key observation again lies in the intersection delay variability seen between replicate runs. Even with the increased complexity the offline control continues to provide predictable operations. On the lower demand North-South arterial, COP and Constrained COP also both provide predictable performance, with an average delay almost always lower than the offline solution. The heavier demand on the East-West arterial leads to less consistent adaptive control performance, with intersection 3 (where the two roads cross) providing some of the widest ranges and highest delay values.

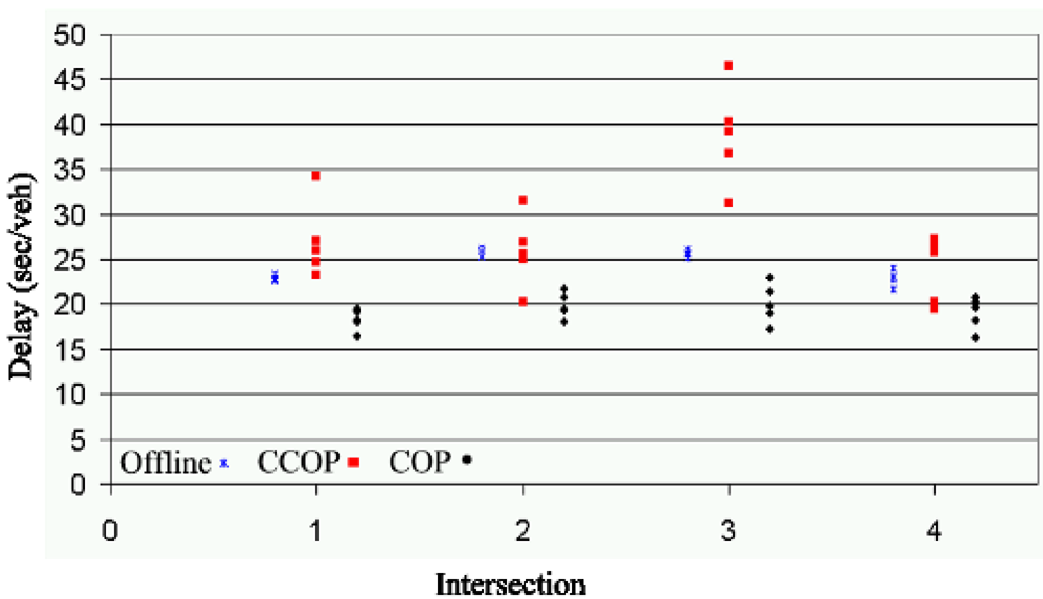
Network 3 Results – Network 3 is the most complicated network tested. From 0 and 0 behavior is seen similar to that for network 2. The best overall performance is provided by COP with very similar overall performance from Constrained COP and the offline control. Also, it is again seen that the adaptive control strategies provide a significantly wider range of delays at the intersections over the replicate runs. Some of the largest ranges in intersection replicate run delays are seen on this network.

Results Summary – As seen in all of the networks the COP adaptive control approach provided the best average network delay with Constrained COP and the Offline control alternating as second best, but for the most part providing very similar average network delays. The more interesting observations were seen at the intersection level where the offline control consistently provided very predictable delays while the adaptive control intersection delay replications could cover a wide range. Interestingly, as the complexity of the networks increased variability in the offline solutions remains fairly constant while the adaptive control tended to increased variability with increase complexity.

As has been stated in this chapter and discussed in the previous these results are inline with other adaptive control studies. This has further increased the confidence in Open-TS3's ability to accurately capture traffic system performance, even when modeling advance traffic control. Unfortunately Constrained COP did not meet one of is underlying goals – a reduction in the performance variability. There are many potential causes for this including: the limited number of phases, length of planning horizon, selection of background cycle, length of offset window, etc. It is also possible that such variability will simply always be inherent in adaptive control strategies. This initial testing of Constrained COP (and adaptive control in general) has definitely demonstrated the need for additional research and investigation. Fortunately, Open-TS3 has shown its usefulness in such an investigation and will be an important part in the efficient conduct of future adaptive control research.



(A) Network 1A - Average Intersection Delay with 95% Confidence Interval



(B) Network 1A - Average Intersection Delay for Each Replicate Run

Figure 6.8 Network 1A Intersection Delay, (A) Intersection Average with 95% C.I., (B) Average for each Replicate Run

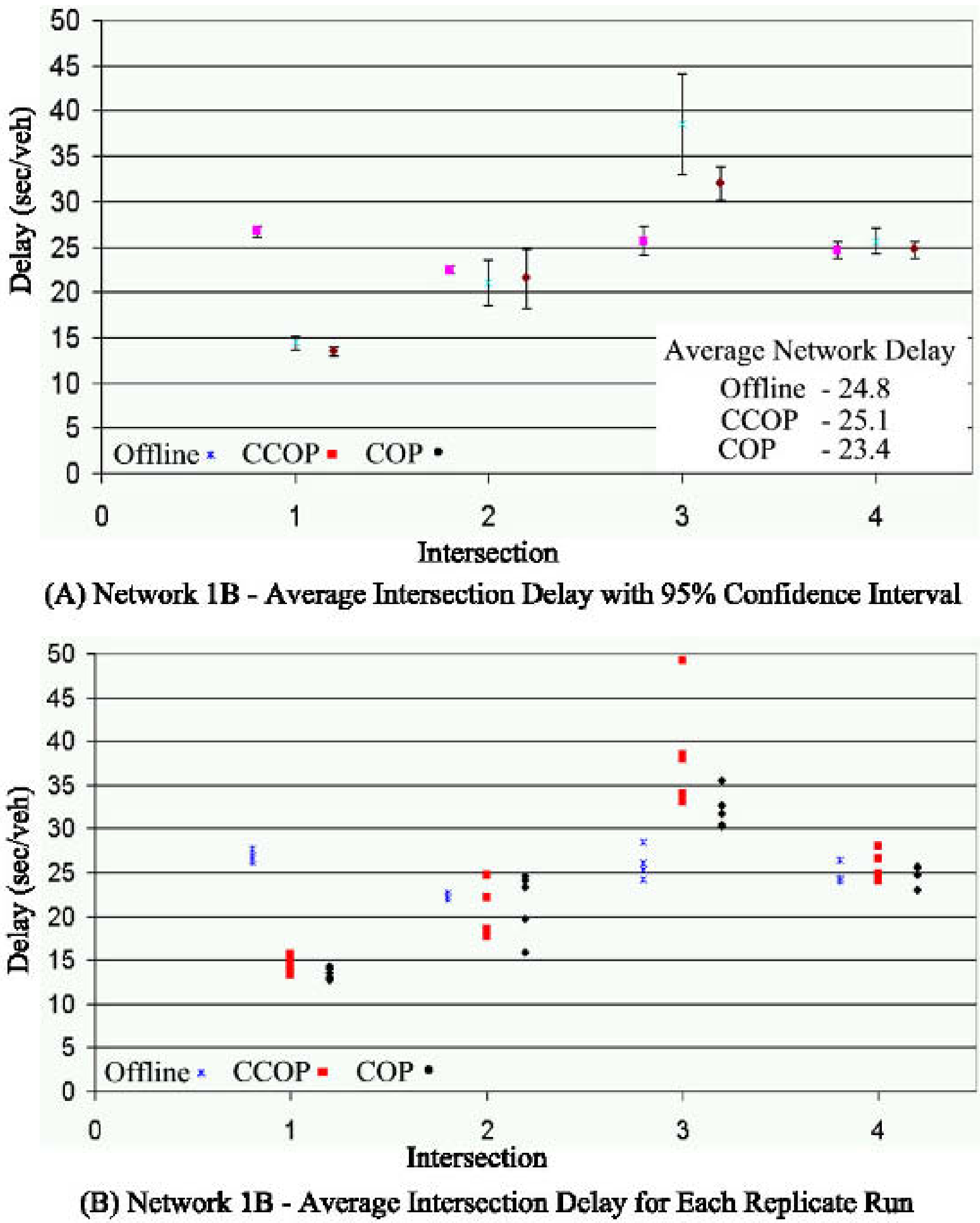
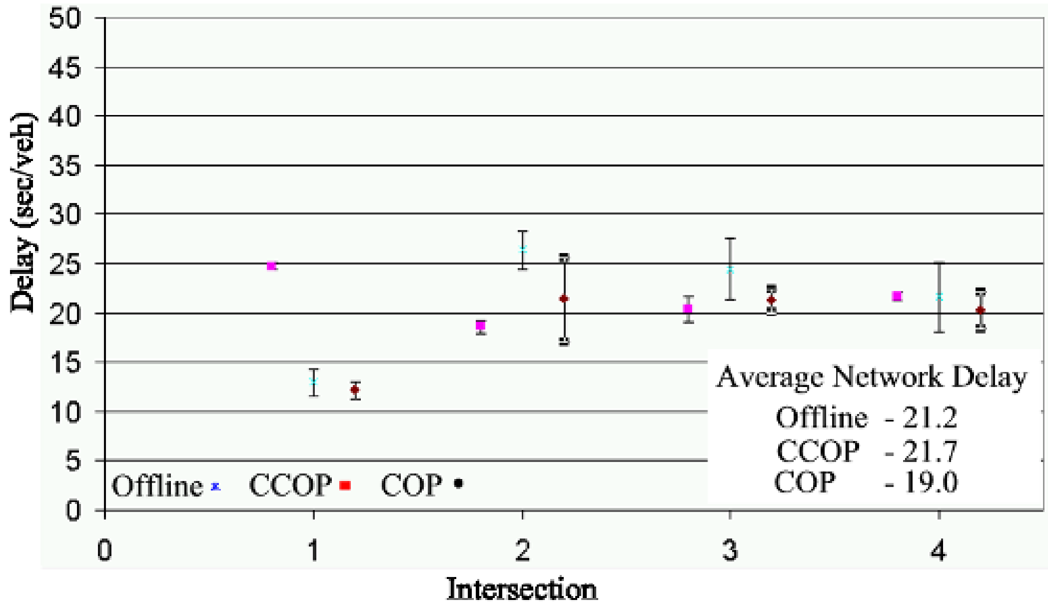
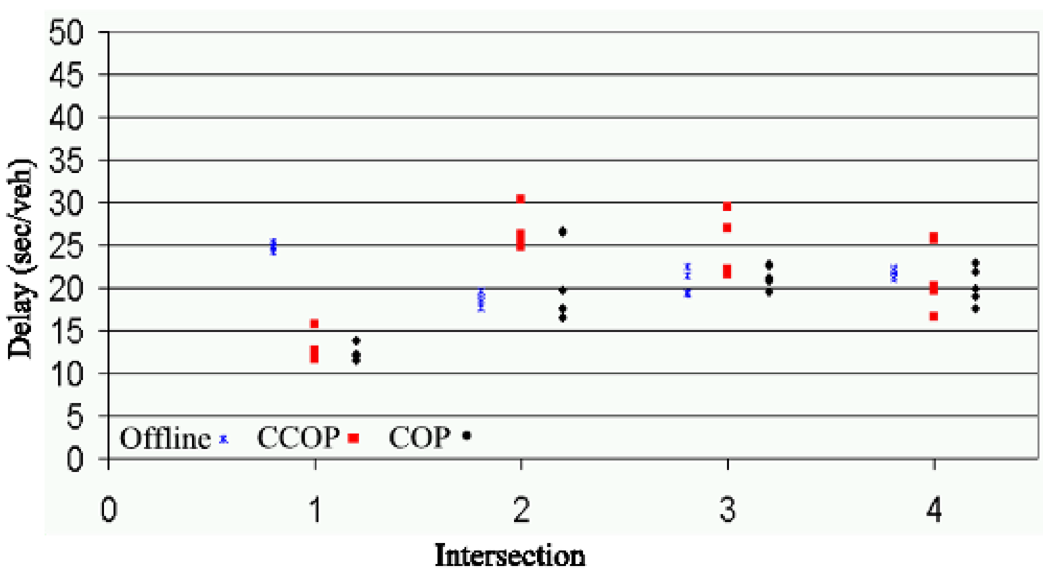


Figure 6.9 Network 1B Intersection Delay, (A) Intersection Average with 95% C.I., (B) Average for each Replicate Run

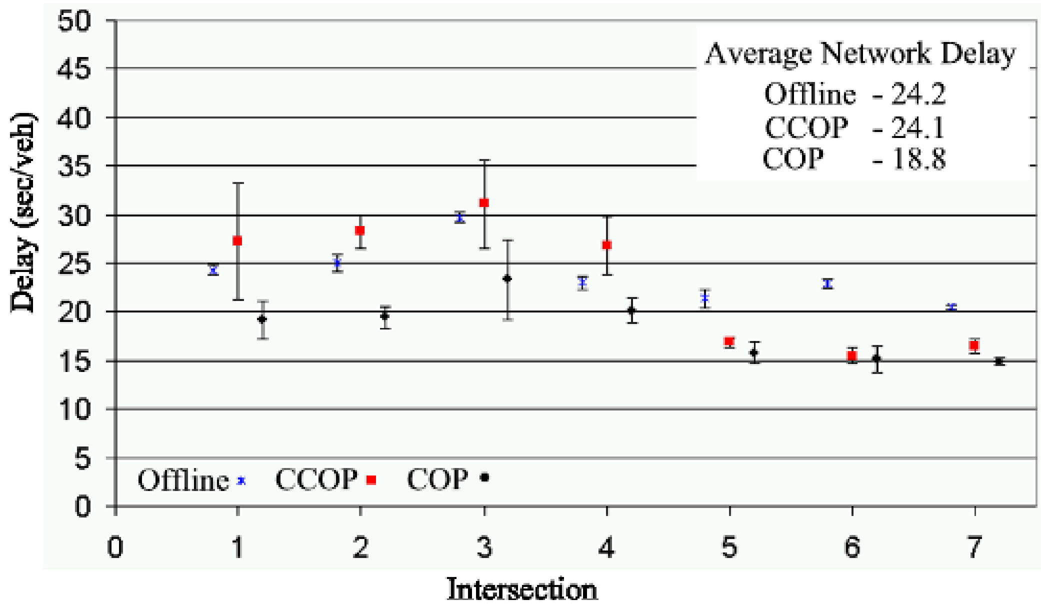


(A) Network 1C - Average Intersection Delay with 95% Confidence Interval

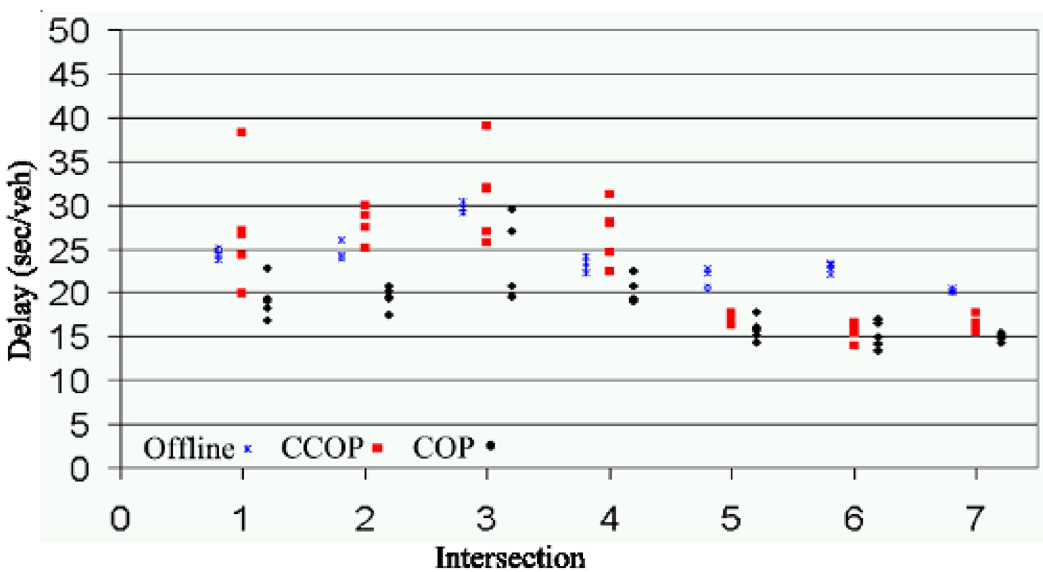


(B) Network 1C - Average Intersection Delay for Each Replicate Run

Figure 6.10 Network 1C Intersection Delay, (A) Intersection Average with 95% C.I., (B) Average for each Replicate Run

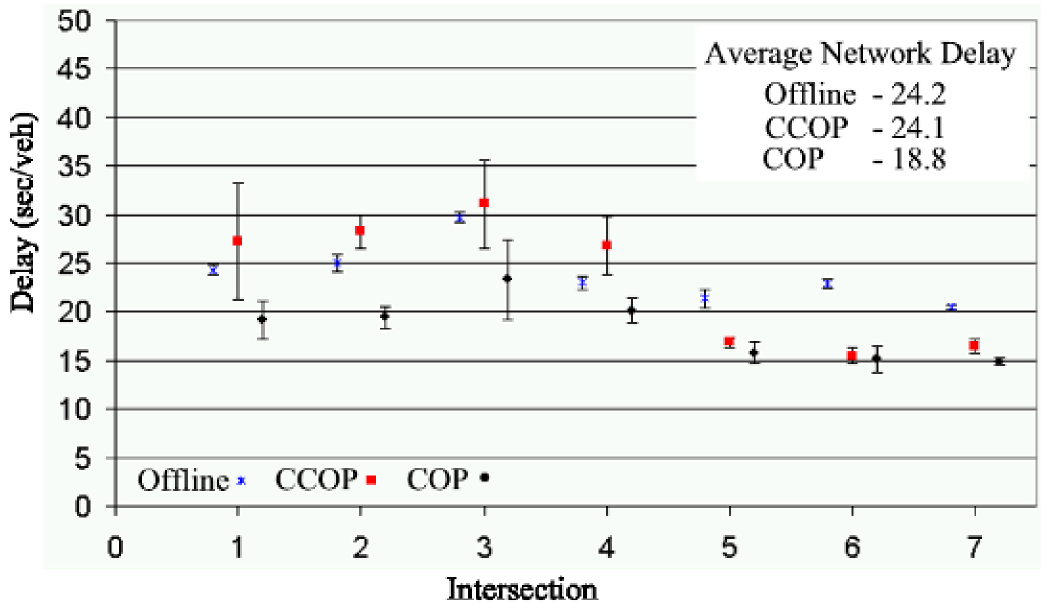


(A) Network 2A - Average Intersection Delay with 95% Confidence Interval

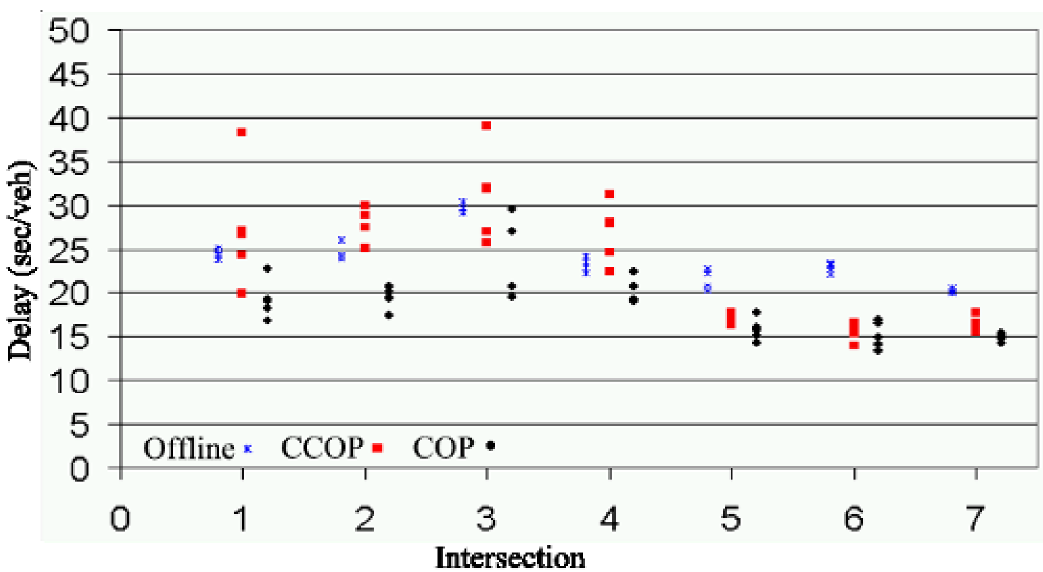


(B) Network 2A - Average Intersection Delay for Each Replicate Run

Figure 6.11 Network 2A Intersection Delay, (A) Intersection Average with 95% C.I., (B) Average for each Replicate Run

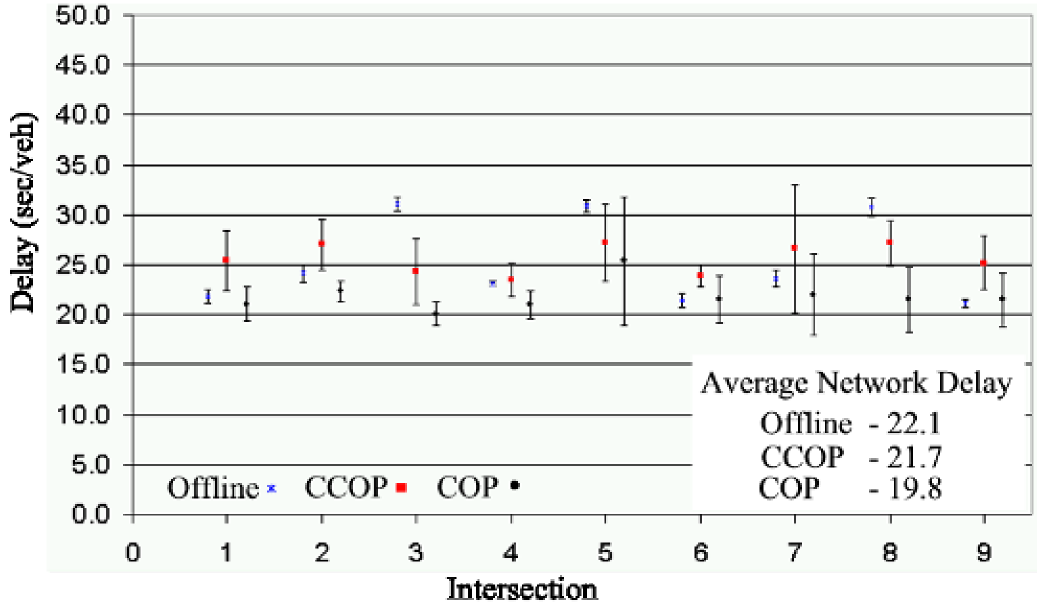


(A) Network 2A - Average Intersection Delay with 95% Confidence Interval

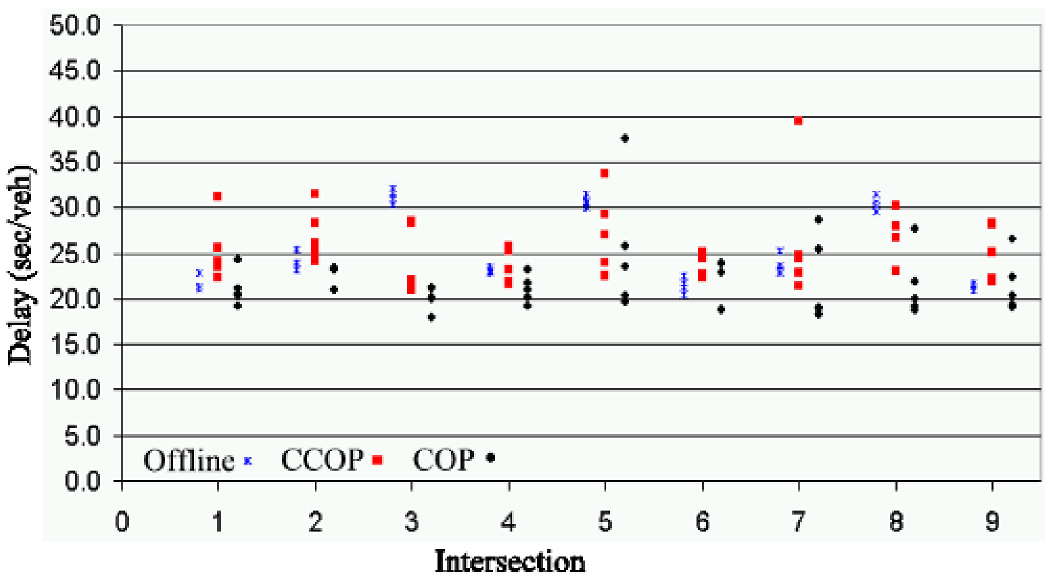


(B) Network 2A - Average Intersection Delay for Each Replicate Run

Figure 6.12 Network 2B Intersection Delay, (A) Intersection Average with 95% C.I., (B) Average for each Replicate Run

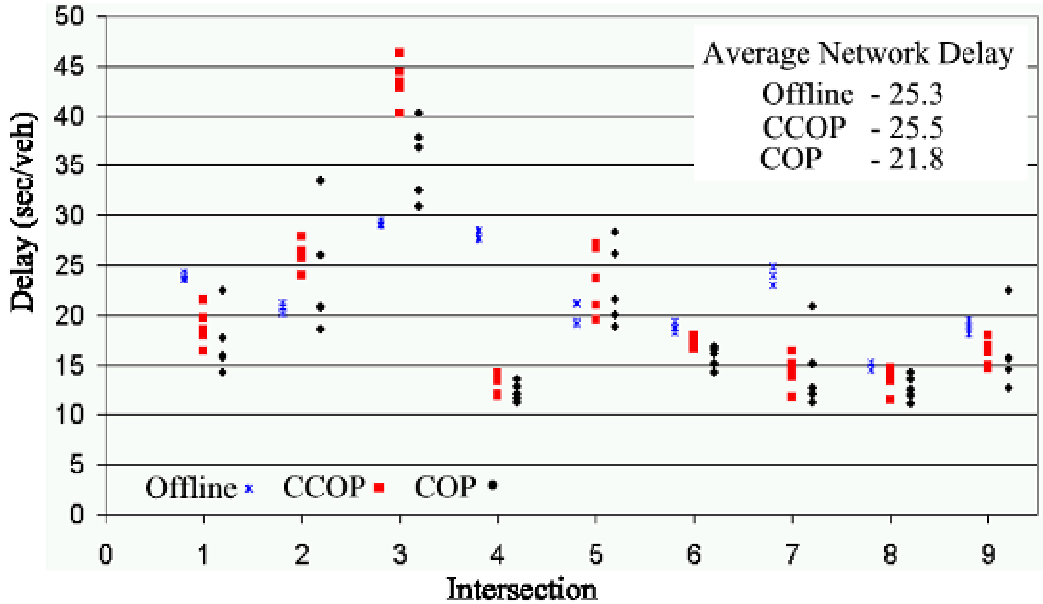


(A) Network 3A - Average Intersection Delay with 95% Confidence Interval

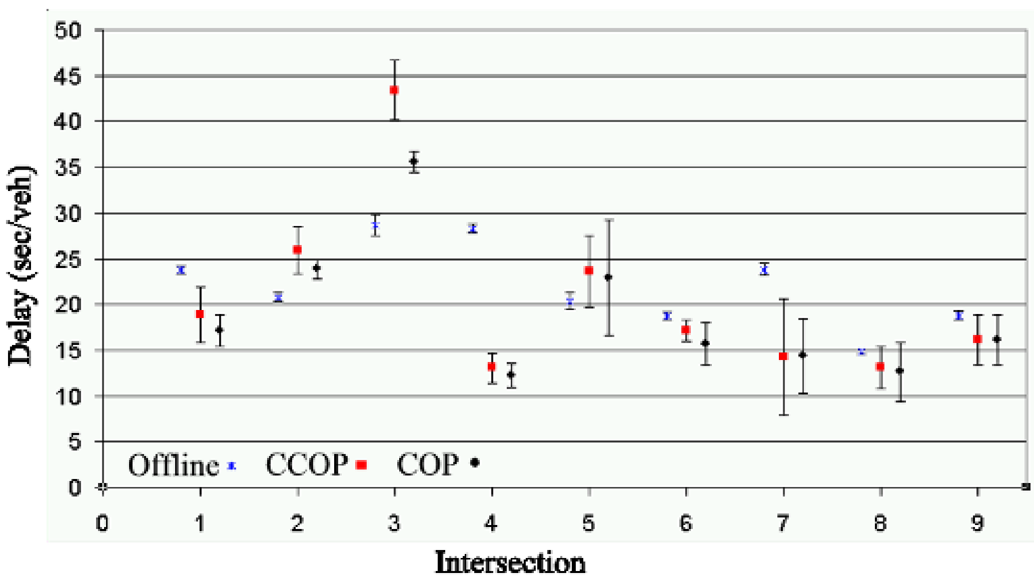


(B) Network 3A - Average Intersection Delay for Each Replicate Run

Figure 6.13 Network 3A Intersection Delay, (A) Intersection Average with 95% C.I., (B) Average for each Replicate Run



(A) Network 3B - Average Intersection Delay with 95% Confidence Interval



(B) Network 3B - Average Intersection Delay for Each Replicate Run

Figure 6.14 Network 3B Intersection Delay, (A) Intersection Average with 95% C.I., (B) Average for each Replicate Run

CONCLUSION

Through out this report the development and testing of Open-TS3 has been presented. This chapter further demonstrated that Open-TS3 is reliable and flexible, able to be used as the underling tool for the analysis of advance traffic control strategies. Presented in this chapter was dynamic programming, the rolling horizon concept, and two adaptive control strategies: COP and the developed Constrained COP. Importantly, it was seen that these concepts could all be implemented into Open-TS3 and readily tested. Once the basic intersection model block was developed, creating the test networks and demand scenarios was a relatively simple and straightforward task. As a result of these initial adaptive control tests it was seen that adaptive control can provide superior overall performance, but can also have a significantly greater range of variability than that of pre-timed offline control. This implementation of adaptive control into Open-TS3 has clearly demonstrated Open-TS3's flexibility, openness, and reliability.

CHAPTER 7 CONCLUSIONS AND RECOMMENDATIONS

INTRODUCTION

At the beginning of this report, the current state of simulation in transportation was discussed. It was seen that over the years, several trade-offs have been made to gain the speed and efficiency required for simulations to be cost-effective and gain general acceptance in transportation practice. Transportation simulation packages have become “black boxes” and little knowledge of the underlying principles, strengths, and weaknesses of the models is required for a user to create simulations. Even when detailed, up-to-date documentation is desired, it is often difficult to obtain and difficult to comprehend. Also, today’s transportation simulation packages are often inflexible in terms of the type of scenarios they are able to evaluate. The likelihood of an end user being able to conceptualize, design, and develop a simulation beyond the bounds set by the simulation package is extremely low.

The objective of this research effort was investigate the feasibility of constructing a transportation simulation approach that minimizes the black box problem and increases modeling flexibility while still providing an easy-to-use package in which typical, straightforward models may be quickly and accurately built. The developed approach is Open-TS3 (Open Architecture Transportation Simulation, 2003). Open-TS3 satisfies the stated objective through four main accomplishments, which are described below. Following these accomplishments is a discussion of the limitations of Open-TS3 and the general modeling approach undertaken, followed by the next steps that may be undertaken in the development of Open-TS3

ACCOMPLISHMENTS

Review of Existing Transportation Simulations And Signal Control Methodologies

The first task completed was a review of simulation practice in transportation and simulation in general (Chapter 2). This review provided an understanding of the strengths and weaknesses of transportation simulation as practiced today. Importantly, this review also presented general simulation concepts, beyond strict transportation applications. It is from this general review of simulation that the foundation for Open-TS3 arises, with the simulation language SIMAN and application package ARENA.

Also conducted is a review of existing research and practice in signal control, including pre-timed, semi-actuated, actuated, and adaptive control (Chapter 5). The current focus of Open-TS3 is arterial simulation, thus an understanding of the primary elements of arterial traffic control is crucial. Also, real-time adaptive signal control, an area of advanced research in traffic control, highlights the difficulties with existing transportation simulation. Utilizing existing transportation packages for the analysis of adaptive signal control has proven extremely difficult and time-consuming. The implementation of adaptive control in Open-TS3 would be a clear indication of the value of Open-TS3.

Development Of A Simulation Test Bed (Open-TS3) With Intuitive User Interfaces And Transparent, Ready Access To Internal Simulation Model Operations

This accomplishment is the main contribution of this research effort: the development of Open-TS3. Open-TS3 utilizes the SIMAN simulation modeling language, and ARENA, a graphical user interface for SIMAN. Open-TS3 is an intuitive and flexible simulation modeling architecture, utilizing a hierarchical object-oriented approach to simulation.

Open-TS3 may be briefly described as follows:

Transportation – Currently individual intersections, arterials and networks may be analyzed with Open-TS3, under pre-timed, actuated, and adaptive control. Future extensions of Open-TS3 are likely to include freeways, toll plazas and HOV/HOT lanes, ports, and multi-modal facilities.

Hierarchical – Open-TS3 consists of three tiers of blocks (objects). The top tier blocks may be combined to create an intersection, arterial, or network simulation model. Each of these tier 1 blocks is constructed from hierarchy tier 2 and tier 3 blocks. Tier 3 blocks (the lowest tier) are the basic SIMAN constructs.

Event-Based – Time progresses from event to event, not continuously or through pre-selected time steps.

Object-Oriented – In this approach, one first considers the simulation in terms of objects and how those objects interact with each other, rather than coding structures or program implementation.

Stochastic – While a Open-TS3 model may be specified to be completely deterministic, it is also possible for a user to introduce randomness.

Simulation – Finally, Open-TS3 is a simulation, i.e. a computer representation of a real world system (currently existing or planned) that attempts to mimic the behavior of the real world system.

In constructing a simulation, users can employ the developed blocks, or develop new modeling constructs. Currently ten primary blocks have been developed which users can use to readily and efficiently model a typical traffic network.

ENTER – All vehicles enter a model through this block.

EXIT – All vehicles leave a model through this block and final vehicle statistics are collected.

PRETIMED – One of the four intersection blocks, this block utilizes a two or three phase pre-timed signal logic, modeling three-way and four-way signalized intersections and protected-only left turn phasing.

PRETIMED8P – Another intersection block, however, this block incorporates an eight-phase dual-ring pre-timed signal logic.

ACTUATED8P – This intersection block models an eight-phase dual-ring actuated signal logic.

SIGNAL – This last intersection block allows for the simulation of adaptive signal control logic.

QUEUECHANGE – This block contains the logic by which a vehicle may choose to change lanes to enter a shorter queue.

TURNBAY – This block controls the logic by which vehicles are assigned to a turn movement at an intersection approach.

LANEADD – This block may be used to increase the number of lanes on a link.

LANEDROP – This block is used to reduce the number of lanes on a link.

Other lower level blocks have also been developed and are utilized within the ten blocks listed above.

As a result of its approach to simulation, Open-TS3 has many advantages. Firstly, within the limitations in the following section, it is readily expandable to other aspects of the transportation system. Much of this expansion potential is a result of Open-TS3's hierarchical, object-oriented structure. Users may develop their own blocks and employ them with the existing Open-TS3 blocks. This "open architecture" approach frees a user from a dependence on Open-TS3 developers.

The hierarchical nature of Open-TS3 also allows for a minimal learning curve to initial model construction. One may quickly become efficient with tier 1 blocks, learning as little or as much as desired about the underlying logic (tiers 2 and 3) and still be able to construct a realistic, usable model. As users desire to expand beyond the default tier 1 blocks, they can learn and experiment with tier 2 and 3 blocks, performing more unique analyses.

Finally, the object-oriented approach to modeling represents a more "common sense" approach to simulation, utilizing people's existing natural mechanisms for viewing their surroundings. This increases the likelihood of creating a more intuitive, understandable, efficient, and accurate simulation software package (Brown, 1997).

Validation of Simulation Through The Use of Real World Data

The purpose of this task was to insure that Open-TS3 fulfilled its primary initial requirement: reasonably modeling real-world signalized networks. Utilizing the current Open-TS3 model blocks, Open-TS3 was successfully compared against both the well-respected CORSIM model and real-world data. The CORSIM validation studies included comparisons of simulated volumes, delays, queues, and speeds. Excellent agreement was seen between Open-TS3 and CORSIM, except in two instances: overcapacity demand scenarios and potentially left-turn operations.

As stated, Open-TS3 was also compared against real-world conditions. Based on data collected for a twelve-intersection network in downtown Chicago, Open-TS3 was able to reasonably capture real-world conditions. As part of this validation, Open-TS3 was also useful in identifying potential pitfalls in the data set. The real-world validation process demonstrated that one must be careful to understand the limitations and nature of real-world data. Also, it was seen that there is a great dearth in usable, available data sets.

Development and Implementation of Adaptive Control Strategies in The Simulation and Test On Several Networks Under Several Traffic Flow Conditions

As a final step in this initial Open-TS3 development, the openness and flexibility of Open-TS3 was highlighted by incorporating adaptive signal control. The feasibility of implementing an existing adaptive control strategy and enhancements into Open-TS3 was demonstrated. Once the basic adaptive control intersection model block was developed, it is a relatively straightforward task to create adaptive control test networks. Based on these test networks, adaptive control was shown to be capable of providing superior average network performance when compared to pre-timed control. However, adaptive control can also have a significantly greater range of performance variability.

LIMITATIONS.

The development and testing of Open-TS3 has led to an understanding of limitations in the model, both in real-world traffic features (either not captured or poorly captured), and in general limitations to the simulation approach.

Traffic Features

Currently Open-TS3 is limited to typical arterial networks. Features not modeled include: permissive phasing, sign control (i.e. yield and stop signs), vehicle overtaking (i.e. vehicles may not pass a slower moving vehicle in the same lane), and pedestrian or alternative modes. Also, the lane changing and shared lane logic are based upon idealized assumptions, not directly modeling vehicle conflicts. As stated, the transportation system beyond typical arterial features is not captured, including freeway operations, roundabouts, and ITS based information and control concepts. Open-TS3 is still far from the capabilities of commercial transportation simulation packages, such as CORSIM and VISSIM, and should not be considered a competitor in the commercial market place. The primary area of current usage for Open-TS3 is in simulation research and development.

Vehicle Queuing

Perhaps one of the most significant limitations to the arterial model is the current use of vertical queuing. Vertical queues fail to adequately capture several aspects of realistic system performance. A vertical queue fails to limit the queue length by the approach link length, potentially inflating the link delays. In addition, the upstream intersection movements feeding the downstream queue may have a lower delay as vehicles will be allowed to enter a downstream link even when the downstream queue length exceeds the link length. Likewise, the upstream crossing movements would not be blocked by spillback. On the approach, the vertical queue allows for vehicles to enter a turn bay when in actuality the turn bay may be full or the vehicle may be trapped in its current queue, upstream of the turn bay. The lack of a horizontal queuing model is one of the primary contributing factors to the current inability of Open-TS3 to accurately model congestion on networks and must be addressed before the model has the possibility of becoming useful to a wide audience.

Event-Based

Possibly the most daunting general limitation is the underlying event-based nature of this approach. While event-based simulation is well suited to modeling signal control, it is not nearly as apt at capturing the interactions that take place at a microscopic level in traffic flow. This weakness will become particularly constraining when attempting to model freeways and to implement detailed car-following models. Also, as stated previously, Open-TS3 currently utilizes vertical queuing on the approaches. The event-based nature of SIMAN is the underlying reason for the use of this queuing model rather than horizontal queuing. Future effort on Open-TS3 will include the incorporation of time-based simulation. Initial efforts will center on incorporating time-based modeling into the current ARENA platform although if this is not possible it will be necessary to move the hierarchical, object-oriented constructs, to an alternative platform, if the simulation approach is to be further advanced. Until this drawback is overcome Open-TS3 will not be able to fully contribute to, or take advantage of, larger simulation efforts, such as NGSIM (Next Generation Simulation).

Scalability

Another limitation is the scalability of the model. Both the computational time and potential maximum model size must be considered. The additional overhead of the ARENA simulation package will not allow for the construction of simulations of the size and speed that may be generated directly by lower level languages such as C++. Nonetheless, while not currently optimized for run-time efficiency initial investigations of computational speed lead to performance similar to that of CORSIM. For example, for the Chicago network (23 intersections), 30 replications in CORSIM required 180.25 seconds of CPU time while 30 replications in Open-TS3 required 190.5 seconds (runtimes are for simulation without the generation of animation, utilizing internal scripts for generation of replications). For the smaller networks utilized in the comparison of Open-TS3 and CORSIM, similar runtimes were also seen. (CPU time measurements where conducted on a Dell Optiplex GX260, with a 2.26 Ghz processor and 1GB of RAM.)

Maximum model size is more difficult to gauge. ARENA has no technical limit on the number of blocks or entities that may be incorporated into a model. The model size limitations are dependent on the available computer resourses. For example a sixteen-intersection model, on the same computer as in the above CPU times, utilized approximately 15% of the available RAM. A reasonable expectation of the limitation in model size is approximately 100 intersections. As models become more complex, particularly items requiring dynamic link libraries and additional coding (such as adaptive control) the maximum model size will decrease. It is unlikely, even with optimization of the Open-TS3 object logic, that network sizes comparable to those in transportation specific simulations will be achievable. (For example CORSIM allows for up to 500 intersections with up to 20,000 vehicles on a network at any instant.)

Open Architecture

While this approach attempts to open the “black box” by allowing the user to add to and alter the underlying objects, it must not be assumed that this will be a simple task. To fully understand the model constructs, the user will have to devote time and effort into gaining an understanding of general simulation development and the underlying SIMAN language. Without this effort a user may still construct complex models using the tier 1 blocks, but the model will be no less a “black box” than the other available simulation packages.

Also, as users take advantage of the open structure of the model it will become increasingly difficult to maintain the model and compare results from different Open-TS3 models. Each user will essentially be developing their own version. Validation becomes even more difficult as one user’s results may differ form another due to individual model logic changes.

Validation Data

As a last point, the effort to obtain real-world data to compare to the simulation highlighted a lack of available real-world data. There is a clear need for field studies with the expressed goal of developing data collection guidelines and obtaining data sets for the validation of transportation analytical and simulation models. This would not only be useful for validation of the proposed simulation model but also would provide a means for direct validation of the many other simulations that are used in practice today.

In addition guidelines are needed to gauge how well a model captures real-world performance. This effort relied on the traffic engineering expertise and judgment of the author. The development of an evaluation methodology in which practical significance, representing the difference between real-world data and simulation results, can be evaluated in an unbiased manner is important area in which future research efforts should be undertaken.

RECOMMENDATION FOR FUTURE OPEN-TS3 DEVELOPMENT

There are numerous steps that may be undertaken in future Open-TS3 model development. The following is a listing and brief discussion of a few of the possible updates.

Incorporation of Time-Based Modeling – As stated in the limitations section the event-based nature of the model has lead to numerous other limitations. One of the first areas of future development must be incorporation of some form of time-based simulation.

Refine for Overcapacity Demand Scenarios - Overcapacity scenarios were one of the primary areas of difficulty for Open-TS3. Future study should be conducted to improve overcapacity demand model operations. This work should include the development of a horizontal queuing model to replace the vertical queuing model, and ability of the model to capture spillback into upstream intersections.

Study Left-Turn Operation - A study of left-turn operation should be included in any real world traffic validation. In the validation studies, CORSIM was significantly more aggressive in processing left-turn vehicles than Open-TS3 in higher demand scenarios. It is important that the Open-TS3 left-turn model be validated with real world data and, if necessary, reconstructed to provide superior left-turn treatment.

Permissive Signal Control and Sign Control - Currently, the only intersection control modeled in Open-TS3 is protected-only phasing. Modules should be developed for permissive signal control and sign (stop and yield) control. This is a critical step that must be accomplished before Open-TS3 will experience widespread use.

Shared Lane Use - In conjunction with permissive signal control, the ability to model shared lane use should also be developed. Currently, each lane at an intersection is limited to one movement type. The usefulness of Open-TS3 will greatly increase when shared lanes (i.e thru/right, thru/left, left/right) are permissible.

Increase Driver/Vehicle Diversity - A relatively simple improvement to future versions of Open-TS3 will be to offer additional vehicle calculations. Currently all driver and vehicle variation is captured in a single driver aggressiveness factor. Additional study may lead to the implementation of a driver / vehicle population that more realistically reflects the real world.

Origin / Destination Matrices – Future versions of Open-TS3 could be set to accept O-D matrices, when such data is available. Instead of randomly assigning a vehicle's movement at each intersection, vehicle paths could be assigned a priori. Such an improvement creates access to additional methods of data collection and network analysis and a greater possible set of recommended system improvements.

LINK Block - Possibly one of the most fundamental enhancements to Open-TS3 object set is the creation of a LINK block. Currently, upon exiting an intersection a vehicle is assigned a travel time to the next downstream block. Possible lane changing and interaction between vehicles on the link is not directly modeled, except for guaranteeing that the downstream order of vehicle arrivals matches upstream departures and that minimum headway constraints are satisfied. A LINK block would allow for a direct modeling of the vehicles as they travel between intersections. Such a LINK block could utilize many different forms, including simple Newtonian mechanics, car-following, rule-based algorithms, or Cellular Automata.

Inclusion of Other Transportation System Features – Open-TS3 modules can be created to simulate much more than the traditional traffic network, including features such as toll booths, parking lots, freeways, weigh stations, HOV lanes, transit, rail, and air.

Additional Real World Traffic Validation Study

In addition to Open-TS3 advance general efforts should be made in the area of traffic data collection. While the real-world validation study undertaken in this report was extremely useful, it also highlighted the great lack of quality data that exists today. Data sets that include delays, saturation flows, acceleration/deceleration, speeds, volumes, and signal behavior will be of use not only to the validation of Open-TS3 but also to the validation of other simulations and transportation simulations.

CONCLUSION AND CONTRIBUTION

The work that has been presented in this report makes several contributions to the transportation field. Initially these contributions have the potential to aid academics and researches while they lay the foundation for future contributions to engineering practitioners.

Academics and researchers are aided in several ways. Open-TS3 provides an intuitive, flexible, and open simulation modeling architecture that may be applied to problems other than those studied in this report, outside of traditional transportation analysis. The capability to expand the abilities of Open-TS3 is not limited to any particular developer. Open-TS3 opens the "black box" of transportation simulation modeling while maintaining the capabilities for which the box was initially created. Also, Open-TS3 has proven to be capable for the study of adaptive signal control. As technology available in the field increases and new approaches to traffic control are developed, a simulation with the characteristics of Open-TS3 will be invaluable.

As additional features are generated transportation engineers and modelers will be increasing benefited. Using the Open-TS3 approach, transportation professionals will be able to readily and efficiently construct simulation models (such as those built with current "black box" models) using the existing Open-TS3 blocks, while maintaining the ability to develop model objects for transportation features not currently included. This flexibility and ease of use is not currently found among common transportation analysis tools.

Importantly, transport system users (drivers, passengers, pedestrians, etc.) also may receive long-term benefits from this research. As stated, Open-TS3 may aid the transportation professional (academics, researchers, and practitioners) leading to the creation of an improved transportation system, the ultimate goal.

Appendix A

Random Number Generation in ARENA

A.1 INTRODUCTION

The random number generator in ARENA is expected to create a stream of independent and identically distributed (IID) observations, X_1, X_2, X_3... from a continuous, uniform distribution between 0 and 1 (U(0,1)). In an ARENA simulation these observations are referred to as a random number stream. Random number streams are crucial because they are used in the generation of random variates (draws from a particular distribution, i.e., Poisson, Exponential, etc.). Random variate generation from a random number stream is typically accomplished through the application of an inverted cumulative distribution algorithm (224,225). A critical result of this process is that the quality of random variates generated during an ARENA simulation is directly related to the quality of random number streams. If the random number streams do not resemble a uniform distribution, or are correlated in some manner, then the generated random variates also become highly suspicious, skewing simulation results. Therefore, it is crucial before using a simulation that the random number generator be tested to assure the generated streams appear as IID U(0,1) random variates. Thus, this appendix presents the ARENA random number generator and the results of several empirical tests on the random number streams.

A.2 ARENA RANDOM NUMBER GENERATOR

The random number generator used by ARENA is a Linear Congruential Generator (LCG) (224). A LCG generates a sequence of random integers Z_1, Z_2, Z_3... using the recursion listed in equation (1).

$$Z_i = (aZ_{i-1} + c) \bmod m \quad (1)$$

where: a, c, and m are user supplied constants

Z_o is a user supplied initialization seed

The modulus m operation on $(aZ_{i-1} + c)$ in the recursion formula guarantees that Z_i is an integer, where $0 <= Z_i <= m-1$. Thus, dividing Z_i by m results in a sequence of real numbers, U_1, U_2, U_3..., where $0 <= U_i < 1$. If a, c and m are chosen well, this sequence will resemble a set of (pseudo-) random numbers, IID U(0,1). The constant values utilized in ARENA are $m = 2^{31} - 1$, a = 16807 and c = 0. ARENA supplies 10 different default seeds (Z_o) providing ten different streams of random numbers. A user may select additional random number seeds if desired.

A.3 EMPIRICAL TESTS OF ARENA RANDOM NUMBERS

Four different empirical tests are performed on the default random number streams to test how well the generated random numbers resemble IID U(0,1) random variates. These tests examine (1) if the random numbers appear uniformly distributed between 0 and 1, (2) how well uniformity is held in two and three dimensions, (3) the independence of consecutive numbers in a stream, and (4) if the stream displays any discernible correlation. The remainder of this appendix presents a brief discussion of each of these tests, followed by a presentation and discussion of results for the ten streams tested. Although, before this, one general caveat to these tests should be mentioned.

It must be remembered that these tests are all empirical. That is, they are tests on actual streams of random numbers produced by the generator and therefore do not supply global guarantees. If a particular stream(s) is found to resemble IID U(0,1), nothing may be definitively concluded about all possible generated number streams. While global (theoretical based) tests do exist they unfortunately have the opposite problem in that they do not definitely indicate how well a specific stream will behave. It is arguable whether global or local tests are preferable (225) although for this research the empirical tests where chosen as they will test the actual random number streams to be used. Also, as numerous generated streams are found to be acceptable, one's confidence in the overall generator certainly increases. For references to global theoretical tests on the LCG used by ARENA the reader is referred to Kelton, Sadowski, and Sadowski. The primary source for sections A.2.1 through A.2.4, presentation of the empirical test methodologies, is Law and Kelton(225), chapter 7. The reader is referred to this text if further discussion is desired.

A.3.1 Empirical Test 1, Random Numbers Uniformly Distributed between 0 and 1

The first empirical test examines if the random numbers (U_1, U_2, U_3...) appear to be uniformly distributed over the range [0,1]. This test is a chi-squared test with all parameters known, U(0,1). In this test the range [0,1] is divided into k subintervals of equal length. A stream of n random numbers is generated and fj, where j = 1 to k, is taken to be the number of random numbers that fall into subinterval j. For the null hypothesis that the random numbers are IID U(0,1) random variates the test statistic x^2, shown in equation (2), will have a chi-squared distribution with k-1 degrees of freedom. Thus, if the value of $x^2 > x^2_{k-1,1-a}$ the null hypothesis may not be accepted and the random variates would be said to not be uniform, at a level of 1- a.

$$x^2 = \frac{k}{n}\sum_{j=1}^{k}\left(f_i - \frac{n}{k}\right)^2 \qquad (2)$$

This test was applied to the ten generated random number streams. For each test k was taken as 4096 ($k = 2^{12}$, the most significant 12 bits of the random numbers are being examined for uniformity) and n as 32,768 ($n = 2^{15}$, allowing for an average of eight numbers per interval). For an a of 0.10 (test at a level of 90%), the chi-squared critical point is, $X^2_{4095,0.90} = 4211.4$.

The first row of Table A.l gives the results of this test on ten different random number streams. Discussion of these results is reserved until after the presentation of the other test methodologies.

A.3.2 Empirical Test 2, Uniformity In Two and Three Dimensions

The test for uniformity in multiple dimensions, often referred to as a serial test, is essentially an extension of the first test. For the serial test the random numbers are divided into non-overlapping d-tuples (U1 = (U_1, U_2, U_3,..., U_d), U_2 = (U_{d+1}, U_{d+2}, U_{d+3},..., U_{2d}),...). If the random variates are IID U(0,1) them these d-tuples (U_1, U_2,...) should be random vectors uniformly distributed over a d-dimensional space.

The performance of the serial test is as follows. The range [0,1] is divided into k equal sized subintervals, j = 1 to k, and the random number stream is divided into a stream of d-tuples. Now, $f_{j1j2....jd}$ is taken to be the number of generated d-tuples (U_i's) with the first component falling into subinterval j_1, the second component into subinterval $j2$, and so on. Then the test statistic, shown in equation (3), has an (approximately) chi-squared distribution with k^d - 1 degrees of freedom.

$$x^2\text{-(d)}= \frac{k^d}{n}\sum_{j_1=1}^{k}\ \sum_{j_2=1}^{k}\ .....\sum_{j_d=1}^{k}\left(f_{j1j2...jd}-\frac{n}{k^d}\right) \quad (3)$$

The ten random number streams being tested were subjected to both 2 and 3 dimensional tests, the results of which are given in Table A.1, as Test 2a and Test 2b, respectively. For the two-dimensional tests, k was taken as 64, resulting in 64^2 -1 = 4095 degrees of freedom, and n = 32,768 vectors were generated, requiring 2n = 65536 random variates from each stream. For the three-dimensional test k was taken as 16, again resulting in 4095 (64^3 -1) degrees of freedom, and n = 32,768 vectors were generated, requiring 3n = 98304 random variates from each stream. For both tests the critical chi-squared value at an a = 0.10 is 4211.4, as in the first empirical test.

A.3.3 Empirical Test 3, Independence Of Consecutive Stream Numbers

The third test is not a test of uniformity per se, but instead a test of independence. This test involves counting the runs-up (or runs-down) in a random number stream of length n. A run-up is a random number stream sequence where the numbers increase monotonically, such that r; is number of runs of length i, where i = 1,2,3,4,5, and r_6 is the number of runs of length 6 or more. For example, the sequence (0.86, 0.11, 0.23, 0.03, 0.13, 0.06, 0.55, 0.64, 0.87, 0.10) contains the following runs, (0.86) with a length of 1, (.11, .23) with a length of two, (0.03, 0.13) with a length of 2, (0.06, 0.55, 0.64, 0.87) with a length of 4 and (0.10) with a length of 1, thus r_l = 2, r_2 = 2, r_3 = 0, r_4 = 1, r_5 = 0, and r_6 = 0.

The test statistic for the runs-up (or runs-down) test is given in equation (4). In this equation a_{ij}

$$R = \frac{1}{n}\sum_{i=1}^{6}\ \sum_{j=1}^{6}\ a_{ij}\left(r_i - nb_i\right)\left(r_j - nb_j\right) \quad (4)$$

and b_j are constants
[See Law and Kelton for a listing of these constants].

The null hypothesis for this test statistic is that the number streams are IID random variables. Where n is large (4000+) this test statistic has been shown to approximately follow a chi-square distribution with 6 degrees of freedom. For the ten number streams under study both a runs-up (Test 3a) and a runs-down (Test 3b) test was performed. Each test used the first 5000 numbers (n = 5000) of the random numbers stream. The critical chi-square value for 6 degrees of freedom at an a=0.l0 is 10.6.

A.3.4 Empirical Test 4, Correlation in a Number Stream

The final empirical test applied to the ten random number streams assessed whether there was a discernible correlation between the random numbers at different lag (number of observations between the two observations under consideration) lengths. This test assumes a covariance stationary process generated the number stream. In a covariance stationary process the mean and variance are unchanging for each observation and the covariance between two observations is dependent on the lag between the observations regardless of the location in the stream of the observations. In a covariance stationary process the correlation p between two observations with a lag j is defined as $p_j = C_j / C_o$ where $C_o = Var(X_i)$ and C_i is defined as in equation (5).

$$Cj = Cov(X_i, X_{i+j}) = E(X_i,X_{i+j}) - E(X_i)E(X_{i+j})$$

It may be shown (225) that for random numbers generated from a U(0,1) that p_j may be directly estimated from equation (6) and have an estimated variance as shown in equation (7), where: h = [(n-1)/j]-1.

$$\hat{p} = \frac{12}{h+1}\sum_{k=0}^{h} U_{1+kj}U_{1+(k+1)_j} - 3 \qquad (6)$$

$$\text{Var}(\hat{p}) = \frac{13h+7}{(h+1)^2} \qquad (7)$$

Now, to test if the generated random numbers have a no discernable correlation, a characteristic of a IID U(0,1), the null hypothesis that $p_j = 0$ is tested using the test statistic Aj, equation (8). For sufficiently large n, this test statistic will have an approximately standard normal distribution. Thus, a rejection of the null hypothesis, at some level a, leads to an inability to accepts there being no correlation and therefore an inability to accept the numbers as IID U(0,1).

$$\text{Aj} = \frac{\hat{p}}{\sqrt{Var(\hat{p})}} \qquad (8)$$

For the ten different generated streams correlations at lags of 1,2,3,4,5 and 6 (Tests 6a, 6b, 6c, 6d, 6e, and 6f, respectively) were tested at level a = 0.10, thus H_o is not accepted if ~A;J > z,-a,z = 1.645. The results of these tests may be seen in Table A. 1.

A.4 DISCUSSION OF TEST RESULTS

Table A.1Empirical Test Results (highlighted values exceed critical statistic point value)

Test No.	Random Number Stream										Test Stat. (α =0.10)
	1	2	3	4	5	6	7	8	9	10	
1	4252.00	4095.50	3967.25	4053.75	4015.75	4165.25	4118.50	4063.00	3932.00	4085.00	4211.4
2a	4086.75	4001.00	4070.25	4154.25	4278.50	4031.50	4162.50	4099.25	3987.75	4097.75	4211.4
2b	4256.75	4061.50	4057.50	4116.25	4044.25	4075.25	4020.50	259.50	4206.75	4173.75	4211.4
3a	9.93	3.67	27.20	8.01	4.39	5.35	4.45	6.13	4.17	3.06	10.6
3b	7.11	4.64	5.25	5.18	8.30	7.22	7.57	9.80	3.93	7.22	10.6
4a	0.82	-1.37	0.83	1.42	-0.58	0.49	l0.13	0.26	-0.18	-0.99	1.65
4b	1.29	-0.19	-0.06	0.04	-0.38	0.11	-0.98	-0.73	-0.28	-0.39	1.65
4c	-0.32	-2.12	0.64	2.06	-0.10	0.74	2.10	-0.03	-0.14	0.56	1.65
4d	0.48	0.22	0.97	-0.64	0.38	-0.96	-1.17	-2.02	-0.76	-1.34	1.65
4e	1.68	-0.24	-0.79	1.93	0.92	1.73	0.31	-0.75	0.49	0.88	1.65
4f	-0.36	-0.04	0.48	0.78	-0.66	0.53	0.20	0.41	-0.43	0.48	1.65

As stated, Table A.1 gives the results from the empirical tests. The first test null hypothesis was that the generated random number streams are uniformly distributed. Nine of the number streams test statistic do not allow for a rejection of the null hypothesis, i.e. not showing a statistical indication of not being uniformly distributed. Only stream one has a test statistic value which exceeds the critical value for a = 0.10. When considering the first stream rejection of H_o one should recall that a is the probability of rejecting the null hypothesis when it is true. Therefore, considering that ten streams were tested it would be expected that, if the random number generator did create IID U(0,1) streams, that 10% of the time the null hypothesis would be rejected. Thus, having one or two of the ten streams reject the null hypothesis is not reason to doubt the random number generator, but an expected outcome of a series of tests. To cast doubt on the generator at least several random number streams would need to lead to a rejection of H_o. Also, it is noted that the stream one test statistic barely rejects H_o, leading to further confidence in the generator.

Tests 2a and 2b test for uniformity in two and three dimensions, respectively. As with the first empirical test, Test 2a shows one failure to accept the null hypothesis, while Test 2b show two failures. Following the reasoning of the Test 1 discussion, these failures do not lead to a doubt in the generator, as a limited number of rejections are expected, even for a random number generator that generates IID U(0,1).

Tests 3a and 3b utilize the runs-up and runs-down empirical tests, respectively, to test the null hypothesis that the generated numbers in a stream are not sequence dependent (a = 0.10). Test 3a fails to accept the null hypothesis in only one case (stream 3) and test 3b accepts the null hypothesis for all the streams. Again, following the previous reasoning, one may be conclude that the generated numbers are most likely not sequence dependent. Finally, tests 4a through 4f test for a null hypothesis of zero lag correlation for lags of length 1 through 6, respectively. Again, these test are performed for a = 0.10. Of the 60 individual tests, only once is the null hypothesis not accepted, (stream 4 with a lag of 3). Thus, as with the previous tests, it is seen that the generated number streams likely meet the tested null hypothesis.

In conclusion, based on the performed empirical tests and reviewed literature on the ARENA LCG, it may be stated that the ARENA random number generator may be used with confidence. Again, while not every possible random number stream has been tested it is still reasonable to conclude, based on the number of tests completed for this appendix and the literature on this generator, that the ARENA random number generator provides a reasonable approximation of a IID U(0,1) distribution.

Appendix B

Adaptive Control Test Networks, 1, 2, and 3 Performance Measures

Network 1 - Pre-Timed Control - Replication 1

Movement	EB T 1	EB T 2	WB T 5	WB T 6	EB L 9	WB L 10	NB T 11	NB R 12	SB T 13	SB R 14
Int 1										
Volume (vph)	170	164	227	240	51	71	70	30	119	73
Delay Avg (spv)	17.2	19.4	23.0	23.3	28.4	39.3	27.1	24.2	31.4	28.4
Delay Max (spv)	50.0	49.1	50.9	51.2	68.0	89.4	66.7	63.5	67.5	66.4
Q Avg (v)	1.9	2.1	3.4	3.6	0.9	1.8	1.3	0.5	2.6	1.4
Q Max (v)	8	8	11	10	3	9	4	2	7	4
Int 2										
Volume (vph)	165	160	235	245	29	95	174	67	191	49
Delay Avg (spv)	5.7	6.3	7.6	7.4	58.2	56.6	28.4	25.2	30.0	24.8
Delay Max (spv)	53.6	54.2	54.2	54.9	74.1	113.4	61.2	61.9	70.6	62.2
Q Avg (v)	0.6	0.6	1.3	1.3	1.1	3.6	3.2	1.1	3.7	0.8
Q Max (v)	4	5	5	6	5	9	10	4	10	3
Int 3										
Volume (vph)	176	184	270	270	43	57	245	34	247	49
Delay Avg (spv)	10.7	9.2	12.6	12.2	48.1	73.7	25.9	17.7	25.7	17.8
Delay Max (spv)	54.9	50.8	54.7	54.8	84.6	144.4	57.2	54.0	56.0	56.4
Q Avg (v)	1.2	1.1	2.2	2.2	1.3	2.7	4.3	0.4	4.2	0.6
Q Max (v)	7	7	7	8	4	7	12	2	11	2
Int 4										
Volume (vph)	160	183	250	241	46	74	200	43	165	103
Delay Avg (spv)	4.9	3.7	24.8	23.5	56.1	36.4	27.9	19.2	24.7	24.9
Delay Max (spv)	53.9	53.3	54.1	53.6	72.4	75.9	58.7	58.0	59.2	58.8
Q Avg (v)	0.5	0.4	4.1	3.8	1.8	1.9	3.6	0.5	2.6	1.7
Q Max (v)	3	3	12	11	5	5	11	2	8	6

Network 1 - Constrained COP - Replication 1

Movement	EB T 1	EB T 2	WB T 5	WB T 6	EB L 9	WB L 10	NB T 11	NB R 12	SB T 13	SB R 14
Int Volume	167	160	232	243	48	69	73	32	125	76
Delay Avg	6.6	6.9	4.9	5.7	27.1	49.0	15.9	6.1	18.4	13.9
Delay Max	35.2	29.7	32.9	36.4	74.8	131.1	56.4	28.0	53.5	47.0
Q Avg	0.7	0.7	0.7	0.9	0.9	2.7	0.7	0.1	1.5	0.7
Q Max	5	5	5	6	3	9	4	1	6	3
Int Volume	159	178	251	248	23	52	158	66	187	48
Delay Avg	8.5	6.6	5.9	6.0	84.0	522.5	15.2	9.2	17.9	12.3
Delay Max	39.2	36.8	43.3	36.4	327.5	805.3	53.0	48.8	64.4	43.2
Q Avg	0.9	0.7	0.9	0.9	1.3	31.3	1.5	0.4	2.2	0.4
Q Max	5	5	7	7	7	47	6	2	12	2
Int Volume	178	189	273	268	35	43	245	34	246	50
Delay Avg	7.5	7.0	12.6	13.6	165.9	166.9	24.1	11.2	21.0	10.3
Delay Max	33.3	47.0	42.3	52.4	346.3	310.6	69.0	56.5	76.7	62.2
Q Avg	0.8	0.8	2.2	2.4	4.3	6.0	3.9	0.2	3.4	0.3
Q Max	6	6	8	8	10	14	12	2	16	3
Int Volume	166	176	248	237	51	74	198	42	160	100
Delay Avg	7.3	7.3	11.3	10.7	62.6	147.5	19.5	9.4	13.6	10.2
Delay Max	34.6	32.7	36.6	35.4	148.3	235.2	54.7	47.2	48.9	42.0
Q Avg	0.8	0.8	1.8	1.7	2.1	7.5	2.5	0.2	1.4	0.6
Q Max	5	7	7	7	7	13	8	2	6	4

Network 1 - COP - Replication 1

Movement	EB T 1	EB T 2	WB T 5	WB T 6	EB L 9	WB L 10	NB T 11	NB R 12	SB T 13	SB R 14
Int Volume	168	161	234	238	50	72	73	32	124	76
Delay Avg	7.3	6.4	4.7	5.0	27.6	51.8	13.7	10.9	16.9	11.2
Delay Max	31.5	26.4	24.4	31.9	82.2	120.5	43.7	44.9	44.0	45.0
Q Avg	0.8	0.6	0.7	0.8	0.9	2.5	0.6	0.2	1.4	0.5
Q Max	5	4	7	6	3	9	2	1	5	3
Int Volume	166	165	235	260	26	68	165	65	186	48
Delay Avg	6.3	6.8	6.0	5.7	54.6	337.2	16.2	10.5	19.4	10.1
Delay Max	46.0	33.0	40.8	36.5	119.1	610.0	49.5	46.8	54.5	38.1
Q Avg	0.7	0.7	0.9	1.0	0.9	20.4	1.7	0.4	2.4	0.3
Q Max	5	5	7	6	4	34	8	2	8	2
Int Volume	177	178	255	277	47	46	245	35	249	52
Delay Avg	7.3	6.9	12.3	10.9	107.0	130.6	19.0	8.5	20.8	9.6
Delay Max	42.7	38.8	42.3	35.3	248.0	250.0	53.8	45.9	54.9	42.0
Q Avg	0.8	0.8	2.0	1.9	3.4	5.7	3.1	0.2	3.4	0.3
Q Max	5	4	8	8	11	15	11	2	10	2
Int Volume	173	178	251	242	42	60	196	42	162	101
Delay Avg	4.1	4.9	10.5	10.2	63.8	209.4	17.5	7.7	12.5	10.5
Delay Max	33.3	35.2	29.5	37.7	242.7	431.0	43.2	31.4	35.9	38.0
Q Avg	0.4	0.5	1.7	1.5	1.8	10.9	2.2	0.2	1.3	0.7
Q Max	5	6	7	7	8	23	7	1	5	4

Network 1 - Pre-Timed Control - Replication 2

Movement	EB T 1	EB T 2	WB T 5	WB T 6	EB L 9	WB L 10	NB T 11	NB R 12	SB T 13	SB R 14
Int 1										
Volume (vph)	172	167	233	245	50	78	73	32	123	69
Delay Avg (spv)	18.6	17.4	25.6	23.4	30.6	38.1	28.3	26.7	30.1	26.4
Delay Max (spv)	48.8	49.4	51.2	50.3	71.2	59.9	67.5	65.0	66.2	66.4
Q Avg (v)	2.1	1.9	3.8	3.7	1.0	2.1	1.4	0.6	2.6	1.2
Q Max (v)	8	7	11	10	3	11	5	2	8	4
Int 2										
Volume (vph)	169	154	239	256	32	87	170	71	183	51
Delay Avg (spv)	6.4	6.7	8.0	8.8	50.9	64.5	27.8	25.3	28.7	21.7
Delay Max (spv)	54.3	53.5	54.2	55.5	73.5	129.5	62.5	62.4	62.4	60.9
Q Avg (v)	0.7	0.7	1.3	1.5	1.1	3.8	3.1	1.1	3.4	0.7
Q Max (v)	4	4	5	6	5	9	9	4	10	3
Int 3										
Volume (vph)	176	181	262	272	49	64	242	32	242	48
Delay Avg (spv)	10.3	9.6	12.4	11.9	60.9	88.1	26.9	19.2	28.4	18.8
Delay Max (spv)	54.4	52.3	58.1	56.9	170.0	241.7	62.1	54.5	58.9	53.5
Q Avg (v)	1.2	1.1	2.2	2.1	1.9	4.2	4.3	0.4	4.6	0.6
Q Max (v)	8	7	7	6	7	13	13	2	12	2
Int 4										
Volume (vph)	160	171	245	254	48	74	211	42	171	105
Delay Avg (spv)	5.5	5.9	23.4	25.1	58.8	39.7	27.2	19.3	24.7	23.3
Delay Max (spv)	54.0	53.8	53.5	61.9	73.4	86.0	60.2	54.4	58.1	59.0
Q Avg (v)	0.6	0.7	3.8	4.3	2.0	2.0	3.7	0.5	2.8	1.6
Q Max (v)	4	4	11	12	6	5	11	2	9	6

Network 1 - Constrained COP - Replication 2

Movement	EB T 1	EB T 2	WB T 5	WB T 6	EB L 9	WB L 10	NB T 11	NB R 12	SB T 13	SB R 14
Int Volume	170	166	223	233	49	83	76	33	128	71
Delay Avg	7.0	6.3	5.9	6.8	26.5	54.5	12.9	9.3	16.0	12.9
Delay Max	31.5	30.0	31.6	30.0	90.2	180.0	44.0	33.1	49.4	39.2
Q Avg	0.7	0.6	0.8	1.0	0.8	3.0	0.6	0.2	1.3	0.6
Q Max	5	5	6	6	3	9	3	1	6	3
Int Volume	171	164	235	252	27	71	173	70	181	51
Delay Avg	9.3	8.9	4.7	5.0	49.9	285.3	19.8	13.5	17.9	12.7
Delay Max	37.2	38.6	42.1	39.5	133.6	599.2	65.1	53.1	66.5	40.5
Q Avg	1.0	0.9	0.7	0.8	0.9	21.5	2.2	0.6	2.1	0.4
Q Max	6	6	6	6	6	48	10	3	8	2
Int Volume	171	177	270	276	53	54	244	33	243	48
Delay Avg	6.2	7.1	11.0	11.1	94.5	122.6	23.5	15.1	32.5	11.2
Delay Max	36.0	41.6	52.0	47.5	229.3	236.2	77.0	61.0	86.2	56.5
Q Avg	0.7	0.8	1.9	2.0	3.4	4.4	3.8	0.3	5.2	0.4
Q Max	6	6	10	9	10	12	10	2	15	2
Int Volume	165	184	248	258	33	65	209	41	169	104
Delay Avg	5.7	5.5	11.1	11.9	53.4	114.9	17.9	8.8	13.7	10.1
Delay Max	43.8	35.9	39.9	37.3	132.1	210.0	59.5	36.5	45.2	34.4
Q Avg	0.6	0.6	1.8	2.0	1.4	6.5	2.4	0.2	1.5	0.7
Q Max	6	5	8	7	5	16	10	2	6	4

Network 1 - COP - Replication 2

Movement	EB T	EB T	WB T	WB T	EB L	WB L	NB T	NB R	SB T	SB R
	1	2	5	6	9	10	11	12	13	14
Int Volume	170	165	244	242	50	81	75	32	128	72
Delay Avg	7.5	6.5	5.1	5.8	26.6	50.9	16.0	14.1	17.7	12.7
Delay Max	31.4	35.9	33.0	38.1	74.0	142.8	43.1	34.0	51.0	40.8
Q Avg	0.8	0.7	0.8	0.9	0.8	3.0	0.8	0.3	1.5	0.6
Q Max	5	4	6	6	3	9	3	1	5	3
Int Volume	169	156	263	265	37	73	177	70	181	51
Delay Avg	7.5	6.7	7.7	7.9	46.7	121.6	21.7	11.3	17.5	9.4
Delay Max	34.9	31.5	36.7	37.8	189.9	215.8	61.5	37.9	46.8	39.2
Q Avg	0.8	0.7	1.3	1.3	1.1	5.9	2.5	0.5	2.0	0.3
Q Max	5	5	8	8	6	13	7	2	6	2
Int Volume	174	172	273	284	47	49	243	34	243	50
Delay Avg	8.3	8.6	12.5	11.4	131.2	113.6	21.8	10.1	22.1	11.7
Delay Max	35.4	37.8	60.7	43.7	352.6	277.7	65.0	45.0	73.8	47.8
Q Avg	0.9	0.9	2.2	2.1	4.2	4.1	3.5	0.2	3.5	0.4
Q Max	6	7	9	9	12	13	11	2	15	2
Int Volume	159	176	242	255	39	66	208	42	169	104
Delay Avg	6.8	5.4	10.4	13.4	51.8	147.3	19.0	7.7	14.5	11.2
Delay Max	42.4	36.7	38.5	40.9	165.8	365.0	54.2	31.1	44.0	42.8
Q Avg	0.7	0.6	1.6	2.2	1.3	8.2	2.6	0.2	1.6	0.7
Q Max	6	6	8	9	5	19	8	2	6	4

Network 1 - Pre-Timed Control - Replication 3

Movement	EB T 1	EB T 2	WB T 5	WB T 6	EB L 9	WB L 10	NB T 11	NB R 12	SB T 13	SB R 14
Int 1										
Volume (vph)	174	169	246	249	54	74	74	33	123	74
Delay Avg (spv)	16.5	16.6	23.9	24.7	30.9	37.2	28.5	26.8	30.0	27.4
Delay Max (spv)	49.4	48.8	51.1	51.0	71.5	64.8	67.0	64.2	67.0	66.4
Q Avg (v)	1.9	1.8	3.8	4.0	1.0	1.7	1.4	0.6	2.5	1.4
Q Max (v)	7	7	10	10	4	7	5	2	7	5
Int 2										
Volume (vph)	162	173	231	272	29	86	165	69	187	53
Delay Avg (spv)	6.0	6.0	9.9	8.9	48.8	79.2	30.1	23.0	28.7	23.4
Delay Max (spv)	53.2	53.2	54.1	54.8	64.8	159.5	62.0	60.1	62.5	60.9
Q Avg (v)	0.7	0.7	1.5	1.6	0.9	4.7	3.2	1.0	3.5	0.8
Q Max (v)	4	4	5	5	3	14	8	4	10	3
Int 3										
Volume (vph)	176	181	264	279	59	57	250	32	246	48
Delay Avg (spv)	10.4	9.5	10.7	10.1	42.7	64.3	29.2	15.7	27.7	18.2
Delay Max (spv)	52.4	53.1	54.7	53.6	139.5	150.0	69.5	55.5	64.3	56.0
Q Avg (v)	1.2	1.1	1.8	1.8	1.6	2.3	4.9	0.3	4.6	0.6
Q Max (v)	8	7	7	8	5	6	14	2	12	3
Int 4										
Volume (vph)	158	181	252	242	42	71	198	42	166	101
Delay Avg (spv)	3.2	4.8	22.6	23.0	61.3	40.0	26.5	21.1	25.8	21.9
Delay Max (spv)	52.8	53.8	54.8	53.0	74.5	87.0	59.7	56.0	59.7	58.1
Q Avg (v)	0.4	0.6	3.8	3.7	1.8	2.0	3.4	0.5	2.8	1.4
Q Max (v)	4	3	11	11	4	5	11	2	8	5

Network 1 - Constrained COP - Replication 3

Movement	EB T 1	EB T 2	WB T 5	WB T 6	EB L 9	WB L 10	NB T 11	NB R 12	SB T 13	SB R 14
Int Volume	174	167	228	235	51	83	75	34	126	75
Delay Avg	5.9	5.6	5.1	5.8	29.8	62.8	10.1	9.9	16.7	13.1
Delay Max	32.0	31.9	26.3	35.9	85.0	160.9	41.1	32.5	47.9	38.7
Q Avg	0.6	0.6	0.7	0.9	1.0	3.5	0.5	0.2	1.4	0.7
Q Max	4	4	5	5	4	9	3	1	6	3
Int Volume	169	169	240	247	32	60	160	66	183	52
Delay Avg	6.5	6.2	6.8	7.3	44.4	369.4	14.8	11.0	17.9	10.8
Delay Max	36.3	33.9	53.9	45.4	135.3	681.5	47.9	41.7	49.1	42.7
Q Avg	0.7	0.7	1.0	1.1	0.9	19.7	1.5	0.5	2.1	0.3
Q Max	5	5	7	8	6	35	7	3	8	2
Int Volume	175	184	269	261	56	60	243	32	238	48
Delay Avg	8.7	9.3	11.4	11.8	110.7	236.1	23.7	8.6	25.2	8.7
Delay Max	36.7	42.5	52.9	55.0	243.5	395.4	69.3	30.1	64.2	38.9
Q Avg	1.0	1.1	2.0	2.0	4.0	11.0	4.0	0.2	4.3	0.3
Q Max	6	6	9	10	11	24	14	1	13	2
Int Volume	161	169	249	239	50	67	197	42	163	100
Delay Avg	4.5	4.3	11.6	11.5	45.8	183.4	16.8	11.1	11.0	9.9
Delay Max	40.7	30.2	40.5	37.9	122.0	393.1	62.2	34.4	36.7	39.1
Q Avg	0.4	0.5	1.9	1.8	1.5	9.3	2.1	0.3	1.1	0.6
Q Max	5	5	7	7	5	19	9	1	5	4

Network 1 - COP - Replication 3

Movement	EB T 1	EB T 2	WB T 5	WB T 6	EB L 9	WB L 10	NB T 11	NB R 12	SB T 13	SB R 14
Int Volume	172	166	248	229	52	84	75	34	124	74
Delay Avg	8.2	8.6	7.7	8.9	21.5	51.9	14.6	10.0	19.2	16.1
Delay Max	37.2	39.5	59.4	61.0	75.5	149.4	55.1	44.9	59.5	52.7
Q Avg	0.9	0.9	1.2	1.3	0.7	2.9	0.7	0.2	1.6	0.8
Q Max	6	5	9	9	3	8	3	1	6	4
Int Volume	166	166	248	248	34	73	165	66	182	51
Delay Avg	6.8	9.0	6.7	8.6	57.1	93.7	19.0	11.1	19.1	10.8
Delay Max	35.6	37.2	38.2	53.3	132.7	193.3	56.7	41.0	54.2	30.0
Q Avg	0.7	1.0	1.1	1.4	1.1	4.7	2.1	0.5	2.3	0.4
Q Max	7	7	7	8	5	12	7	2	8	2
Int Volume	165	179	265	266	49	54	253	33	249	51
Delay Avg	10.1	9.8	12.1	11.5	104.5	118.5	21.0	7.1	20.3	10.3
Delay Max	38.8	41.7	43.3	44.5	241.9	224.5	83.8	42.0	67.4	54.7
Q Avg	1.1	1.1	2.1	2.0	3.8	5.1	3.5	0.1	3.3	0.3
Q Max	7	8	8	8	10	14	14	2	11	2
Int Volume	153	177	249	239	42	72	196	42	163	98
Delay Avg	5.7	5.4	11.6	9.9	51.7	126.7	16.1	8.4	12.6	10.9
Delay Max	31.7	37.2	42.9	33.7	115.6	279.2	53.0	40.7	47.1	53.0
Q Avg	0.5	0.6	1.9	1.5	1.4	6.3	2.0	0.2	1.3	0.7
Q Max	6	5	9	7	7	13	8	2	6	3

Network 1 - Pre-Timed Control - Replication 4

Movement	EB T 1	EB T 2	WB T 5	WB T 6	EB L 9	WB L 10	NB T 11	NB R 12	SB T 13	SB R 14
Int 1										
Volume (vph)	173	168	236	249	51	75	73	31	118	72
Delay Avg (spv)	18.4	17.8	24.9	24.4	31.3	39.9	28.3	26.2	30.8	26.8
Delay Max (spv)	49.5	49.4	51.4	50.5	70.8	71.6	67.0	62.0	67.5	67.1
Q Avg (v)	2.1	1.9	3.8	3.9	1.0	1.9	1.4	0.6	2.5	1.3
Q Max (v)	7	8	10	10	3	7	5	2	7	4
Int 2										
Volume (vph)	174	161	237	264	28	87	163	70	200	50
Delay Avg (spv)	5.7	6.3	8.3	7.8	61.1	63.2	26.8	23.5	31.1	20.6
Delay Max (spv)	53.7	54.2	53.4	54.7	74.3	135.8	61.0	60.4	64.5	61.2
Q Avg (v)	0.7	0.7	1.3	1.4	1.1	3.8	2.8	1.0	4.0	0.6
Q Max (v)	4	4	6	7	4	10	9	4	10	3
Int 3										
Volume (vph)	175	187	265	278	49	67	252	31	249	48
Delay Avg (spv)	10.5	10.0	12.5	11.9	50.2	101.9	30.4	19.4	29.4	18.6
Delay Max (spv)	53.0	53.6	55.4	54.6	83.6	229.3	71.2	56.0	66.2	53.0
Q Avg (v)	1.2	1.2	2.1	2.2	1.5	4.3	5.1	0.4	4.8	0.6
Q Max (v)	8	8	7	7	4	9	15	2	13	3
Int 4										
Volume (vph)	159	178	248	255	45	74	199	42	172	104
Delay Avg (spv)	4.8	5.6	23.8	24.5	56.3	38.6	28.0	18.2	25.8	22.9
Delay Max (spv)	53.1	54.0	53.5	53.4	74.8	83.5	59.0	52.5	59.4	59.0
Q Avg (v)	0.5	0.7	3.9	4.1	1.8	2.0	3.6	0.5	2.9	1.5
Q Max (v)	4	4	11	12	6	5	10	2	8	6

Network 1 - Constrained COP - Replication 4

Movement	EB T 1	EB T 2	WB T 5	WB T 6	EB L 9	WB L 10	NB T 11	NB R 12	SB T 13	SB R 14
Int Volume	169	165	235	237	48	81	76	33	123	74
Delay Avg	6.9	6.6	7.1	7.7	26.4	88.7	14.7	11.8	21.5	15.5
Delay Max	29.0	32.2	44.1	43.8	81.5	178.0	50.7	41.2	60.2	55.1
Q Avg	0.7	0.7	1.1	1.2	0.9	5.4	0.7	0.2	1.7	0.7
Q Max	5	5	7	7	3	14	4	1	6	4
Int Volume	163	173	254	257	25	68	163	68	193	49
Delay Avg	8.4	7.3	7.9	7.4	61.2	270.4	20.2	13.1	20.6	11.4
Delay Max	44.6	40.9	41.2	38.2	227.4	421.2	63.5	48.7	66.7	50.9
Q Avg	0.9	0.8	1.3	1.2	1.0	16.1	2.2	0.6	2.6	0.3
Q Max	6	5	8	8	4	29	8	3	10	2
Int Volume	185	186	270	272	37	53	255	32	252	49
Delay Avg	8.3	8.8	10.9	11.1	130.8	324.4	33.8	11.5	23.6	14.0
Delay Max	34.5	56.5	52.4	50.4	280.0	485.4	85.4	50.0	69.5	70.5
Q Avg	1.0	1.1	1.9	2.0	3.1	12.8	5.7	0.2	3.8	0.4
Q Max	6	6	10	11	7	24	16	2	12	3
Int Volume	185	175	247	254	47	63	193	42	168	101
Delay Avg	7.1	6.0	9.2	10.2	83.9	304.8	20.4	9.0	14.4	11.7
Delay Max	36.8	34.7	35.4	36.5	164.9	466.2	60.7	34.4	50.7	56.0
Q Avg	0.8	0.6	1.5	1.7	2.7	15.5	2.6	0.2	1.6	0.7
Q Max	5	5	8	7	9	24	8	2	6	4

Network 1 - COP - Replication 4

Movement	EB T 1	EB T 2	WB T 5	WB T 6	EB L 9	WB L 10	NB T 11	NB R 12	SB T 13	SB R 14
Int Volume	170	164	250	255	49	71	77	33	123	74
Delay Avg	7.2	6.8	5.9	5.9	20.5	54.2	12.6	9.5	17.1	12.4
Delay Max	32.8	33.1	33.4	31.4	80.2	157.9	49.9	40.9	45.4	42.5
Q Avg	0.8	0.7	0.9	0.9	0.7	2.6	0.6	0.2	1.4	0.6
Q Max	5	4	8	7	3	9	4	2	5	3
Int Volume	166	170	254	271	27	67	169	69	196	49
Delay Avg	7.7	7.5	5.8	6.5	51.4	321.9	21.3	12.1	21.8	12.8
Delay Max	40.2	41.6	37.5	37.4	105.7	469.5	61.7	50.7	62.7	61.5
Q Avg	0.8	0.8	1.0	1.1	0.9	17.2	2.3	0.5	2.8	0.4
Q Max	6	6	8	7	5	25	8	3	9	3
Int Volume	178	177	275	282	44	51	259	33	252	51
Delay Avg	10.1	9.4	11.7	11.6	98.2	161.6	28.7	10.3	20.3	12.1
Delay Max	39.9	50.1	46.1	51.4	276.8	365.6	87.4	35.9	60.1	45.5
Q Avg	1.2	1.1	2.1	2.1	3.2	5.2	4.8	0.2	3.3	0.4
Q Max	7	9	10	10	9	12	17	1	10	2
Int Volume	170	169	247	253	44	71	194	41	169	101
Delay Avg	4.6	4.0	12.3	13.1	64.0	193.8	15.6	9.4	15.0	13.6
Delay Max	42.7	34.2	40.4	41.0	146.2	370.7	52.1	31.5	56.2	49.2
Q Avg	0.5	0.5	2.0	2.2	1.9	9.8	2.0	0.2	1.6	0.9
Q Max	6	7	9	9	6	19	7	2	7	4

Network 1 - Pre-Timed Control - Replication 5

Movement	EB T 1	EB T 2	WB T 5	WB T 6	EB L 9	WB L 10	NB T 11	NB R 12	SB T 13	SB R 14
Int 1										
Volume (vph)	170	170	237	237	49	91	69	31	123	73
Delay Avg (spv)	17.3	17.5	23.3	23.3	35.7	45.4	29.3	27.7	31.1	27.7
Delay Max (spv)	49.4	49.2	50.0	50.4	73.2	112.2	67.0	67.1	67.2	64.8
Q Avg (v)	1.9	1.9	3.6	3.6	1.1	3.0	1.4	0.6	2.6	1.4
Q Max (v)	7	8	10	10	3	13	4	2	7	4
Int 2										
Volume (vph)	166	164	249	262	33	79	171	66	186	50
Delay Avg (spv)	5.6	4.0	9.2	8.5	52.2	58.3	27.3	23.3	29.0	22.0
Delay Max (spv)	52.3	52.7	58.3	54.3	68.7	130.5	62.2	61.0	61.1	60.0
Q Avg (v)	0.6	0.4	1.5	1.5	1.2	3.0	3.0	1.0	3.5	0.7
Q Max (v)	4	4	7	7	5	10	8	4	10	3
Int 3										
Volume (vph)	174	187	267	278	43	56	244	32	254	50
Delay Avg (spv)	9.8	8.5	10.5	11.4	51.3	65.4	25.5	19.4	29.6	19.7
Delay Max (spv)	51.4	53.6	55.6	55.3	91.7	150.1	59.5	54.0	70.2	54.8
Q Avg (v)	1.1	1.0	1.9	2.1	1.4	2.2	4.2	0.4	5.0	0.7
Q Max (v)	7	7	6	6	6	6	12	2	14	3
Int 4										
Volume (vph)	165	185	249	250	43	74	208	43	165	100
Delay Avg (spv)	4.4	4.6	24.3	22.6	61.7	38.0	27.8	21.0	24.8	21.4
Delay Max (spv)	52.7	53.9	54.3	55.5	76.8	77.4	60.5	56.0	59.4	57.0
Q Avg (v)	0.5	0.6	4.0	3.7	1.7	1.9	3.7	0.6	2.7	1.4
Q Max (v)	3	3	11	11	6	5	11	2	8	5

Network 1 - Constrained COP - Replication 5

Movement	EB T 1	EB T 2	WB T 5	WB T 6	EB L 9	WB L 10	NB T 11	NB R 12	SB T 13	SB R 14
Int Volume	170	169	236	234	48	79	69	33	126	74
Delay Avg	7.6	6.6	5.8	6.3	23.1	57.0	13.8	10.9	19.1	16.3
Delay Max	33.9	39.1	40.1	36.6	80.5	132.4	44.5	33.8	51.9	44.4
Q Avg	0.8	0.7	0.9	0.9	0.7	2.9	0.6	0.2	1.6	0.8
Q Max	4	5	7	9	3	11	3	1	5	3
Int Volume	159	171	252	245	39	75	160	65	181	49
Delay Avg	8.1	6.9	5.5	5.5	46.5	263.7	18.8	13.3	22.9	12.7
Delay Max	38.7	33.1	43.5	41.7	133.4	552.4	59.5	60.4	58.4	58.0
Q Avg	0.8	0.7	0.9	0.9	1.2	19.4	2.0	0.5	2.7	0.4
Q Max	6	6	6	6	5	42	8	3	8	2
Int Volume	167	188	282	276	39	44	243	33	254	52
Delay Avg	8.0	7.6	13.1	13.9	130.5	118.0	21.6	10.2	29.1	9.9
Delay Max	50.7	52.6	56.4	54.1	336.3	269.3	64.5	46.5	77.2	40.0
Q Avg	0.9	0.9	2.4	2.5	3.7	3.6	3.5	0.2	4.9	0.3
Q Max	7	7	12	12	9	10	13	2	12	2
Int Volume	165	179	245	246	40	73	206	42	162	98
Delay Avg	4.5	5.1	10.9	14.0	56.9	249.3	26.6	11.8	13.1	10.7
Delay Max	30.0	33.1	43.0	48.8	222.2	354.1	62.5	37.0	39.7	43.0
Q Avg	0.5	0.6	1.8	2.3	1.5	12.8	3.5	0.3	1.3	0.6
Q Max	6	6	8	10	8	18	10	2	6	3

Network 1 - COP - Replication 5

Movement	EB T 1	EB T 2	WB T 5	WB T 6	EB L 9	WB L 10	NB T 11	NB R 12	SB T 13	SB R 14
Int Volume	170	169	225	248	48	76	70	33	128	75
Delay Avg	6.7	7.4	5.0	5.6	26.7	51.7	13.9	9.2	17.7	14.7
Delay Max	26.0	34.9	34.7	35.0	87.0	120.9	48.7	28.0	49.9	47.1
Q Avg	0.7	0.8	0.7	0.9	0.8	2.7	0.6	0.2	1.5	0.7
Q Max	5	5	9	7	3	11	3	1	5	3
Int Volume	170	180	248	263	26	57	168	65	181	50
Delay Avg	5.8	6.5	5.2	5.9	75.9	238.1	16.0	11.3	16.9	11.6
Delay Max	28.6	36.1	35.7	36.9	181.7	452.6	46.8	40.8	55.0	38.0
Q Avg	0.6	0.7	0.8	1.0	1.5	13.1	1.7	0.5	2.0	0.3
Q Max	5	5	6	6	5	28	7	2	7	2
Int Volume	183	185	269	274	43	52	243	33	250	51
Delay Avg	9.8	8.8	12.0	12.0	63.6	213.4	25.9	5.5	21.0	10.1
Delay Max	49.3	47.9	53.9	56.8	173.2	400.9	74.7	28.7	63.4	65.1
Q Avg	1.1	1.0	2.1	2.1	2.2	10.0	4.2	0.1	3.5	0.3
Q Max	6	7	12	13	6	20	14	1	13	3
Int Volume	186	173	248	251	35	83	201	41	160	97
Delay Avg	5.2	4.7	11.6	12.0	34.5	198.3	22.6	10.1	14.9	11.9
Delay Max	36.0	38.4	35.4	35.8	88.0	330.4	57.2	37.7	46.0	45.9
Q Avg	0.6	0.5	1.9	1.9	0.8	10.5	3.0	0.3	1.6	0.7
Q Max	5	6	9	8	4	17	9	2	6	4

Network 1 - Pre-Timed Control - Average Values

Movement	EB T 1	EB T 2	WB T 5	WB T 6	EB L 9	WB L 10	NB T 11	NB R 12	SB T 13	SB R 14	Overall
Int 1											
Volume (vph)	171.8	167.6	235.8	244.0	51.0	77.8	71.8	31.4	121.2	72.2	1244.6
Delay Avg (spv)	17.6	17.7	24.1	23.8	31.4	40.0	28.3	26.3	30.7	27.3	24.7
Delay Max (spv)	49.4	49.2	50.9	50.7	70.9	79.6	67.0	64.4	67.0	66.2	56.8
Q Avg (v)	2.0	1.9	3.7	3.8	1.0	2.1	1.4	0.6	2.6	1.3	2.6
Q Max (v)	7.4	7.6	10.4	10.0	3.2	9.4	4.6	2.0	7.2	4.2	
Int 2											
Volume (vph)	167.2	162.4	238.2	259.8	30.2	86.8	168.6	68.6	189.4	50.6	1421.8
Delay Avg (spv)	5.9	5.9	8.6	8.3	54.2	64.4	28.1	24.0	29.5	22.5	18.6
Delay Max (spv)	53.4	53.6	54.8	54.8	71.1	133.7	61.7	61.2	64.2	61.0	62.3
Q Avg (v)	0.7	0.6	1.4	1.4	1.1	3.8	3.1	1.0	3.6	0.7	1.8
Q Max (v)	4.0	4.2	5.6	6.2	4.4	10.4	8.8	4.0	10.0	3.0	
Int 3											
Volume (vph)	175.4	184.0	265.6	275.4	48.6	60.2	246.6	32.2	247.6	48.6	1584.2
Delay Avg (spv)	10.4	9.4	11.7	11.5	50.6	78.7	27.6	18.3	28.2	18.6	20.4
Delay Max (spv)	53.2	52.7	55.7	55.0	113.9	183.1	63.9	54.8	63.1	54.7	64.0
Q Avg (v)	1.2	1.1	2.0	2.1	1.6	3.2	4.5	0.4	4.6	0.6	2.6
Q Max (v)	7.6	7.2	6.8	7.0	5.2	8.2	13.2	2.0	12.4	2.6	
Int 4											
Volume (vph)	160.4	179.6	248.8	248.4	44.8	73.4	203.2	42.4	167.8	102.6	1471.4
Delay Avg (spv)	4.6	4.9	23.8	23.8	58.8	38.5	27.5	19.8	25.2	22.9	21.7
Delay Max (spv)	53.3	53.8	54.0	55.5	74.4	81.9	59.6	55.4	59.2	58.4	57.9
Q Avg (v)	0.5	0.6	3.9	3.9	1.8	1.9	3.6	0.5	2.7	1.5	2.5
Q Max (v)	3.6	3.4	11.2	11.4	5.4	5.0	10.8	2.0	8.2	5.6	

Network 1 - Constrained COP - Average Values

Movement	EB T 1	EB T 2	WB T 5	WB T 6	EB L 9	WB L 10	NB T 11	NB R 12	SB T 13	SB R 14	Overall
Int Volume	170.0	165.4	230.8	236.4	48.8	79.0	73.8	33.0	125.6	74.0	1236.8
Delay Avg	6.8	6.4	5.8	6.5	26.6	62.4	13.5	9.6	18.4	14.4	12.9
Delay Max	32.3	32.6	35.0	36.6	82.4	156.5	47.3	33.7	52.6	44.9	47.3
Q Avg	0.7	0.6	0.8	1.0	0.9	3.5	0.6	0.2	1.5	0.7	1.0
Q Max	4.6	4.8	6.0	6.6	3.2	10.4	3.4	1.0	5.8	3.2	
Int Volume	164.2	171.0	246.4	249.8	29.2	65.2	162.8	67.0	185.0	49.8	1390.4
Delay Avg	8.2	7.2	6.2	6.3	57.2	342.2	17.8	12.0	19.4	12.0	27.0
Delay Max	39.2	36.7	44.8	40.2	191.4	611.9	57.8	50.5	61.0	47.1	76.0
Q Avg	0.9	0.8	1.0	1.0	1.1	21.6	1.9	0.5	2.3	0.4	2.2
Q Max	5.6	5.4	6.8	7.0	5.6	40.2	7.8	2.8	9.2	2.0	
Int Volume	175.2	184.8	272.8	270.6	44.0	50.8	246.0	32.8	246.6	49.4	1573.0
Delay Avg	7.8	8.0	11.8	12.3	126.5	193.6	25.3	11.3	26.3	10.8	24.4
Delay Max	38.2	48.1	51.2	51.9	287.1	339.4	73.1	48.8	74.7	53.6	72.5
Q Avg	0.9	0.9	2.1	2.2	3.7	7.6	4.2	0.2	4.3	0.3	2.6
Q Max	6.2	6.2	9.8	10.0	9.4	16.8	13.0	1.8	13.6	2.4	
Int Volume	168.4	176.6	247.4	246.8	44.2	68.4	200.6	41.8	164.4	100.6	1459.2
Delay Avg	5.8	5.7	10.8	11.7	60.5	200.0	20.2	10.0	13.2	10.5	21.6
Delay Max	37.2	33.3	39.1	39.2	157.9	331.7	59.9	37.9	44.2	42.9	59.2
Q Avg	0.6	0.6	1.7	1.9	1.8	10.3	2.6	0.3	1.4	0.7	1.9
Q Max	5.4	5.6	7.6	7.6	6.8	18.0	9.0	1.8	5.8	3.8	

Network 1 - COP - Average Values

Movement	EB T 1	EB T 2	WB T 5	WB T 6	EB L 9	WB L 10	NB T 11	NB R 12	SB T 13	SB R 14	Overall
Int Volume	170.0	165.0	240.2	242.4	49.8	76.8	74.0	32.8	125.4	74.2	1250.6
Delay Avg	7.4	7.1	5.7	6.2	24.6	52.1	14.2	10.7	17.7	13.4	12.1
Delay Max	31.8	34.0	37.0	39.5	79.8	138.3	48.1	38.5	50.0	45.6	46.8
Q Avg	0.8	0.7	0.9	1.0	0.8	2.7	0.7	0.2	1.4	0.6	1.0
Q Max	5.2	4.4	7.8	7.0	3.0	9.2	3.0	1.2	5.2	3.2	
Int Volume	167.4	167.4	249.6	261.4	30.0	67.6	168.8	67.0	185.2	49.8	1414.2
Delay Avg	6.8	7.3	6.3	6.9	57.1	222.5	18.8	11.3	18.9	11.0	21.6
Delay Max	37.0	35.9	37.8	40.4	145.8	388.2	55.2	43.4	54.6	41.3	61.7
Q Avg	0.7	0.8	1.0	1.2	1.1	12.3	2.1	0.5	2.3	0.3	1.8
Q Max	5.6	5.6	7.2	7.0	5.0	22.4	7.4	2.2	7.6	2.2	
Int Volume	175.4	178.2	267.4	276.6	46.0	50.4	248.6	33.6	248.6	51.0	1575.8
Delay Avg	9.1	8.7	12.1	11.5	100.9	147.5	23.3	8.3	20.9	10.7	21.2
Delay Max	41.2	43.2	49.3	46.3	258.5	303.7	72.9	39.5	63.9	51.0	67.3
Q Avg	1.0	1.0	2.1	2.0	3.4	6.0	3.8	0.2	3.4	0.3	2.4
Q Max	6.2	7.0	9.4	9.6	9.6	14.8	13.4	1.6	11.8	2.2	
Int Volume	168.2	174.6	247.4	248.0	40.4	70.4	199.0	41.6	164.6	100.2	1454.4
Delay Avg	5.3	4.9	11.3	11.7	53.2	175.1	18.2	8.6	13.9	11.6	20.2
Delay Max	37.2	36.3	37.3	37.8	151.6	355.3	51.9	34.5	45.9	45.8	59.3
Q Avg	0.5	0.5	1.8	1.9	1.5	9.1	2.4	0.2	1.5	0.7	1.8
Q Max	5.6	6.0	8.4	8.0	6.0	18.2	7.8	1.8	6.0	3.8	

Network 1 - Pre-Timed Control - Average Value Standard Deviation

Movement	EB T 1	EB T 2	WB T 5	WB T 6	EB L 9	WB L 10	NB T 11	NB R 12	SB T 13	SB R 14
Int 1										
Volume (vph)	1.8	2.3	6.9	5.4	1.9	7.8	2.2	1.1	2.5	1.9
Delay Avg (spv)	0.9	1.0	1.1	0.7	2.7	3.2	0.8	1.3	0.6	0.8
Delay Max (spv)	0.4	0.2	0.5	0.4	1.9	21.4	0.3	1.9	0.5	0.8
Q Avg (v)	0.1	0.1	0.2	0.2	0.1	0.5	0.0	0.0	0.0	0.1
Q Max (v)	0.5	0.5	0.5	0.0	0.4	2.6	0.5	0.0	0.4	0.4
Int 2										
Volume (vph)	4.5	6.9	6.7	10.1	2.2	5.7	4.5	2.1	6.6	1.5
Delay Avg (spv)	0.3	1.1	0.9	0.7	5.2	8.9	1.3	1.1	1.1	1.6
Delay Max (spv)	0.7	0.7	2.0	0.4	4.2	16.7	0.7	1.0	3.8	0.8
Q Avg (v)	0.0	0.1	0.1	0.1	0.1	0.6	0.2	0.1	0.2	0.1
Q Max (v)	0.0	0.4	0.9	0.8	0.9	2.1	0.8	0.0	0.0	0.0
Int 3										
Volume (vph)	0.9	3.0	3.0	4.1	6.5	5.0	4.2	1.1	4.4	0.9
Delay Avg (spv)	0.4	0.6	1.0	0.8	6.6	16.1	2.1	1.6	1.6	0.7
Delay Max (spv)	1.4	1.2	1.4	1.2	39.0	48.1	6.2	0.9	5.7	1.5
Q Avg (v)	0.0	0.1	0.2	0.1	0.2	1.0	0.4	0.0	0.3	0.0
Q Max (v)	0.5	0.4	0.4	1.0	1.3	2.9	1.3	0.0	1.1	0.5
Int 4										
Volume (vph)	2.7	5.5	2.6	6.6	2.4	1.3	5.9	0.5	3.4	2.1
Delay Avg (spv)	0.9	0.9	0.8	1.0	2.7	1.4	0.6	1.3	0.6	1.4
Delay Max (spv)	0.6	0.3	0.6	3.7	1.7	5.0	0.8	2.1	0.6	0.8
Q Avg (v)	0.1	0.1	0.1	0.3	0.1	0.1	0.1	0.0	0.1	0.1
Q Max (v)	0.5	0.5	0.4	0.5	0.9	0.0	0.4	0.0	0.4	0.5

Network 1 - Constrained COP - Average Value Standard Deviation

Movement	EB T 1	EB T 2	WB T 5	WB T 6	EB L 9	WB L 10	NB T 11	NB R 12	SB T 13	SB R 14
Int Volume	2.5	3.4	5.4	4.0	1.3	5.8	2.9	0.7	1.8	1.9
Delay Avg	0.6	0.5	0.9	0.8	2.4	15.5	2.2	2.2	2.2	1.5
Delay Max	2.4	3.8	7.1	4.9	5.7	23.8	6.2	4.7	4.8	6.7
Q Avg	0.1	0.0	0.1	0.1	0.1	1.1	0.1	0.0	0.2	0.1
Q Max	0.5	0.4	1.0	1.5	0.4	2.2	0.5	0.0	0.4	0.4
Int Volume	5.6	5.1	8.4	4.8	6.4	9.2	6.0	2.0	5.1	1.6
Delay Avg	1.0	1.1	1.2	1.1	16.3	109.3	2.6	1.9	2.3	0.8
Delay Max	3.2	3.3	5.2	3.5	86.1	143.4	7.2	6.9	7.5	7.3
Q Avg	0.1	0.1	0.2	0.2	0.2	5.8	0.3	0.1	0.3	0.0
Q Max	0.5	0.5	0.8	1.0	1.1	8.1	1.5	0.4	1.8	0.0
Int Volume	6.9	4.8	5.4	6.3	9.7	7.2	5.1	0.8	6.5	1.7
Delay Avg	1.0	1.0	1.0	1.3	26.7	87.2	4.8	2.4	4.6	2.0
Delay Max	7.1	6.4	5.3	3.0	53.0	101.0	8.2	11.9	8.4	13.9
Q Avg	0.1	0.1	0.2	0.2	0.5	4.1	0.9	0.0	0.7	0.1
Q Max	0.4	0.4	1.5	1.6	1.5	6.7	2.2	0.4	1.8	0.5
Int Volume	9.5	5.5	1.5	9.1	7.6	4.9	6.7	0.4	3.9	2.2
Delay Avg	1.4	1.1	0.9	1.5	14.4	76.9	3.8	1.4	1.3	0.7
Delay Max	5.3	2.2	3.1	5.4	39.5	107.8	3.2	5.3	6.0	8.1
Q Avg	0.2	0.1	0.2	0.3	0.5	3.8	0.5	0.0	0.2	0.1
Q Max	0.5	0.9	0.5	1.3	1.8	4.1	1.0	0.4	0.4	0.4

Network 1 - COP - Average Value Standard Devaition

Movement	EB T 1	EB T 2	WB T 5	WB T 6	EB L 9	WB L 10	NB T 11	NB R 12	SB T 13	SB R 14
Int Volume	1.4	2.9	10.5	9.9	1.5	5.6	2.6	0.8	2.4	1.5
Delay Avg	0.6	0.9	1.2	1.5	3.3	1.3	1.2	2.0	0.9	2.0
Delay Max	4.0	4.8	13.2	12.3	5.2	16.9	4.9	7.4	6.1	4.6
Q Avg	0.1	0.1	0.2	0.2	0.1	0.2	0.1	0.0	0.1	0.1
Q Max	0.4	0.5	1.3	1.2	0.0	1.1	0.7	0.4	0.4	0.4
Int Volume	1.9	8.7	10.2	8.5	5.1	6.5	4.9	2.3	6.4	1.3
Delay Avg	0.8	1.0	0.9	1.3	11.2	111.9	2.7	0.6	1.9	1.3
Delay Max	6.5	3.9	1.9	7.3	37.8	178.6	6.8	5.2	5.6	11.8
Q Avg	0.1	0.1	0.2	0.2	0.3	6.9	0.3	0.0	0.3	0.0
Q Max	0.9	0.9	0.8	1.0	0.7	9.6	0.5	0.4	1.1	0.4
Int Volume	6.7	4.7	7.9	7.1	2.4	3.0	7.1	0.9	3.4	0.7
Delay Avg	1.3	1.1	0.3	0.4	24.4	41.3	3.9	2.0	0.7	1.1
Delay Max	5.2	5.5	7.8	8.2	64.9	76.0	13.8	7.2	7.2	9.1
Q Avg	0.1	0.1	0.1	0.1	0.7	2.3	0.7	0.0	0.1	0.0
Q Max	0.8	1.9	1.7	2.1	2.3	3.1	2.5	0.5	2.2	0.4
Int Volume	12.8	3.6	3.4	7.1	3.5	8.5	5.7	0.5	4.2	2.8
Delay Avg	1.0	0.6	0.8	1.6	12.1	36.0	2.8	1.1	1.2	1.2
Delay Max	5.1	1.7	5.2	3.2	58.9	55.9	5.3	4.4	7.3	5.8
Q Avg	0.1	0.1	0.1	0.3	0.5	1.9	0.4	0.0	0.2	0.1
Q Max	0.5	0.7	0.9	1.0	1.6	3.6	0.8	0.4	0.7	0.4

Network 2 - Pre -Timed Control - Replication 1

Movement	EB T 1	EB T 2	WB T 5	WB T 6	EB L 9	WB L 10	NB T 11	NB R 12	SB T 13	SB R 14
Int 1										
Volume (vph)	205	194	224	234	94	94	70	115	119	73
Delay Avg (spv)	19.0	18.7	27.0	28.4	32.5	35.1	29.5	30.5	32.8	27.3
Delay Max (spv)	50.6	50.1	51.1	52.4	71.3	105.2	68.0	67.4	68.4	64.4
Q Avg (v)	2.6	2.4	3.9	4.3	2.0	2.2	1.4	2.4	2.7	1.4
Q Max (v)	10	9	11	11	6	11	5	7	7	4
Int 2										
Volume (vph)	212	231	244	257	69	79	162	168	191	49
Delay Avg (spv)	15.1	12.4	13.0	12.6	58.3	57.7	27.3	28.5	29.7	20.9
Delay Max (spv)	53.5	54.4	55.2	55.3	120.9	127.7	61.4	64.6	67.9	62.0
Q Avg (v)	2.1	1.9	2.2	2.2	2.8	3.2	2.9	3.1	3.7	0.7
Q Max (v)	9	9	7	7	7	11	8	9	10	3
Int 3										
Volume (vph)	257	262	268	262	66	52	266	52	264	44
Delay Avg (spv)	23.6	22.4	13.8	14.7	52.4	61.9	37.0	14.8	23.8	11.1
Delay Max (spv)	67.4	76.6	59.1	63.3	94.6	124.6	74.6	57.3	63.8	55.7
Q Avg (v)	4.2	4.0	2.5	2.6	2.2	2.0	6.5	0.5	4.3	0.3
Q Max (v)	14	13	8	8	6	6	13	4	10	3
Int 4										
Volume (vph)	233	241	254	245	105	73	199	75	164	103
Delay Avg (spv)	13.5	14.3	26.3	24.1	68.6	34.2	31.0	21.5	25.9	26.2
Delay Max (spv)	56.1	56.2	57.9	53.7	145.1	74.1	69.7	58.4	61.7	60.1
Q Avg (v)	2.1	2.3	4.4	3.8	5.2	1.7	4.1	1.0	2.8	1.8
Q Max (v)	6	7	12	12	12	6	12	4	8	6
Int 5										
Volume (vph)	249	0	221	0	62	104	233	54	258	30
Delay Avg (spv)	18.4	0.0	11.8	0.0	54.6	98.7	26.3	16.0	30.7	18.4
Delay Max (spv)	74.5	0.0	66.5	0.0	85.6	180.9	56.4	54.4	72.1	56.1
Q Avg (v)	3.1	0.0	1.7	0.0	2.2	6.8	4.0	0.5	5.1	0.3
Q Max (v)	14	0	8	0	6	13	11	3	14	2
Int 6										
Volume (vph)	258	0	209	0	47	38	173	64	217	49
Delay Avg (spv)	23.0	0.0	22.0	0.0	36.2	43.7	23.5	18.5	25.6	19.5
Delay Max (spv)	53.0	0.0	54.0	0.0	71.9	98.6	56.4	56.4	58.4	56.4
Q Avg (v)	3.9	0.0	3.2	0.0	1.2	1.2	2.6	0.8	3.6	0.6
Q Max (v)	12	0	9	0	3	5	9	3	10	3
Int 7										
Volume (vph)	242	0	256	0	64	66	183	48	164	48
Delay Avg (spv)	20.6	0.0	24.5	0.0	56.4	36.5	27.2	21.6	25.9	21.4
Delay Max (spv)	54.3	0.0	53.1	0.0	122.2	77.4	59.1	59.4	61.4	59.4
Q Avg (v)	3.3	0.0	4.2	0.0	2.3	1.5	3.3	0.7	2.8	0.7
Q Max (v)	11	0	12	0	8	4	10	3	10	3

Network 2 - Constrained COP - Replication 1

Movement	EB T 1	EB T 2	WB T 5	WB T 6	EB L 9	WB L 10	NB T 11	NB R 12	SB T 13	SB R 14
Int 1										
Volume (vph)	204	193	237	232	90	87	74	118	124	76
Delay Avg (spv)	7.7	7.5	7.8	8.0	36.8	48.9	13.2	14.7	14.9	12.8
Delay Max (spv)	33.1	28.4	32.9	39.1	95.5	117.5	34.4	50.1	52.5	43.1
Q Avg (v)	1.0	0.9	1.2	1.2	2.2	3.0	0.6	1.1	1.2	0.6
Q Max (v)	6	4	8	7	6	11	3	5	4	3
Int 2										
Volume (vph)	213	223	260	250	70	74	166	166	188	48
Delay Avg (spv)	10.1	9.9	8.4	8.3	143.4	116.2	19.3	19.0	26.6	12.2
Delay Max (spv)	44.1	42.2	50.9	53.9	275.6	217.5	70.5	57.5	103.9	44.1
Q Avg (v)	1.4	1.4	1.4	1.3	7.2	6.2	2.1	2.0	3.2	0.4
Q Max (v)	7	7	6	7	16	16	9	8	12	2
Int 3										
Volume (vph)	252	260	267	266	52	52	262	56	257	51
Delay Avg (spv)	15.1	15.4	10.4	10.4	380.3	295.0	35.1	11.2	29.3	13.7
Delay Max (spv)	50.3	65.5	52.3	47.9	561.6	574.0	89.6	53.3	100.6	57.0
Q Avg (v)	2.5	2.7	1.8	1.8	18.1	10.2	6.0	0.4	5.3	0.5
Q Max (v)	12	12	10	9	31	22	19	4	21	6
Int 4										
Volume (vph)	238	257	252	242	80	75	194	74	162	101
Delay Avg (spv)	8.4	8.4	13.4	13.0	134.5	55.2	34.4	15.0	18.9	13.1
Delay Max (spv)	51.4	50.8	48.6	52.0	282.0	152.5	84.1	55.4	59.0	57.1
Q Avg (v)	1.3	1.4	2.2	2.0	8.1	2.9	4.4	0.7	2.0	0.8
Q Max (v)	7	6	10	10	15	9	11	3	7	5
Int 5										
Volume (vph)	258	0	236	0	60	74	224	52	249	29
Delay Avg (spv)	27.1	0.0	23.0	0.0	51.9	125.3	18.1	8.7	23.1	10.1
Delay Max (spv)	80.6	0.0	81.9	0.0	192.7	317.7	55.5	40.1	70.5	39.9
Q Avg (v)	4.7	0.0	3.6	0.0	2.1	6.1	2.7	0.3	3.8	0.2
Q Max (v)	14	0	16	0	8	18	11	3	14	1
Int 6										
Volume (vph)	253	0	231	0	50	35	173	64	215	50
Delay Avg (spv)	13.3	0.0	12.8	0.0	53.3	57.9	11.9	8.6	14.1	6.1
Delay Max (spv)	37.5	0.0	53.2	0.0	143.0	141.2	34.7	33.3	37.5	25.4
Q Avg (v)	2.2	0.0	2.0	0.0	1.8	1.3	1.3	0.3	1.9	0.2
Q Max (v)	9	0	9	0	5	5	7	3	7	2
Int 7										
Volume (vph)	236	0	256	0	73	69	183	48	166	49
Delay Avg (spv)	13.5	0.0	12.6	0.0	57.7	35.7	16.7	8.4	13.4	8.9
Delay Max (spv)	52.8	0.0	38.0	0.0	133.1	84.1	57.5	33.4	57.0	33.3
Q Avg (v)	2.0	0.0	2.1	0.0	2.7	1.5	2.0	0.2	1.4	0.3
Q Max (v)	11	0	8	0	7	5	8	2	7	2

Network 2 - COP - Replication 1

Movement	EB T 1	EB T 2	WB T 5	WB T 6	EB L 9	WB L 10	NB T 11	NB R 12	SB T 13	SB R 14
Int 1										
Volume (vph)	200	191	231	233	91	80	73	120	125	76
Delay Avg (spv)	8.8	7.9	6.3	6.0	44.8	46.6	14.4	17.6	17.0	12.5
Delay Max (spv)	46.0	35.6	36.8	38.8	111.5	105.2	41.3	49.1	66.0	38.0
Q Avg (v)	1.1	0.9	0.9	0.9	2.7	2.4	0.6	1.3	1.4	0.6
Q Max (v)	7	5	7	7	8	9	3	4	5	3
Int 2										
Volume (vph)	216	226	246	247	67	77	166	167	189	49
Delay Avg (spv)	11.5	10.7	8.6	8.0	60.5	165.0	16.7	17.1	20.8	11.2
Delay Max (spv)	57.4	40.3	48.0	47.4	151.4	364.8	55.0	51.4	54.6	47.4
Q Avg (v)	1.6	1.6	1.4	1.3	2.7	11.5	1.7	1.8	2.5	0.3
Q Max (v)	11	9	8	6	8	26	6	6	8	2
Int 3										
Volume (vph)	258	257	271	255	59	55	243	69	244	63
Delay Avg (spv)	14.3	15.3	10.9	12.4	268.2	118.6	25.4	12.4	24.7	10.0
Delay Max (spv)	53.5	52.3	52.3	53.4	442.2	264.6	78.2	74.9	78.9	69.1
Q Avg (v)	2.5	2.6	2.0	2.1	14.3	4.7	4.0	0.6	4.1	0.4
Q Max (v)	11	11	10	10	23	14	13	4	16	4
Int 4										
Volume (vph)	241	251	250	242	73	79	199	73	163	100
Delay Avg (spv)	7.4	7.8	12.3	11.9	136.1	70.3	29.1	11.1	16.5	12.3
Delay Max (spv)	52.0	55.3	41.9	47.3	362.2	205.1	72.7	48.0	54.0	44.4
Q Avg (v)	1.1	1.2	2.0	1.9	10.8	3.5	3.8	0.5	1.7	0.8
Q Max (v)	9	9	8	9	31	10	11	3	8	4
Int 5										
Volume (vph)	258	0	233	0	68	73	224	52	251	29
Delay Avg (spv)	25.9	0.0	17.7	0.0	48.0	53.4	19.3	9.2	30.4	11.1
Delay Max (spv)	69.9	0.0	63.0	0.0	141.5	174.6	61.9	38.4	72.4	41.4
Q Avg (v)	4.4	0.0	2.7	0.0	2.1	2.6	2.9	0.3	5.1	0.2
Q Max (v)	13	0	10	0	9	9	10	2	13	1
Int 6										
Volume (vph)	255	0	216	0	54	44	170	62	212	48
Delay Avg (spv)	12.1	0.0	9.0	0.0	48.8	59.7	12.4	7.9	13.3	8.2
Delay Max (spv)	41.5	0.0	44.3	0.0	119.5	163.6	35.4	29.5	42.2	30.1
Q Avg (v)	2.0	0.0	1.3	0.0	1.6	1.7	1.3	0.3	1.8	0.2
Q Max (v)	8	0	7	0	4	6	6	2	6	2
Int 7										
Volume (vph)	230	0	258	0	67	66	182	48	164	48
Delay Avg (spv)	8.5	0.0	13.0	0.0	51.3	30.3	15.3	7.2	12.7	10.6
Delay Max (spv)	43.9	0.0	45.4	0.0	149.3	69.4	49.3	31.4	43.0	39.1
Q Avg (v)	1.2	0.0	2.2	0.0	2.2	1.3	1.8	0.2	1.3	0.3
Q Max (v)	9	0	9	0	6	4	8	1	6	2

Network 2 - Pre -Timed Control - Replication 2

Movement	EB T	EB T	WB T	WB T	EB L	WB L	NB T	NB R	SB T	SB R
	1	2	5	6	9	10	11	12	13	14
Int 1										
Volume (vph)	204	200	248	243	98	79	74	121	124	69
Delay Avg (spv)	20.9	19.7	29.5	30.9	34.7	30.3	28.9	31.1	31.6	28.9
Delay Max (spv)	50.0	51.0	51.1	52.2	71.6	60.5	68.1	68.1	67.3	66.9
Q Avg (v)	2.8	2.6	4.8	4.9	2.1	1.5	1.4	2.6	2.7	1.4
Q Max (v)	9	9	14	14	7	8	4	7	7	4
Int 2										
Volume (vph)	224	230	259	265	74	71	172	163	183	51
Delay Avg (spv)	15.3	14.1	12.3	13.9	48.7	48.9	27.5	27.3	28.9	21.8
Delay Max (spv)	52.6	53.0	55.0	55.1	73.8	91.3	63.0	63.9	62.4	61.5
Q Avg (v)	2.3	2.1	2.1	2.4	2.4	2.3	3.1	2.9	3.5	0.7
Q Max (v)	7	7	7	8	8	7	9	8	10	3
Int 3										
Volume (vph)	267	270	255	267	64	72	259	49	258	51
Delay Avg (spv)	21.3	22.9	13.4	13.8	56.4	79.2	26.2	14.7	25.9	17.6
Delay Max (spv)	75.3	72.7	58.4	60.1	122.2	145.9	68.8	56.6	70.1	58.2
Q Avg (v)	3.8	4.2	2.3	2.5	2.4	3.6	4.6	0.5	4.4	0.6
Q Max (v)	13	12	7	7	8	8	10	2	13	4
Int 4										
Volume (vph)	247	261	250	259	87	75	206	76	170	106
Delay Avg (spv)	15.2	15.1	24.2	26.5	62.1	32.7	30.9	24.4	26.3	24.9
Delay Max (spv)	58.9	56.0	55.6	62.2	139.2	74.4	73.3	61.1	60.9	61.4
Q Avg (v)	2.5	2.7	3.9	4.5	3.7	1.7	4.3	1.2	3.0	1.7
Q Max (v)	7	7	11	13	14	5	12	4	9	6
Int 5										
Volume (vph)	242	0	217	0	67	95	232	52	255	30
Delay Avg (spv)	17.7	0.0	8.7	0.0	44.9	68.6	24.3	19.4	26.2	19.6
Delay Max (spv)	70.2	0.0	57.3	0.0	83.3	147.0	58.4	53.0	68.9	55.3
Q Avg (v)	2.9	0.0	1.3	0.0	2.0	4.3	3.7	0.6	4.3	0.4
Q Max (v)	12	0	7	0	5	11	12	3	13	2
Int 6										
Volume (vph)	254	0	211	0	46	33	164	63	214	49
Delay Avg (spv)	23.7	0.0	21.6	0.0	37.5	46.4	24.8	17.9	25.8	18.0
Delay Max (spv)	54.1	0.0	53.3	0.0	77.4	124.2	57.4	51.4	58.1	57.4
Q Avg (v)	4.0	0.0	3.2	0.0	1.2	1.0	2.6	0.7	3.6	0.6
Q Max (v)	11	0	9	0	3	5	9	3	11	3
Int 7										
Volume (vph)	243	0	253	0	72	68	196	49	162	49
Delay Avg (spv)	22.4	0.0	22.3	0.0	59.0	39.1	27.1	22.0	27.3	18.5
Delay Max (spv)	53.6	0.0	52.7	0.0	132.0	77.0	61.0	59.1	59.1	52.4
Q Avg (v)	3.6	0.0	3.8	0.0	2.8	1.7	3.5	0.7	3.0	0.6
Q Max (v)	11	0	11	0	8	5	10	3	9	3

Network 2 - Constrained COP - Replication 2

Movement	EB T	EB T	WB T	WB T	EB L	WB L	NB T	NB R	SB T	SB R
	1	2	5	6	9	10	11	12	13	14
Int 1										
Volume (vph)	200	198	224	238	100	76	75	124	129	71
Delay Avg (spv)	9.9	8.8	7.6	8.3	44.7	42.9	13.3	15.6	17.4	13.6
Delay Max (spv)	34.4	39.6	35.1	41.9	142.1	129.6	41.9	55.0	52.1	39.4
Q Avg (v)	1.3	1.1	1.1	1.3	2.7	2.0	0.6	1.3	1.4	0.6
Q Max (v)	6	5	8	8	9	8	3	5	5	3
Int 2										
Volume (vph)	228	237	235	246	59	88	167	161	180	50
Delay Avg (spv)	11.5	11.0	9.5	10.3	62.6	161.3	21.5	20.7	22.6	14.9
Delay Max (spv)	46.5	51.0	52.1	47.4	197.3	315.8	62.5	58.4	62.5	60.1
Q Avg (v)	1.7	1.7	1.4	1.6	2.2	11.4	2.3	2.1	2.6	0.5
Q Max (v)	8	8	9	11	7	24	8	7	11	3
Int 3										
Volume (vph)	249	283	262	257	37	39	257	56	245	58
Delay Avg (spv)	12.3	10.4	8.5	9.4	542.1	505.0	37.9	14.9	21.0	10.0
Delay Max (spv)	53.5	47.3	48.2	55.2	1019.5	852.5	119.7	58.7	76.6	54.4
Q Avg (v)	2.1	2.0	1.5	1.6	28.6	25.2	6.3	0.5	3.4	0.4
Q Max (v)	12	11	9	8	56	44	20	6	19	3
Int 4										
Volume (vph)	242	263	243	252	83	73	206	77	168	106
Delay Avg (spv)	9.2	9.1	13.7	14.1	217.4	72.2	35.5	15.8	18.7	14.9
Delay Max (spv)	47.1	51.4	47.9	47.1	501.7	206.5	83.5	60.4	64.4	59.5
Q Avg (v)	1.4	1.5	2.2	2.4	14.7	3.7	4.9	0.7	2.1	1.0
Q Max (v)	8	8	9	10	26	11	12	4	8	6
Int 5										
Volume (vph)	242	0	192	0	68	83	231	50	253	30
Delay Avg (spv)	23.1	0.0	13.5	0.0	45.7	108.4	16.2	11.1	18.5	7.1
Delay Max (spv)	84.9	0.0	69.9	0.0	165.4	226.6	53.1	65.3	60.4	40.9
Q Avg (v)	3.8	0.0	1.7	0.0	2.0	6.0	2.4	0.3	3.0	0.1
Q Max (v)	14	0	11	0	8	11	10	3	10	1
Int 6										
Volume (vph)	254	0	186	0	47	34	161	61	211	48
Delay Avg (spv)	14.2	0.0	7.7	0.0	57.5	58.5	9.3	8.7	13.5	7.2
Delay Max (spv)	42.4	0.0	39.1	0.0	124.4	206.2	34.0	25.1	40.1	31.4
Q Avg (v)	2.3	0.0	0.9	0.0	1.8	1.3	0.9	0.3	1.8	0.2
Q Max (v)	8	0	6	0	5	4	5	2	7	2
Int 7										
Volume (vph)	212	0	249	0	57	68	193	50	165	49
Delay Avg (spv)	13.1	0.0	14.6	0.0	46.7	38.8	15.7	6.5	13.1	9.9
Delay Max (spv)	55.9	0.0	45.7	0.0	115.4	90.4	43.9	32.0	36.4	42.4
Q Avg (v)	1.8	0.0	2.4	0.0	1.8	1.7	2.0	0.2	1.4	0.3
Q Max (v)	9	0	9	0	7	5	6	2	6	2

Network 2 - COP - Replication 2

Movement	EB T 1	EB T 2	WB T 5	WB T 6	EB L 9	WB L 10	NB T 11	NB R 12	SB T 13	SB R 14
Int 1										
Volume (vph)	204	200	234	243	95	75	76	124	128	72
Delay Avg (spv)	9.5	9.0	5.5	5.8	34.0	42.4	13.2	18.9	18.7	13.3
Delay Max (spv)	34.0	31.0	32.1	33.7	100.5	114.9	47.4	62.1	55.5	49.4
Q Avg (v)	1.2	1.1	0.8	0.9	2.2	2.1	0.6	1.5	1.5	0.6
Q Max (v)	6	6	5	7	7	8	3	6	6	3
Int 2										
Volume (vph)	238	223	235	255	74	88	162	158	177	50
Delay Avg (spv)	10.6	11.6	7.9	6.9	64.4	119.9	19.2	20.6	20.6	12.6
Delay Max (spv)	40.2	42.4	40.7	39.5	129.5	267.3	48.3	50.1	51.4	45.9
Q Avg (v)	1.6	1.7	1.2	1.1	3.0	7.6	2.1	2.1	2.4	0.4
Q Max (v)	7	7	7	8	8	16	7	7	7	2
Int 3										
Volume (vph)	278	274	275	275	52	56	240	60	247	49
Delay Avg (spv)	17.2	16.6	9.5	9.6	213.4	130.8	31.8	16.2	34.2	11.3
Delay Max (spv)	62.5	61.1	56.5	58.5	442.6	369.6	92.2	72.6	96.8	45.4
Q Avg (v)	3.2	3.0	1.7	1.7	9.7	5.2	5.1	0.6	5.7	0.4
Q Max (v)	13	14	9	8	23	14	16	4	15	4
Int 4										
Volume (vph)	239	249	239	247	86	82	202	74	167	103
Delay Avg (spv)	6.8	7.4	13.4	13.1	248.9	63.0	30.9	12.0	16.7	14.3
Delay Max (spv)	39.1	41.9	42.7	40.9	473.7	167.1	71.9	45.5	51.1	53.1
Q Avg (v)	1.1	1.2	2.1	2.2	16.8	3.2	4.3	0.6	1.8	0.9
Q Max (v)	6	8	8	9	36	9	12	3	8	4
Int 5										
Volume (vph)	256	0	207	0	63	73	225	49	244	30
Delay Avg (spv)	27.1	0.0	9.5	0.0	43.2	102.8	17.2	11.0	19.1	10.0
Delay Max (spv)	74.0	0.0	56.8	0.0	123.1	205.6	45.8	39.4	47.7	44.5
Q Avg (v)	4.6	0.0	1.3	0.0	1.7	5.9	2.6	0.3	3.1	0.2
Q Max (v)	14	0	7	0	6	15	9	2	12	1
Int 6										
Volume (vph)	254	0	194	0	46	38	162	61	211	48
Delay Avg (spv)	13.0	0.0	7.9	0.0	32.5	66.8	11.8	7.0	12.4	5.3
Delay Max (spv)	42.0	0.0	48.3	0.0	87.0	156.2	35.3	32.4	38.7	25.4
Q Avg (v)	2.1	0.0	1.0	0.0	1.0	1.7	1.2	0.3	1.7	0.1
Q Max (v)	8	0	11	0	4	5	5	2	7	2
Int 7										
Volume (vph)	232	0	258	0	68	66	193	50	164	50
Delay Avg (spv)	10.7	0.0	15.5	0.0	47.9	34.2	17.1	6.0	11.3	9.2
Delay Max (spv)	51.6	0.0	45.1	0.0	140.3	97.3	42.1	28.0	32.0	38.9
Q Avg (v)	1.7	0.0	2.5	0.0	2.1	1.5	2.1	0.2	1.2	0.3
Q Max (v)	10	0	9	0	7	5	7	2	5	2

Network 2 - Pre -Timed Control - Replication 3

Movement	EB T 1	EB T 2	WB T 5	WB T 6	EB L 9	WB L 10	NB T 11	NB R 12	SB T 13	SB R 14
Int 1										
Volume (vph)	206	204	238	239	98	84	73	122	123	73
Delay Avg (spv)	19.4	19.5	30.4	29.3	33.6	36.3	31.1	32.1	28.7	27.7
Delay Max (spv)	50.4	51.0	51.6	51.3	71.4	100.2	68.4	67.5	67.1	68.4
Q Avg (v)	2.6	2.6	4.7	4.6	2.1	2.0	1.6	2.7	2.4	1.4
Q Max (v)	9	9	14	12	6	10	5	7	7	4
Int 2										
Volume (vph)	236	222	250	261	77	81	164	171	187	53
Delay Avg (spv)	14.2	14.2	12.6	12.9	49.3	48.1	29.0	28.6	30.8	23.7
Delay Max (spv)	53.1	54.4	58.7	56.1	80.2	123.2	62.6	62.3	64.4	62.0
Q Avg (v)	2.2	2.1	2.1	2.3	2.6	2.6	3.1	3.2	3.8	0.8
Q Max (v)	8	8	8	8	7	7	10	8	10	3
Int 3										
Volume (vph)	264	271	264	264	72	57	244	55	253	49
Delay Avg (spv)	24.8	24.4	11.6	11.9	109.8	63.8	19.8	13.6	23.7	8.5
Delay Max (spv)	71.0	71.5	56.9	59.0	238.9	141.3	63.4	57.3	63.8	57.6
Q Avg (v)	4.4	4.5	2.0	2.1	5.2	2.4	3.2	0.5	4.0	0.3
Q Max (v)	11	11	8	7	12	6	10	3	11	2
Int 4										
Volume (vph)	237	251	258	247	111	71	197	77	165	100
Delay Avg (spv)	12.9	14.1	23.6	24.6	86.5	34.6	27.7	20.9	28.3	23.8
Delay Max (spv)	55.7	56.6	57.1	54.3	151.0	73.1	59.9	58.1	62.3	59.1
Q Avg (v)	2.1	2.4	3.9	3.9	6.7	1.7	3.6	1.0	3.1	1.6
Q Max (v)	7	7	11	11	13	5	11	4	9	6
Int 5										
Volume (vph)	241	0	220	0	58	82	234	52	259	29
Delay Avg (spv)	19.7	0.0	17.2	0.0	50.7	67.8	28.8	18.0	28.6	13.4
Delay Max (spv)	70.5	0.0	77.7	0.0	79.1	143.4	60.8	54.5	65.0	43.4
Q Avg (v)	3.2	0.0	2.5	0.0	1.9	3.7	4.4	0.6	4.8	0.3
Q Max (v)	11	0	15	0	7	8	12	3	13	2
Int 6										
Volume (vph)	248	0	201	0	46	41	173	64	227	52
Delay Avg (spv)	21.2	0.0	22.0	0.0	36.3	50.8	24.4	20.5	25.2	21.8
Delay Max (spv)	53.1	0.0	54.1	0.0	80.0	123.8	57.3	56.5	57.4	57.4
Q Avg (v)	3.5	0.0	3.0	0.0	1.2	1.5	2.7	0.8	3.7	0.7
Q Max (v)	12	0	9	0	3	6	9	3	11	3
Int 7										
Volume (vph)	246	0	243	0	58	67	180	50	166	49
Delay Avg (spv)	20.3	0.0	25.0	0.0	44.7	34.7	25.9	21.7	28.3	18.3
Delay Max (spv)	54.3	0.0	52.4	0.0	74.6	76.9	60.3	59.5	60.6	55.1
Q Avg (v)	3.3	0.0	4.0	0.0	1.7	1.5	3.1	0.7	3.1	0.6
Q Max (v)	10	0	11	0	7	4	9	3	9	3

Network 2 - Constrained COP - Replication 3

Movement	EB T 1	EB T 2	WB T 5	WB T 6	EB L 9	WB L 10	NB T 11	NB R 12	SB T 13	SB R 14
Int 1										
Volume (vph)	203	201	212	227	99	85	76	129	128	78
Delay Avg (spv)	8.6	8.2	7.7	7.2	64.3	53.9	12.9	16.2	16.7	13.1
Delay Max (spv)	41.0	36.4	33.9	32.9	145.4	175.6	43.4	66.4	53.4	44.9
Q Avg (v)	1.1	1.0	1.0	1.0	4.0	3.0	0.6	1.3	1.4	0.6
Q Max (v)	8	6	7	7	11	10	3	7	6	3
Int 2										
Volume (vph)	232	230	236	242	69	81	165	170	187	52
Delay Avg (spv)	12.8	12.8	8.1	8.3	85.0	138.7	19.8	22.3	20.4	16.8
Delay Max (spv)	53.3	54.3	52.5	55.2	202.8	299.3	72.9	67.0	61.4	63.3
Q Avg (v)	1.9	1.9	1.3	1.3	3.9	9.5	2.1	2.4	2.4	0.6
Q Max (v)	10	10	6	6	11	31	9	9	9	3
Int 3										
Volume (vph)	275	272	263	266	62	63	234	57	227	49
Delay Avg (spv)	11.9	12.5	9.2	9.7	328.8	181.6	40.3	17.0	24.3	16.2
Delay Max (spv)	46.7	48.8	64.9	53.1	480.9	365.9	134.8	109.5	140.2	97.8
Q Avg (v)	2.2	2.3	1.6	1.7	15.1	7.5	6.4	0.6	3.8	0.5
Q Max (v)	10	10	10	9	25	17	19	4	15	5
Int 4										
Volume (vph)	247	255	250	241	77	75	196	74	163	97
Delay Avg (spv)	5.9	6.6	12.8	13.6	236.2	54.1	30.5	16.9	21.9	17.2
Delay Max (spv)	43.6	43.8	45.5	49.0	499.4	150.4	91.6	63.4	59.3	63.9
Q Avg (v)	0.9	1.1	2.1	2.1	17.6	2.6	3.9	0.8	2.3	1.1
Q Max (v)	6	8	11	10	38	8	12	4	8	6
Int 5										
Volume (vph)	223	0	201	0	76	90	232	52	255	30
Delay Avg (spv)	26.2	0.0	14.6	0.0	90.0	80.3	20.8	11.9	25.9	7.9
Delay Max (spv)	80.1	0.0	70.7	0.0	176.0	152.7	60.4	53.3	82.0	38.1
Q Avg (v)	4.1	0.0	2.0	0.0	4.5	4.8	3.2	0.4	4.3	0.1
Q Max (v)	15	0	10	0	12	10	12	3	17	1
Int 6										
Volume (vph)	251	0	197	0	46	29	170	62	221	51
Delay Avg (spv)	13.1	0.0	9.3	0.0	59.0	64.7	12.9	8.0	15.1	10.1
Delay Max (spv)	41.1	0.0	48.0	0.0	152.1	135.1	48.1	31.0	46.0	39.1
Q Avg (v)	2.1	0.0	1.2	0.0	1.8	1.3	1.4	0.3	2.2	0.3
Q Max (v)	8	0	10	0	5	5	7	2	8	2
Int 7										
Volume (vph)	233	0	241	0	56	67	182	52	169	50
Delay Avg (spv)	13.3	0.0	13.0	0.0	37.5	29.3	14.7	9.0	12.1	10.2
Delay Max (spv)	54.4	0.0	41.4	0.0	103.0	92.1	43.3	40.0	41.0	36.5
Q Avg (v)	2.0	0.0	2.0	0.0	1.4	1.3	1.7	0.3	1.3	0.3
Q Max (v)	9	0	9	0	7	4	6	2	6	2

Network 2 - COP - Replication 3

Movement	EB T 1	EB T 2	WB T 5	WB T 6	EB L 9	WB L 10	NB T 11	NB R 12	SB T 13	SB R 14
Int 1										
Volume (vph)	202	199	223	241	98	74	78	131	131	80
Delay Avg (spv)	9.4	9.5	5.7	7.1	37.6	35.5	13.9	18.7	17.0	12.8
Delay Max (spv)	32.4	35.7	27.0	34.1	94.1	80.9	55.0	50.1	62.4	40.4
Q Avg (v)	1.2	1.2	0.8	1.1	2.3	1.7	0.7	1.5	1.4	0.6
Q Max (v)	5	6	7	8	7	7	3	5	6	3
Int 2										
Volume (vph)	215	228	236	254	82	86	165	170	185	52
Delay Avg (spv)	9.3	9.6	8.4	10.1	78.7	127.0	21.7	20.5	21.4	13.4
Delay Max (spv)	39.5	43.5	48.0	45.0	200.9	285.3	61.1	63.1	62.4	50.0
Q Avg (v)	1.3	1.4	1.3	1.7	4.4	8.0	2.4	2.2	2.6	0.4
Q Max (v)	7	8	9	9	15	18	8	9	9	2
Int 3										
Volume (vph)	273	267	264	279	51	49	232	59	238	52
Delay Avg (spv)	14.2	15.6	9.6	9.6	314.0	253.0	27.2	15.4	35.1	17.4
Delay Max (spv)	41.9	44.2	42.4	45.9	473.9	461.0	97.3	64.8	110.1	62.2
Q Avg (v)	2.5	2.7	1.6	1.7	17.1	9.1	4.3	0.6	5.6	0.6
Q Max (v)	11	12	10	8	34	17	17	7	17	5
Int 4										
Volume (vph)	245	249	250	240	79	75	195	75	163	99
Delay Avg (spv)	8.5	8.1	14.3	11.8	137.7	48.8	25.9	13.8	17.3	15.3
Delay Max (spv)	37.5	37.5	44.1	41.0	326.8	127.1	71.9	48.4	60.9	45.5
Q Avg (v)	1.3	1.4	2.4	1.9	7.9	2.4	3.3	0.7	1.8	1.0
Q Max (v)	8	8	11	8	18	6	9	3	7	4
Int 5										
Volume (vph)	248	0	220	0	57	63	231	52	254	30
Delay Avg (spv)	22.8	0.0	11.4	0.0	81.2	57.9	17.2	11.2	24.4	10.4
Delay Max (spv)	60.6	0.0	48.4	0.0	229.3	163.8	51.9	56.3	64.1	53.5
Q Avg (v)	3.6	0.0	1.6	0.0	3.3	2.5	2.6	0.4	4.1	0.2
Q Max (v)	12	0	8	0	8	8	10	3	12	2
Int 6										
Volume (vph)	246	0	212	0	47	31	171	62	222	51
Delay Avg (spv)	14.5	0.0	8.5	0.0	53.3	45.1	11.0	7.9	13.1	8.3
Delay Max (spv)	43.9	0.0	43.6	0.0	152.1	146.5	30.5	33.0	41.4	28.1
Q Avg (v)	2.3	0.0	1.2	0.0	1.7	0.9	1.2	0.3	1.9	0.3
Q Max (v)	9	0	8	0	5	4	6	2	8	2
Int 7										
Volume (vph)	214	0	243	0	65	65	183	53	169	51
Delay Avg (spv)	12.7	0.0	12.0	0.0	49.1	30.1	14.3	7.0	11.2	8.0
Delay Max (spv)	47.0	0.0	40.7	0.0	135.8	78.3	37.8	29.0	35.9	29.4
Q Avg (v)	1.8	0.0	1.9	0.0	2.2	1.3	1.6	0.2	1.2	0.2
Q Max (v)	10	0	8	0	8	5	6	2	6	2

Network 2 - Pre -Timed Control - Replication 4

Movement	EB T 1	EB T 2	WB T 5	WB T 6	EB L 9	WB L 10	NB T 11	NB R 12	SB T 13	SB R 14
Int 1										
Volume (vph)	209	200	245	242	96	98	74	120	118	71
Delay Avg (spv)	19.1	18.6	28.7	29.8	33.1	33.5	27.9	30.0	32.7	30.4
Delay Max (spv)	50.8	50.3	52.2	52.1	71.4	101.9	67.4	68.4	68.4	68.4
Q Avg (v)	2.6	2.4	4.6	4.7	2.1	2.1	1.4	2.5	2.7	1.5
Q Max (v)	9	9	11	12	6	10	5	7	7	4
Int 2										
Volume (vph)	229	224	270	264	73	77	165	170	200	50
Delay Avg (spv)	13.6	13.7	13.7	15.5	62.5	51.9	28.1	29.6	30.1	22.3
Delay Max (spv)	53.1	53.4	55.1	54.8	143.9	118.5	61.4	75.0	67.0	62.4
Q Avg (v)	2.1	2.0	2.4	2.7	3.1	2.8	3.0	3.3	3.9	0.7
Q Max (v)	8	8	8	8	9	8	8	9	11	3
Int 3										
Volume (vph)	264	262	265	277	78	55	241	66	251	49
Delay Avg (spv)	24.0	23.9	13.1	11.9	76.8	60.4	22.6	16.8	18.1	13.5
Delay Max (spv)	73.0	72.8	56.2	55.4	164.1	130.4	59.9	58.0	60.9	56.9
Q Avg (v)	4.2	4.2	2.3	2.2	3.9	2.1	3.6	0.7	3.0	0.4
Q Max (v)	12	13	8	8	9	7	9	4	9	3
Int 4										
Volume (vph)	261	263	255	260	81	75	196	76	170	103
Delay Avg (spv)	16.1	15.5	27.0	26.4	54.3	33.9	28.9	22.7	29.2	24.7
Delay Max (spv)	56.8	57.0	57.4	61.7	131.1	74.3	60.4	61.3	62.3	61.1
Q Avg (v)	2.8	2.7	4.4	4.4	3.1	1.7	3.8	1.1	3.3	1.7
Q Max (v)	8	8	13	13	9	5	10	4	9	6
Int 5										
Volume (vph)	237	0	234	0	58	67	248	54	265	30
Delay Avg (spv)	18.0	0.0	14.8	0.0	47.7	60.7	26.4	21.6	30.7	18.0
Delay Max (spv)	65.6	0.0	67.8	0.0	76.2	133.8	62.4	55.4	76.3	48.4
Q Avg (v)	2.9	0.0	2.3	0.0	1.7	2.7	4.2	0.7	5.3	0.4
Q Max (v)	11	0	10	0	5	8	12	3	14	2
Int 6										
Volume (vph)	236	0	222	0	52	38	171	63	217	52
Delay Avg (spv)	22.7	0.0	26.3	0.0	40.7	46.7	23.2	20.4	27.8	20.5
Delay Max (spv)	52.9	0.0	54.0	0.0	80.1	116.0	57.0	55.1	59.2	55.0
Q Avg (v)	3.6	0.0	3.9	0.0	1.4	1.3	2.6	0.8	3.9	0.7
Q Max (v)	10	0	11	0	4	6	8	4	12	2
Int 7										
Volume (vph)	227	0	252	0	66	71	181	48	165	48
Delay Avg (spv)	20.1	0.0	23.0	0.0	55.8	35.2	26.6	21.9	25.4	25.5
Delay Max (spv)	53.8	0.0	51.7	0.0	126.5	75.5	61.0	59.9	61.0	58.4
Q Avg (v)	3.1	0.0	3.9	0.0	2.3	1.6	3.2	0.7	2.8	0.8
Q Max (v)	9	0	13	0	8	5	9	3	8	3

Network 2 - Constrained COP - Replication 4

Movement	EB T 1	EB T 2	WB T 5	WB T 6	EB L 9	WB L 10	NB T 11	NB R 12	SB T 13	SB R 14
Int 1										
Volume (vph)	205	195	223	233	102	81	76	123	122	74
Delay Avg (spv)	8.5	8.5	8.5	9.0	65.4	34.4	13.7	18.3	17.3	14.0
Delay Max (spv)	30.3	36.0	36.5	39.7	193.0	92.8	40.5	44.4	54.4	49.1
Q Avg (v)	1.1	1.0	1.2	1.3	4.1	1.9	0.6	1.5	1.4	0.6
Q Max (v)	6	7	9	10	11	9	3	6	6	3
Int 2										
Volume (vph)	224	219	233	261	75	80	164	167	194	48
Delay Avg (spv)	10.2	10.3	9.4	9.5	94.7	191.1	18.9	17.6	22.1	12.6
Delay Max (spv)	47.5	46.3	44.9	47.8	241.7	380.9	62.4	55.4	57.1	59.4
Q Avg (v)	1.5	1.5	1.4	1.6	4.8	12.8	2.0	1.9	2.8	0.4
Q Max (v)	10	11	9	9	12	29	8	7	8	2
Int 3										
Volume (vph)	264	269	267	282	39	38	250	52	246	45
Delay Avg (spv)	11.2	11.8	8.6	8.3	289.3	171.4	42.3	18.0	37.3	17.1
Delay Max (spv)	45.2	48.6	46.1	43.7	794.2	518.2	122.8	89.2	90.9	85.2
Q Avg (v)	2.0	2.1	1.5	1.5	20.4	8.5	7.0	0.6	6.0	0.5
Q Max (v)	10	9	10	11	46	25	19	5	23	5
Int 4										
Volume (vph)	243	254	248	255	80	81	190	75	169	101
Delay Avg (spv)	6.8	6.1	13.6	13.4	95.7	72.3	30.0	15.9	22.0	17.0
Delay Max (spv)	46.0	47.5	49.4	50.0	218.3	191.1	80.1	54.4	63.3	60.3
Q Avg (v)	1.1	1.0	2.2	2.2	6.0	3.6	3.8	0.8	2.4	1.1
Q Max (v)	5	6	9	9	15	10	11	4	9	6
Int 5										
Volume (vph)	233	0	210	0	67	69	244	53	263	31
Delay Avg (spv)	24.5	0.0	14.7	0.0	54.5	86.9	18.8	9.5	18.8	10.2
Delay Max (spv)	67.5	0.0	61.3	0.0	136.3	225.5	58.0	43.4	71.0	47.4
Q Avg (v)	3.8	0.0	2.0	0.0	2.2	4.1	3.0	0.3	3.2	0.2
Q Max (v)	14	0	8	0	7	10	10	2	12	2
Int 6										
Volume (vph)	235	0	201	0	54	40	165	62	211	51
Delay Avg (spv)	13.4	0.0	9.2	0.0	61.5	48.6	11.3	7.3	10.8	8.3
Delay Max (spv)	48.4	0.0	36.9	0.0	197.5	172.3	38.0	27.4	35.7	27.1
Q Avg (v)	2.1	0.0	1.3	0.0	2.2	1.3	1.2	0.3	1.5	0.3
Q Max (v)	8	0	9	0	8	4	7	2	6	2
Int 7										
Volume (vph)	209	0	249	0	68	71	185	48	170	50
Delay Avg (spv)	11.5	0.0	15.8	0.0	62.4	28.1	13.8	9.0	13.2	9.0
Delay Max (spv)	59.0	0.0	40.8	0.0	188.3	68.1	59.1	33.4	51.6	25.5
Q Avg (v)	1.6	0.0	2.6	0.0	2.8	1.3	1.6	0.3	1.4	0.3
Q Max (v)	8	0	10	0	10	4	6	2	6	2

Network 2 - COP - Replication 4

Movement	EB T 1	EB T 2	WB T 5	WB T 6	EB L 9	WB L 10	NB T 11	NB R 12	SB T 13	SB R 14
Int 1										
Volume (vph)	205	194	231	239	94	72	76	125	123	74
Delay Avg (spv)	7.7	8.1	4.9	6.2	47.8	44.6	12.6	18.1	18.9	14.2
Delay Max (spv)	33.4	33.0	27.9	40.1	102.4	96.4	44.0	48.0	48.4	46.0
Q Avg (v)	1.0	1.0	0.7	0.9	3.0	2.3	0.6	1.5	1.5	0.6
Q Max (v)	6	6	7	7	7	8	3	6	5	3
Int 2										
Volume (vph)	234	222	251	257	73	83	160	163	191	48
Delay Avg (spv)	10.1	9.9	8.4	7.2	57.0	131.6	16.5	18.7	21.7	13.1
Delay Max (spv)	57.9	46.2	55.3	55.2	158.6	251.9	71.6	72.9	65.4	69.1
Q Avg (v)	1.5	1.4	1.3	1.2	2.6	8.6	1.8	2.1	2.8	0.4
Q Max (v)	9	9	9	8	8	19	9	9	12	2
Int 3										
Volume (vph)	267	256	271	261	59	59	261	44	245	44
Delay Avg (spv)	14.0	14.8	12.7	13.7	384.6	169.0	40.8	20.8	31.7	14.6
Delay Max (spv)	62.7	62.7	54.0	60.5	658.0	364.6	108.8	76.2	93.3	67.5
Q Avg (v)	2.5	2.6	2.3	2.4	19.6	6.4	7.0	0.6	5.1	0.4
Q Max (v)	12	11	16	16	32	20	20	5	17	4
Int 4										
Volume (vph)	252	236	247	255	81	79	187	73	166	100
Delay Avg (spv)	7.9	8.0	12.3	13.0	108.1	54.8	23.5	14.5	19.9	15.7
Delay Max (spv)	44.8	53.6	47.4	44.8	255.1	132.4	67.7	52.5	58.4	48.9
Q Avg (v)	1.3	1.2	2.0	2.2	6.4	2.8	3.0	0.7	2.2	1.0
Q Max (v)	10	10	7	11	17	8	10	3	7	4
Int 5										
Volume (vph)	232	0	238	0	67	63	243	52	262	31
Delay Avg (spv)	17.2	0.0	27.6	0.0	81.8	47.8	16.6	10.7	19.4	8.2
Delay Max (spv)	66.3	0.0	111.9	0.0	209.3	121.9	45.3	30.1	55.3	38.3
Q Avg (v)	2.5	0.0	4.5	0.0	3.6	2.0	2.6	0.3	3.3	0.1
Q Max (v)	10	0	17	0	13	6	9	2	11	1
Int 6										
Volume (vph)	235	0	223	0	53	38	164	62	209	51
Delay Avg (spv)	11.9	0.0	10.4	0.0	49.3	49.3	10.9	7.7	13.6	7.7
Delay Max (spv)	39.3	0.0	40.8	0.0	119.1	123.9	36.5	41.4	39.0	31.1
Q Avg (v)	1.8	0.0	1.5	0.0	1.7	1.3	1.1	0.3	1.9	0.2
Q Max (v)	7	0	9	0	5	7	5	2	7	2
Int 7										
Volume (vph)	226	0	252	0	78	70	179	49	168	50
Delay Avg (spv)	9.8	0.0	13.7	0.0	52.4	27.5	17.5	9.1	14.9	10.9
Delay Max (spv)	53.1	0.0	41.0	0.0	144.1	72.1	52.0	35.0	43.4	43.4
Q Avg (v)	1.4	0.0	2.2	0.0	2.6	1.3	2.1	0.3	1.6	0.3
Q Max (v)	11	0	8	0	9	4	8	2	7	2

Network 2 - Pre -Timed Control - Replication 5

Movement	EB T 1	EB T 2	WB T 5	WB T 6	EB L 9	WB L 10	NB T 11	NB R 12	SB T 13	SB R 14
Int 1										
Volume (vph)	209	203	227	228	94	90	69	117	123	73
Delay Avg (spv)	20.2	18.7	26.5	28.2	33.3	37.2	29.9	32.6	29.8	27.4
Delay Max (spv)	51.2	50.4	51.9	51.1	71.4	107.0	68.4	68.1	66.0	64.0
Q Avg (v)	2.8	2.5	3.9	4.2	2.0	2.1	1.4	2.7	2.5	1.4
Q Max (v)	9	9	10	12	5	11	5	7	8	4
Int 2										
Volume (vph)	228	224	236	251	74	97	171	166	186	50
Delay Avg (spv)	15.7	15.6	12.7	13.3	53.3	60.4	27.6	27.5	28.6	22.1
Delay Max (spv)	53.4	53.4	54.2	55.7	114.5	131.9	62.0	62.3	59.1	58.9
Q Avg (v)	2.4	2.3	2.0	2.2	2.8	4.1	3.1	3.0	3.5	0.7
Q Max (v)	9	9	7	7	8	10	9	9	9	3
Int 3										
Volume (vph)	270	264	253	268	65	60	252	53	259	52
Delay Avg (spv)	21.7	21.7	10.8	11.8	50.8	65.8	23.0	10.3	19.1	15.7
Delay Max (spv)	70.4	60.2	54.9	55.1	83.3	133.5	65.7	57.3	59.7	57.5
Q Avg (v)	3.9	3.9	1.8	2.1	2.2	2.8	3.8	0.3	3.1	0.6
Q Max (v)	11	11	7	6	6	8	10	2	9	3
Int 4										
Volume (vph)	243	252	254	255	98	75	204	78	165	100
Delay Avg (spv)	13.3	15.0	25.1	23.8	68.1	33.4	29.8	26.2	26.1	23.3
Delay Max (spv)	55.9	55.7	55.2	58.6	138.7	73.9	61.4	61.4	61.4	60.9
Q Avg (v)	2.2	2.6	4.1	3.9	4.5	1.7	4.1	1.3	2.8	1.5
Q Max (v)	7	7	11	12	13	5	11	5	8	6
Int 5										
Volume (vph)	246	0	224	0	61	82	234	49	252	29
Delay Avg (spv)	18.5	0.0	11.5	0.0	48.5	63.3	27.3	21.2	30.7	14.6
Delay Max (spv)	74.8	0.0	64.7	0.0	77.5	137.9	62.0	56.3	70.4	51.3
Q Avg (v)	3.1	0.0	1.7	0.0	1.9	3.3	4.2	0.7	5.0	0.3
Q Max (v)	14	0	9	0	5	7	13	3	13	2
Int 6										
Volume (vph)	249	0	213	0	50	41	168	65	211	51
Delay Avg (spv)	22.8	0.0	24.5	0.0	35.0	47.5	23.2	18.8	26.0	18.0
Delay Max (spv)	52.1	0.0	53.9	0.0	75.0	98.4	57.0	57.0	57.4	57.4
Q Avg (v)	3.8	0.0	3.5	0.0	1.2	1.4	2.5	0.8	3.6	0.6
Q Max (v)	11	0	10	0	4	5	8	3	12	3
Int 7										
Volume (vph)	242	0	248	0	67	67	194	48	169	47
Delay Avg (spv)	22.8	0.0	23.9	0.0	53.7	36.3	28.4	19.2	26.7	22.1
Delay Max (spv)	55.4	0.0	52.4	0.0	132.0	75.3	59.8	59.1	60.1	59.4
Q Avg (v)	3.7	0.0	3.9	0.0	2.3	1.6	3.6	0.6	3.0	0.7
Q Max (v)	11	0	11	0	7	4	11	3	10	3

Network 2 - Constrained COP - Replication 5

Movement	EB T 1	EB T 2	WB T 5	WB T 6	EB L 9	WB L 10	NB T 11	NB R 12	SB T 13	SB R 14
Int 1										
Volume (vph)	206	200	214	229	93	94	71	123	129	76
Delay Avg (spv)	8.0	7.4	7.1	7.5	35.8	118.0	12.8	15.7	17.6	12.9
Delay Max (spv)	27.1	27.5	31.5	34.7	91.9	240.1	49.1	42.4	55.1	47.1
Q Avg (v)	1.0	0.9	1.0	1.1	2.2	7.4	0.6	1.2	1.4	0.6
Q Max (v)	5	5	9	7	6	18	3	5	5	3
Int 2										
Volume (vph)	225	225	247	252	76	86	170	164	184	50
Delay Avg (spv)	10.6	11.5	7.7	7.6	67.8	178.4	23.2	21.8	23.2	16.5
Delay Max (spv)	48.8	45.6	55.4	46.9	157.1	332.9	67.0	58.4	71.9	55.5
Q Avg (v)	1.5	1.7	1.2	1.2	3.5	10.6	2.6	2.3	2.8	0.5
Q Max (v)	7	8	6	7	9	18	9	7	9	3
Int 3										
Volume (vph)	255	269	271	270	46	45	256	51	259	45
Delay Avg (spv)	15.4	14.1	8.7	8.8	500.9	256.9	34.7	11.3	27.2	8.4
Delay Max (spv)	57.2	50.3	51.2	51.2	807.1	448.4	97.6	63.6	76.3	77.1
Q Avg (v)	2.6	2.5	1.5	1.6	27.0	9.0	5.8	0.4	4.7	0.2
Q Max (v)	12	10	9	9	47	18	20	4	18	3
Int 4										
Volume (vph)	231	252	246	248	79	77	204	77	162	98
Delay Avg (spv)	6.4	5.4	14.3	14.2	218.5	64.0	32.8	14.2	19.5	14.7
Delay Max (spv)	49.1	46.7	48.8	49.1	402.5	167.3	72.3	65.3	66.5	58.9
Q Avg (v)	1.0	0.9	2.3	2.3	12.9	3.2	4.4	0.7	2.0	0.9
Q Max (v)	7	6	10	10	23	10	10	4	7	5
Int 5										
Volume (vph)	250	0	221	0	59	62	233	49	248	29
Delay Avg (spv)	26.2	0.0	13.2	0.0	61.6	79.2	18.6	8.2	19.1	8.5
Delay Max (spv)	83.5	0.0	79.8	0.0	170.7	166.7	55.7	35.3	53.9	50.0
Q Avg (v)	4.4	0.0	1.9	0.0	2.3	3.9	2.8	0.2	3.1	0.1
Q Max (v)	14	0	11	0	8	13	11	2	11	1
Int 6										
Volume (vph)	250	0	213	0	51	40	163	63	204	50
Delay Avg (spv)	12.6	0.0	13.8	0.0	35.0	80.9	9.6	6.9	12.0	7.2
Delay Max (spv)	38.3	0.0	54.8	0.0	103.1	202.0	46.9	27.4	37.7	25.0
Q Avg (v)	2.0	0.0	1.9	0.0	1.2	2.3	1.0	0.3	1.6	0.2
Q Max (v)	7	0	8	0	5	8	5	2	6	2
Int 7										
Volume (vph)	226	0	247	0	61	65	196	50	172	50
Delay Avg (spv)	11.5	0.0	11.6	0.0	51.0	44.5	17.1	8.6	14.2	9.6
Delay Max (spv)	51.1	0.0	38.9	0.0	123.8	135.3	51.3	44.0	44.0	37.1
Q Avg (v)	1.7	0.0	1.8	0.0	2.1	1.9	2.1	0.3	1.6	0.3
Q Max (v)	10	0	8	0	8	6	9	2	8	2

Network 2 - COP - Replication 5

Movement	EB T 1	EB T 2	WB T 5	WB T 6	EB L 9	WB L 10	NB T 11	NB R 12	SB T 13	SB R 14
Int 1										
Volume (vph)	203	198	233	230	95	68	69	120	126	74
Delay Avg (spv)	9.8	7.6	7.4	7.8	43.8	36.8	12.8	16.2	16.2	12.8
Delay Max (spv)	35.1	32.0	42.5	41.4	110.9	92.4	45.3	47.4	40.5	41.4
Q Avg (v)	1.3	1.0	1.1	1.1	2.7	1.6	0.6	1.3	1.3	0.6
Q Max (v)	5	5	7	8	7	7	3	5	5	3
Int 2										
Volume (vph)	233	238	237	234	60	82	168	164	183	50
Delay Avg (spv)	10.7	11.3	7.6	7.8	53.5	218.0	19.0	16.9	20.2	15.4
Delay Max (spv)	44.6	46.4	41.0	42.1	111.8	379.0	51.4	61.3	54.1	48.1
Q Avg (v)	1.6	1.7	1.2	1.2	2.1	14.2	2.1	1.8	2.4	0.5
Q Max (v)	9	8	9	9	6	24	8	7	7	2
Int 3										
Volume (vph)	274	288	257	276	55	56	239	55	240	49
Delay Avg (spv)	16.2	16.0	12.5	12.4	185.7	330.7	29.4	17.1	32.4	21.7
Delay Max (spv)	67.3	64.9	57.5	60.6	398.2	538.2	87.8	80.8	104.7	73.7
Q Avg (v)	2.9	3.0	2.1	2.3	8.2	13.8	4.8	0.6	5.5	0.7
Q Max (v)	12	13	11	11	18	20	15	5	22	9
Int 4										
Volume (vph)	231	266	250	253	76	75	207	78	166	100
Delay Avg (spv)	5.3	5.7	13.0	14.0	238.0	67.3	35.7	14.2	18.5	15.4
Delay Max (spv)	45.6	40.2	45.0	47.4	538.3	159.0	92.5	48.4	70.9	53.1
Q Avg (v)	0.8	1.0	2.1	2.3	19.8	3.5	4.8	0.7	2.0	1.0
Q Max (v)	8	8	8	8	42	9	13	4	8	4
Int 5										
Volume (vph)	245	0	222	0	76	71	229	48	243	28
Delay Avg (spv)	22.4	0.0	10.0	0.0	48.3	42.5	17.1	8.5	23.1	11.5
Delay Max (spv)	73.5	0.0	67.8	0.0	135.6	94.6	65.1	43.4	68.9	36.0
Q Avg (v)	3.7	0.0	1.5	0.0	2.1	1.9	2.6	0.3	3.7	0.2
Q Max (v)	13	0	10	0	7	6	12	2	13	1
Int 6										
Volume (vph)	250	0	211	0	49	38	164	65	208	50
Delay Avg (spv)	13.1	0.0	9.0	0.0	37.3	41.4	10.7	6.7	11.2	8.6
Delay Max (spv)	36.4	0.0	40.5	0.0	88.1	120.3	34.0	31.4	41.4	30.5
Q Avg (v)	2.1	0.0	1.3	0.0	1.3	1.1	1.1	0.3	1.5	0.3
Q Max (v)	7	0	7	0	4	4	6	2	6	2
Int 7										
Volume (vph)	226	0	246	0	64	69	194	50	171	49
Delay Avg (spv)	8.9	0.0	14.4	0.0	47.6	33.6	13.7	11.7	12.7	8.1
Delay Max (spv)	51.2	0.0	51.1	0.0	107.1	80.4	43.8	38.4	44.4	30.1
Q Avg (v)	1.3	0.0	2.3	0.0	2.0	1.4	1.7	0.4	1.4	0.2
Q Max (v)	8	0	9	0	8	5	8	2	6	2

Network 2 - Pre -Timed Control - Average Values

Movement	EB T 1	EB T 2	WB T 5	WB T 6	EB L 9	WB L 10	NB T 11	NB R 12	SB T 13	SB R 14	Overall
Int 1											
Volume (vph)	206.6	200.2	236.4	237.2	96.0	89.0	72.0	119.0	121.4	71.8	1449.6
Delay Avg (spv)	19.7	19.0	28.4	29.3	33.4	34.5	29.5	31.3	31.1	28.4	27.2
Delay Max (spv)	50.6	50.5	51.6	51.8	71.4	95.0	68.1	67.9	67.4	66.4	59.5
Q Avg (v)	2.7	2.5	4.4	4.5	2.1	2.0	1.5	2.6	2.6	1.4	3.0
Q Max (v)	9.2	9.0	12.0	12.2	6.0	10.0	4.8	7.0	7.2	4.0	
Int 2											
Volume (vph)	225.8	226.2	251.8	259.6	73.4	81.0	166.8	167.6	189.4	50.6	1692.2
Delay Avg (spv)	14.8	14.0	12.8	13.6	54.4	53.4	27.9	28.3	29.6	22.2	22.3
Delay Max (spv)	53.1	53.7	55.6	55.4	106.7	118.5	62.1	65.6	64.2	61.4	63.0
Q Avg (v)	2.2	2.1	2.2	2.4	2.7	3.0	3.0	3.1	3.7	0.7	2.6
Q Max (v)	8.2	8.2	7.4	7.6	7.8	8.6	8.8	8.6	10.0	3.0	
Int 3											
Volume (vph)	264.4	265.8	261.0	267.6	69.0	59.2	252.4	55.0	257.0	49.0	1800.4
Delay Avg (spv)	23.1	23.1	12.6	12.8	69.2	66.2	25.7	14.0	22.1	13.3	22.9
Delay Max (spv)	71.4	70.7	57.1	58.6	140.6	135.1	66.5	57.3	63.7	57.1	69.5
Q Avg (v)	4.1	4.1	2.2	2.3	3.2	2.6	4.3	0.5	3.8	0.4	3.3
Q Max (v)	12.2	12.0	7.6	7.2	8.2	7.0	10.4	3.0	10.4	3.0	
Int 4											
Volume (vph)	244.2	253.6	254.2	253.2	96.4	73.8	200.4	76.4	166.8	102.4	1721.4
Delay Avg (spv)	14.2	14.8	25.2	25.1	67.9	33.7	29.7	23.1	27.2	24.6	25.4
Delay Max (spv)	56.7	56.3	56.7	58.1	141.0	74.0	64.9	60.1	61.7	60.5	64.1
Q Avg (v)	2.3	2.5	4.1	4.1	4.7	1.7	4.0	1.1	3.0	1.6	3.2
Q Max (v)	7.0	7.2	11.6	12.2	12.2	5.2	11.2	4.2	8.6	6.0	
Int 5											
Volume (vph)	243.0	0.0	223.2	0.0	61.2	86.0	236.2	52.2	257.8	29.6	1189.2
Delay Avg (spv)	18.4	0.0	12.8	0.0	49.3	71.8	26.6	19.3	29.4	16.8	26.8
Delay Max (spv)	71.1	0.0	66.8	0.0	80.3	148.6	60.0	54.7	70.5	50.9	72.8
Q Avg (v)	3.0	0.0	1.9	0.0	1.9	4.2	4.1	0.6	4.9	0.3	3.3
Q Max (v)	12.4	0.0	9.8	0.0	5.6	9.4	12.0	3.0	13.4	2.0	
Int 6											
Volume (vph)	249.0	0.0	211.2	0.0	48.2	38.2	169.8	63.8	217.2	50.6	1048.0
Delay Avg (spv)	22.7	0.0	23.3	0.0	37.1	47.0	23.8	19.2	26.1	19.6	24.9
Delay Max (spv)	53.0	0.0	53.8	0.0	76.9	112.2	57.0	55.3	58.1	56.7	58.5
Q Avg (v)	3.8	0.0	3.4	0.0	1.2	1.3	2.6	0.8	3.7	0.6	2.9
Q Max (v)	11.2	0.0	9.6	0.0	3.4	5.4	8.6	3.2	11.2	2.8	
Int 7											
Volume (vph)	240.0	0.0	250.4	0.0	65.4	67.8	186.8	48.6	165.2	48.2	1072.4
Delay Avg (spv)	21.3	0.0	23.7	0.0	53.9	36.4	27.0	21.3	26.7	21.2	26.6
Delay Max (spv)	54.3	0.0	52.5	0.0	117.5	76.4	60.2	59.4	60.4	56.9	61.4
Q Avg (v)	3.4	0.0	4.0	0.0	2.3	1.6	3.3	0.7	2.9	0.7	3.0
Q Max (v)	10.4	0.0	11.6	0.0	7.6	4.4	9.8	3.0	9.2	3.0	

Network 2 - Constrained COP - Average Values

Movement	EB T 1	EB T 2	WB T 5	WB T 6	EB L 9	WB L 10	NB T 11	NB R 12	SB T 13	SB R 14	Overall
Int 1											
Volume (vph)	203.6	197.4	222.0	231.8	96.8	84.6	74.4	123.4	126.4	75.0	1435.4
Delay Avg (spv)	8.6	8.1	7.7	8.0	49.4	59.6	13.2	16.1	16.8	13.3	15.9
Delay Max (spv)	33.1	33.6	34.0	37.7	133.6	151.1	41.9	51.6	53.5	44.7	52.2
Q Avg (v)	1.1	1.0	1.1	1.2	3.0	3.4	0.6	1.3	1.4	0.6	1.3
Q Max (v)	6.2	5.4	8.2	7.8	8.6	11.2	3.0	5.6	5.2	3.0	
Int 2											
Volume (vph)	224.4	226.8	242.2	250.2	69.8	81.8	166.4	165.6	186.6	49.6	1663.4
Delay Avg (spv)	11.0	11.1	8.6	8.8	90.7	157.2	20.5	20.3	23.0	14.6	24.2
Delay Max (spv)	48.1	47.9	51.2	50.2	214.9	309.3	67.1	59.3	71.4	56.5	74.5
Q Avg (v)	1.6	1.6	1.4	1.4	4.3	10.1	2.2	2.2	2.8	0.5	2.3
Q Max (v)	8.4	8.8	7.2	8.0	11.0	23.6	8.6	7.6	9.8	2.6	
Int 3											
Volume (vph)	259.0	270.6	266.0	268.2	47.2	47.4	251.8	54.4	246.8	49.6	1761.0
Delay Avg (spv)	13.2	12.8	9.1	9.3	408.3	282.0	38.1	14.5	27.8	13.1	35.4
Delay Max (spv)	50.6	52.1	52.5	50.2	732.7	551.8	112.9	74.9	96.9	74.3	99.7
Q Avg (v)	2.3	2.3	1.6	1.6	21.8	12.1	6.3	0.5	4.6	0.4	3.7
Q Max (v)	11.2	10.4	9.6	9.2	41.0	25.2	19.4	4.6	19.2	4.4	
Int 4											
Volume (vph)	240.2	256.2	247.8	247.6	79.8	76.2	198.0	75.4	164.8	100.6	1686.6
Delay Avg (spv)	7.3	7.1	13.6	13.7	180.5	63.5	32.6	15.6	20.2	15.4	24.9
Delay Max (spv)	47.4	48.0	48.1	49.4	380.8	173.5	82.3	59.8	62.5	59.9	76.2
Q Avg (v)	1.1	1.2	2.2	2.2	11.9	3.2	4.3	0.7	2.2	1.0	2.5
Q Max (v)	6.6	6.8	9.8	9.8	23.4	9.6	11.2	3.8	7.8	5.6	
Int 5											
Volume (vph)	241.2	0.0	212.0	0.0	66.0	75.6	232.8	51.2	253.6	29.8	1162.2
Delay Avg (spv)	25.4	0.0	15.8	0.0	60.7	96.0	18.5	9.9	21.1	8.8	26.8
Delay Max (spv)	79.3	0.0	72.7	0.0	168.2	217.9	56.5	47.4	67.6	43.3	82.7
Q Avg (v)	4.2	0.0	2.3	0.0	2.6	5.0	2.8	0.3	3.5	0.2	3.1
Q Max (v)	14.2	0.0	11.2	0.0	8.6	12.4	10.8	2.6	12.8	1.2	
Int 6											
Volume (vph)	248.6	0.0	205.6	0.0	49.6	35.6	166.4	62.4	212.4	50.0	1030.6
Delay Avg (spv)	13.3	0.0	10.6	0.0	53.3	62.1	11.0	7.9	13.1	7.8	15.4
Delay Max (spv)	41.5	0.0	46.4	0.0	144.0	171.4	40.3	28.8	39.4	29.6	49.9
Q Avg (v)	2.2	0.0	1.5	0.0	1.8	1.5	1.2	0.3	1.8	0.2	1.5
Q Max (v)	8.0	0.0	8.4	0.0	5.6	5.2	6.2	2.2	6.8	2.0	
Int 7											
Volume (vph)	223.2	0.0	248.4	0.0	63.0	68.0	187.8	49.6	168.4	49.6	1058.0
Delay Avg (spv)	12.6	0.0	13.5	0.0	51.1	35.3	15.6	8.3	13.2	9.5	16.8
Delay Max (spv)	54.6	0.0	40.9	0.0	132.7	94.0	51.0	36.6	46.0	34.9	54.8
Q Avg (v)	1.8	0.0	2.2	0.0	2.1	1.5	1.9	0.3	1.4	0.3	1.7
Q Max (v)	9.4	0.0	8.8	0.0	7.8	4.8	7.0	2.0	6.6	2.0	

Network 2 - COP - Average Values

Movement	EB T 1	EB T 2	WB T 5	WB T 6	EB L 9	WB L 10	NB T 11	NB R 12	SB T 13	SB R 14	Overall
Int 1											
Volume (vph)	202.8	196.4	230.4	237.2	94.6	73.8	74.4	124.0	126.6	75.2	1435.4
Delay Avg (spv)	9.0	8.4	6.0	6.6	41.6	41.2	13.4	17.9	17.6	13.1	13.8
Delay Max (spv)	36.2	33.5	33.2	37.6	103.9	98.0	46.6	51.3	54.5	43.0	47.0
Q Avg (v)	1.2	1.0	0.9	1.0	2.6	2.0	0.6	1.4	1.4	0.6	1.2
Q Max (v)	5.8	5.6	6.6	7.4	7.2	7.8	3.0	5.2	5.4	3.0	
Int 2											
Volume (vph)	227.2	227.4	241.0	249.4	71.2	83.2	164.2	164.4	185.0	49.8	1662.8
Delay Avg (spv)	10.5	10.6	8.2	8.0	62.8	152.3	18.6	18.8	20.9	13.1	22.0
Delay Max (spv)	47.9	43.7	46.6	45.8	150.4	309.7	57.5	59.7	57.6	52.1	67.6
Q Avg (v)	1.5	1.6	1.3	1.3	3.0	10.0	2.0	2.0	2.5	0.4	2.1
Q Max (v)	8.6	8.2	8.4	8.0	9.0	20.6	7.6	7.6	8.6	2.0	
Int 3											
Volume (vph)	270.0	268.4	267.6	269.2	55.2	55.0	243.0	57.4	242.8	51.4	1780.0
Delay Avg (spv)	15.2	15.7	11.0	11.5	273.1	200.4	30.9	16.4	31.6	15.0	32.2
Delay Max (spv)	57.6	57.1	52.5	55.8	483.0	399.6	92.9	73.9	96.8	63.6	91.1
Q Avg (v)	2.7	2.8	1.9	2.0	13.8	7.8	5.0	0.6	5.2	0.5	3.5
Q Max (v)	11.8	12.2	11.2	10.6	26.0	17.0	16.2	5.0	17.4	5.2	
Int 4											
Volume (vph)	241.6	250.2	247.2	247.4	79.0	78.0	198.0	74.6	165.0	100.4	1681.4
Delay Avg (spv)	7.2	7.4	13.1	12.7	173.7	60.9	29.0	13.1	17.8	14.6	23.5
Delay Max (spv)	43.8	45.7	44.2	44.3	391.2	158.1	75.4	48.6	59.1	49.0	71.6
Q Avg (v)	1.1	1.2	2.1	2.1	12.3	3.1	3.8	0.6	1.9	0.9	2.4
Q Max (v)	8.2	8.6	8.4	9.0	28.8	8.4	11.0	3.2	7.6	4.0	
Int 5											
Volume (vph)	247.8	0.0	224.0	0.0	66.2	68.6	230.4	50.6	250.8	29.6	1168.0
Delay Avg (spv)	23.1	0.0	15.3	0.0	60.5	60.9	17.5	10.1	23.3	10.3	24.0
Delay Max (spv)	68.8	0.0	69.6	0.0	167.8	152.1	54.0	41.5	61.7	42.7	73.2
Q Avg (v)	3.8	0.0	2.3	0.0	2.6	3.0	2.7	0.3	3.9	0.2	2.9
Q Max (v)	12.4	0.0	10.4	0.0	8.6	8.8	10.0	2.2	12.2	1.2	
Int 6											
Volume (vph)	248.0	0.0	211.2	0.0	49.8	37.8	166.2	62.4	212.4	49.6	1037.4
Delay Avg (spv)	12.9	0.0	9.0	0.0	44.2	52.5	11.4	7.5	12.7	7.6	14.2
Delay Max (spv)	40.6	0.0	43.5	0.0	113.2	142.1	34.3	33.5	40.5	29.0	46.4
Q Avg (v)	2.1	0.0	1.2	0.0	1.5	1.4	1.2	0.3	1.7	0.2	1.4
Q Max (v)	7.8	0.0	8.4	0.0	4.4	5.2	5.6	2.0	6.8	2.0	
Int 7											
Volume (vph)	225.6	0.0	251.4	0.0	68.4	67.2	186.2	50.0	167.2	49.6	1065.6
Delay Avg (spv)	10.1	0.0	13.7	0.0	49.6	31.2	15.6	8.2	12.6	9.4	16.0
Delay Max (spv)	49.4	0.0	44.7	0.0	135.3	79.5	45.0	32.3	39.7	36.2	52.0
Q Avg (v)	1.5	0.0	2.2	0.0	2.2	1.4	1.9	0.2	1.3	0.3	1.6
Q Max (v)	9.6	0.0	8.6	0.0	7.6	4.6	7.4	1.8	6.0	2.0	

Network 2 - Pre -Timed Control - Average Value Standard Deviations

Movement	EB T 1	EB T 2	WB T 5	WB T 6	EB L 9	WB L 10	NB T 11	NB R 12	SB T 13	SB R 14
Int 1										
Volume (vph)	2.3	3.9	10.6	6.2	2.0	7.6	2.3	2.9	2.7	1.8
Delay Avg (spv)	0.8	0.5	1.6	1.1	0.8	2.7	1.2	1.1	1.8	1.3
Delay Max (spv)	0.5	0.4	0.5	0.6	0.1	19.4	0.4	0.4	1.0	2.1
Q Avg (v)	0.1	0.1	0.4	0.3	0.1	0.3	0.1	0.1	0.1	0.0
Q Max (v)	0.4	0.0	1.9	1.1	0.7	1.2	0.4	0.0	0.4	0.0
Int 2										
Volume (vph)	8.8	4.0	13.2	5.7	2.9	9.7	4.4	3.2	6.6	1.5
Delay Avg (spv)	0.9	1.1	0.5	1.1	5.9	5.4	0.7	0.9	0.9	1.0
Delay Max (spv)	0.3	0.6	1.8	0.5	29.3	16.0	0.7	5.4	3.6	1.4
Q Avg (v)	0.1	0.2	0.1	0.2	0.3	0.7	0.1	0.2	0.2	0.1
Q Max (v)	0.8	0.8	0.5	0.5	0.8	1.8	0.8	0.5	0.7	0.0
Int 3										
Volume (vph)	4.8	4.4	6.6	5.8	5.9	7.7	10.4	6.5	5.1	3.1
Delay Avg (spv)	1.5	1.1	1.3	1.3	24.9	7.5	6.7	2.4	3.4	3.6
Delay Max (spv)	3.0	6.2	1.7	3.4	63.2	8.5	5.6	0.5	4.0	1.0
Q Avg (v)	0.2	0.2	0.3	0.2	1.4	0.6	1.3	0.1	0.7	0.1
Q Max (v)	1.3	1.0	0.5	0.8	2.5	1.0	1.5	1.0	1.7	0.7
Int 4										
Volume (vph)	10.8	8.8	2.9	6.9	12.4	1.8	4.4	1.1	2.9	2.5
Delay Avg (spv)	1.4	0.6	1.4	1.3	11.9	0.8	1.4	2.2	1.5	1.2
Delay Max (spv)	1.3	0.5	1.2	4.0	7.5	0.5	6.1	1.7	0.6	0.9
Q Avg (v)	0.3	0.2	0.2	0.3	1.4	0.0	0.3	0.1	0.2	0.1
Q Max (v)	0.7	0.4	0.9	0.8	1.9	0.4	0.8	0.4	0.5	0.0
Int 5										
Volume (vph)	4.6	0.0	6.5	0.0	3.7	14.1	6.6	2.0	4.9	0.5
Delay Avg (spv)	0.7	0.0	3.3	0.0	3.6	15.4	1.6	2.3	2.0	2.7
Delay Max (spv)	3.7	0.0	7.3	0.0	4.0	18.7	2.6	1.2	4.1	5.2
Q Avg (v)	0.2	0.0	0.5	0.0	0.2	1.6	0.3	0.1	0.4	0.1
Q Max (v)	1.5	0.0	3.1	0.0	0.9	2.5	0.7	0.0	0.5	0.0
Int 6										
Volume (vph)	8.3	0.0	7.6	0.0	2.7	3.3	3.8	0.8	6.0	1.5
Delay Avg (spv)	0.9	0.0	2.0	0.0	2.2	2.6	0.7	1.2	1.0	1.6
Delay Max (spv)	0.7	0.0	0.3	0.0	3.5	12.9	0.4	2.3	0.7	1.1
Q Avg (v)	0.2	0.0	0.3	0.0	0.1	0.2	0.1	0.1	0.1	0.1
Q Max (v)	0.8	0.0	0.9	0.0	0.5	0.5	0.5	0.4	0.8	0.4
Int 7										
Volume (vph)	7.4	0.0	5.0	0.0	5.1	1.9	7.6	0.9	2.6	0.8
Delay Avg (spv)	1.3	0.0	1.1	0.0	5.5	1.7	0.9	1.2	1.2	3.0
Delay Max (spv)	0.7	0.0	0.5	0.0	24.3	0.9	0.8	0.3	0.9	3.1
Q Avg (v)	0.3	0.0	0.2	0.0	0.4	0.1	0.2	0.0	0.1	0.1
Q Max (v)	0.9	0.0	0.9	0.0	0.5	0.5	0.8	0.0	0.8	0.0

Network 2 - Constrained COP - Average Value Standard Deviations

Movement	EB T 1	EB T 2	WB T 5	WB T 6	EB L 9	WB L 10	NB T 11	NB R 12	SB T 13	SB R 14
Int 1										
Volume (vph)	2.3	3.4	9.9	4.2	5.1	6.7	2.1	3.9	3.2	2.6
Delay Avg (spv)	0.9	0.6	0.5	0.7	14.5	33.4	0.4	1.4	1.1	0.5
Delay Max (spv)	5.2	5.3	1.9	3.8	41.6	58.1	5.3	9.6	1.2	3.7
Q Avg (v)	0.1	0.1	0.1	0.1	0.9	2.3	0.0	0.1	0.1	0.0
Q Max (v)	1.1	1.1	0.8	1.3	2.5	4.0	0.0	0.9	0.8	0.0
Int 2										
Volume (vph)	7.1	6.9	11.3	7.2	6.8	5.5	2.3	3.4	5.2	1.7
Delay Avg (spv)	1.2	1.1	0.8	1.1	32.2	30.1	1.8	2.0	2.3	2.1
Delay Max (spv)	3.4	4.8	3.9	4.0	45.3	59.7	4.7	4.5	19.0	7.5
Q Avg (v)	0.2	0.2	0.1	0.2	1.9	2.5	0.2	0.2	0.3	0.1
Q Max (v)	1.5	1.6	1.6	2.0	3.4	6.6	0.5	0.9	1.6	0.5
Int 3										
Volume (vph)	10.6	8.3	3.6	9.1	10.2	10.4	10.8	2.7	12.7	5.4
Delay Avg (spv)	1.9	2.0	0.8	0.8	109.3	135.0	3.3	3.2	6.1	3.8
Delay Max (spv)	4.9	7.6	7.3	4.5	214.6	185.2	18.7	23.8	26.3	18.5
Q Avg (v)	0.3	0.3	0.1	0.1	5.8	7.4	0.5	0.1	1.1	0.1
Q Max (v)	1.1	1.1	0.5	1.1	12.7	11.0	0.5	0.9	3.0	1.3
Int 4										
Volume (vph)	6.1	4.2	3.5	6.1	2.2	3.0	6.8	1.5	3.4	3.5
Delay Avg (spv)	1.4	1.5	0.6	0.5	61.7	8.8	2.4	1.0	1.6	1.7
Delay Max (spv)	3.0	3.1	1.5	1.8	127.8	24.6	7.0	4.8	3.3	2.5
Q Avg (v)	0.2	0.3	0.1	0.1	4.8	0.5	0.4	0.0	0.2	0.1
Q Max (v)	1.1	1.1	0.8	0.4	9.5	1.1	0.8	0.4	0.8	0.5
Int 5										
Volume (vph)	13.8	0.0	17.2	0.0	6.9	11.1	7.2	1.6	6.0	0.8
Delay Avg (spv)	1.6	0.0	4.1	0.0	17.3	20.2	1.7	1.6	3.3	1.4
Delay Max (spv)	6.9	0.0	8.3	0.0	20.6	65.1	2.8	11.9	10.8	5.2
Q Avg (v)	0.4	0.0	0.8	0.0	1.1	1.0	0.3	0.1	0.6	0.0
Q Max (v)	0.4	0.0	2.9	0.0	1.9	3.4	0.8	0.5	2.8	0.4
Int 6										
Volume (vph)	7.8	0.0	17.2	0.0	3.2	4.6	5.0	1.1	6.2	1.2
Delay Avg (spv)	0.6	0.0	2.6	0.0	10.6	12.0	1.5	0.8	1.7	1.5
Delay Max (spv)	4.3	0.0	8.1	0.0	35.3	33.1	6.7	3.2	4.0	5.9
Q Avg (v)	0.1	0.0	0.5	0.0	0.4	0.5	0.2	0.0	0.3	0.1
Q Max (v)	0.7	0.0	1.5	0.0	1.3	1.6	1.1	0.4	0.8	0.0
Int 7										
Volume (vph)	12.2	0.0	5.4	0.0	7.3	2.2	6.3	1.7	2.9	0.5
Delay Avg (spv)	1.0	0.0	1.7	0.0	9.7	6.8	1.4	1.0	0.7	0.5
Delay Max (spv)	3.0	0.0	3.0	0.0	33.0	24.9	7.4	5.2	8.3	6.2
Q Avg (v)	0.2	0.0	0.3	0.0	0.6	0.3	0.2	0.0	0.1	0.0
Q Max (v)	1.1	0.0	0.8	0.0	1.3	0.8	1.4	0.0	0.9	0.0

Network 2 - COP - Average Value Standard Deviations

Movement	EB T 1	EB T 2	WB T 5	WB T 6	EB L 9	WB L 10	NB T 11	NB R 12	SB T 13	SB R 14
Int 1										
Volume (vph)	1.9	3.8	4.3	5.5	2.5	4.4	3.5	4.5	3.0	3.0
Delay Avg (spv)	0.9	0.8	1.0	0.8	5.6	4.9	0.8	1.1	1.2	0.7
Delay Max (spv)	5.6	2.1	6.5	3.5	7.4	12.9	5.2	6.1	10.3	4.6
Q Avg (v)	0.1	0.1	0.1	0.1	0.3	0.4	0.0	0.1	0.1	0.0
Q Max (v)	0.8	0.5	0.9	0.5	0.4	0.8	0.0	0.8	0.5	0.0
Int 2										
Volume (vph)	10.8	6.4	7.1	9.4	8.2	4.2	3.2	4.5	5.5	1.5
Delay Avg (spv)	0.8	0.9	0.4	1.2	9.7	40.6	2.1	1.8	0.6	1.5
Delay Max (spv)	9.1	2.6	6.0	6.0	33.7	58.2	9.2	9.4	6.0	9.6
Q Avg (v)	0.1	0.1	0.1	0.2	0.9	2.8	0.3	0.2	0.1	0.1
Q Max (v)	1.7	0.8	0.9	1.2	3.5	4.2	1.1	1.3	2.1	0.0
Int 3										
Volume (vph)	7.8	13.2	7.1	10.5	3.8	3.7	10.8	9.1	3.7	7.1
Delay Avg (spv)	1.4	0.7	1.6	1.9	79.6	89.8	6.0	3.0	4.1	4.8
Delay Max (spv)	10.1	8.6	6.0	6.2	101.5	104.1	11.3	5.9	12.0	11.0
Q Avg (v)	0.3	0.2	0.3	0.3	4.8	3.8	1.2	0.0	0.7	0.1
Q Max (v)	0.8	1.3	2.8	3.3	6.7	3.0	2.6	1.2	2.7	2.2
Int 4										
Volume (vph)	7.7	10.7	4.8	6.6	4.9	3.0	7.5	2.1	1.9	1.5
Delay Avg (spv)	1.2	1.0	0.8	0.9	64.8	8.9	4.7	1.5	1.4	1.4
Delay Max (spv)	5.7	8.1	2.1	3.2	114.0	31.3	9.8	2.5	7.6	4.1
Q Avg (v)	0.2	0.1	0.1	0.2	5.7	0.5	0.7	0.1	0.2	0.1
Q Max (v)	1.5	0.9	1.5	1.2	11.0	1.5	1.6	0.4	0.5	0.0
Int 5										
Volume (vph)	10.4	0.0	12.1	0.0	7.0	5.2	7.6	1.9	7.8	1.1
Delay Avg (spv)	3.8	0.0	7.6	0.0	19.3	24.2	1.1	1.2	4.6	1.3
Delay Max (spv)	5.6	0.0	24.7	0.0	48.0	43.9	9.1	9.6	10.1	6.8
Q Avg (v)	0.8	0.0	1.3	0.0	0.9	1.7	0.1	0.0	0.8	0.0
Q Max (v)	1.5	0.0	3.9	0.0	2.7	3.7	1.2	0.4	0.8	0.4
Int 6										
Volume (vph)	8.1	0.0	10.7	0.0	3.6	4.6	4.0	1.5	5.6	1.5
Delay Avg (spv)	1.0	0.0	0.9	0.0	8.9	10.5	0.7	0.6	1.0	1.3
Delay Max (spv)	2.9	0.0	3.2	0.0	26.9	19.3	2.3	4.6	1.6	2.3
Q Avg (v)	0.2	0.0	0.2	0.0	0.3	0.3	0.1	0.0	0.2	0.0
Q Max (v)	0.8	0.0	1.7	0.0	0.5	1.3	0.5	0.0	0.8	0.0
Int 7										
Volume (vph)	7.0	0.0	6.8	0.0	5.6	2.2	6.8	1.9	3.1	1.1
Delay Avg (spv)	1.7	0.0	1.3	0.0	2.1	2.8	1.7	2.3	1.5	1.4
Delay Max (spv)	3.8	0.0	4.2	0.0	16.6	10.9	5.7	4.3	5.5	6.1
Q Avg (v)	0.2	0.0	0.2	0.0	0.2	0.1	0.2	0.1	0.2	0.0
Q Max (v)	1.1	0.0	0.5	0.0	1.1	0.5	0.9	0.4	0.7	0.0

Network 3 - Pre-Timed Control - Replication 1

Movement	EB T 1	EB T 2	WB T 5	WB T 6	EB L 9	WB L 10	NB T 11	NB R 12	SB T 13	SB R 14
Int 1										
Volume (vph)	227	225	229	248	102	111	150	109	165	98
Delay Avg (spv)	20.3	19.7	18.5	17.6	26.1	54.2	24.3	21.1	21.5	16.3
Delay Max (spv)	48.0	45.0	48.3	47.9	56.5	109.4	56.3	51.3	61.0	50.6
Q Avg (v)	3.1	3.0	2.8	2.9	1.7	3.8	2.4	1.5	2.3	1.1
Q Max (v)	10	10	8	8	5	11	8	7	9	5
Int 2										
Volume (vph)	224	236	230	243	102	97	173	102	172	116
Delay Avg (spv)	33.7	32.3	11.5	12.6	50.5	46.6	20.3	18.8	23.0	19.9
Delay Max (spv)	63.3	45.0	48.4	44.7	106.3	106.9	61.1	50.6	55.6	51.6
Q Avg (v)	5.0	5.1	1.8	2.0	3.4	3.0	2.2	1.2	2.5	1.5
Q Max (v)	12	13	7	7	11	10	8	5	8	6
Int 3										
Volume (vph)	238	233	232	243	93	99	161	93	161	102
Delay Avg (spv)	35.4	36.8	20.1	18.8	26.0	26.9	39.7	32.0	39.4	32.7
Delay Max (spv)	83.2	77.3	44.2	44.0	87.8	55.4	88.1	50.2	83.2	49.7
Q Avg (v)	5.4	5.5	3.0	3.0	1.6	1.7	4.3	2.0	4.3	2.3
Q Max (v)	14	14	9	9	9	5	11	7	10	9
Int 4										
Volume (vph)	231	226	230	241	93	94	171	104	166	108
Delay Avg (spv)	20.7	20.0	11.7	10.7	26.1	56.7	24.0	21.0	36.0	34.8
Delay Max (spv)	49.6	43.2	47.7	44.6	55.2	145.9	50.5	50.1	84.1	50.1
Q Avg (v)	3.2	3.0	1.8	1.7	1.6	3.6	2.6	1.4	3.9	2.4
Q Max (v)	10	9	7	6	4	13	7	5	10	9
Int 5										
Volume (vph)	232	239	246	248	99	77	163	100	168	91
Delay Avg (spv)	32.1	32.3	35.5	35.0	23.0	23.9	22.3	21.3	43.5	34.4
Delay Max (spv)	43.9	44.3	69.3	54.1	79.3	55.0	49.5	48.9	85.9	49.4
Q Avg (v)	4.8	5.0	5.7	5.6	1.5	1.2	2.5	1.4	5.0	2.2
Q Max (v)	12	11	13	12	8	7	8	5	15	8
Int 6										
Volume (vph)	225	239	228	231	91	99	172	103	165	104
Delay Avg (spv)	13.6	12.7	19.2	18.4	46.4	26.9	24.8	21.0	27.4	23.5
Delay Max (spv)	46.7	45.3	44.1	43.1	106.5	56.5	53.1	48.1	61.3	63.0
Q Avg (v)	2.0	2.0	2.9	2.8	3.0	1.8	2.8	1.4	2.9	1.6
Q Max (v)	6	7	8	8	9	5	8	5	9	7
Int 7										
Volume (vph)	235	228	207	218	95	102	176	84	167	109
Delay Avg (spv)	20.5	19.4	12.5	12.9	23.4	46.9	37.6	35.3	23.5	21.2
Delay Max (spv)	43.4	43.5	44.3	45.2	55.4	100.8	87.6	50.0	50.8	50.4
Q Avg (v)	3.2	2.9	1.7	1.9	1.5	3.4	4.2	1.9	2.5	1.5
Q Max (v)	9	9	6	6	4	10	11	7	8	5
Int 8										
Volume (vph)	233	241	220	232	89	116	174	95	159	97
Delay Avg (spv)	33.002	31.4	30.396	29.012	42.76	34.057	45.356	30.876	23.589	21.057
Delay Max (spv)	53.84	57.01	44.07	44.47	133.48	97.575	95.117	49.215	50.165	50.465
Q Avg (v)	4.9424	4.889	4.2419	4.283	2.3831	2.5594	5.3839	1.9386	2.5292	1.382
Q Max (v)	13	13	11	10	11	10	14	7	7	5
Int 9										
Volume (vph)	225	226	229	220	97	97	165	92	173	103
Delay Avg (spv)	11.868	12.381	18.702	19.382	42.906	26.347	26.289	21.545	23.724	21.513
Delay Max (spv)	46.103	47.129	44.46	43.465	94.97	56.165	58.24	52.706	49.555	50.105
Q Avg (v)	1.7704	1.8462	2.8246	2.8388	2.9045	1.7213	2.8493	1.2961	2.6569	1.4103
Q Max (v)	7	6	9	9	10	7	8	6	8	5

Network 3 - Constrained COP - Replication 1

Movement	EB T	EB T	WB T	WB T	EB L	WB L	NB T	NB R	SB T	SB R
	1	2	5	6	9	10	11	12	13	14
Int 1										
Volume (vph)	231	228	219	244	100	93	168	94	152	108
Delay Avg (spv)	13.0	11.0	9.4	9.6	68.1	116.2	27.6	16.6	19.4	15.2
Delay Max (spv)	39.9	39.2	40.0	36.5	159.4	296.0	113.7	56.8	62.9	60.8
Q Avg (v)	1.9	1.6	1.3	1.5	4.5	7.3	3.0	1.0	1.9	1.0
Q Max (v)	9	7	8	8	11	15	12	7	10	5
Int 2										
Volume (vph)	214	231	232	244	86	91	171	93	164	88
Delay Avg (spv)	14.4	12.8	9.4	8.6	98.8	93.6	32.9	15.0	27.9	17.1
Delay Max (spv)	51.0	45.9	47.3	48.8	249.7	219.1	86.7	55.6	71.8	59.8
Q Avg (v)	2.0	1.9	1.4	1.3	6.7	5.7	3.9	0.9	2.9	1.0
Q Max (v)	7	8	6	7	17	14	12	6	12	6
Int 3										
Volume (vph)	210	238	237	233	95	100	145	93	164	103
Delay Avg (spv)	10.2	10.0	10.7	10.6	91.1	81.5	18.5	12.2	20.9	15.3
Delay Max (spv)	41.1	40.2	42.9	42.1	173.5	189.4	57.3	57.4	88.9	73.9
Q Avg (v)	1.4	1.5	1.6	1.6	5.8	5.4	1.8	0.7	2.3	1.0
Q Max (v)	9	9	7	7	15	13	7	5	10	7
Int 4										
Volume (vph)	233	226	226	223	93	96	170	102	151	91
Delay Avg (spv)	13.1	13.5	12.1	12.4	77.1	117.4	24.4	15.8	26.4	17.3
Delay Max (spv)	47.0	44.2	48.5	45.1	196.5	215.8	88.8	84.2	91.5	84.5
Q Avg (v)	2.0	2.0	1.8	1.8	4.8	7.7	2.6	1.0	2.6	1.0
Q Max (v)	8	9	8	11	13	16	9	7	14	7
Int 5										
Volume (vph)	214	228	228	225	110	109	161	99	154	99
Delay Avg (spv)	9.4	9.3	12.3	12.8	220.8	85.3	22.1	16.0	29.9	16.5
Delay Max (spv)	45.0	45.7	46.6	46.3	306.8	202.4	64.8	68.1	88.9	51.8
Q Avg (v)	1.3	1.4	1.8	1.9	16.2	6.1	2.4	1.1	3.0	1.0
Q Max (v)	7	7	8	9	27	16	8	5	10	6
Int 6										
Volume (vph)	221	227	227	233	74	97	168	100	168	96
Delay Avg (spv)	12.4	13.5	14.1	12.9	133.6	80.8	18.2	13.7	21.3	16.2
Delay Max (spv)	47.6	48.6	41.0	43.7	347.2	206.4	66.0	61.0	60.6	69.4
Q Avg (v)	1.8	2.0	2.1	2.0	8.6	5.4	2.0	0.9	2.4	1.0
Q Max (v)	11	10	8	8	31	16	8	5	8	6
Int 7										
Volume (vph)	234	228	216	229	96	89	156	118	167	109
Delay Avg (spv)	11.5	10.4	10.3	9.6	81.8	89.4	21.9	17.4	17.8	13.9
Delay Max (spv)	39.9	41.4	46.1	49.6	190.9	160.0	61.3	55.9	63.0	53.4
Q Avg (v)	1.8	1.5	1.5	1.4	5.2	5.5	2.1	1.3	1.8	0.9
Q Max (v)	7	7	9	7	13	13	9	6	8	5
Int 8										
Volume (vph)	235	248	229	220	95	88	143	110	163	98
Delay Avg (spv)	13.424	13.652	10.175	10.098	119.16	189.46	25.055	15.592	20.842	13.689
Delay Max (spv)	48.535	51.775	38.28	39.905	307.35	364.68	87.024	48.5	61.165	47.805
Q Avg (v)	1.9976	2.1499	1.4172	1.3472	7.6121	13.378	2.3858	1.0737	2.2021	0.8618
Q Max (v)	10	8	10	8	16	31	10	6	8	4
Int 9										
Volume (vph)	225	244	229	225	100	101	148	89	169	100
Delay Avg (spv)	14.841	15.277	12.109	12.833	145.62	74.449	21.363	20.69	25.603	17.639
Delay Max (spv)	74.061	72.86	41.91	42.91	363.19	197.02	72.345	64.155	78.105	56.805
Q Avg (v)	2.2129	2.4263	1.8013	1.8704	13.587	4.8741	2.0889	1.2031	2.8863	1.1432
Q Max (v)	10	11	8	8	33	13	11	6	10	5

Network 3 - COP - Replication 1

Movement	EB T 1	EB T 2	WB T 5	WB T 6	EB L 9	WB L 10	NB T 11	NB R 12	SB T 13	SB R 14
Int 1										
Volume (vph)	229	226	225	238	97	92	159	107	158	103
Delay Avg (spv)	11.4	12.0	8.6	9.5	51.9	106.2	21.3	15.5	17.0	13.8
Delay Max (spv)	36.0	35.2	41.3	41.7	143.1	244.7	57.3	50.2	58.5	47.9
Q Avg (v)	1.7	1.8	1.2	1.4	3.4	7.2	2.2	1.1	1.8	0.9
Q Max (v)	8	8	8	7	10	16	8	7	8	6
Int 2										
Volume (vph)	224	230	221	237	111	109	167	88	169	95
Delay Avg (spv)	11.1	11.1	11.9	11.3	74.1	55.7	21.5	14.7	22.3	12.9
Delay Max (spv)	49.5	44.5	49.1	40.3	159.5	129.6	66.0	57.1	84.2	50.3
Q Avg (v)	1.6	1.7	1.8	1.8	5.0	3.9	2.4	0.8	2.5	0.8
Q Max (v)	9	8	9	9	12	10	9	5	12	7
Int 3										
Volume (vph)	213	241	238	233	100	101	162	94	156	112
Delay Avg (spv)	9.1	9.2	12.3	12.9	89.5	56.1	19.6	11.7	17.1	12.3
Delay Max (spv)	39.3	51.6	36.1	38.5	160.1	127.1	61.6	51.0	53.4	39.5
Q Avg (v)	1.2	1.4	1.9	1.9	5.8	3.6	2.0	0.7	1.7	0.9
Q Max (v)	7	7	7	8	13	9	9	4	8	5
Int 4										
Volume (vph)	233	225	215	221	95	99	172	100	145	100
Delay Avg (spv)	13.3	11.8	7.4	9.7	45.2	119.7	20.4	12.2	15.2	10.6
Delay Max (spv)	39.1	41.9	39.6	53.7	112.5	237.6	54.4	41.5	44.6	56.1
Q Avg (v)	2.0	1.7	1.0	1.4	2.8	8.1	2.3	0.8	1.5	0.7
Q Max (v)	8	8	8	10	9	17	7	4	7	6
Int 5										
Volume (vph)	247	236	212	235	87	99	165	102	166	108
Delay Avg (spv)	13.0	14.1	9.7	9.9	58.6	115.2	18.9	16.1	30.5	15.8
Delay Max (spv)	57.3	54.1	49.3	56.7	171.5	247.0	58.4	60.4	101.5	59.7
Q Avg (v)	2.1	2.1	1.3	1.4	3.2	8.0	2.0	1.0	3.4	1.1
Q Max (v)	11	10	8	8	11	18	8	4	15	9
Int 6										
Volume (vph)	236	243	232	228	93	102	169	99	172	89
Delay Avg (spv)	9.4	8.4	12.2	12.2	70.1	43.6	17.5	12.9	26.2	16.1
Delay Max (spv)	40.0	33.4	38.5	41.0	170.2	119.0	57.5	53.4	66.7	57.9
Q Avg (v)	1.4	1.3	1.8	1.8	4.4	2.9	1.9	0.8	2.9	0.9
Q Max (v)	9	8	6	7	14	9	8	4	10	5
Int 7										
Volume (vph)	236	227	210	237	97	96	161	96	164	108
Delay Avg (spv)	13.3	13.4	9.3	8.0	40.9	73.1	21.5	12.6	16.4	11.7
Delay Max (spv)	48.8	46.1	41.6	47.1	91.0	154.1	74.3	40.2	45.0	39.5
Q Avg (v)	2.0	2.0	1.2	1.2	2.6	5.0	2.3	0.8	1.7	0.8
Q Max (v)	8	9	8	8	7	14	9	6	6	4
Int 8										
Volume (vph)	225	225	223	223	111	110	153	119	163	100
Delay Avg (spv)	9.3256	10.817	11.648	10.599	97.662	61.892	22.356	12.263	17.039	13.041
Delay Max (spv)	42.37	50.645	53.1	60.325	211.4	143.56	76.505	60.75	59.655	48.105
Q Avg (v)	1.3526	1.5801	1.678	1.5351	7.1124	4.2661	2.2899	0.9243	1.8041	0.8242
Q Max (v)	7	7	9	9	18	10	9	6	8	4
Int 9										
Volume (vph)	226	233	231	223	98	101	161	97	170	101
Delay Avg (spv)	10.415	9.0095	11.749	11.649	78.493	58.595	26.03	12.396	18.263	14.795
Delay Max (spv)	38.625	37.47	38.91	36.105	197.78	141.17	79.91	45.395	44.525	42.165
Q Avg (v)	1.5211	1.359	1.7552	1.6784	5.8917	3.8424	2.7337	0.8187	2.0291	0.9468
Q Max (v)	9	8	7	7	17	10	10	6	7	4

Network 3 - Pre-Timed Control - Replication 2

Movement	EB T	EB T	WB T	WB T	EB L	WB L	NB T	NB R	SB T	SB R
	1	2	5	6	9	10	11	12	13	14
Int 1										
Volume (vph)	220	229	233	236	105	95	162	99	161	106
Delay Avg (spv)	20.5	19.5	17.9	17.8	27.2	48.5	23.9	18.5	16.0	19.6
Delay Max (spv)	44.1	43.5	50.0	48.8	56.5	104.1	56.9	50.7	51.7	50.7
Q Avg (v)	3.1	3.0	2.8	2.8	1.8	2.9	2.5	1.2	1.7	1.4
Q Max (v)	10	9	8	8	5	10	8	7	5	5
Int 2										
Volume (vph)	223	247	234	247	86	100	171	92	179	97
Delay Avg (spv)	33.4	31.0	15.5	15.4	25.7	54.2	25.2	17.7	26.0	17.8
Delay Max (spv)	49.6	52.0	50.0	48.5	78.0	152.5	59.2	50.8	62.3	50.7
Q Avg (v)	5.0	5.1	2.4	2.5	1.5	3.7	2.8	1.0	3.1	1.1
Q Max (v)	11	12	7	8	7	13	10	4	8	6
Int 3										
Volume (vph)	230	247	240	241	93	99	163	105	146	112
Delay Avg (spv)	38.0	38.2	20.3	20.0	21.4	25.5	39.5	35.0	35.8	37.0
Delay Max (spv)	88.9	84.4	43.7	43.8	64.6	56.2	89.0	50.4	55.0	50.0
Q Avg (v)	5.7	6.2	3.2	3.1	1.2	1.6	4.4	2.6	3.6	2.8
Q Max (v)	14	14	9	9	6	5	12	8	9	8
Int 4										
Volume (vph)	225	224	228	235	99	87	165	100	164	97
Delay Avg (spv)	18.9	20.1	11.7	14.3	27.1	39.7	25.0	20.9	41.7	31.8
Delay Max (spv)	42.9	45.5	44.9	45.3	56.5	92.4	50.4	49.8	99.8	48.0
Q Avg (v)	2.8	3.0	1.8	2.2	1.8	2.4	2.7	1.4	4.4	2.0
Q Max (v)	9	9	6	7	5	7	8	4	12	7
Int 5										
Volume (vph)	233	231	232	239	95	112	163	98	166	97
Delay Avg (spv)	31.9	33.8	31.5	30.4	34.8	30.2	24.0	21.5	39.0	35.3
Delay Max (spv)	58.9	55.0	45.8	47.6	88.2	81.7	53.3	50.2	89.6	48.7
Q Avg (v)	4.8	5.1	4.7	4.6	2.1	2.1	2.6	1.4	4.5	2.4
Q Max (v)	11	13	12	11	10	8	8	4	12	8
Int 6										
Volume (vph)	227	232	226	228	103	99	174	97	143	108
Delay Avg (spv)	13.5	13.2	18.6	18.3	49.4	27.3	24.3	18.9	22.4	18.2
Delay Max (spv)	51.2	49.6	43.1	43.3	119.4	58.5	51.5	49.0	51.6	51.9
Q Avg (v)	2.0	2.0	2.8	2.8	3.4	1.8	2.7	1.2	2.0	1.3
Q Max (v)	9	9	9	8	11	6	8	4	6	5
Int 7										
Volume (vph)	233	233	220	235	96	97	179	91	173	101
Delay Avg (spv)	18.9	20.0	14.2	14.5	26.6	54.3	38.3	29.6	24.3	19.6
Delay Max (spv)	43.1	54.1	48.1	48.3	56.4	105.5	82.7	50.6	52.2	49.5
Q Avg (v)	2.9	3.1	2.1	2.3	1.7	3.5	4.4	1.7	2.7	1.2
Q Max (v)	9	11	9	8	5	10	10	6	8	5
Int 8										
Volume (vph)	223	234	223	242	105	101	155	94	166	97
Delay Avg (spv)	31.238	30.937	30.786	28.511	39.52	23.675	40.402	30.998	23.175	20.65
Delay Max (spv)	53.065	53.555	44.415	44.229	93.73	71.728	83.142	49.947	48.91	47.955
Q Avg (v)	4.5736	4.7358	4.3753	4.3865	2.6208	1.5622	4.3572	1.9781	2.5842	1.3404
Q Max (v)	12	12	10	11	11	8	12	7	8	4
Int 9										
Volume (vph)	209	217	224	227	109	97	168	86	165	100
Delay Avg (spv)	9.8393	10.099	19.475	19.403	54.536	29.146	23.321	21.568	23.151	20.462
Delay Max (spv)	44.59	47.924	43.755	44.4	105.66	56.165	64.57	50.87	55.91	49.91
Q Avg (v)	1.3771	1.4902	2.8742	2.9208	4.1466	1.8921	2.5789	1.2017	2.4259	1.2899
Q Max (v)	6	7	9	9	11	5	10	7	7	4

Network 3 - Constrained COP - Replication 2

Movement	EB T 1	EB T 2	WB T 5	WB T 6	EB L 9	WB L 10	NB T 11	NB R 12	SB T 13	SB R 14
Int 1										
Volume (vph)	223	233	235	226	105	103	169	92	152	97
Delay Avg (spv)	12.2	12.2	10.1	10.8	77.0	108.4	37.5	17.4	17.3	16.7
Delay Max (spv)	43.9	47.6	49.1	44.4	200.4	222.0	110.8	63.3	54.6	69.1
Q Avg (v)	1.8	1.9	1.6	1.6	5.2	7.4	4.3	1.1	1.7	1.1
Q Max (v)	8	8	7	8	13	15	13	8	7	7
Int 2										
Volume (vph)	235	233	237	240	85	84	170	95	156	97
Delay Avg (spv)	13.0	12.5	9.4	9.8	126.4	141.5	19.5	15.3	21.2	13.9
Delay Max (spv)	47.3	49.1	56.2	52.5	255.6	245.6	55.5	54.2	76.0	81.0
Q Avg (v)	1.9	1.9	1.4	1.5	7.9	10.5	2.1	0.9	2.2	0.9
Q Max (v)	10	10	8	8	16	22	9	6	9	6
Int 3										
Volume (vph)	234	247	235	242	78	94	177	88	155	105
Delay Avg (spv)	9.9	10.5	10.2	10.6	56.9	100.1	25.9	12.4	23.0	16.1
Delay Max (spv)	35.7	57.0	34.1	37.0	157.2	179.5	71.1	68.2	85.9	78.7
Q Avg (v)	1.5	1.7	1.5	1.6	2.9	6.5	3.1	0.7	2.4	1.0
Q Max (v)	10	9	8	8	9	12	14	5	12	6
Int 4										
Volume (vph)	227	226	219	223	100	85	169	100	156	98
Delay Avg (spv)	11.2	11.7	10.2	10.4	87.7	58.2	30.4	15.0	22.0	16.7
Delay Max (spv)	41.4	42.0	40.6	34.5	182.2	165.7	80.2	59.2	86.1	56.9
Q Avg (v)	1.6	1.7	1.4	1.5	5.8	3.3	3.3	0.9	2.2	1.0
Q Max (v)	8	8	9	8	12	11	10	5	9	8
Int 5										
Volume (vph)	218	232	217	214	101	102	165	100	147	91
Delay Avg (spv)	10.3	10.1	10.0	10.9	64.8	204.7	23.1	13.7	23.9	12.9
Delay Max (spv)	52.7	51.9	48.6	41.9	162.0	320.6	69.8	50.7	80.0	46.1
Q Avg (v)	1.4	1.5	1.4	1.5	4.0	16.4	2.5	0.9	2.3	0.8
Q Max (v)	8	7	8	9	12	29	8	4	9	5
Int 6										
Volume (vph)	224	235	232	227	98	101	168	95	153	101
Delay Avg (spv)	13.1	14.0	12.4	12.1	74.7	113.2	25.3	15.9	19.2	15.3
Delay Max (spv)	47.7	46.1	49.8	47.0	220.4	199.2	85.5	52.4	67.9	57.8
Q Avg (v)	1.9	2.1	1.9	1.8	4.8	7.6	2.9	1.0	1.8	1.0
Q Max (v)	9	10	8	7	14	15	12	4	8	7
Int 7										
Volume (vph)	232	232	231	239	96	82	176	96	174	100
Delay Avg (spv)	12.2	10.6	11.4	12.0	104.2	64.4	40.3	16.4	23.5	15.5
Delay Max (spv)	43.4	41.0	53.3	54.6	238.1	142.4	112.2	51.9	71.7	70.8
Q Avg (v)	1.8	1.6	1.7	1.9	6.8	3.6	4.6	1.0	2.6	1.0
Q Max (v)	7	10	9	8	16	10	16	8	9	5
Int 8										
Volume (vph)	223	221	223	230	100	91	160	104	170	99
Delay Avg (spv)	10.02	9.043	11.8	12.103	94.858	70.659	27.212	19.231	21.823	16.92
Delay Max (spv)	48.085	45.61	52.405	47.855	226.35	206.7	86.581	85.222	75.91	71.015
Q Avg (v)	1.4156	1.2879	1.6834	1.7837	7.5917	4.3167	2.8175	1.2649	2.3918	1.0704
Q Max (v)	9	8	9	9	23	13	10	7	11	6
Int 9										
Volume (vph)	225	226	228	230	85	100	164	90	163	99
Delay Avg (spv)	10.615	12.058	13.252	13.214	64.282	77.575	26.093	16.318	20.748	18.387
Delay Max (spv)	43.9	47.065	50.61	48.015	153.61	199.96	84.315	67.615	63.465	66.105
Q Avg (v)	1.6081	1.8476	1.9518	1.9711	3.6834	5.0889	2.8274	0.9651	2.164	1.1562
Q Max (v)	9	8	8	9	14	14	11	7	8	6

Network 3 - COP - Replication 2

Movement	EB T 1	EB T 2	WB T 5	WB T 6	EB L 9	WB L 10	NB T 11	NB R 12	SB T 13	SB R 14
Int 1										
Volume (vph)	221	231	215	226	101	101	170	98	143	114
Delay Avg (spv)	12.0	14.6	7.8	9.3	50.8	148.4	21.6	14.0	18.9	14.9
Delay Max (spv)	44.1	49.9	40.1	43.1	120.9	298.8	71.4	45.4	61.7	53.1
Q Avg (v)	1.8	2.2	1.1	1.4	3.4	11.0	2.4	0.9	1.8	1.1
Q Max (v)	8	12	7	8	9	20	11	6	11	6
Int 2										
Volume (vph)	227	244	231	240	94	99	164	95	163	99
Delay Avg (spv)	9.1	9.1	11.4	12.0	57.3	130.5	25.7	14.5	19.4	15.0
Delay Max (spv)	53.2	49.3	60.3	60.3	135.5	284.9	78.2	51.9	56.1	64.6
Q Avg (v)	1.3	1.4	1.7	1.8	3.6	8.4	2.8	0.9	2.1	0.9
Q Max (v)	8	7	11	11	10	20	12	6	11	7
Int 3										
Volume (vph)	225	244	233	242	88	100	159	101	166	86
Delay Avg (spv)	10.4	9.5	13.5	13.4	74.2	58.5	23.0	13.3	20.5	13.9
Delay Max (spv)	47.2	50.2	42.3	44.3	176.1	142.0	89.9	41.7	62.7	48.4
Q Avg (v)	1.5	1.5	2.1	2.1	4.4	3.8	2.4	0.9	2.4	0.8
Q Max (v)	8	8	7	9	13	9	11	6	8	5
Int 4										
Volume (vph)	222	223	217	215	105	105	165	99	148	89
Delay Avg (spv)	12.4	13.6	8.9	10.9	48.1	86.6	22.1	15.0	25.8	14.0
Delay Max (spv)	36.2	38.5	40.7	42.6	142.8	194.3	53.6	48.9	74.4	47.8
Q Avg (v)	1.8	2.0	1.2	1.5	3.2	6.4	2.3	0.9	2.5	0.8
Q Max (v)	7	7	7	9	9	16	7	5	11	6
Int 5										
Volume (vph)	227	223	229	228	91	95	168	102	169	93
Delay Avg (spv)	8.3	8.7	10.1	10.9	66.2	80.0	19.8	14.9	21.8	12.0
Delay Max (spv)	35.7	39.1	36.2	40.1	134.4	182.3	54.5	45.5	93.4	41.2
Q Avg (v)	1.2	1.3	1.5	1.6	3.9	4.9	2.1	0.9	2.4	0.7
Q Max (v)	7	8	7	8	10	14	7	4	10	5
Int 6										
Volume (vph)	222	233	228	230	96	105	171	96	169	96
Delay Avg (spv)	8.8	9.5	12.6	11.6	102.0	93.2	20.8	13.6	21.5	16.8
Delay Max (spv)	36.4	42.0	36.0	40.9	277.7	181.0	66.5	51.8	66.8	64.2
Q Avg (v)	1.2	1.4	1.8	1.7	7.9	6.3	2.3	0.8	2.4	1.0
Q Max (v)	8	8	8	7	24	14	9	4	11	8
Int 7										
Volume (vph)	235	234	224	228	102	94	169	90	165	96
Delay Avg (spv)	12.8	11.5	7.1	8.0	68.0	155.9	31.6	12.4	18.8	13.4
Delay Max (spv)	41.2	43.9	45.6	40.3	142.2	275.8	70.5	51.0	59.4	50.0
Q Avg (v)	1.9	1.7	1.0	1.2	4.4	9.8	3.7	0.7	2.1	0.8
Q Max (v)	8	8	6	6	9	18	12	5	8	4
Int 8										
Volume (vph)	220	246	219	236	100	95	172	84	170	99
Delay Avg (spv)	9.0791	9.7917	7.8033	7.9278	71.504	67.041	31.48	12.459	19.859	13.939
Delay Max (spv)	34.555	55.13	36.802	44.54	167.82	187.12	95.29	52.74	51.165	45.015
Q Avg (v)	1.25	1.5257	1.0828	1.1865	4.7057	4.9649	3.6087	0.6745	2.185	0.8726
Q Max (v)	6	8	7	7	16	15	11	5	7	4
Int 9										
Volume (vph)	212	239	223	235	95	98	153	87	158	96
Delay Avg (spv)	8.8854	8.8052	12.407	11.185	78.578	42.459	28.748	14.901	19.117	13.023
Delay Max (spv)	36.61	42.3	41.4	36.655	151.7	121.91	82.888	61.675	52.66	46.4
Q Avg (v)	1.203	1.3562	1.7738	1.6851	4.8875	2.7852	2.9409	0.8217	1.9703	0.795
Q Max (v)	8	9	7	7	13	9	13	7	7	4

Network 3 - Pre-Timed Control - Replication 3

Movement	EB T 1	EB T 2	WB T 5	WB T 6	EB L 9	WB L 10	NB T 11	NB R 12	SB T 13	SB R 14
Int 1										
Volume (vph)	221	234	246	232	104	90	146	118	190	76
Delay Avg (spv)	20.3	21.0	15.8	15.8	26.3	38.1	23.9	22.6	24.2	14.6
Delay Max (spv)	43.5	46.5	45.4	48.5	56.0	55.4	51.1	51.8	58.6	49.7
Q Avg (v)	3.0	3.3	2.5	2.4	1.7	2.2	2.4	1.7	3.1	0.7
Q Max (v)	9	10	8	7	5	8	9	6	9	4
Int 2										
Volume (vph)	243	240	243	221	98	92	177	96	175	104
Delay Avg (spv)	30.9	33.0	11.1	12.4	31.8	40.1	22.4	16.9	22.0	19.4
Delay Max (spv)	75.3	63.3	51.1	49.2	87.8	93.2	63.5	50.7	53.6	51.8
Q Avg (v)	5.0	5.3	1.8	1.8	2.1	2.6	2.5	1.0	2.4	1.3
Q Max (v)	14	12	6	7	9	8	9	4	6	6
Int 3										
Volume (vph)	252	241	238	238	93	99	159	97	171	93
Delay Avg (spv)	35.6	36.9	19.6	19.4	27.7	25.1	35.8	30.7	40.4	33.3
Delay Max (spv)	82.0	76.1	43.5	43.5	79.1	56.5	63.3	49.0	90.8	49.8
Q Avg (v)	5.8	5.8	3.0	3.0	1.6	1.6	3.8	2.0	4.8	2.1
Q Max (v)	14	14	9	9	8	5	10	8	15	6
Int 4										
Volume (vph)	236	230	223	219	101	91	173	102	174	105
Delay Avg (spv)	20.5	19.6	11.5	9.8	28.2	45.1	24.8	21.4	40.0	34.5
Delay Max (spv)	44.0	43.0	48.6	45.9	56.4	96.3	52.2	49.5	91.1	50.1
Q Avg (v)	3.1	3.0	1.7	1.4	1.9	2.7	2.8	1.4	4.5	2.3
Q Max (v)	9	9	6	7	5	9	8	5	11	8
Int 5										
Volume (vph)	239	235	228	231	104	101	158	96	161	97
Delay Avg (spv)	31.8	31.9	30.8	30.8	25.6	30.8	24.7	21.6	38.8	32.7
Delay Max (spv)	44.5	45.1	45.3	45.8	74.7	85.4	56.0	50.4	84.8	49.9
Q Avg (v)	4.9	4.8	4.6	4.6	1.7	2.0	2.6	1.4	4.3	2.2
Q Max (v)	12	12	13	12	7	9	8	5	10	7
Int 6										
Volume (vph)	232	242	230	228	92	98	170	103	181	94
Delay Avg (spv)	13.6	12.9	18.2	18.8	43.8	26.6	22.5	21.6	37.9	20.0
Delay Max (spv)	47.3	44.8	43.2	43.8	94.0	56.0	50.7	49.5	108.6	51.1
Q Avg (v)	2.1	2.1	2.8	2.8	2.7	1.8	2.5	1.4	4.5	1.2
Q Max (v)	6	6	10	9	8	5	7	5	12	5
Int 7										
Volume (vph)	235	222	217	233	103	104	152	107	169	108
Delay Avg (spv)	20.4	18.3	14.8	14.1	26.3	49.5	34.5	32.6	24.0	20.6
Delay Max (spv)	45.0	43.4	48.5	45.5	55.8	99.4	87.0	50.0	50.5	49.5
Q Avg (v)	3.2	2.7	2.1	2.2	1.8	3.4	3.3	2.3	2.6	1.4
Q Max (v)	10	9	7	6	6	10	10	7	7	5
Int 8										
Volume (vph)	220	254	235	234	92	109	173	94	160	97
Delay Avg (spv)	35.472	31.669	30.808	32.164	27.127	29.095	41.031	35.539	22.932	21.386
Delay Max (spv)	88.11	83.2	50.33	48.21	70.455	82.302	85.622	48.77	50.165	50.465
Q Avg (v)	5.1165	5.2366	4.6385	4.8286	1.578	2.0698	4.7739	2.3266	2.48	1.3988
Q Max (v)	15	15	12	13	7	10	12	8	7	4
Int 9										
Volume (vph)	216	244	230	234	102	100	160	97	174	103
Delay Avg (spv)	12.077	13.591	20.414	20.321	57.003	27.195	25.593	18.645	23.939	21.297
Delay Max (spv)	45.525	47.57	43.655	43.4	120.79	59.525	52.04	49.809	50.105	50.465
Q Avg (v)	1.7168	2.1568	3.0991	3.1289	3.8536	1.7963	2.6775	1.1782	2.675	1.3903
Q Max (v)	7	8	9	10	12	5	9	6	8	5

Network 3 - Constrained COP - Replication 3

Movement	EB T	EB T	WB T	WB T	EB L	WB L	NB T	NB R	SB T	SB R
	1	2	5	6	9	10	11	12	13	14
Int 1										
Volume (vph)	220	234	226	216	105	100	160	98	158	95
Delay Avg (spv)	12.3	12.4	9.4	9.6	77.2	65.1	24.1	17.2	29.1	16.3
Delay Max (spv)	38.4	42.4	45.3	44.7	189.5	171.9	69.3	55.2	95.0	65.6
Q Avg (v)	1.8	1.9	1.4	1.3	5.2	4.2	2.5	1.1	3.1	1.0
Q Max (v)	8	9	7	8	14	10	13	7	12	5
Int 2										
Volume (vph)	229	233	228	223	95	103	165	91	176	93
Delay Avg (spv)	13.3	13.1	9.4	9.2	80.7	213.4	25.7	17.7	23.5	16.7
Delay Max (spv)	56.8	45.5	42.8	45.1	158.7	339.9	93.2	67.1	70.7	55.4
Q Avg (v)	2.0	2.0	1.4	1.3	5.3	15.9	2.7	1.0	2.7	1.0
Q Max (v)	11	11	8	7	14	28	9	7	12	7
Int 3										
Volume (vph)	214	222	230	231	99	99	176	84	163	102
Delay Avg (spv)	10.1	12.3	10.7	10.7	175.2	61.4	30.2	14.2	29.9	14.3
Delay Max (spv)	42.3	44.8	41.6	40.8	333.2	128.8	82.9	64.5	83.8	53.7
Q Avg (v)	1.4	1.8	1.6	1.6	12.4	4.0	3.5	0.7	3.3	0.9
Q Max (v)	7	8	7	7	22	9	14	7	13	6
Int 4										
Volume (vph)	233	230	228	229	94	90	167	100	154	98
Delay Avg (spv)	11.9	11.5	9.0	10.7	140.7	75.1	21.2	13.5	30.6	13.8
Delay Max (spv)	39.0	41.5	42.7	44.0	252.1	163.0	56.2	54.5	92.6	45.1
Q Avg (v)	1.8	1.7	1.3	1.6	9.5	4.9	2.3	0.8	3.1	0.9
Q Max (v)	7	7	8	7	20	14	8	5	12	7
Int 5										
Volume (vph)	233	239	224	240	98	93	160	94	174	94
Delay Avg (spv)	9.9	9.8	11.0	10.8	100.7	77.8	21.1	13.6	25.6	12.6
Delay Max (spv)	37.2	45.7	47.2	38.0	229.9	238.9	65.9	52.7	85.4	52.3
Q Avg (v)	1.5	1.5	1.6	1.7	6.6	5.1	2.3	0.9	3.1	0.9
Q Max (v)	7	7	9	9	18	18	8	4	11	6
Int 6										
Volume (vph)	240	232	230	229	97	107	165	102	149	110
Delay Avg (spv)	11.2	10.9	12.6	14.1	58.1	128.6	21.2	18.0	24.5	18.3
Delay Max (spv)	45.2	45.5	40.8	48.8	139.7	291.0	71.0	73.8	83.5	85.0
Q Avg (v)	1.7	1.6	1.9	2.1	3.8	8.6	2.3	1.2	2.4	1.3
Q Max (v)	8	9	8	8	12	22	8	7	9	6
Int 7										
Volume (vph)	235	224	221	231	104	93	150	106	168	104
Delay Avg (spv)	11.7	10.6	8.6	9.3	94.3	87.7	38.3	20.3	23.1	16.4
Delay Max (spv)	47.5	44.2	38.9	39.2	213.0	163.8	129.7	76.8	87.1	69.1
Q Avg (v)	1.8	1.5	1.2	1.4	6.6	5.9	4.1	1.4	2.5	1.1
Q Max (v)	9	7	9	9	15	14	15	8	10	5
Int 8										
Volume (vph)	222	226	222	236	98	100	165	100	171	99
Delay Avg (spv)	11.623	11.148	9.5474	9.6049	148.47	93.981	24.387	15.545	19.785	13.075
Delay Max (spv)	39.12	37.214	37.29	38.91	221.17	196.47	63.98	50.711	58.415	47.805
Q Avg (v)	1.6701	1.6353	1.3672	1.4598	10.517	6.5704	2.6391	0.9532	2.063	0.8195
Q Max (v)	8	8	9	7	22	15	10	7	7	4
Int 9										
Volume (vph)	214	222	233	238	96	96	168	99	173	101
Delay Avg (spv)	9.3698	9.6227	11.714	11.33	161.34	99.624	25.583	18.347	24.205	14.448
Delay Max (spv)	42.575	44.505	40.66	40.465	272.26	233.53	75.369	63.38	69.91	63.105
Q Avg (v)	1.3007	1.4015	1.7572	1.7345	11.941	6.6175	2.8325	1.1755	2.7015	0.9325
Q Max (v)	9	8	8	8	22	16	10	10	9	6

Network 3 - COP - Replication 3

Movement	EB T 1	EB T 2	WB T 5	WB T 6	EB L 9	WB L 10	NB T 11	NB R 12	SB T 13	SB R 14
Int 1										
Volume (vph)	222	236	225	229	102	86	165	104	182	86
Delay Avg (spv)	11.6	13.4	11.0	11.0	59.8	62.8	26.5	18.8	26.7	13.7
Delay Max (spv)	45.1	47.4	52.7	48.0	206.0	174.5	76.4	69.4	81.6	48.6
Q Avg (v)	1.7	2.1	1.7	1.7	4.0	4.0	2.8	1.3	3.2	0.8
Q Max (v)	8	9	10	9	16	13	12	6	10	5
Int 2										
Volume (vph)	227	235	236	226	101	103	176	93	185	90
Delay Avg (spv)	10.2	11.2	12.8	12.3	69.3	103.3	21.3	11.8	24.0	12.3
Delay Max (spv)	48.9	48.3	42.1	51.8	209.9	232.6	72.8	44.6	92.4	56.8
Q Avg (v)	1.5	1.7	2.0	1.9	4.8	7.5	2.4	0.7	2.9	0.7
Q Max (v)	6	8	10	9	13	16	10	5	10	6
Int 3										
Volume (vph)	220	233	240	225	93	95	163	103	150	121
Delay Avg (spv)	10.1	10.5	11.6	12.1	63.5	46.3	22.3	12.3	18.5	9.9
Delay Max (spv)	42.4	53.0	34.7	37.8	137.5	94.8	73.4	54.3	72.4	52.4
Q Avg (v)	1.4	1.6	1.8	1.8	4.3	3.0	2.4	0.8	1.7	0.8
Q Max (v)	8	8	7	6	10	7	11	6	9	7
Int 4										
Volume (vph)	235	232	203	227	105	105	162	100	172	90
Delay Avg (spv)	12.6	12.7	13.1	11.3	44.7	116.3	19.9	13.3	19.4	20.9
Delay Max (spv)	42.3	37.1	42.2	41.9	106.4	253.0	58.0	51.7	56.3	59.1
Q Avg (v)	1.9	1.9	1.7	1.6	3.0	9.1	2.1	0.8	2.2	1.3
Q Max (v)	9	7	9	9	8	21	8	4	10	7
Int 5										
Volume (vph)	217	238	216	222	106	101	159	94	174	103
Delay Avg (spv)	10.2	8.9	7.9	7.0	281.3	97.2	24.3	15.5	41.1	15.2
Delay Max (spv)	40.8	38.3	39.7	37.3	424.6	177.7	72.8	53.4	138.5	58.1
Q Avg (v)	1.4	1.3	1.1	1.0	22.1	7.4	2.6	1.0	4.9	1.0
Q Max (v)	8	8	8	8	31	19	9	4	14	6
Int 6										
Volume (vph)	206	215	226	231	99	100	168	103	160	92
Delay Avg (spv)	7.4	7.4	12.3	12.4	148.6	54.6	17.2	12.9	23.5	13.9
Delay Max (spv)	43.3	37.6	37.4	37.9	379.3	135.2	48.8	53.0	71.4	42.8
Q Avg (v)	1.0	1.0	1.8	1.9	13.2	3.6	1.9	0.8	2.5	0.8
Q Max (v)	6	7	7	7	29	10	7	4	8	6
Int 7										
Volume (vph)	231	219	207	222	106	104	166	105	168	105
Delay Avg (spv)	14.3	12.0	10.0	10.0	70.1	178.8	22.3	20.0	18.2	14.4
Delay Max (spv)	49.2	38.8	39.0	40.6	173.0	329.2	63.6	72.8	58.2	50.1
Q Avg (v)	2.2	1.8	1.4	1.5	4.9	12.9	2.4	1.3	2.0	1.0
Q Max (v)	10	8	10	9	11	25	13	7	8	4
Int 8										
Volume (vph)	215	233	216	226	107	109	168	107	168	100
Delay Avg (spv)	11.186	10.945	9.491	8.8886	63.084	172.09	27.77	17.264	19.172	13.957
Delay Max (spv)	44.89	44.544	45.185	45.779	193.98	303.19	82.25	52.165	58.015	48.525
Q Avg (v)	1.5642	1.6638	1.3582	1.3271	4.2757	13.384	3.0163	1.2096	2.0007	0.8816
Q Max (v)	8	8	9	9	12	23	11	8	8	4
Int 9										
Volume (vph)	207	238	226	232	107	101	166	91	169	99
Delay Avg (spv)	10.888	9.8068	12.637	12.18	130.55	59.289	43.904	14.107	18.648	14.405
Delay Max (spv)	50.493	40.04	43.655	38.165	252.77	151.02	109.95	50.5	68.955	59.91
Q Avg (v)	1.4567	1.5006	1.8905	1.8816	9.4567	3.9732	4.8768	0.8246	2.0524	0.9163
Q Max (v)	7	8	7	7	23	12	16	6	8	5

Network 3 - Pre-Timed Control - Replication 4

Movement	EB T 1	EB T 2	WB T 5	WB T 6	EB L 9	WB L 10	NB T 11	NB R 12	SB T 13	SB R 14
Int 1										
Volume (vph)	236	219	233	257	104	88	162	105	181	90
Delay Avg (spv)	21.3	21.1	17.9	19.6	26.6	36.4	25.0	18.3	31.5	18.3
Delay Max (spv)	48.4	44.1	49.4	51.4	54.5	86.5	52.2	51.1	105.9	53.3
Q Avg (v)	3.4	3.1	2.8	3.3	1.8	2.0	2.7	1.3	3.8	1.1
Q Max (v)	10	9	8	10	5	7	8	5	12	6
Int 2										
Volume (vph)	222	239	237	244	109	80	163	112	179	103
Delay Avg (spv)	32.5	30.4	13.8	14.9	38.0	39.2	24.3	18.0	23.3	16.3
Delay Max (spv)	57.0	49.2	48.5	48.3	101.4	93.7	62.5	51.2	58.1	50.7
Q Avg (v)	4.7	4.8	2.2	2.4	2.9	2.2	2.6	1.3	2.6	1.1
Q Max (v)	13	12	7	7	12	8	8	5	8	6
Int 3										
Volume (vph)	254	246	239	242	81	98	152	94	181	86
Delay Avg (spv)	39.3	40.2	20.4	20.5	19.8	25.6	38.3	35.1	44.0	33.9
Delay Max (spv)	82.2	77.9	42.9	45.7	36.7	56.0	80.1	50.0	99.8	50.5
Q Avg (v)	6.4	6.4	3.1	3.2	1.0	1.6	3.9	2.3	5.3	2.1
Q Max (v)	16	15	10	10	7	5	11	8	14	7
Int 4										
Volume (vph)	229	232	218	223	99	106	174	101	161	107
Delay Avg (spv)	20.0	19.1	9.8	12.2	26.1	48.1	24.7	20.6	43.8	36.3
Delay Max (spv)	43.6	43.4	44.5	46.2	55.5	111.2	52.0	50.0	96.8	57.6
Q Avg (v)	3.0	2.9	1.4	1.9	1.7	3.4	2.8	1.3	4.6	2.6
Q Max (v)	10	9	8	7	5	11	8	5	15	9
Int 5										
Volume (vph)	229	234	233	240	99	105	161	100	153	100
Delay Avg (spv)	33.3	32.5	31.0	31.0	28.5	30.5	23.2	19.4	36.1	36.6
Delay Max (spv)	44.2	61.3	44.4	44.7	82.2	89.0	50.0	48.9	80.1	82.7
Q Avg (v)	5.0	5.0	4.7	4.8	1.8	2.0	2.5	1.3	3.7	2.5
Q Max (v)	13	13	12	11	7	9	8	5	11	10
Int 6										
Volume (vph)	225	235	237	228	90	101	160	105	177	106
Delay Avg (spv)	12.1	13.3	19.4	20.1	47.4	26.8	23.2	20.7	40.6	21.9
Delay Max (spv)	45.3	46.4	43.7	45.9	104.3	56.4	50.0	50.2	100.1	50.9
Q Avg (v)	1.7	2.0	3.0	3.0	2.9	1.8	2.4	1.4	4.8	1.5
Q Max (v)	6	7	9	9	9	5	7	4	14	6
Int 7										
Volume (vph)	230	232	217	228	96	116	165	107	165	104
Delay Avg (spv)	20.2	19.9	13.1	14.6	27.8	63.1	39.8	35.0	23.6	21.7
Delay Max (spv)	47.3	50.6	45.0	45.6	56.2	168.3	89.7	49.3	50.1	50.2
Q Avg (v)	3.1	3.1	1.9	2.2	1.8	5.1	4.2	2.4	2.5	1.4
Q Max (v)	9	10	6	7	5	16	12	8	7	5
Int 8										
Volume (vph)	240	239	231	241	94	101	154	109	165	100
Delay Avg (spv)	31.241	29.987	32.608	30.552	25.782	32.305	38.197	35.201	23.76	20.894
Delay Max (spv)	46.05	47.07	54.005	84.49	79.358	86.314	88.862	83.209	50.105	50.465
Q Avg (v)	4.8913	4.6804	4.8289	4.7051	1.6423	2.1452	4.0224	2.6029	2.664	1.4071
Q Max (v)	11	12	13	13	8	8	13	10	8	5
Int 9										
Volume (vph)	250	261	233	229	69	105	154	90	171	101
Delay Avg (spv)	17.646	16.933	18.861	19.71	35.324	30.063	23.286	19.212	24.876	19.205
Delay Max (spv)	55.72	53.403	43.62	44.655	76.829	56.955	63.07	51.732	57.655	48.165
Q Avg (v)	2.9006	2.8962	2.8994	2.9743	1.7159	2.1386	2.2829	1.0982	2.703	1.2131
Q Max (v)	9	10	9	9	7	6	8	4	8	5

Network 3 - Constrained COP - Replication 4

Movement	EB T 1	EB T 2	WB T 5	WB T 6	EB L 9	WB L 10	NB T 11	NB R 12	SB T 13	SB R 14
Int 1										
Volume (vph)	237	221	234	251	101	91	169	91	156	106
Delay Avg (spv)	14.1	13.7	12.6	12.7	145.8	138.3	31.7	19.6	20.5	14.2
Delay Max (spv)	49.7	44.5	47.4	45.8	222.0	250.8	76.8	54.2	84.6	85.6
Q Avg (v)	2.2	2.0	1.9	2.1	9.8	9.6	3.5	1.1	2.1	1.0
Q Max (v)	10	7	11	12	16	20	15	7	9	6
Int 2										
Volume (vph)	213	225	237	254	109	94	151	118	169	98
Delay Avg (spv)	15.8	14.5	11.0	10.4	142.3	75.3	29.5	17.2	28.2	15.5
Delay Max (spv)	47.3	43.5	44.3	47.8	242.7	181.5	104.4	55.7	114.3	58.2
Q Avg (v)	2.2	2.2	1.7	1.7	10.9	4.3	2.8	1.3	3.2	1.0
Q Max (v)	9	9	7	8	19	12	12	6	12	5
Int 3										
Volume (vph)	232	229	228	237	86	100	142	105	157	116
Delay Avg (spv)	12.5	12.5	11.2	11.3	54.0	91.9	23.2	18.3	24.8	17.9
Delay Max (spv)	46.3	42.4	36.6	32.5	133.6	165.0	69.3	59.4	73.9	72.5
Q Avg (v)	1.9	1.9	1.7	1.7	3.0	5.9	2.1	1.2	2.5	1.3
Q Max (v)	9	8	8	7	10	11	7	8	12	8
Int 4										
Volume (vph)	229	232	234	245	95	75	161	98	155	93
Delay Avg (spv)	12.3	10.1	9.0	10.1	88.2	81.3	23.7	15.2	24.1	18.1
Delay Max (spv)	43.7	43.8	52.6	43.5	207.8	181.2	77.7	58.5	66.9	62.9
Q Avg (v)	1.8	1.5	1.4	1.6	5.9	4.5	2.5	1.0	2.4	1.1
Q Max (v)	8	8	10	11	15	15	9	5	10	7
Int 5										
Volume (vph)	219	236	222	234	101	106	167	103	156	105
Delay Avg (spv)	12.2	11.9	11.5	11.9	74.2	96.2	20.6	16.0	23.3	22.3
Delay Max (spv)	47.2	36.8	47.3	45.3	199.4	185.6	66.2	68.5	70.8	75.1
Q Avg (v)	1.8	1.8	1.7	1.8	4.7	6.5	2.2	1.0	2.4	1.5
Q Max (v)	10	11	8	8	13	16	8	4	10	9
Int 6										
Volume (vph)	224	252	235	240	78	89	157	99	159	93
Delay Avg (spv)	9.4	8.9	11.2	10.2	75.6	133.6	18.1	13.9	21.8	15.7
Delay Max (spv)	38.9	36.7	41.7	42.2	176.7	229.7	61.7	48.2	61.5	75.6
Q Avg (v)	1.3	1.4	1.7	1.6	4.5	9.1	1.8	0.9	2.3	0.9
Q Max (v)	6	7	7	7	14	17	7	4	9	6
Int 7										
Volume (vph)	233	237	233	236	91	94	170	98	163	102
Delay Avg (spv)	11.9	12.4	9.2	8.9	215.4	244.6	37.1	18.9	19.4	14.1
Delay Max (spv)	42.3	46.4	42.0	42.4	316.5	363.7	92.9	81.4	74.8	62.9
Q Avg (v)	1.8	1.9	1.4	1.4	13.9	16.2	4.3	1.2	2.0	0.9
Q Max (v)	7	8	8	9	22	28	15	8	8	5
Int 8										
Volume (vph)	223	228	217	236	95	98	160	99	176	102
Delay Avg (spv)	10.762	10.704	11.46	11.54	109.33	142.87	24.779	16.273	22.878	16.68
Delay Max (spv)	45.825	43.495	44.635	45.075	228.11	259.22	86.576	62.136	75.3	65.955
Q Avg (v)	1.5484	1.5714	1.5555	1.7516	7.5444	9.6553	2.5791	1.0241	2.5143	1.0903
Q Max (v)	8	8	9	8	19	18	12	7	8	5
Int 9										
Volume (vph)	218	228	223	237	93	99	142	87	170	100
Delay Avg (spv)	12.786	12.424	12.486	11.377	73.674	106.41	26.613	20.712	25.789	16.154
Delay Max (spv)	52.116	50.91	47.4	48.465	173.91	234.11	92.555	60.009	88.465	65.91
Q Avg (v)	1.836	1.8739	1.8089	1.7827	4.9233	7.6666	2.4021	1.1496	2.8125	1.0056
Q Max (v)	9	10	8	8	14	16	9	7	12	5

Network 3 - COP - Replication 4

Movement	EB T	EB T	WB T	WB T	EB L	WB L	NB T	NB R	SB T	SB R
	1	2	5	6	9	10	11	12	13	14
Int 1										
Volume (vph)	240	223	245	249	100	95	177	93	155	93
Delay Avg (spv)	13.5	11.3	9.0	9.6	52.7	87.0	27.3	10.3	19.8	11.9
Delay Max (spv)	38.2	35.2	41.5	42.9	125.8	198.3	62.3	56.2	66.3	59.6
Q Avg (v)	2.1	1.6	1.4	1.5	3.6	6.3	3.2	0.6	2.0	0.7
Q Max (v)	8	6	7	7	8	16	12	4	8	5
Int 2										
Volume (vph)	223	227	236	246	97	103	145	113	165	114
Delay Avg (spv)	10.7	10.5	8.9	8.7	90.4	126.2	13.7	11.6	17.1	17.4
Delay Max (spv)	53.7	50.2	46.1	40.8	289.9	282.6	60.9	51.6	55.0	67.2
Q Avg (v)	1.5	1.6	1.4	1.4	6.7	8.4	1.3	0.9	1.9	1.3
Q Max (v)	9	9	8	8	23	18	6	8	8	7
Int 3										
Volume (vph)	228	249	238	239	89	96	165	91	159	108
Delay Avg (spv)	9.9	10.3	12.0	12.5	122.4	50.8	20.1	11.9	17.6	15.5
Delay Max (spv)	47.1	43.4	36.0	37.6	278.7	125.0	78.5	59.8	66.1	60.5
Q Avg (v)	1.5	1.6	1.8	1.9	7.3	3.3	2.2	0.7	1.8	1.1
Q Max (v)	8	8	9	10	17	9	10	6	7	6
Int 4										
Volume (vph)	228	230	228	229	97	91	170	101	163	85
Delay Avg (spv)	12.9	11.5	10.1	9.8	60.4	72.3	21.1	14.8	24.7	10.1
Delay Max (spv)	34.8	34.9	47.8	44.5	167.9	175.1	63.6	46.0	97.7	43.6
Q Avg (v)	1.9	1.7	1.5	1.5	4.0	4.5	2.3	0.9	2.6	0.5
Q Max (v)	8	6	8	7	12	12	8	4	10	6
Int 5										
Volume (vph)	236	229	225	223	96	96	168	100	160	98
Delay Avg (spv)	7.5	8.8	8.8	9.6	89.8	80.2	21.1	15.0	17.3	11.2
Delay Max (spv)	38.5	37.7	40.1	40.0	179.5	200.2	70.8	55.7	60.5	61.3
Q Avg (v)	1.1	1.3	1.3	1.4	5.6	5.2	2.2	1.0	1.8	0.7
Q Max (v)	7	8	8	8	13	11	10	4	8	7
Int 6										
Volume (vph)	237	230	233	240	97	105	155	100	171	87
Delay Avg (spv)	8.7	9.5	13.1	12.9	62.1	43.4	16.9	13.3	30.6	11.3
Delay Max (spv)	47.2	48.7	37.4	39.2	127.5	105.0	48.5	40.5	93.2	51.0
Q Avg (v)	1.3	1.4	2.0	2.0	4.4	2.9	1.7	0.9	3.4	0.6
Q Max (v)	7	7	7	7	11	7	6	4	12	5
Int 7										
Volume (vph)	232	235	219	237	100	86	168	108	163	103
Delay Avg (spv)	12.7	11.2	8.0	8.5	46.3	98.5	21.7	14.7	15.4	12.3
Delay Max (spv)	37.8	34.5	43.9	47.2	113.8	201.9	62.2	53.7	41.7	37.2
Q Avg (v)	1.9	1.7	1.1	1.3	3.0	6.4	2.4	1.0	1.6	0.8
Q Max (v)	8	7	8	8	9	14	10	7	6	4
Int 8										
Volume (vph)	243	236	216	237	93	115	145	91	167	102
Delay Avg (spv)	11.856	11.907	7.8442	7.922	44.06	74.895	21.49	12.054	19.377	13.431
Delay Max (spv)	43.215	43.035	39.184	39.57	101.88	128.95	60.546	49.31	50.015	37.4
Q Avg (v)	1.8544	1.8062	1.0832	1.1973	2.738	5.73	2.0385	0.6974	2.1297	0.8628
Q Max (v)	8	10	6	6	9	11	9	4	7	4
Int 9										
Volume (vph)	226	241	238	232	96	102	168	84	165	96
Delay Avg (spv)	9.5632	9.3997	11.346	12.089	53.359	91.857	42.046	11.971	19.004	12.617
Delay Max (spv)	35.595	38.56	37.91	39.165	143.91	174.47	104.25	62.506	49.165	44.655
Q Avg (v)	1.42	1.491	1.7186	1.7935	3.4986	6.3918	4.7744	0.6248	2.0478	0.7936
Q Max (v)	8	9	8	8	10	12	12	4	6	4

Network 3 - Pre-Timed Control - Replication 5

Movement	EB T 1	EB T 2	WB T 5	WB T 6	EB L 9	WB L 10	NB T 11	NB R 12	SB T 13	SB R 14
Int 1										
Volume (vph)	220	239	214	217	102	92	173	91	165	93
Delay Avg (spv)	20.7	21.0	14.1	14.2	29.1	44.7	26.2	17.1	23.0	13.3
Delay Max (spv)	43.2	44.1	45.8	47.6	57.0	102.9	58.6	50.8	59.1	50.6
Q Avg (v)	3.1	3.3	2.0	2.0	1.9	2.8	3.0	1.0	2.6	0.8
Q Max (v)	9	10	8	7	6	10	8	4	7	4
Int 2										
Volume (vph)	215	241	225	224	97	104	158	106	168	100
Delay Avg (spv)	31.2	29.0	10.4	12.0	40.8	43.0	19.5	17.6	24.3	17.6
Delay Max (spv)	44.3	45.9	47.7	45.8	89.4	87.0	54.8	50.6	63.6	49.3
Q Avg (v)	4.4	4.6	1.6	1.8	2.7	3.0	2.1	1.2	2.6	1.1
Q Max (v)	11	12	7	7	11	8	7	4	9	5
Int 3										
Volume (vph)	220	230	234	231	113	104	166	102	158	102
Delay Avg (spv)	35.3	33.8	19.7	19.7	42.0	26.0	43.4	34.5	39.2	32.0
Delay Max (spv)	47.7	51.9	43.4	43.5	95.0	55.9	99.3	50.2	92.5	50.3
Q Avg (v)	5.1	5.1	3.0	3.0	3.1	1.7	4.8	2.4	4.2	2.2
Q Max (v)	13	12	9	8	12	5	15	7	11	7
Int 4										
Volume (vph)	224	232	212	231	96	93	172	102	164	91
Delay Avg (spv)	18.6	19.8	12.3	12.6	26.7	52.8	24.4	20.5	39.9	32.3
Delay Max (spv)	43.4	43.5	44.8	45.2	56.2	104.2	54.1	49.7	90.4	53.7
Q Avg (v)	2.7	3.0	1.8	1.9	1.7	3.8	2.7	1.3	4.3	1.9
Q Max (v)	9	9	9	9	5	13	8	5	11	7
Int 5										
Volume (vph)	232	234	216	240	98	114	165	100	159	108
Delay Avg (spv)	30.8	31.7	30.2	28.7	28.3	35.5	23.2	21.0	35.7	33.9
Delay Max (spv)	44.8	45.1	44.5	46.0	82.3	86.2	52.5	50.0	51.8	48.8
Q Avg (v)	4.7	4.8	4.2	4.4	1.7	2.5	2.6	1.4	3.9	2.4
Q Max (v)	13	11	11	12	9	11	7	5	10	9
Int 6										
Volume (vph)	227	235	227	232	101	94	168	103	169	104
Delay Avg (spv)	12.1	11.7	20.4	19.4	48.0	27.7	23.5	20.1	28.8	21.8
Delay Max (spv)	44.2	48.8	43.4	44.2	104.9	56.2	50.5	49.5	64.5	51.2
Q Avg (v)	1.8	1.8	3.1	2.9	3.2	1.8	2.5	1.3	3.2	1.4
Q Max (v)	7	7	8	8	10	5	7	5	10	5
Int 7										
Volume (vph)	229	227	228	241	99	94	169	107	175	99
Delay Avg (spv)	20.3	20.3	12.7	13.8	26.1	38.0	39.7	33.3	23.8	21.8
Delay Max (spv)	44.1	43.4	44.3	44.4	54.8	56.0	90.3	49.5	60.8	50.1
Q Avg (v)	3.1	3.1	1.9	2.2	1.7	2.5	4.4	2.3	2.7	1.3
Q Max (v)	9	9	6	7	5	7	11	7	9	5
Int 8										
Volume (vph)	226	243	226	235	98	98	154	97	160	98
Delay Avg (spv)	32.518	29.377	30.362	27.705	30.402	32.502	36.188	30.404	23.153	20.937
Delay Max (spv)	53.555	44.949	44.719	45.455	93.32	88.6	88.25	49.595	49.91	50.015
Q Avg (v)	4.7904	4.6619	4.4241	4.1817	1.9578	2.2471	3.7566	2.0332	2.4804	1.3694
Q Max (v)	12	12	11	11	10	11	11	8	7	5
Int 9										
Volume (vph)	217	242	225	225	97	96	171	108	169	108
Delay Avg (spv)	10.157	11.131	20.36	19.993	44.687	26.234	35.602	22.266	22.785	20.371
Delay Max (spv)	45.555	44.494	43.465	43.465	108.42	55.465	62.31	56.48	50.465	49.655
Q Avg (v)	1.434	1.7917	3.0314	2.9846	2.87	1.7121	4.0368	1.5497	2.481	1.3809
Q Max (v)	6	5	9	10	10	5	10	7	7	5

Network 3 - Constrained COP - Replication 5

Movement	EB T 1	EB T 2	WB T 5	WB T 6	EB L 9	WB L 10	NB T 11	NB R 12	SB T 13	SB R 14
Int 1										
Volume (vph)	222	237	212	230	104	106	157	102	162	107
Delay Avg (spv)	12.3	13.8	12.0	11.2	99.0	81.9	20.3	12.2	21.5	14.9
Delay Max (spv)	43.6	42.8	47.4	44.8	190.0	181.0	66.3	56.1	60.7	53.3
Q Avg (v)	1.8	2.2	1.7	1.7	6.6	5.6	2.1	0.8	2.3	1.1
Q Max (v)	7	9	10	10	13	13	9	5	11	5
Int 2										
Volume (vph)	226	236	227	233	106	97	168	99	161	102
Delay Avg (spv)	13.4	15.0	10.2	10.9	93.2	69.0	30.2	19.1	22.1	14.0
Delay Max (spv)	51.6	51.3	51.1	45.4	253.5	166.2	99.4	57.3	67.9	56.3
Q Avg (v)	2.0	2.3	1.5	1.7	6.6	4.8	3.4	1.2	2.3	0.9
Q Max (v)	9	10	7	8	17	14	12	7	8	5
Int 3										
Volume (vph)	212	225	234	230	100	100	131	117	156	107
Delay Avg (spv)	11.2	10.1	10.8	10.3	194.8	69.7	16.8	16.9	23.7	15.6
Delay Max (spv)	34.2	37.3	34.0	32.2	331.0	167.9	63.3	74.7	64.1	63.1
Q Avg (v)	1.5	1.5	1.6	1.5	15.1	4.7	1.4	1.3	2.4	1.0
Q Max (v)	11	9	6	6	24	11	7	8	11	7
Int 4										
Volume (vph)	226	233	217	220	97	91	158	98	174	83
Delay Avg (spv)	11.8	12.6	13.5	13.4	70.2	60.3	20.9	15.7	40.3	15.7
Delay Max (spv)	43.1	43.8	55.9	56.7	132.8	133.8	73.3	54.1	114.9	54.1
Q Avg (v)	1.7	1.9	1.9	1.9	4.6	3.7	2.2	1.0	4.7	0.9
Q Max (v)	8	8	10	9	10	12	9	4	16	7
Int 5										
Volume (vph)	216	225	222	214	105	102	169	102	165	98
Delay Avg (spv)	8.8	9.2	12.5	12.9	144.3	89.1	22.6	13.2	19.9	15.7
Delay Max (spv)	41.1	37.7	48.4	47.6	238.1	217.8	74.5	48.4	59.9	49.2
Q Avg (v)	1.2	1.4	1.8	1.8	10.1	5.9	2.5	0.8	2.1	1.0
Q Max (v)	7	9	10	10	18	14	9	5	8	6
Int 6										
Volume (vph)	218	235	231	228	83	94	169	101	172	90
Delay Avg (spv)	10.1	10.6	13.0	13.2	79.3	99.7	21.2	14.6	21.7	13.9
Delay Max (spv)	38.7	45.0	49.4	50.5	230.1	179.1	58.8	54.7	83.7	65.2
Q Avg (v)	1.5	1.6	2.0	2.0	5.0	6.3	2.3	0.9	2.5	0.8
Q Max (v)	7	8	8	9	18	13	8	5	10	5
Int 7										
Volume (vph)	232	230	218	222	101	102	152	101	172	96
Delay Avg (spv)	10.4	11.5	8.9	8.9	59.1	115.1	20.2	18.1	24.0	15.3
Delay Max (spv)	38.7	39.0	45.8	39.4	127.5	213.6	71.9	83.8	63.7	48.4
Q Avg (v)	1.6	1.7	1.2	1.3	3.9	8.3	2.0	1.2	2.7	0.9
Q Max (v)	7	7	6	7	10	18	9	8	8	5
Int 8										
Volume (vph)	230	229	225	231	87	97	165	109	162	100
Delay Avg (spv)	12.251	11.875	12.072	12.451	100.98	122.31	35.511	21.102	25.905	15.012
Delay Max (spv)	52.936	50.325	58.97	60.41	228.13	241.93	105.25	70.965	74.91	48.525
Q Avg (v)	1.7576	1.6761	1.7097	1.8011	7.4897	8.6545	3.9238	1.4928	2.7495	0.9563
Q Max (v)	8	7	10	10	21	22	16	8	9	4
Int 9										
Volume (vph)	234	225	229	226	96	95	138	88	163	105
Delay Avg (spv)	10.509	11.42	12.208	12.941	109.99	59.943	20.035	14.331	17.785	14.618
Delay Max (spv)	44.847	45.142	45.3	51.465	221.56	139.4	60.254	54.29	55.415	50.465
Q Avg (v)	1.5778	1.6724	1.7983	1.8952	7.338	3.9002	1.8112	0.8186	1.9235	0.9781
Q Max (v)	9	9	9	8	17	10	8	5	7	5

Network 3 - COP - Replication 5

Movement	EB T	EB T	WB T	WB T	EB L	WB L	NB T	NB R	SB T	SB R
	1	2	5	6	9	10	11	12	13	14
Int 1										
Volume (vph)	224	242	211	233	101	99	165	98	151	120
Delay Avg (spv)	11.6	12.7	9.5	9.7	44.7	92.5	17.6	11.4	14.7	13.5
Delay Max (spv)	42.1	45.7	44.2	38.2	116.0	193.1	52.4	48.7	48.1	49.8
Q Avg (v)	1.7	2.0	1.3	1.4	2.9	6.4	1.9	0.7	1.5	1.0
Q Max (v)	7	8	9	9	8	15	6	6	9	7
Int 2										
Volume (vph)	232	246	221	231	90	101	160	102	151	98
Delay Avg (spv)	10.3	9.4	8.9	9.9	71.9	73.3	31.9	12.5	21.3	19.1
Delay Max (spv)	41.7	45.2	49.4	49.4	164.3	164.0	109.9	47.8	83.8	65.6
Q Avg (v)	1.5	1.5	1.3	1.5	4.4	5.2	3.7	0.8	2.1	1.2
Q Max (v)	10	10	8	9	13	15	16	6	11	6
Int 3										
Volume (vph)	239	251	228	230	76	98	164	100	161	108
Delay Avg (spv)	10.9	10.1	12.3	12.4	84.3	75.3	24.7	17.7	19.4	12.9
Delay Max (spv)	70.4	41.9	49.2	53.2	207.4	220.9	69.4	65.1	61.7	58.6
Q Avg (v)	1.7	1.6	1.8	1.9	5.1	5.1	2.6	1.1	2.1	0.9
Q Max (v)	12	12	8	9	17	14	14	8	8	5
Int 4										
Volume (vph)	223	232	213	239	97	103	165	98	162	80
Delay Avg (spv)	13.1	12.1	11.4	9.6	49.6	60.9	21.4	14.3	21.8	9.9
Delay Max (spv)	47.9	41.6	39.9	40.3	119.4	136.7	51.9	45.1	69.3	47.8
Q Avg (v)	1.9	1.8	1.6	1.5	3.2	4.4	2.4	0.9	2.4	0.5
Q Max (v)	8	7	7	8	9	11	8	4	8	4
Int 5										
Volume (vph)	222	230	229	238	106	109	167	101	151	95
Delay Avg (spv)	9.3	9.1	13.4	11.3	146.6	60.6	20.7	15.8	24.9	16.4
Delay Max (spv)	54.7	49.6	57.7	50.8	291.3	153.7	56.2	44.7	77.6	56.0
Q Avg (v)	1.3	1.3	2.0	1.7	10.4	4.7	2.3	1.0	2.5	1.0
Q Max (v)	8	8	7	8	22	15	7	4	10	6
Int 6										
Volume (vph)	206	231	237	228	95	96	162	99	132	122
Delay Avg (spv)	7.7	8.0	11.9	12.1	162.0	46.5	17.4	11.5	16.1	14.5
Delay Max (spv)	39.5	40.7	38.0	41.3	324.7	105.1	50.5	39.0	61.0	55.2
Q Avg (v)	1.0	1.2	1.8	1.8	12.6	3.0	1.8	0.7	1.4	1.2
Q Max (v)	7	7	7	7	28	8	7	3	8	6
Int 7										
Volume (vph)	231	229	234	231	103	100	149	111	174	96
Delay Avg (spv)	12.0	12.5	7.9	8.2	62.5	57.8	20.2	17.5	20.1	13.5
Delay Max (spv)	42.5	38.0	36.8	36.8	142.0	133.6	74.2	52.6	57.0	44.1
Q Avg (v)	1.8	1.9	1.2	1.2	4.2	3.8	2.0	1.3	2.3	0.8
Q Max (v)	7	8	7	7	10	13	11	6	8	4
Int 8										
Volume (vph)	217	238	228	232	106	90	167	88	165	102
Delay Avg (spv)	11.341	9.6049	10.238	10.079	70.784	45.827	26.145	12.885	19.452	15.73
Delay Max (spv)	54.585	54.345	51.73	44.93	149.5	130.29	73.049	62.952	58.91	55.955
Q Avg (v)	1.6157	1.4788	1.5296	1.5243	5.001	2.7361	2.8912	0.7199	2.0518	1.0041
Q Max (v)	10	8	9	9	12	9	10	7	8	4
Int 9										
Volume (vph)	219	226	224	229	94	100	155	89	167	107
Delay Avg (spv)	8.8099	9.0193	12.467	11.511	65.841	41.684	34.313	12.824	18.019	13.077
Delay Max (spv)	49.48	47.78	37.955	32.4	170.99	101.17	128.03	50.1	55.465	47.165
Q Avg (v)	1.2338	1.2967	1.8068	1.7037	4.2377	2.6877	3.4306	0.7093	1.937	0.8648
Q Max (v)	7	7	9	8	11	7	18	6	7	4

Network 3 - Pre-Timed Control - Average Values

Movement	EB T 1	EB T 2	WB T 5	WB T 6	EB L 9	WB L 10	NB T 11	NB R 12	SB T 13	SB R 14	Overall
Int 1											
Volume (vph)	224.8	229.2	231.0	238.0	103.4	95.2	158.6	104.4	172.4	92.6	1649.6
Delay Avg (spv)	20.6	20.5	16.9	17.0	27.1	44.4	24.6	19.5	23.2	16.4	21.7
Delay Max (spv)	45.4	44.6	47.8	48.8	56.1	91.6	55.0	51.1	67.3	51.0	53.4
Q Avg (v)	3.1	3.1	2.6	2.7	1.8	2.7	2.6	1.4	2.7	1.0	2.6
Q Max (v)	9.6	9.6	8.0	8.0	5.2	9.2	8.2	5.8	8.4	4.8	
Int 2											
Volume (vph)	225.4	240.6	233.8	235.8	98.4	94.6	168.4	101.6	174.6	104.0	1677.2
Delay Avg (spv)	32.4	31.2	12.5	13.4	37.4	44.6	22.3	17.8	23.7	18.2	24.1
Delay Max (spv)	57.9	51.1	49.1	47.3	92.6	106.7	60.2	50.8	58.6	50.8	58.4
Q Avg (v)	4.8	5.0	1.9	2.1	2.5	2.9	2.4	1.1	2.6	1.2	2.9
Q Max (v)	12.2	12.2	6.8	7.2	10.0	9.4	8.4	4.4	7.8	5.8	
Int 3											
Volume (vph)	238.8	239.4	236.6	239.0	94.6	99.8	160.2	98.2	163.4	99.0	1669.0
Delay Avg (spv)	36.7	37.2	20.0	19.7	27.4	25.8	39.4	33.5	39.7	33.8	31.0
Delay Max (spv)	76.8	73.5	43.5	44.1	72.6	56.0	84.0	50.0	84.3	50.1	63.7
Q Avg (v)	5.7	5.8	3.1	3.1	1.7	1.6	4.3	2.3	4.4	2.3	3.8
Q Max (v)	14.2	13.8	9.2	9.0	8.4	5.0	11.8	7.6	11.8	7.4	
Int 4											
Volume (vph)	229.0	228.8	222.2	229.8	97.6	94.2	171.0	101.8	165.8	101.6	1641.8
Delay Avg (spv)	19.7	19.7	11.4	11.9	26.8	48.5	24.6	20.9	40.3	33.9	23.1
Delay Max (spv)	44.7	43.7	46.1	45.4	55.9	110.0	51.8	49.8	92.4	51.9	55.6
Q Avg (v)	3.0	3.0	1.7	1.8	1.8	3.2	2.7	1.4	4.3	2.2	2.5
Q Max (v)	9.4	9.0	7.2	7.2	4.8	10.6	7.8	4.8	11.8	8.0	
Int 5											
Volume (vph)	233.0	234.6	231.0	239.6	99.0	101.8	162.0	98.8	161.4	98.6	1659.8
Delay Avg (spv)	32.0	32.4	31.8	31.2	28.0	30.2	23.5	21.0	38.6	34.6	30.9
Delay Max (spv)	47.3	50.1	49.9	47.6	81.3	79.4	52.3	49.7	78.4	55.9	56.3
Q Avg (v)	4.8	4.9	4.8	4.8	1.8	2.0	2.6	1.4	4.3	2.3	3.8
Q Max (v)	12.2	12.0	12.2	11.6	8.2	8.8	7.8	4.8	11.6	8.4	
Int 6											
Volume (vph)	227.2	236.6	229.6	229.4	95.4	98.2	168.8	102.2	167.0	103.2	1657.6
Delay Avg (spv)	13.0	12.7	19.2	19.0	47.0	27.0	23.7	20.5	31.4	21.1	21.3
Delay Max (spv)	47.0	47.0	43.5	44.1	105.8	56.7	51.1	49.3	77.2	53.6	54.1
Q Avg (v)	1.9	2.0	2.9	2.9	3.0	1.8	2.6	1.3	3.5	1.4	2.4
Q Max (v)	6.8	7.2	8.8	8.4	9.4	5.2	7.4	4.6	10.2	5.6	
Int 7											
Volume (vph)	232.4	228.4	217.8	231.0	97.8	102.6	168.2	99.2	169.8	104.2	1651.4
Delay Avg (spv)	20.1	19.6	13.4	14.0	26.0	50.4	38.0	33.2	23.8	21.0	23.6
Delay Max (spv)	44.6	47.0	46.1	45.8	55.7	106.0	87.5	49.9	52.9	49.9	55.6
Q Avg (v)	3.1	3.0	2.0	2.1	1.7	3.6	4.1	2.1	2.6	1.4	2.6
Q Max (v)	9.2	9.6	6.8	6.8	5.0	10.6	10.8	7.0	7.8	5.0	
Int 8											
Volume (vph)	228.4	242.2	227.0	236.8	95.6	105.0	162.0	97.8	162.0	97.8	1654.6
Delay Avg (spv)	32.7	30.7	31.0	29.6	33.1	30.3	40.2	32.6	23.3	21.0	30.7
Delay Max (spv)	58.9	57.2	47.5	53.4	94.1	85.3	88.2	56.1	49.9	49.9	61.3
Q Avg (v)	4.9	4.8	4.5	4.5	2.0	2.1	4.5	2.2	2.5	1.4	3.8
Q Max (v)	12.6	12.8	11.4	11.6	9.4	9.4	12.4	8.0	7.4	4.6	
Int 9											
Volume (vph)	223.4	238.0	228.2	227.0	94.8	99.0	163.6	94.6	170.4	103.0	1642.0
Delay Avg (spv)	12.3	12.8	19.6	19.8	46.9	27.8	26.8	20.6	23.7	20.6	21.0
Delay Max (spv)	47.5	48.1	43.8	43.9	101.3	56.9	60.0	52.3	52.7	49.7	52.4
Q Avg (v)	1.8	2.0	2.9	3.0	3.1	1.9	2.9	1.3	2.6	1.3	2.4
Q Max (v)	7.0	7.2	9.0	9.4	10.0	5.6	9.0	6.0	7.6	4.8	

Network 3 - Constrained COP - Average Values

Movement	EB T 1	EB T 2	WB T 5	WB T 6	EB L 9	WB L 10	NB T 11	NB R 12	SB T 13	SB R 14	Overall
Int 1											
Volume (vph)	226.6	230.6	225.2	233.4	103.0	98.6	164.6	95.4	156.0	102.6	1636.0
Delay Avg (spv)	12.8	12.6	10.7	10.8	93.4	102.0	28.2	16.6	21.6	15.4	25.4
Delay Max (spv)	43.1	43.3	45.9	43.2	192.2	224.3	87.4	57.1	71.6	66.9	73.3
Q Avg (v)	1.9	1.9	1.6	1.7	6.3	6.8	3.1	1.0	2.2	1.0	2.4
Q Max (v)	8.4	8.0	8.6	9.2	13.4	14.6	12.4	6.8	9.8	5.6	
Int 2											
Volume (vph)	223.4	231.6	232.2	238.8	96.2	93.8	165.0	99.2	165.2	95.6	1641.0
Delay Avg (spv)	14.0	13.6	9.9	9.8	108.3	118.6	27.6	16.9	24.6	15.4	26.9
Delay Max (spv)	50.8	47.1	48.3	47.9	232.0	230.5	87.9	58.0	80.1	62.1	78.2
Q Avg (v)	2.0	2.1	1.5	1.5	7.5	8.3	3.0	1.1	2.7	0.9	2.6
Q Max (v)	9.2	9.6	7.2	7.6	16.6	18.0	10.8	6.4	10.6	5.8	
Int 3											
Volume (vph)	220.4	232.2	232.8	234.6	91.6	98.6	154.2	97.4	159.0	106.6	1627.4
Delay Avg (spv)	10.8	11.1	10.8	10.7	114.4	80.9	22.9	14.8	24.5	15.8	24.0
Delay Max (spv)	39.9	44.4	37.8	36.9	225.7	166.1	68.8	64.9	79.3	68.4	67.9
Q Avg (v)	1.5	1.7	1.6	1.6	7.9	5.3	2.4	0.9	2.6	1.1	2.3
Q Max (v)	9.2	8.6	7.2	7.0	16.0	11.2	9.8	6.6	11.6	6.8	
Int 4											
Volume (vph)	229.6	229.4	224.8	228.0	95.8	87.4	165.0	99.6	158.0	92.6	1610.2
Delay Avg (spv)	12.0	11.9	10.7	11.4	92.8	78.5	24.1	15.0	28.7	16.3	23.5
Delay Max (spv)	42.8	43.0	48.1	44.8	194.3	171.9	75.2	62.1	90.4	60.7	70.1
Q Avg (v)	1.8	1.8	1.6	1.7	6.1	4.8	2.6	1.0	3.0	1.0	2.3
Q Max (v)	7.8	8.0	9.0	9.2	14.0	13.6	9.0	5.2	12.2	7.2	
Int 5											
Volume (vph)	220.0	232.0	222.6	225.4	103.0	102.4	164.4	99.6	159.2	97.4	1626.0
Delay Avg (spv)	10.1	10.1	11.4	11.9	121.0	110.6	21.9	14.5	24.5	16.0	27.1
Delay Max (spv)	44.6	43.6	47.6	43.8	227.2	233.1	68.2	57.7	77.0	54.9	75.2
Q Avg (v)	1.4	1.5	1.7	1.7	8.3	8.0	2.4	0.9	2.6	1.0	2.5
Q Max (v)	7.8	8.2	8.6	9.0	17.6	18.6	8.2	4.4	9.6	6.4	
Int 6											
Volume (vph)	225.4	236.2	231.0	231.4	86.0	97.6	165.4	99.4	160.2	98.0	1630.6
Delay Avg (spv)	11.3	11.6	12.6	12.5	84.2	111.2	20.8	15.2	21.7	15.9	24.0
Delay Max (spv)	43.6	44.4	44.5	46.4	222.8	221.1	68.6	58.0	71.4	70.6	72.1
Q Avg (v)	1.7	1.8	1.9	1.9	5.3	7.4	2.2	1.0	2.3	1.0	2.3
Q Max (v)	8.2	8.8	7.8	7.8	17.8	16.6	8.6	5.0	8.8	6.0	
Int 7											
Volume (vph)	233.2	230.2	223.8	231.4	97.6	92.0	160.8	103.8	168.8	102.2	1643.8
Delay Avg (spv)	11.6	11.1	9.7	9.7	111.0	120.2	31.5	18.2	21.6	15.1	26.6
Delay Max (spv)	42.4	42.4	45.2	45.1	217.2	208.7	93.6	70.0	72.0	60.9	73.8
Q Avg (v)	1.7	1.7	1.4	1.5	7.3	7.9	3.4	1.2	2.3	1.0	2.5
Q Max (v)	7.4	7.8	8.2	8.0	15.2	16.6	12.8	7.6	8.6	5.0	
Int 8											
Volume (vph)	226.6	230.4	223.2	230.6	95.0	94.8	158.6	104.4	168.4	99.6	1631.6
Delay Avg (spv)	11.6	11.3	11.0	11.2	114.6	123.9	27.4	17.5	22.2	15.1	27.2
Delay Max (spv)	46.9	45.7	46.3	46.4	242.2	253.8	85.9	63.5	69.1	56.2	77.7
Q Avg (v)	1.7	1.7	1.5	1.6	8.2	8.5	2.9	1.2	2.4	1.0	2.5
Q Max (v)	8.6	7.8	9.4	8.4	20.2	19.8	11.6	7.0	8.6	4.6	
Int 9											
Volume (vph)	223.2	229.0	228.4	231.2	94.0	98.2	152.0	90.6	167.6	101.0	1615.2
Delay Avg (spv)	11.6	12.2	12.4	12.3	111.0	83.6	23.9	18.1	22.8	16.2	25.0
Delay Max (spv)	51.5	52.1	45.2	46.3	236.9	200.8	77.0	61.9	71.1	60.5	75.4
Q Avg (v)	1.7	1.8	1.8	1.9	8.3	5.6	2.4	1.1	2.5	1.0	2.5
Q Max (v)	9.2	9.2	8.2	8.2	20.0	13.8	9.8	7.0	9.2	5.4	

Network 3 - COP - Average Values

Movement	EB T 1	EB T 2	WB T 5	WB T 6	EB L 9	WB L 10	NB T 11	NB R 12	SB T 13	SB R 14	Overall
Int 1											
Volume (vph)	227.2	231.6	224.2	235.0	100.2	94.6	167.2	100.0	157.8	103.2	1641.0
Delay Avg (spv)	12.0	12.8	9.2	9.8	52.0	99.4	22.8	14.0	19.4	13.6	20.9
Delay Max (spv)	41.1	42.7	44.0	42.8	142.4	221.9	63.9	54.0	63.2	51.8	64.5
Q Avg (v)	1.8	1.9	1.3	1.5	3.4	7.0	2.5	0.9	2.0	0.9	2.1
Q Max (v)	7.8	8.6	8.2	8.0	10.2	16.0	9.8	5.8	9.2	5.8	
Int 2											
Volume (vph)	226.6	236.4	229.0	236.0	98.6	103.0	162.4	98.2	166.6	99.2	1656.0
Delay Avg (spv)	10.3	10.3	10.8	10.8	72.6	97.8	22.8	13.0	20.8	15.3	22.3
Delay Max (spv)	49.4	47.5	49.4	48.5	191.8	218.7	77.6	50.6	74.3	60.9	74.0
Q Avg (v)	1.5	1.6	1.6	1.7	4.9	6.7	2.5	0.8	2.3	1.0	2.2
Q Max (v)	8.4	8.4	9.2	9.2	14.2	15.8	10.6	6.0	10.4	6.6	
Int 3											
Volume (vph)	225.0	243.6	235.4	233.8	89.2	98.0	162.6	97.8	158.4	107.0	1650.8
Delay Avg (spv)	10.1	9.9	12.3	12.7	86.8	57.4	21.9	13.4	18.6	12.9	20.1
Delay Max (spv)	49.3	48.0	39.7	42.3	191.9	142.0	74.6	54.4	63.3	51.9	64.2
Q Avg (v)	1.5	1.6	1.9	1.9	5.4	3.8	2.3	0.9	1.9	0.9	2.0
Q Max (v)	8.6	8.6	7.6	8.4	14.0	9.6	11.0	6.0	8.0	5.6	
Int 4											
Volume (vph)	228.2	228.4	215.2	226.2	99.8	100.6	166.8	99.6	158.0	88.8	1611.6
Delay Avg (spv)	12.8	12.3	10.2	10.3	49.6	91.2	21.0	13.9	21.4	13.1	21.0
Delay Max (spv)	40.0	38.8	42.0	44.6	129.8	199.3	56.3	46.6	68.4	50.9	61.7
Q Avg (v)	1.9	1.8	1.4	1.5	3.3	6.5	2.3	0.9	2.2	0.8	2.1
Q Max (v)	8.0	7.0	7.8	8.6	9.4	15.4	7.6	4.2	9.2	5.8	
Int 5											
Volume (vph)	229.8	231.2	222.2	229.2	97.2	100.0	165.4	99.8	164.0	99.4	1638.2
Delay Avg (spv)	9.6	9.9	10.0	9.7	128.5	86.7	21.0	15.5	27.1	14.1	25.0
Delay Max (spv)	45.4	43.8	44.6	45.0	240.3	192.2	62.5	51.9	94.3	55.3	73.1
Q Avg (v)	1.4	1.5	1.4	1.4	9.0	6.0	2.2	1.0	3.0	0.9	2.4
Q Max (v)	8.2	8.4	7.6	8.0	17.4	15.4	8.2	4.0	11.4	6.6	
Int 6											
Volume (vph)	221.4	230.4	231.2	231.4	96.0	101.6	165.0	99.4	160.8	97.2	1634.4
Delay Avg (spv)	8.4	8.6	12.4	12.2	109.0	56.3	18.0	12.8	23.6	14.5	21.5
Delay Max (spv)	41.3	40.5	37.5	40.1	255.9	129.1	54.4	47.5	71.8	54.2	64.0
Q Avg (v)	1.2	1.3	1.8	1.8	8.5	3.7	1.9	0.8	2.5	0.9	2.1
Q Max (v)	7.4	7.4	7.0	7.0	21.2	9.6	7.4	3.8	9.8	6.0	
Int 7											
Volume (vph)	233.0	228.8	218.8	231.0	101.6	96.0	162.6	102.0	166.8	101.6	1642.2
Delay Avg (spv)	13.0	12.1	8.5	8.5	57.6	112.8	23.5	15.5	17.8	13.1	21.9
Delay Max (spv)	43.9	40.3	41.4	42.4	132.4	218.9	69.0	54.0	52.2	44.2	62.5
Q Avg (v)	2.0	1.8	1.2	1.3	3.8	7.6	2.5	1.0	1.9	0.8	2.1
Q Max (v)	8.2	8.0	7.8	7.6	9.2	16.8	11.0	6.2	7.2	4.0	
Int 8											
Volume (vph)	224.0	235.6	220.4	230.8	103.4	103.8	161.0	97.8	166.6	100.6	1644.0
Delay Avg (spv)	10.6	10.6	9.4	9.1	69.4	84.3	25.8	13.4	19.0	14.0	21.3
Delay Max (spv)	43.9	49.5	45.2	47.0	164.9	178.6	77.5	55.6	55.6	47.0	66.8
Q Avg (v)	1.5	1.6	1.3	1.4	4.8	6.2	2.8	0.8	2.0	0.9	2.1
Q Max (v)	7.8	8.2	8.0	8.0	13.4	13.6	10.0	6.0	7.6	4.0	
Int 9											
Volume (vph)	218.0	235.4	228.4	230.2	98.0	100.4	160.6	89.6	165.8	99.8	1626.2
Delay Avg (spv)	9.7	9.2	12.1	11.7	81.4	58.8	35.0	13.2	18.6	13.6	21.4
Delay Max (spv)	42.2	41.2	40.0	36.5	183.4	137.9	101.0	54.0	54.2	48.1	63.4
Q Avg (v)	1.4	1.4	1.8	1.7	5.6	3.9	3.8	0.8	2.0	0.9	2.1
Q Max (v)	7.8	8.2	7.6	7.4	14.8	10.0	13.8	5.8	7.0	4.2	

Network 3 - Pre-Timed Control - Average Value Standard Deviations

Movement	EB T 1	EB T 2	WB T 5	WB T 6	EB L 9	WB L 10	NB T 11	NB R 12	SB T 13	SB R 14
Int 1										
Volume (vph)	6.9	7.8	11.5	15.3	1.3	9.2	10.8	10.2	12.5	11.1
Delay Avg (spv)	0.4	0.8	1.8	2.1	1.2	7.4	0.9	2.2	5.6	2.6
Delay Max (spv)	2.6	1.2	2.1	1.5	1.0	22.0	3.2	0.4	21.9	1.4
Q Avg (v)	0.1	0.2	0.3	0.5	0.1	0.7	0.3	0.3	0.8	0.3
Q Max (v)	0.5	0.5	0.0	1.2	0.4	1.6	0.4	1.3	2.6	0.8
Int 2										
Volume (vph)	10.5	4.0	6.8	12.3	8.4	9.3	7.7	7.9	4.7	7.2
Delay Avg (spv)	1.3	1.6	2.1	1.5	9.4	6.1	2.4	0.7	1.5	1.5
Delay Max (spv)	12.1	7.3	1.4	1.9	11.3	26.6	3.5	0.3	4.3	1.0
Q Avg (v)	0.3	0.3	0.3	0.4	0.7	0.6	0.3	0.1	0.3	0.2
Q Max (v)	1.3	0.4	0.4	0.4	2.0	2.2	1.1	0.5	1.1	0.4
Int 3										
Volume (vph)	14.5	7.6	3.4	4.8	11.5	2.4	5.3	5.2	13.3	9.9
Delay Avg (spv)	1.8	2.3	0.4	0.6	8.8	0.7	2.7	2.0	2.9	1.9
Delay Max (spv)	16.5	12.5	0.5	0.9	23.1	0.4	13.4	0.6	17.4	0.3
Q Avg (v)	0.5	0.5	0.1	0.1	0.8	0.1	0.4	0.2	0.7	0.3
Q Max (v)	1.1	1.1	0.4	0.7	2.3	0.0	1.9	0.5	2.6	1.1
Int 4										
Volume (vph)	4.8	3.6	7.4	8.9	3.1	7.1	3.5	1.5	4.9	7.3
Delay Avg (spv)	0.9	0.4	0.9	1.7	0.9	6.6	0.4	0.4	2.9	1.9
Delay Max (spv)	2.8	1.0	1.9	0.6	0.6	21.3	1.5	0.3	6.1	3.8
Q Avg (v)	0.2	0.0	0.1	0.3	0.1	0.6	0.1	0.0	0.3	0.3
Q Max (v)	0.5	0.0	1.3	1.1	0.4	2.6	0.4	0.4	1.9	1.0
Int 5										
Volume (vph)	3.7	2.9	10.8	6.0	3.2	14.8	2.6	1.8	5.9	6.2
Delay Avg (spv)	0.9	0.8	2.1	2.3	4.4	4.1	0.9	0.9	3.1	1.5
Delay Max (spv)	6.5	7.6	10.9	3.8	4.9	13.9	2.7	0.7	15.3	15.0
Q Avg (v)	0.1	0.1	0.5	0.5	0.2	0.5	0.1	0.0	0.5	0.1
Q Max (v)	0.8	1.0	0.8	0.5	1.3	1.5	0.4	0.4	2.1	1.1
Int 6										
Volume (vph)	2.9	3.9	4.4	1.9	6.1	2.6	5.4	3.0	14.8	5.4
Delay Avg (spv)	0.8	0.6	0.8	0.8	2.1	0.4	0.9	1.1	7.6	2.0
Delay Max (spv)	2.7	2.1	0.4	1.1	9.0	1.0	1.2	0.8	25.4	5.3
Q Avg (v)	0.1	0.1	0.1	0.1	0.3	0.0	0.2	0.1	1.1	0.2
Q Max (v)	1.3	1.1	0.8	0.5	1.1	0.4	0.5	0.5	3.0	0.9
Int 7										
Volume (vph)	2.8	4.4	7.5	8.6	3.3	8.5	10.6	11.0	4.1	4.3
Delay Avg (spv)	0.7	0.8	1.0	0.7	1.6	9.3	2.2	2.3	0.3	0.9
Delay Max (spv)	1.7	5.0	2.1	1.5	0.6	40.2	3.0	0.5	4.5	0.4
Q Avg (v)	0.1	0.2	0.2	0.1	0.1	0.9	0.5	0.3	0.1	0.1
Q Max (v)	0.4	0.9	1.3	0.8	0.7	3.3	0.8	0.7	0.8	0.0
Int 8										
Volume (vph)	8.1	7.4	6.0	4.4	6.2	7.4	10.5	6.4	3.2	1.3
Delay Avg (spv)	1.7	1.0	0.9	1.8	7.6	4.1	3.4	2.5	0.3	0.3
Delay Max (spv)	16.6	15.3	4.5	17.5	24.1	9.4	4.5	15.1	0.5	1.1
Q Avg (v)	0.2	0.2	0.2	0.3	0.5	0.4	0.6	0.3	0.1	0.0
Q Max (v)	1.5	1.3	1.1	1.3	1.8	1.3	1.1	1.2	0.5	0.5
Int 9										
Volume (vph)	15.9	17.1	3.7	5.1	15.2	3.7	6.7	8.5	3.6	3.1
Delay Avg (spv)	3.1	2.6	0.8	0.4	8.9	1.7	5.1	1.6	0.8	0.9
Delay Max (spv)	4.6	3.3	0.4	0.6	16.5	1.6	5.1	2.6	3.8	0.9
Q Avg (v)	0.6	0.5	0.1	0.1	1.0	0.2	0.7	0.2	0.1	0.1
Q Max (v)	1.2	1.9	0.0	0.5	1.9	0.9	1.0	1.2	0.5	0.4

Network 3 - Constrained COP - Average Value Standard Deviations

Movement	EB T 1	EB T 2	WB T 5	WB T 6	EB L 9	WB L 10	NB T 11	NB R 12	SB T 13	SB R 14
Int 1										
Volume (vph)	7.2	6.3	9.8	14.1	2.3	6.4	5.7	4.6	4.2	6.1
Delay Avg (spv)	0.8	1.1	1.5	1.3	31.4	28.8	6.7	2.7	4.5	1.0
Delay Max (spv)	4.4	3.1	3.5	3.8	22.6	51.1	23.0	3.6	17.3	12.0
Q Avg (v)	0.2	0.2	0.2	0.3	2.1	2.0	0.9	0.1	0.5	0.1
Q Max (v)	1.1	1.0	1.8	1.8	1.8	3.6	2.2	1.1	1.9	0.9
Int 2										
Volume (vph)	9.6	4.1	4.8	11.6	11.1	7.0	8.2	10.9	7.7	5.3
Delay Avg (spv)	1.1	1.1	0.7	0.9	25.3	60.2	5.2	1.7	3.3	1.5
Delay Max (spv)	3.9	3.1	5.4	3.0	41.3	68.7	19.2	5.2	19.3	10.7
Q Avg (v)	0.1	0.2	0.1	0.2	2.1	4.9	0.7	0.2	0.4	0.1
Q Max (v)	1.5	1.1	0.8	0.5	1.8	6.8	1.6	0.5	1.9	0.8
Int 3										
Volume (vph)	11.6	10.2	3.7	4.9	9.4	2.6	21.0	13.5	4.2	5.6
Delay Avg (spv)	1.1	1.2	0.4	0.4	66.4	15.8	5.5	2.7	3.4	1.3
Delay Max (spv)	5.0	7.6	4.2	4.6	98.2	23.0	9.6	7.0	10.2	10.0
Q Avg (v)	0.2	0.2	0.0	0.1	5.6	1.0	0.9	0.3	0.4	0.2
Q Max (v)	1.5	0.5	0.8	0.7	6.8	1.5	3.8	1.5	1.1	0.8
Int 4										
Volume (vph)	3.3	3.3	6.9	10.0	2.8	8.0	5.2	1.7	9.1	6.2
Delay Avg (spv)	0.7	1.3	2.0	1.4	27.8	23.9	3.8	0.9	7.2	1.6
Delay Max (spv)	2.9	1.2	6.4	7.9	43.2	29.9	12.1	12.6	17.2	14.8
Q Avg (v)	0.1	0.2	0.3	0.2	2.0	1.7	0.4	0.1	1.0	0.1
Q Max (v)	0.4	0.7	1.0	1.8	3.8	2.1	0.7	1.1	2.9	0.4
Int 5										
Volume (vph)	7.5	5.7	4.0	11.7	4.6	6.0	3.8	3.5	10.5	5.3
Delay Avg (spv)	1.3	1.1	1.0	1.0	63.8	53.0	1.0	1.4	3.7	3.9
Delay Max (spv)	5.9	6.3	0.9	3.9	53.6	52.7	4.0	9.8	11.7	11.5
Q Avg (v)	0.2	0.2	0.2	0.1	5.0	4.7	0.1	0.1	0.4	0.3
Q Max (v)	1.3	1.8	0.9	0.7	5.9	6.0	0.4	0.5	1.1	1.5
Int 6										
Volume (vph)	8.5	9.4	2.9	5.3	11.0	6.8	4.9	2.7	9.7	7.8
Delay Avg (spv)	1.5	2.1	1.0	1.5	28.8	21.6	2.9	1.8	1.9	1.6
Delay Max (spv)	4.5	4.5	4.7	3.5	78.4	43.1	10.5	10.0	11.5	10.3
Q Avg (v)	0.2	0.3	0.2	0.2	1.9	1.5	0.4	0.1	0.3	0.2
Q Max (v)	1.9	1.3	0.4	0.8	7.7	3.4	1.9	1.2	0.8	0.7
Int 7										
Volume (vph)	1.3	4.8	7.7	6.6	5.0	7.3	11.5	8.8	4.3	4.8
Delay Avg (spv)	0.7	0.8	1.2	1.3	60.8	71.8	9.7	1.5	2.8	1.0
Delay Max (spv)	3.4	2.9	5.4	6.8	69.0	90.6	28.1	14.9	9.8	9.8
Q Avg (v)	0.1	0.2	0.2	0.2	3.9	5.0	1.2	0.1	0.4	0.1
Q Max (v)	0.9	1.3	1.3	1.0	4.4	7.0	3.5	0.9	0.9	0.0
Int 8										
Volume (vph)	5.7	10.3	4.4	6.5	4.9	5.1	9.1	5.0	5.9	1.5
Delay Avg (spv)	1.3	1.7	1.1	1.2	21.0	45.8	4.7	2.5	2.3	1.7
Delay Max (spv)	5.1	5.8	9.3	8.6	36.5	67.0	14.6	15.1	8.6	11.3
Q Avg (v)	0.2	0.3	0.2	0.2	1.3	3.4	0.6	0.2	0.3	0.1
Q Max (v)	0.9	0.4	0.5	1.1	2.8	7.1	2.6	0.7	1.5	0.9
Int 9										
Volume (vph)	7.7	8.7	3.6	6.1	5.6	2.6	13.3	4.8	4.4	2.3
Delay Avg (spv)	2.2	2.0	0.6	0.9	42.7	19.1	3.0	2.8	3.5	1.8
Delay Max (spv)	13.1	11.9	4.0	4.5	84.2	38.6	12.2	5.0	12.8	6.7
Q Avg (v)	0.3	0.4	0.1	0.1	4.3	1.5	0.5	0.2	0.4	0.1
Q Max (v)	0.4	1.3	0.4	0.4	8.0	2.5	1.3	1.9	1.9	0.5

Network 3 - COP - Average Value Standard Deviations

Movement	EB T 1	EB T 2	WB T 5	WB T 6	EB L 9	WB L 10	NB T 11	NB R 12	SB T 13	SB R 14
Int 1										
Volume (vph)	7.8	7.6	13.2	9.0	1.9	5.9	6.7	5.5	14.7	14.1
Delay Avg (spv)	0.9	1.3	1.2	0.7	5.4	31.6	4.0	3.4	4.5	1.1
Delay Max (spv)	3.9	7.0	5.1	3.5	37.0	50.2	9.9	9.5	12.2	4.8
Q Avg (v)	0.2	0.2	0.2	0.1	0.4	2.6	0.5	0.3	0.7	0.2
Q Max (v)	0.4	2.2	1.3	1.0	3.3	2.5	2.7	1.1	1.3	0.8
Int 2										
Volume (vph)	3.5	8.4	7.6	7.8	8.0	3.7	11.4	9.7	12.3	9.0
Delay Avg (spv)	0.7	1.0	1.8	1.5	11.9	32.7	6.7	1.5	2.7	2.9
Delay Max (spv)	4.8	2.5	6.8	8.4	61.1	70.0	19.2	4.7	17.5	7.1
Q Avg (v)	0.1	0.1	0.3	0.2	1.2	2.0	0.9	0.1	0.4	0.3
Q Max (v)	1.5	1.1	1.3	1.1	5.1	3.8	3.7	1.2	1.5	0.5
Int 3										
Volume (vph)	9.7	7.1	4.9	6.8	8.8	2.5	2.3	5.1	5.9	12.9
Delay Avg (spv)	0.7	0.5	0.7	0.5	22.3	11.1	2.1	2.5	1.4	2.1
Delay Max (spv)	12.2	5.0	6.1	6.7	54.8	47.3	10.6	8.9	6.9	8.4
Q Avg (v)	0.2	0.1	0.1	0.1	1.2	0.8	0.2	0.2	0.3	0.1
Q Max (v)	1.9	1.9	0.9	1.5	3.0	2.6	1.9	1.4	0.7	0.9
Int 4										
Volume (vph)	5.8	4.2	9.0	9.0	4.8	5.9	4.1	1.1	11.2	7.4
Delay Avg (spv)	0.4	0.9	2.2	0.8	6.4	26.2	0.9	1.2	4.3	4.7
Delay Max (spv)	5.2	3.0	3.4	5.3	25.4	47.1	4.7	3.9	20.1	6.4
Q Avg (v)	0.1	0.1	0.3	0.1	0.4	2.1	0.1	0.1	0.5	0.3
Q Max (v)	0.7	0.7	0.8	1.1	1.5	4.0	0.5	0.4	1.6	1.1
Int 5										
Volume (vph)	11.9	6.0	7.8	7.1	8.6	5.6	3.8	3.3	8.9	6.1
Delay Avg (spv)	2.1	2.3	2.1	1.7	92.1	20.6	2.0	0.5	9.2	2.3
Delay Max (spv)	9.9	7.6	8.8	8.3	118.5	34.8	8.6	6.7	29.3	8.1
Q Avg (v)	0.4	0.4	0.3	0.3	7.8	1.5	0.2	0.0	1.2	0.2
Q Max (v)	1.6	0.9	0.5	0.0	9.0	3.2	1.3	0.0	3.0	1.5
Int 6										
Volume (vph)	15.3	10.0	4.3	5.0	2.2	3.8	6.5	2.5	16.8	14.3
Delay Avg (spv)	0.8	0.9	0.5	0.5	45.1	21.1	1.6	0.8	5.4	2.1
Delay Max (spv)	4.1	5.7	0.9	1.5	105.2	31.6	7.7	7.2	12.5	8.0
Q Avg (v)	0.2	0.2	0.1	0.1	4.3	1.5	0.2	0.0	0.8	0.2
Q Max (v)	1.1	0.5	0.7	0.0	8.2	2.7	1.1	0.4	1.8	1.2
Int 7										
Volume (vph)	2.3	6.4	10.9	6.4	3.4	6.8	8.2	8.7	4.4	5.4
Delay Avg (spv)	0.8	0.9	1.2	0.8	13.2	52.5	4.6	3.3	1.9	1.1
Delay Max (spv)	5.0	4.7	3.6	4.6	31.2	82.4	5.8	11.8	8.3	5.9
Q Avg (v)	0.2	0.1	0.1	0.1	1.0	3.7	0.7	0.3	0.3	0.1
Q Max (v)	1.1	0.7	1.5	1.1	1.5	5.0	1.6	0.8	1.1	0.0
Int 8										
Volume (vph)	11.3	7.6	5.1	6.1	7.0	10.7	11.5	14.7	2.7	1.3
Delay Avg (spv)	1.3	0.9	1.6	1.2	19.3	50.2	4.1	2.2	1.1	1.0
Delay Max (spv)	7.2	5.5	7.3	7.8	42.5	73.5	12.7	5.9	4.6	6.7
Q Avg (v)	0.2	0.1	0.3	0.2	1.6	4.2	0.6	0.2	0.1	0.1
Q Max (v)	1.5	1.1	1.4	1.4	3.6	5.7	1.0	1.6	0.5	0.0
Int 9										
Volume (vph)	8.5	6.0	6.2	4.5	5.2	1.5	6.6	4.9	4.8	4.5
Delay Avg (spv)	0.9	0.4	0.5	0.4	29.4	20.3	7.9	1.2	0.5	1.0
Delay Max (spv)	7.2	4.1	2.5	2.6	44.0	27.9	20.0	7.6	9.2	6.9
Q Avg (v)	0.1	0.1	0.1	0.1	2.3	1.5	1.0	0.1	0.1	0.1
Q Max (v)	0.8	0.8	0.9	0.5	5.3	2.1	3.2	1.1	0.7	0.4

REFERENCES

Abu-Lebdeh G. and Benekohal, R. F., Dynamic Signal Coordination along Oversaturated Arterials. Proc. Traffic Congestion and Traffic Safety in the 21s' Century, Challenges Innovations, and Opportunities. Urban Transportation Division, ASCE. Highway Division, ASCE. Chicago, Illinois, June 8-11, 1997.

Albright, D., Elements of Success: TRANSIMS Our National Laboratories and Transportation Research, New Mexico State Highway and Transportation Department Research Bureau, March 1997, Report No. FHWA-HPR-NM-91-13.

Athanailos, E. G. The Integration of the Highway Capacity Manual and the TRAF-NETSIM Simulation Model, Institute of Transportation Engineers Journal, May 1994, pp. 33 - 38.

Bretherton, D., Current Developments in SCOOT: Version 3. Transportation Research Record 1554, TRB, National Research Council, Washington, D.C., 1996, pp. 48 - 52.

Bretherton, D., Wood, K. and Raha, N., Traffic Monitoring and Congestion Management in the SCOOT Urban Traffic Control System. Transportation Research Record 1634, TRB, National Research Council, Washington, D.C., 1998. pp. 118 - 122.

Bretherton, R. D. and Bowen, G. T., Recent Enhancements to SCOOT - SCOOT Version 2.4. Third International Conference on Road Traffic Control. Computing and Control Division of the Institution of Electrical Engineers. London. May 1990 pp 95 - 98.

Brown, D., An Introduction to Object-Oriented Analysis, Objects in Plain English. Northern Alberta Institute of Technology, John Wiley and Sons, inc, 1997, pp. 700.

Bullen, A,G.R., Norman, H., Bryer, R. and Nekmat, R. EVIPAS: A Computer Model for the Optimal Design of a Vehicle-Actuated Traffic Signal, Transportation Research Record I Lee, Grayson, and Copeland, 1977, Washington, D.C. pp 103-110.

Bullock, D. and Catarella, A., A Real Time Simulation Environment for Evaluating Traffic Signal Systems. Transportation Research Record 1634, TRB, National Research Council, Washington, D.C., 1998. pp. 130 - 135.

Bullock, D., Implementation Vision for Distributed Control of Traffic Signal Subsystems. Transportation Research Record 1554, TRB, National Research Council, Washington, D.C., 1996. pp. 43 - 47.

Chang, E.C., Guidelines for Actuated Controllers in Coordinated Systems, Transportation Research Record 1554, TRB, National Research Council, Washington, D.C., 1991. pp. 61 - 73.

Chang, E. C., and Koothrappally, J., Field Verification of Coordinated Actuated Control, Transportation Research Record 1456, TRB, National Research Council, Washington, D.C., 1995. pp. 83 - 90.

Chang, G. Vasudevan, M. and Su, C., Bus Preemption Under Adaptive Signal Control Environments. Transportation Research Record 1494, TRB, National Research Council, Washington, D.C., 1995. pp. 146 - 154.

Change, G. and Williams, J. Estimation of Independence of Vehicle Arrivals at Signalized Intersections: A Modeling Methodology, Transportation Research Record, 1194, Washington D.C. 1988.

Chaudhary, N. A. and Messer C. J., PASSER IV: A Program for Optimizing Signal Timing in Grid Networks. Transportation Research Record 1421, TRB, National Research Council, Washington, D.C., 1993. pp. 82 - 93.

Chaudhary, N. A., Pinnoi, A. and Messer, C. J.. Proposed Enhancements to MAXBAND 86 Program. Transportation Research Record 1324, TRB, National Research Council, Washington, D.C., 1991. pp. 98 - 104.

Chen, H., Cohen, S. L., Gartner, N. H., and Liu, C.C., Simulation Study of OPAC: A Demand-Responsive Strategy for Traffic Signal Control, in Gartner and Wilson (eds.) Proceedings of the Tenth International Symposium on Transportation and Traffic Theory, Elsevier, pp. 233-249 (1987)

Cohen, S. L., Concurrent Use of MAXBAND and TRANSYT Signal Timing Programs for Arterial Signal Optimization. Transportation Research Record 906, TRB, National Research Council, Washington, D.C., 1983. pp. 81 - 84.

Conrad, M., Dion, F. and Yagar, S., Real-Time Traffic Signal Optimization with Transit Priority, Recent Advances in the Signal Priority Procedure for Optimization in Real-Time Model, Transportation Research Record 1634, TRB, National Research Council, Washington, D.C., 1998. pp. 100 -109.

Dell'Olmo, P. and Mirchandani P. B., A Model for Real-Time Traffic Coordination Using Simulation Based Optimization. Advanced Methods in Transportation Analysis. Springer-Verlag, Germany, pp. 525-546.

Dell'Olmo P. and Mirchandani, P. B., REALBAND: An Approach for Real-Time Coordination of Traffic Flows on Networks. Transportation Research Record 1494, TRB, National Research Council, Washington, D.C., 1991. pp. 106 - Chang, Vasudevan, and Su, 1995.

Denny, R. W. and Chase, M. J., True Distributed Processing in Modular Traffic Signal Systems - San Antonio Downtown System. Transportation Research Record 1324, TRB, National Research Council, Washington, D.C., 1991. pp. 130 - 136.

Dreyfus, S. E. and Law, A. M., The Art and Theory of Dynamic Programming, Mathematics in Science and Engineering, A Series in Monographs and Textbooks, Vol. 130, Academic Press Inc. New York, ISBN 0-12-221860-4.

Elahi, S. M., Radwan, A. E. and Goul, K. M., Traffic Signal Using Mixed Controller Operations. Journal of Transportation Engineering, Vol. 118, No. 6, November/December, 1992. pp. 866 - 880.

Elahi, S. M., Radwan, A. E. and Goul, K. M., Knowledge-Based System for Adaptive Traffic Signal Control. Transportation Research Record 1324, TRB, National Research Council, Washington, D.C., 1991. pp. 115 - 122.

Farges, J. L., Khoudor, L. and Lesort, J. B., PRODYN: On Site Evaluation. Third International Conference on Road Traffic Control. Computing and Control Division of the Institution of Electrical Engineers. London. May 1990 pp 62-66.

Findler, N. V. and Stapp, J., Distributed Approach to Optimized Control of Street Traffic Signals. Journal of Transportation Engineering, Vol. 118, No. l, January/February, 1992. pp. 99 - 110.

Gal-Tzar, A., Mahalel, D. and Prashker, J. N., Decision Support System for Controlling Traffic Signals. Transportation Research Record 1421, TRB, National Research Council, Washington, D.C., 1993. pp. 69 - 75.

Gartner, N. H. and Al-Malik, M., Combined Model for Signal Control and Route Choice in Urban Traffic Networks. Transportation Research Record 1454, TRB, National Research Council, Washington, D.C., 1996. pp. 27 - 35.

Gartner, N. H. and Stamatiadis, C., Integration of Dynamic Traffic Assignment with Real-Time Traffic Adaptive Control System. Transportation Research Record 1644, TRB, National Research Council, Washington, D.C., 1998. pp. 150 - 164.

Gartner, N. H., Assmann, S. F. Lasaga, F., and Hou, D. L., A Multi-Band Approach to Arterial Traffic Signal Optimization. Transportation Research B, Vol. 25B, No 1. 1991. pp. 55 - 74.

Gartner, N. H., Tarnoff, P. J., and Andrews, C.M., Evaluation of Optimized Policies for Adaptive Control Strategy, Transportation Research Record 1324, TRB, National Research Council, Washington, D.C., 1991, pp. 105-Lee, Grayson, and Copeland, 1977.

Gartner, N. H., Stamatiadis, C., and Tarnoff, P. J., Development of Advanced Traffic Signal Control Strategies for Intelligent Transportation Systems: MultiLevel Design, Transportation Research Record 1494, TRB, National Research Council, Washington, D.C., 1995, pp. 98-105.

Gartner, N. H., Todd, K., Prescription for Demand-Responsive Urban Traffic Control, Transportation Research Record 1494, TRB, National Research Council, Washington, D.C., 1995, pp. 98-105.

Gordon, R. L., Reiss, R. A., Haenel, H., Case, E. R., French, R. L., Mohaddes, A. and Wolcott, R. Traffic Control and Systems Handbook. Department of Transportation, Federal Highway Administration, Report No. FHWA-SA-95-032. February, 1996.

Gulewicz, V. and Danko, J. A Simulation Based Approach to Evaluating Optimal Toll Plaza Lane Staffing Requirements, Presented at the Transportation Research Board 74 `h Annual Meeting, Washington D.C. Preprint 950196.

Hadi, M. A. and Wallace, C. E. Optimization of Signal Phasing and Timing Using Cauchy Simulated Annealing. Transportation Research Record 1456, TRB, National Research Council, Washington, D.C., 1994. pp. 64 - 71.

Hadi, M. A. and Wallace, C. E., Hybrid Genetic Algorithm To Optimize Signal Phasing and Timing. Transportation Research Record 1683, TRB, National Research Council, Washington, D.C., 1998. pp. 104 - 112.

Hagen, L. Signal Timing Using TRANSYT-7 f, McTrans, Dec 1995, pp 2

Han, B. and Yagar, S. A Procedure For Real-Time Signal Control that Considers Transit Interference and Priority. Transportation Research Part B. Vol. 28B, No. 4, August 1994. pp. 315 - 331.

Head, K. L., Mirchandani, P. B. and Shelby, S. The RHODES Prototype: A Description and Some Results. pp. 12

Head, K. L. An Event-Based Short-Term Traffic Flow Prediction Model. Transportation Research Record 1510, TRB, National Research Council, Washington, D.C., 1998. pp. 45 - 52.

Helali, K.N. and Khan, A. M. Macroscopic Delay Models for TSM Planners, Institute of Transportation Journal, June 1994, pp. 42-46.

Highway Capacity Manual, Transportation Research Board HCM 2000, TRB National Research Council, Washington, D.C., HE336.H48 H54 2000

Hounsel, N. B., McLoed, F. N. and Burton. P. SCOOT: A Traffic Database. Third International Conference on Road Traffic Control. Computing and Control Division of the Institution of Electrical Engineers. London. May 1990 pp 99-103.

Hunt, P. B., Robertson, D. L, Bretherton, R. D., and Winton, R. L, SCOOT - A Traffic Responsive Method of Coordinating Signals, Transport and Road Research Laboratory, TRRL Laboratory Report 1014, Crowthorne, Berkshire, 1981, ISSN 0305-1293.

Karaboga, D. and Pham, D.T., Intelligent Optimization Techniques, Genetic Algorithms, Tabu Search, Simulated Annealing and Neural Networks. Springer-Verlag London Limited 2000. Printed in Great Britain. 1998. pp 258.

Kelton, W. D., Sadowski, R. P., Sadowski, D. A., Simulation With ARENA, McGraw-Hill, QA76.9.C95 K45,1998

KLDAssociates, http://www.kldassociates.comiwatsim.htm#Features, WATSim website, March 2002.

Kwon, E. and Stephanedes, Y. J., Development of an Adaptive Control Strategy in a Live Intersection Laboratory. Transportation Research Record 1634, TRB, National Research Council, Washington, D.C., 1998. pp. 123 - 129.

Law, A. M., Kelton, W. D., Simulation Modeling and Analysis, Second Edition, McGraw-Hill, QA76.9.C67L38, 1991

Lee, C.E., Rioux, T.W., Copeland, C.R., The TEXAS Model for Intersection Traffic - Development, Center for Transportation Research, December 1977, Report No. FHWATX78 - 184 - 1.

Lee, C.E., Grayson, G.E., Copeland, C.R., Miller, J.W., Riouw, T.W., Savur, V.S., The TEXAS Model for Intersection Traffic - User's Guide, Center for Transportation Reseach, July 1977, Report No. FHWATX78-184-3.

Lee, C. and Machemehl, R. B., Local and Iterative Searches for Combined Signal Control and Assignment Problem, Implementation and Numerical Examples. Transportation Research Record 1683, TRB, National Research Council, Washington, D.C., 1998. pp. 102 - 109.

Lee, S., Hazelton, M., Stochastic Optimisation of Combined Traffic Assignment and Signal Control Junction Modeling. Transportation and Traffic Theory. Proc. of 13`h International Symposium on Trans. and Traffic Theory. Lyon, France, July 1996. pp. 713 - 735.

Li, M. and Gan, A.C. Signal Timing Optimization for Oversaturated Networks Using TRANSYT-7 f Transportation Research Record 1683, Washington D.C., 1999, pp. 118 - 126.

Lin, F. and Vijayakumar, S., Adaptive Control at Isolated Intersections. Journal of Transportation Engineering, Vol. Lee, Grayson, and Copeland, 1977, No. 5, September 1988. pp. 555 - 573.

Lin, F., Knowledge Base on Semi-Actuated Traffic-Signal Control, Journal of Transportation Engineering, Vol. 117, No. 4, July/August 1991, pp. 398-417.

Linkenheld, J. S., and Benekohal, R. H., and Garrett Jr. J. H., Knowledge-Based System for Design of Signalized Intersections. Journal of Transportation Engineering, Vol. 118, No. 2, March/April, 1992. pp. 241 - 257.

List, G. and Troutbeck, R. Advancing the Frontier of Simulation as a Capacity and Quality of Service Tool, Department of Civil Engineering, Rensselaer Polytechnic Institute, Troy , N.Y. Unpublished paper as of 1999.

List, G. and Mitchell, B., Incremental Benefits and Costs from Signal Timing Control Hardware Upgrades, Final Report., The New York State Energy Research and Development Authority, Contract 1033-EEED-AEP-88, June 30, 1993.

Liu Chang, Chaudhary, N.A., Simeonidis H. C., and Sirigiri, S. Pioneer Application of Passer IV in the Houston Metro-RCTSS Project, Transportation Research Record 1494, Washington D.C., pp. 129-145

Logic, D. M. W., Computer-Aided Design and Evaluation of Traffic Systems, Proceedings of the International Symposium on Traffic Control Systems, Institute of Transportation Studies and Federal Highway Administration, Volume 2D - Analysis and Evaluation, Berkeley Cal. pp. 161-183.

Los Alamos National Laboratory, TRANSIMS: Transportation Analysis Simulation System, Version: TRANSIMS - 3.0, Volume Three - Modules, March 2002, 1992, Los Alamos National Laboratory, LA-UR-00-1725.

Luk, J. Y. K., Two Traffic-Responsive Area Traffic Control Methods: SCAT and SCOOT. Traffic Engineering and Control. Vol. 25, No. l. January 1984. pp. 14 - 18.

Mahmassani, H,, DYNASMART-P, Intelligent Transportation Network Planning tool, Powered by Dynamic Traffic Assignment Technology, http://www.dynasmart.umd.edu/dynasmartp/concept methodology.html.

Malakapalli, M. P. and Messer, C. J. Enhancements to the PASSER II-90 Delay Estimation Procedures, Transportation Research Record, 1421, Washington D.C. 1993.

Massoumi, R., Luba, C. W. and Menaker, P. WATsim Micro-Simulation: I-780/I-6801I-80 Corridor, Proceed. Traffic Congestion and Traffic Safety in the 215` Century, Challenges, Innovations, and Opportunities, American Society of Civil Engineers, New York, New York, pp 90 - 98.

May, A. D., Traffic Flow Fundamentals. Prentice Hall, Englewood Cliffs, New Jersey 07632, 1990. pp. 464

Memon, G. Q, and Bullen A. G. R., Multivariate Optimization Strategies for Real-Time Traffic Control. Transportation Research Record 1554, TRB, National Research Council, Washington, D.C., 1996. pp. 36 - 42.

Mirchandani, P. and Head, L., Rhodes: A Real-Time Traffic Signal Control System: Architecture, Algorithms and Analysis. Proceeding Tristan III. San Juan Puerto Rico, June 17-23, 1998. Vo12. pp. 15.

Morales, J.M. Improving Traffic Signal Operations, A Primer, U.S. Department of Transportation, Federal Highway Administration, 1995.

Mystkowski, C. and Sarosh K. Estimating Queue Lenghts by Using SIGNAL94, SYNCHR03, TRANSYT-7f, PASSER II-90, and CORSIM, Transportation Research Record 1683, Washington D.C. 1999, pp 110 - 117.

Nakatsuji, T. and Kaku, T., Development of a Self-Orginizing Traffic Control System Using Neural Network Models. Transportation Research Record 1324, TRB, National Research Council, Washington, D.C., 1991. pp. 137 -145.

Newell, G. F., Theory of Highway Traffic Signals, University of California at Berkeley Institute of Transportation Studies, Course Notes UCB-ITS-CN-89-1

Nguyen, V. N., Evaluation of SCATSIM-RTA Adaptive Traffic Network Simulation Model. Transportation Research Record 1566, TRB, National Research Council, Washington, D.C., 1996. pp. 8 - 19.

Owen, L.E. and Stallard C. M., Rule Based Approach to Real-Time Distributed Adaptive Signal Control. Transportation Research Record 1683, TRB, National Research Council, Washington, D.C., 1999. pp. 95 - 101.

Owen, L.E., Stallard, C. M., and Steiger, J., Evaluation of the Near North Chicago RT-TRACS Field test, Prepared By ITT industries, FHWA Travel Management R&D, Oct 5. 2001

Papola, N. and Fusco, G., Maximal Bandwidth Problems: A New Algorithm Based on the Properties of Periodicity of the System. Transportation Research B, Vol. 32, No 4. 1998. pp. 277 - 288.

Papola, N, and Fusco, G., Relationship Between Travel Time Minimization and Bandwidth Maximization Problems. Proceedings Tristan III. San Juan Puerto Rico, June 17-13, 1998, Vol. 2. pp. 11.

Park, B., Messer, C. J. and Urbank II, T., Initial Evaluations of New TRANSYT-7f Version 8.1 Program. Transportation Research Record 1683, TRB, National Research Council, Washington, D.C., 1999. pp. 127 - 132.

Park, B., Messer, C. J. and Urbank II, T., Traffic Signal Optimization for Oversaturated Conditions, Genetic Algorithm Approach, Transportation Research Record 1683, TRB, National Research Council, Washington, D.C., 1999. pp. 133 - 141.

Park, B., Messer, C. J. and Urbank II, T., Enhanced Genetic Algorithm for Signal Timing Optimization of Oversaturated Intersections, Paper Presented at 79`h Annual Meeting of the of the Transportation Research Board. Washington, D.C., January 2000. pp. 20

Peck, C. and Gorton P. T. W., The Application of SCOOT in Developing Countries. Third International Conference on Road Traffic Control. Computing and Control Division of the Institution of Electrical Engineers. London. May 1990 pp 104 - 109.

Pegden, C. D, Shannon, E. R, Sadowski, R. P., Introduction to Simulation using SIMAN, Second Edition, McGraw-Hill, 1995

Peirce, J. R. and Webb, P. J., MOVA Control of Isolated Traffic Signals - Recent Experience. Third International Conference on Road Traffic Control. Computing and Control Division of the Institution of Electrical Engineers. London. May 1990 pp 110 - 113.

Peterson, A., Torsdten, B. and Steen, K., LHOVRA - A New Traffic Signal Control Strategy for Isolated Junctions. Second International Conference on Road Traffic Control. Computing and Control Division of the Institution of Electrical Engineers. April , 1986. pp. 98 - 101

Pillai, R. S., Rathi, A. K. and Cohen, S. L., A Restricted Branch and Bound Approach for Generating Maximum Bandwidth Signal Timing Plans for Traffic Networks. Transportation Research B, Vol. 32, No 8. 1998. pp. 517-529.

Pooran, F. J., Tarnoff, P. J., Kalaputapu, R., RT TRACS: Development of the Real-Time Control Logic.

Rathi, A. and Santiago, A., The new NETSIM simulation program, Traffic Engineering and Control, Vol. 31 No. 5, May 1990, pp. 317- 320.

Rakha, H. A. and Van Aerde, M. W. Comparison of Simulation Modules of TRANSYT and INTEGRATION Models, Transportation Research Record 1566, Washington D.C, 1996, pp 1-7.

Rakha, H. and Van Aerde, M., REALTRAN: An Off-Line Emulator for Estimating the Effects of SCOOT. Transportation Research Record 1494, TRB, National Research Council, Washington, D.C., 1995. pp. 124 - 128.

Relijic, S., Multicriteria Optimal Control in a Network of Signalized Intersections. Transportation and Traffic Theory. Proc. of 13`h International Symposium on Trans. and Traffic Theory. Lyon, France, July 1996. pp. 615 - 627.

Robertson, D. L, and Bretherton, R. D., Optimizing Networks of Traffic Signals in Real Time - The SCOOT Method, IEEE Transactions on Vehicular Technology, Vol. 40 No. l, February 1991, pp. 11-15.

Rogness, R. O. and Messer, C. J., Heuristic Programming Approach to Arterial Signal Timing. Transportation Research Record 906, TRB, National Research Council, Washington, D.C., 1983. pp. 67 - 75.

Sabra, Z. A. and Stockfish, C. R. Computerized Signal Timing Techniques: States of the Practice, McTrans, June 1994, pp. 2-5.

Schaffer, S. J. and LaRue, W. W., BONeS DESIGNER: A Graphical Environment for Discrete-Event Modeling and Simulation, Proceed. of the Second International Workshop on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, IEEE Computer Society, Durham North Carolina, 1994

Schriber, T. J. "The Nature and Role of Simulation in the Design of Manufacturing Systems," in Simulation in CIM and Artificial Intelligence Techniques

Sen, S. and Head, K. L., Controlled Optimization of Phases at an Intersection, Transportation Science, Institute for Operations Research and the Management Sciences, Vol. 31, No. 1, Feb 1997, pp. 5-17.

Sims, A. G. and Dobinson, K. W., S.C.A.T. The Sydney Co-ordinated Adaptive Traffic System Philosophy and Benefits. Proceedings of the International Symposium on Traffic Control Systems. Volume 2B. Institute of Transportation Studies and Federal Highway Administration. Univ. of Cal. Berkeley, Dec. 1979 pp. 19 - 42

Shoup, G. E. and Bullock, D., Dynamic Offset Tuning Procedure Using Travel Time Data. Transportation Research Record 1683, TRB, National Research Council, Washington, D.C., 1999. pp. 84 - 94.

Skabardonis, A., Determination of Timings in Signal Systems with Traffic-Actuated Controllers, Transportation Research Record 1554, TRB, National Research Council, Washington, D.C., 1991. pp. 18-26.

Skabardonis, A., Gallagher, B. R. and Patel, K. P. Determining Capacity Benefits of Real-Time Signal Control at an Intersection. Transportation Research Record 1683, TRB, National Research Council, Washington, D.C., 1999. pp. 78 - 83.

Skabardonis, A., Bertini, R. L. and Gallagher, R., Development and Application of Control Strategies for Signalized Intersections in Coordinated Systems. Transportation Research Record 1634, TRB, National Research Council, Washington, D.C., 1998. pp. 110 - 117.

Sripathi, H. K., Gartner, N. H., and Stamatiadis, C., Uniform and Variable Bandwidth Arterial Progression Schemes, . Transportation Research Record 1494, TRB, National Research Council, Washington, D.C., 1995. pp. 106 -- Chang, Vasudevan, and Su, 1995.

Stamatiadis, C. and Gartner, N. H., MULTIBAND-96: A Program for Variable-Bandwidth Progression Optimization of Multiarterial Traffic Networks. Transportation Research Record 1554, TRB, National Research Council, Washington, D.C., 1991. pp. 9 - 17.

Stewart, J. A. and Van Aerde, M., An Assessment of Adaptive Co-Ordination of Traffic Signal Offsets Within Integration. Paper Presented at Annual Meeting of the of the Transportation Research Board. Washington, D.C., pp. 23.

Sunkari, S. R., Beasley, P. S., Urbanik II, T. and Fambro, D. B., Model to Evaluate the Impacts of Bus Priority On Signalized Intersections, Transportation Research Record 1494, TRB, National Research Council, Washington, D.C., 1995. pp. 117 - 123.

Tarnoff P. J. and Gartner, N., Real-Time, Traffic Adaptive Signal Control.

U.S. Department of Transportation, Improving Traffic Signal Operations, a Primer. U.S.D.O.T. Federal Highway Administration. Prepared by the Institute of Transportation Engineers under a grant from the Federal Highway Administration, U.S. Department of Transportation, 1995. pp. 16

Van Aerde, M., and Hellinga, B., An Overview of a simulation study of the Highway 401 freeway traffic management system, Canadian Journal of Civil Engineering, Vo121, 1994, pp 439-454.

Van Aerde, M. and Yager, S., Dynamic Integrated Freeway / Traffic Signal Networks: Problems and Proposed Solutions, Transportation Research Record Part A, Vol 22A, No. 6, 1988, pp 435-443.

Vincent, R. A. and Young, C. P., Self-Optimising Traffic Signal Control Using Microprocessors - The TRRL "MOVA" Strategy for Isolated Intersections. Second International Conference on Road Traffic Control. Computing and Control Division of the Institution of Electrical Engineers. April, 1986. pp. 102 - 105

VISSIM Traffic/Transit Simulation Model, ITC-VISSIM, http:/jwww.itc-world.`vissim.htrn, Software Summary, 2003.

Vincent, R. A. and Young, C. P., Self-Optimising Traffic Signal Control Using Microprocessors - The TRRL "MOVA" Strategy for Isolated Intersections. Traffic Engineering and Control. Vol. 27, No. 7/8, July/August 1986. pp. 385-387.

Wallace, C.E. and Courage, K.G., Arterial Progression - New Design Approach, Transportation Research Record 1494, TRB, National Research Council, Washington, D.C., 1995, pp. 53-58.

Wang, Y., and Prevedouros, P. D. Comparison of INTEGRATION, TSIS/CORSIM, and WATsim in Replicating volumes and Speeds on Three Small Networks, Transportation Research Record 1644, Washington D.C., 1998, pp. 80 - 92.

Wolshon, B. and Taylor, W. C., Impact of Adaptive Signal Control on Major and Minor Approach Delay. Journal of Transportation Engineering, Vol. 125, No. l, January/February, 1999. pp. 30 - 38.

Wong, S., TRAF-NETSIM: How it Works, What it Does, Institute of Transportation Engineers Journal, Volume 60, No. 4, April 1990, pp. 22-27.

Yagar, S. and Dion, F. Distributed Approach to Real-Time Control of Complex Signalized Networks. Transportation Research Record 1554, TRB, National Research Council, Washington, D.C., 1996. pp. 1 - 8.

Yu, L., Platoon Dispersion and Calibration Under Advanced Traffic Control Strategies. Proc. Traffic Congestion and Traffic Safety in the 21" Century, Challenges Innovations, and Opportunities. Urban Transportation Division, ASCE. Highway Division, ASCE. Chicago, Illinois, June 8-11, 1997. pp. 507-513

Zeigler, B.P. Theory of Modeling and Simulation, John Wiley and Sons, Inc., New York, NY. 1976