Safety and Operations of Large-Area Rural/Urban Intermodal Systems

SOLARIS

A Tier 1 university transportation center led by the University of Nevada, Reno.

Final Report

June 2016

INTEGRATING MESO- AND MICRO-SIMULATION MODELS TO EVALUATE TRAFFIC MANAGEMENT STRATEGIES – Year 1

SOLARIS Consortium, Tier 1 University Transportation Center
Center for Advanced Transportation Education and Research
Department of Civil and Environmental Engineering
University of Nevada, Reno
Reno, NV 89557

Pitu B. Mirchandani, Sc.D., Principal Investigator
Pengfei Li, Ph.D., Program Manager
Xuesong Zhou, Ph.D., Co-Principal Investigator
Ira A. Fulton Schools of Engineering
Arizona State University
Tempe, AZ 85281

DISCLAIMER:

The contents of this report reflect the views of the authors, who are responsible for the facts and accuracy of the information presented herein. This document is disseminated under the sponsorship of the U.S. Department of Transportation's University Transportation Centers Program, in the interest of information exchange. The U.S. Government assumes no liability for the contents or use thereof.

EXECUTIVE SUMMARY

In this project we developed a hierarchical multi-resolution traffic simulation system for metropolitan areas, referred to as MetroSim. Categorically, we focus on integrating two types of simulation: microscopic simulation in which individual vehicle's behaviors and vehicles' interactions are simulated with high fidelity; mesoscopic simulation (a dynamic traffic assignment (DTA) type simulator) in which individual vehicle's behaviors and vehicles' interactions are simplified. It has been noticed that some drawbacks exist with the DTA-type simulators that are used to simulate large-scale urban traffic networks where traffic signal systems play an essential role. Specifically, DTA-type simulators can hardly emulate a traffic signal control mechanism close to the reality. Instead, in most cases, the DTA-type simulators approximate signal control effects at intersections just by reducing the adjacent link capacities or increasing the corresponding link travel times. The lack of high-fidelity traffic signal control in DTA simulator may bring significant bias to the simulation results. At the other end of the spectrum, microscopic simulator cannot be applied to large-scale simulation because of the demand on computing resources necessary to run the simulation model. To address the aforementioned issues and provide a flexible simulation platform applicable to both planning projects and operations analysis, (in particular, to develop and evaluate innovative adaptive signal control strategies for large networks), the ASU research team developed a prototype of multi-resolution traffic simulation platform, *MetroSim*.

Keywords: Key words: Multi-resolution simulation, Traffic operations, Dynamic traffic assignment, Traffic signal systems, Traffic simulation, Regional Traffic Management

List of Figures

Table of Contents

1. Development of Multi-resolution Simulation Platform

1.1 Introduction

The concept of *MetroSim* was first proposed in 2014 by the ASU project team. We presented a framework for a hierarchical multi-resolution traffic simulation system for metropolitan areas. Most multi-resolution traffic simulation systems are designed as single-level and different simulators represent different portions within the same network. By contrast, MetroSim has two hierarchical levels including an open-source mesoscopic traffic simulator (Level One) for regional dynamic traffic assignment, DTALite, and a popular commercial microscopic traffic simulator (Level Two), PTV VISSIM, for intersection and corridor traffic operations. Unlike the traditional network modeling method, MetroSim starts from the microscopic simulation network and then develops into a mesoscopic network aggregating lane capacities into link capacities while reserving all the high-fidelity details, such as geometric design or signal head locations. DTALite network and VISSIM network represent two independent but consistent networks and that exchange data while both simulation engines are running to overcome some long-standing issues in other single-level systems, including the consistency of network representation and route choice, computing efficiency and precisely simulating some subareas of special analysis interest. The MetroSim platform aims to provide insights on the development of multi-resolution systems take advantage of parallel computing, and multi-resolution data integration. For more details of the concept, the readers are suggested to refer to the related publication in the appendix, which was one of three finalists is the TRB network modeling committee paper award.

In light of the above concept, the ASU researchers developed the prototype of the MetroSim platform containing both adaptive ramp metering and adaptive signal control. The purpose of MetroSim project is twofold: (1) proof of concept to examine if the proposed architecture is realistically possible for the practice; and (2) provide a tool to evaluate the individual and joint impact of adaptive signal control and adaptive ramp metering which are much needed in many cities where urban freeways is a major part of traffic networks. More specifically, the ASU researchers maintain an adaptive signal control system, RHODES, as well as an adaptive ramp metering system, MILOS. in the first year of the MetroSim Project, RHODES was fully integrated. Figure 1 shows the architecture of MetroSim Platform. Extensive evaluation of adaptive signal control and adaptive ramp metering will be conducted in the second year of this project. This report only discusses the architecture of Metrosim and the interfacing of its various components.

1.2 Metrosim Arcitecture

Figure 1 shows there are three major components in MetroSim: DTALite/NEXTA, the high-definition DTA simulator, ASC3 signal control emulator (in VISSIM) with or without

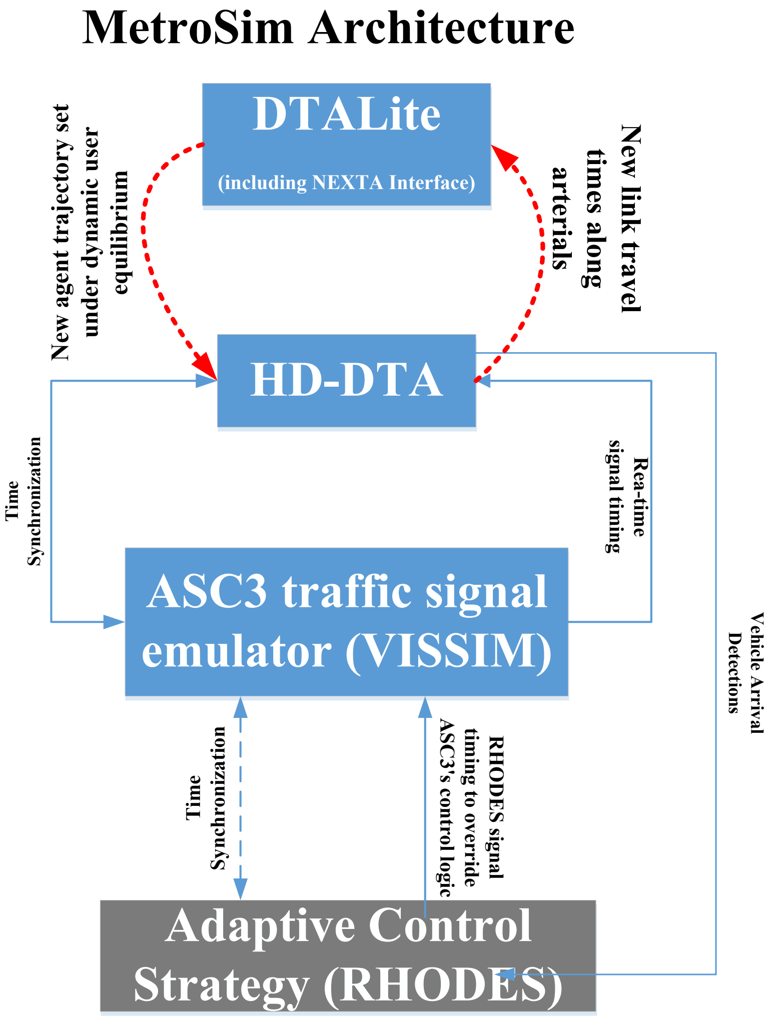

MetroSim Architecture
DTALite
(including NEXTA Interface)
New agent trajectory set under dynamic user equilibrium
New link travel times along arterials
HD-DTA
Time Synchronization
Rea-time signal timing
ASC3 traffic signal emulator (VISSIM)
Vehicle Arrival Detections
Time Synchronization
RHODES signal timing to override ASC3's control logic
Adaptive Control Strategy (RHODES)

Figure 1 Architecture of Metrosim Setup for Simulating Adaptive Control

external control. In the figure an external component is shown, for example adaptive control system (in this case RHODES). The HD-DTA simulates traffic propagation similar to DTALite/NexTA but with more detailed mechanisms at the intersections. To realize traffic signal control, some links within intersections in the HD-DTA network are set as capacity-changeable external high-fidelity signal control emulators. Those links are referred to as signal links; the microscopic VISSIM provides high-fidelity signal emulator (ASC/3). A mapping between the signal links in HD-DTA and signal phases in VISSIM is established in the Metrosim framework. In other words, whenever the ASC/3 changes the signal timing, the corresponding signal links in HD-DTA will be set as open or closed as well. In this way, all the high-fidelity signal control mechanism in ASC/3 can be reflected in HD-DTA simulator. The challenge of coupling VISSIM and HD-DTA is time synchronization and so a special synchronous connection is established between HD-DTA and VISSIM to synchronize the clock while two simulators are launching.

RHODES system was setup was enhanced to be able to work with any NTCIP-compliant traffic signal controllers. A special version of RHODES was also developed to retrieve traffic signal status while VISSIM is running. In the meantime, in this version of RHODES also obtains vehicle detections generated according to the traffic propagation in HD-DTA. Then RHODES make signal decision based on the incoming signal status and detector calls to override the inherent signal timing in ASC/3. These new RHODES timings are also fed back into HD-DTA via the path “RHODES→ ASC/3 (VISSIM)→ Signal Links (HD-DTA)” to reflect adaptive signal control mechanism.

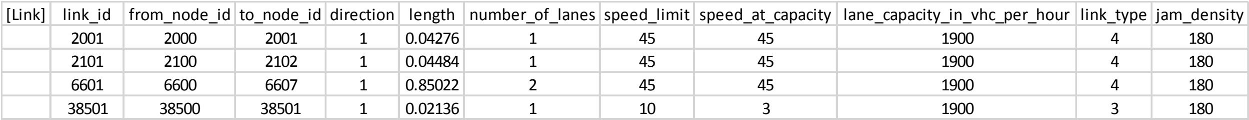
2. Data Structure and Input Files for Multi-Resolution Simulation Platform

In this section, the major data structure and input files for MetroSim are explained in details. Some new data structures and input files were developed to fulfill the research needs, such as how to couple different simulators or how to synchronize simulation time among simulators with different fidelities, etc.

2.1 Input files for DTALite/HD-DTA simulators

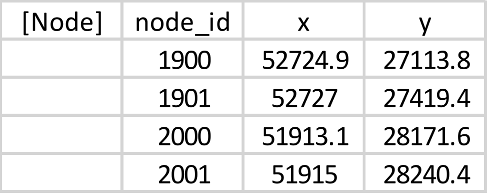
There are four input files for DTALite/HD-DTA simulators: Input_link.csv, input_node.csv, input_signal.csv, input_detector.csv. As illustrated in Figure 2, input_link.csv and input_node.csv defines the traffic network; input_signal.csv defines the mapping relationship between traffic signal phases in ASC/3 and signal links/stop-bar links in DTA simulators. It also specifies the IP address and port number to retrieve real-time phase information from the populated ASC/3 signal emulator in VISSIM.

Input_detector.csv defines the number and locations of detectors in the HD-DTA simulator. Whenever a DTA agent is about to enter a link, whether and when it will reach a detector on that link will be calculated according to the detector’s distance from the link’s from node. This detector will be then scheduled to be activated while the clock moves forward. Such detector call events will be eventually sent the signal strategy being tested to determine new traffic signal timings.



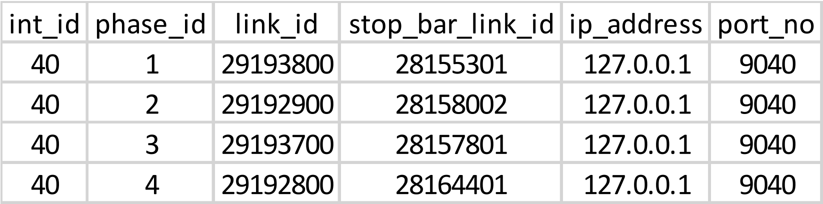
[Link]	link_id	from_node_id	to_node_id	direction	length	number_of_lanes	speed_limit	speed_at_capacity	lane_capacity_in_vhc_per_hour	link_type	jam_density
	2001	2000	2001	1	0.04276	1	45	45	1900	4	180
	2101	2100	2102	1	0.04484	1	45	45	1900	4	180
	6601	6600	6607	1	0.85022	2	45	45	1900	4	180
	38501	38500	38501	1	0.02136	1	10	3	1900	3	180

Figure 2-A input_link.csv



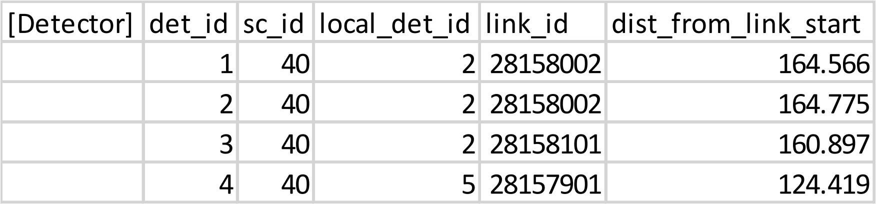
[Node]	node_id	x	y
	1900	52724.9	27113.8
	1901	52727	27419.4
	2000	51913.1	28171.6
	2001	51915	28240.4

Figure 2-B input_node.csv



int_id	phase_id	link_id	stop_bar_link_id	ip_address	port_no
40	1	29193800	28155301	127.0.0.1	9040
40	2	29192900	28158002	127.0.0.1	9040
40	3	29193700	28157801	127.0.0.1	9040
40	4	29192800	28164401	127.0.0.1	9040

Figure 2-C input_signal.csv



[Detector]	det_id	sc_id	local_det_id	link_id	dist_from_link_start
	1	40	2	28158002	164.566
	2	40	2	28158002	164.775
	3	40	2	28158101	160.897
	4	40	5	28157901	124.419

Figure 2-D input_detector.csv

2.2 Interfacing Data Structures to Couple Various Simulators/emulators

There are several coupling links within METROSIM to synchronize simulation clocks and establish real-time data exchange. The following explains how those coupling links are set up:

Link 1: Time synchronization between HD-DTA and ASC/3 (VISSIM)

Using any of APIs provided in VISSIM, such as signal API, driving behavior API or emission API, it is possible to open a synchronous connection and continuously listen to any connections while VISSIM is launching. In the meantime, HD-DTA also populates a synchronous port to couple with VISSIM while launching. Through correct configuration, at each simulated second, HD-DTA need to correctly connect and communicate with VISSIM in order to proceed both HD-DTA and VISSIM. Through this synchronous connection, the clock synchronization is achieved.

Link 2: Real-time signal timing exchange between HD-DTA and ASC/3 (VISSIM)

The latest version of ASC/3 in VISSIM also includes a fully functional communication module like in the real hardware ASC/3 signal controller. This new feature makes efficiently exacting real-time signal status in ASC/3 using external programs possible. Taking advantages of this feature, at each time stamp, the HD-DTA collects signal status from ASC/3 using NTCIP commands and then translate them into the open or close status of the corresponding signal links. If a signal link is open, then vehicles are allowed to enter intersections whereas if a signal link is close, then vehicles will have to wait at stop lines the signal link is re-opened.

Link 3: Time synchronization between ASC/3 (VISSIM) and an external signal control module (e.g., RHODES)

There are additional challenges to synchronize the clocks between ASC/3 (VISSIM) and any external signal control such as RHODES; since there might be many externally controlled intersections. If we set up an independent synchronous connection for each intersection, the communication overheads might significantly slow down the simulation speed. To address this issue, a different solution was adopted. From preliminary experiments, it was found out that microscopic simulation engine, VISSIM, is almost always slower than RHODES' speed. In other words, most of time RHODES optimization routines have to wait for VISSIM to finish its current simulation step to proceed. This phenomenon provides us with the possibility of setting up an asynchronous server to broadcast VISSIM's simulation step and RHODES does not proceed until it is notified to do so.

Specifically, any APIs provided in VISSIM can be used to establish a separate connection to broadcast the current simulation step. In the RHODES experiments, all RHODES-controlled intersections continuously monitor that broadcast simulation time to decide if it's ready to proceed. In this way, the clock synchronization is established between VISSIM and all RHODES routines.

Link 4: Data Exchange between external controllers like RHODES and ASC/3 and HD-DTA

In general, RHODES needs to two data sources to fulfill its optimization task: the on-going traffic signal status and the newly incoming vehicle detections. Via NTCIP commands, RHODES retrieves real-time traffic signal status from ASC/3 (VISSIM) which is also the on-going signal timing in HD-DTA simulation. On the other hand, RHODES sets up another data exchange link with HD-DTA. As described before, HD-DTA schedules detector calls according to agent movements and detector configurations on certain links. At each time step, the HD-DTA sends all detector events reaching the scheduled time at that time step to the corresponding RHODES controller. RHODES then translates the incoming detector calls into vehicle arrivals on various approaches to estimate the queue lengths of left-turn, through and right-turn on each approach.

Link 5: Data Exchange between HD-DTA and DTALite

At this time, the DTALite for adaptive ramp metering and HD-DTA for adaptive traffic signal control are not coupled through automated data exchange. Instead, the data exchange between DTALite and HD-DTA is established through manual file exchange. Specifically, in each iteration, DTALite provides a new agent trajectory set satisfying the dynamic user equilibrium. The agent trajectory set is loaded into HD-DTA network to create new travel demand along the arterial. Since HD-DTA can provide high-fidelity signal control mechanism and the resulting travel time along the arterial is likely to be changed (mostly reduced in our experiments). At the end of that simulation run, the HD-DTA generates updated link travel times along the arterial. Then some of the link travel times in DTALite are updated based on the HD-DTA output and a new user equilibrium will be reached and a new set of agent trajectories will be created as well. This process is iteratively manually repeated between DTALite and HD-DTA until certain equilibrium tolerance is satisfied.

Appendix: Publication on the concept of MetroSim under this project

MetroSim: A Hierarchical Multi-Resolution Traffic Simulator for Metropolitan Areas: Architecture, Challenges and Solutions

(Authors' Copy)

Pengfei Li, Ph.D., P.ENG
Arizona State University, BYENG 553, 699 S. Mill Ave., Tempe, AZ, 85281
Tel: (480)965/1051
Email: pengfeili@asu.edu

Pitu Mirchandani, Ph.D.
Arizona State University, BYENG 553, 699 S. Mill Ave., Tempe, AZ, 85281
Tel: (520)5911/882
Email: pitu@asu.edu

Xuesong Zhou, Ph.D.
Associate Professor, School of Sustainable Engineering and the Built Environment
Arizona State University, ISTB2 201D, Tempe, AZ, 85281
Email: xuesong.zhou@asu.edu

Reference: Li, Pengfei, Pitu Mirchandani, and Xuesong Zhou. "Hierarchical Multiresolution Traffic Simulator for Metropolitan Areas: Architecture, Challenges, and Solutions." Transportation Research Record: Journal of the Transportation Research Board 2497 (2015): 63-72.

Abstract:

This paper presents a hierarchical multi-resolution traffic simulation system for metropolitan areas, referred to as MetroSim. Most multi-resolution traffic simulation systems are designed as single-level and different simulators represent different portions within the same network. By contrast, MetroSim has two hierarchical levels including an open-source mesoscopic traffic simulator (Level One) for regional dynamic traffic assignment, DTALite, and a popular commercial microscopic traffic simulator (Level Two), PTV VISSIM, for intersection and corridor traffic operations. Unlike the traditional network modeling method, MetroSim starts from the microscopic simulation network and migrates into the mesoscopic network while reserving all the high-fidelity details, such as geometric design or signal head locations. DTALite network and VISSIM network represent two independent but highly consistent networks and they exchange data while both simulation engines are running to overcome some long-standing issues in other single-level systems, including the consistency of network representation and route choice, computing efficiency and precisely simulating the subareas with the greatest interest. This paper aims to provide insights on the development of multi-resolution systems and focuses on topics as network migration, parallel computing, inter-process data integration. Finally, a case study is conducted to demonstrate the potential applications of MetroSim.

Key words: Multi-resolution simulation, Traffic operations, Dynamic traffic assignment, Traffic signal systems, Traffic simulation, Regional Traffic Management

1. Introduction

In the last three decades, simulation tools have increasingly become commonplace in transportation research and practice. Although many traffic simulators have been developed for different purposes, coupling different traffic simulators is never easy. Some successful stories of multi-resolution traffic simulation development have been reported but most literature provides insights for the end users and therefore focuses primarily on the insights to model calibration and validation, data analysis, etc. By contrast, this paper aims to serve simulator developers and researchers, and thus focuses on the challenges in developing such systems. The challenges include how to improve the computing efficiency, how to guarantee robust data exchange between different simulators, how to synchronize different simulators' clocks and how to improve the expandability of MetroSim for additional traffic simulators in the future.

Our vision is that MetroSim and its concept will be a common ground for both planners and engineers and it will fill the gap between planning and engineering activities. If planners wish to better review the traffic operations within certain local areas to identify the reasons for bottlenecks, they can just "zoom in" to the corresponding part of the microscopic traffic networks. At the other end of the spectrum, if engineers wish to better understand the impact of their local operations on the overall metropolitan region, all they need to do is to "zoom out" and review the mesoscopic traffic networks.

MetroSim includes two traffic simulators from different categories of traffic simulations, mesoscopic traffic simulation and microscopic traffic simulation. On the metropolitan level, it is important to understand the time-varying network traffic loading between different origin-destination (O-D) pairs under different traffic management, also referred to as dynamic traffic assignment (DTA). The DTA problems are usually analyzed through mesoscopic simulation models and the resulting traffic assignments, or path-volumes, are derived using the simulated link travel times and shortest path finding algorithm(s). The selected mesoscopic DTA simulator for MetroSim is DTALite which has a sound theoretical foundation, proven simulation performance as well as certain unique features to greatly boost super large-scale DTA simulating speed. DTALite is also highly-customizable without changing the kernel simulation engine. Customization can also be modulized. In the meantime, other DTA tools, such as DYNASMART and DynusT, can also be used if needed.

On the link and corridor level, it is important to understand the possible impact of different traffic control measures, such as traffic signal control, managed lanes or ramp metering and how the local changes will propagate through the whole metropolitan network. Toward that goal, it is necessary to simulate the traffic control measures with a high fidelity which can be achieved only in microscopic traffic simulators. The selected microscopic simulator for MetroSim is PTV VISSIM which provides sufficient application programming interfaces (APIs) for users to develop various add-on programs even though it is proprietary software.

The rest of this paper is organized as follows: literature on multi-solution traffic simulation is reviewed in Section 2 followed by the significance of the research in Section 3; In Section 4, the system architecture of MetroSim will be introduced in details; In Section 5, the challenges we encountered during the development and our solutions are described; In Section 6, a case study is conducted to demonstrate the potentials of MetroSim in solving some long-standing traffic problems.

2. Literature Review

In the last decade, both industry and research community have been dedicating efforts to integrating different traffic simulation engines to overcome the drawbacks of individual traffic simulators. For example, the mesoscopic traffic simulation engines often have difficulty in simulating the complex traffic control mechanism and driver behaviors in urban areas whereas the microscopic traffic simulation engines alone often fail to address the impact of improved local traffic on the whole networks over time. According to the mechanism of data exchange and system architecture, the multi-resolution traffic simulation systems can be divided into two categories, on-line multi-resolution traffic simulator (ONLMRTS) and off-line multi-resolution traffic simulation (OFLMRTS). FIG. 1 demonstrates the concepts of these two categories.

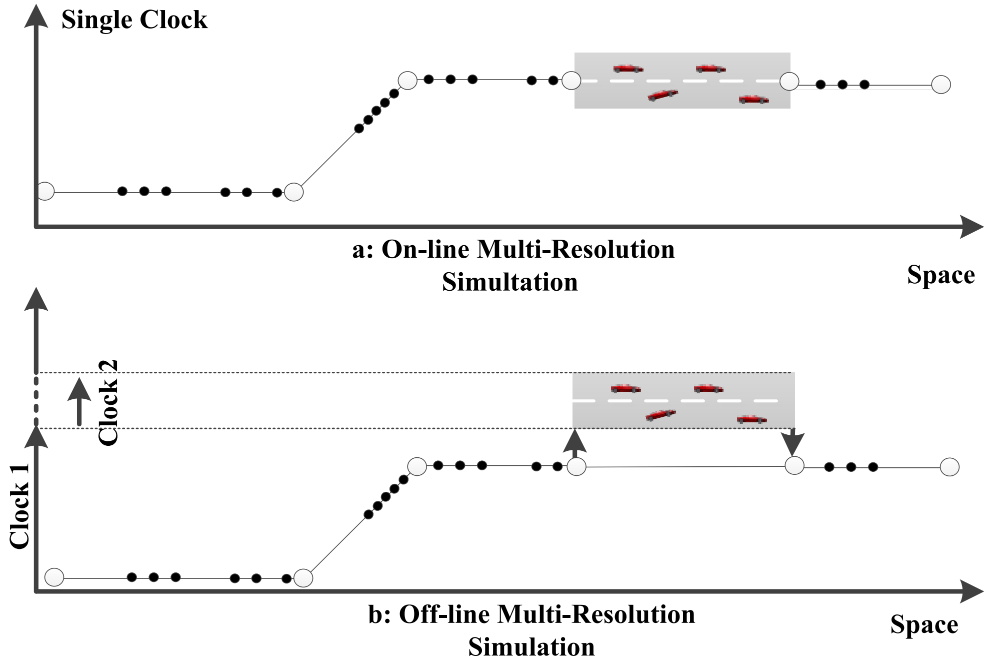

Single Clock
a: On-line Multi-Resolution Simultation
Space
Clock 2
Clock 1
b: Off-line Multi-Resolution Simulation
Space

Figure 1 Demonstration of On-Line Multi-Resolution Simulation and Off-line Multi-Resolution Simulation

ONLMRTS means that multiple traffic simulators represent different sections of the same road network. Those sections of the greatest interest are simulated with microscopic traffic simulation whereas the remaining part of the network is simulated with fast mesoscopic or macroscopic simulations. The synchronized computing clock is determined by the slowest simulator. Harmonizing ONLMRTS system has several major challenges including how to guarantee the consistency of the road network represented by different simulators (e.g., link-based v.s. lane-based), the consistency of vehicles' route choices across simulators, the consistency of traffic dynamics at the boundaries and the consistency of the traffic performance across different simulators. In addition, the efficiencies of computing and data exchange are also pains to system developers. As examples, Burghout et al. developed an ONLMRTS simulator, MiMe, by integrating a mesoscopic traffic simulator, Mezzo and a microscopic Mitsim *(1)*. Some mainstream traffic simulation packages also include the features of on-line multi-resolution, such as Aimsun *(2)* and TransModeler *(3)*. Those commercial software can naturally use the same network and conduct a "hybrid" simulation. Key areas are simulated with high-fidelity car-following and lane-changing models whereas in the other areas, individual vehicle's movements are represented according to aggregated macroscopic traffic models to achieve a faster simulation speed.

The OFLMRTS means that different simulators run iteratively. As shown in FIG.1, the lane-based microscopic network is the same as the corresponding section of the overall mesoscopic link-based network, just with more details. When the mesoscopic simulator runs for a while and generate the necessary data, the mesoscopic simulator will be suspended and will send all the newly generated data down to the microscopic simulator. After the microscopic simulator runs for a while and generate another set of data, it will be suspended and it will send all the data up to the mesoscopic simulator. Then the mesoscopic simulator will resume running based on the latest data from the microscopic simulator and this process continues. Examples of OFLMRS include the commercial VISUM/VISSIM package, which integrates microscopic, mesoscopic and macroscopic simulation products of the software company, PTV AG *(4)*. Chiu and his research team at the University of Arizona have also developed an off-line multi-resolution simulator by integrating their mesoscopic DTA software, DynusT, with the microscopic VISSIM *(5)*.

In theory ONLMRTS is more desirable than OFLMRTS. However, developing and maintaining ONLMRS has many more challenges than OFLMRTS, especially when traffic simulators are not from the same developer. As such, we see both ONLMRTS and OFLMRTS in practice.

3. Significance of the Research

Most ONLMRTS has a single-level architecture. It means the partner simulators represent different portion of the same road network while simulation is running. Such a single-level architecture can bring some problems. First, since all the partner simulators work on the same road network, their computing technology must be compatible. For instance, if one simulator uses the parallel computing to boost the simulating speed, the other partner simulators must be significantly revised to fit this change. In practice, this is nearly impossible. Second, since the network representations and underlying traffic models are quite different from simulator to simulator, it is very difficult to guarantee consistency when vehicles are travelling across simulators. As a result, calibration and validation of ONLMRTS models are often too tricky to form a standard procedure. Third, although traffic signal control systems play a key role in determining the link travel times in metropolitan areas, high-fidelity signal control mechanism can be only simulated in microscopic simulators and a vast majority of mesoscopic road networks rely on oversimplified signal control methods, making the simulation outputs questionable to practitioners.

MetroSim recast the ONLMRTS architecture from single-level to hierarchical. Under the framework of MetroSim, the mesoscopic simulator (DTALite) and microscopic simulator (VISSIM) use two independent but identical networks. Those two independent networks are integrated through synchronous data exchange. One advantage of such hierarchical architecture is that two simulators are totally independent and therefore one simulator can apply any new computing technologies without interfering the partner simulator. In the meantime, even though the MetroSim is ONLMRTS, the hierarchical design will allow us only to address the challenges in OFLMRTS. Second, the high-fidelity signal control mechanism can be realized in DTALite through mapping the signal phase status (green or not) in VISSIM to the link capacities of the corresponding links in DTALite (full capacity or zero). DTALite will also update its traffic assignment partially based on the signal status and notify VISSIM to use the updated travel demand periodically. This feature is particularly useful in simulating metropolitan areas.

4. System Architecture of MetroSim

FIG.2 demonstrates the architecture and network representation of MetroSim. For the network representation, microscopic simulation networks in general are lane-based and contain (require) more details than those link-based mesoscopic simulation networks. Directly trimming and converting a portion of a mesoscopic network into the microscopic network often cannot meet the minimal requirements for road geometry in microscopic simulation. In the meantime, the VISSIM models have been developed in many locations and it would not be difficult to combine them to create a metropolitan network. As such, unlike most other ONLMRTS which construct a microscopic traffic network from its "parent" mesoscopic network, constructing networks in MetroSim starts from VISSIM and the VISSIM network is converted to the open-source DTALite networks through an automated tool developed by the authors.

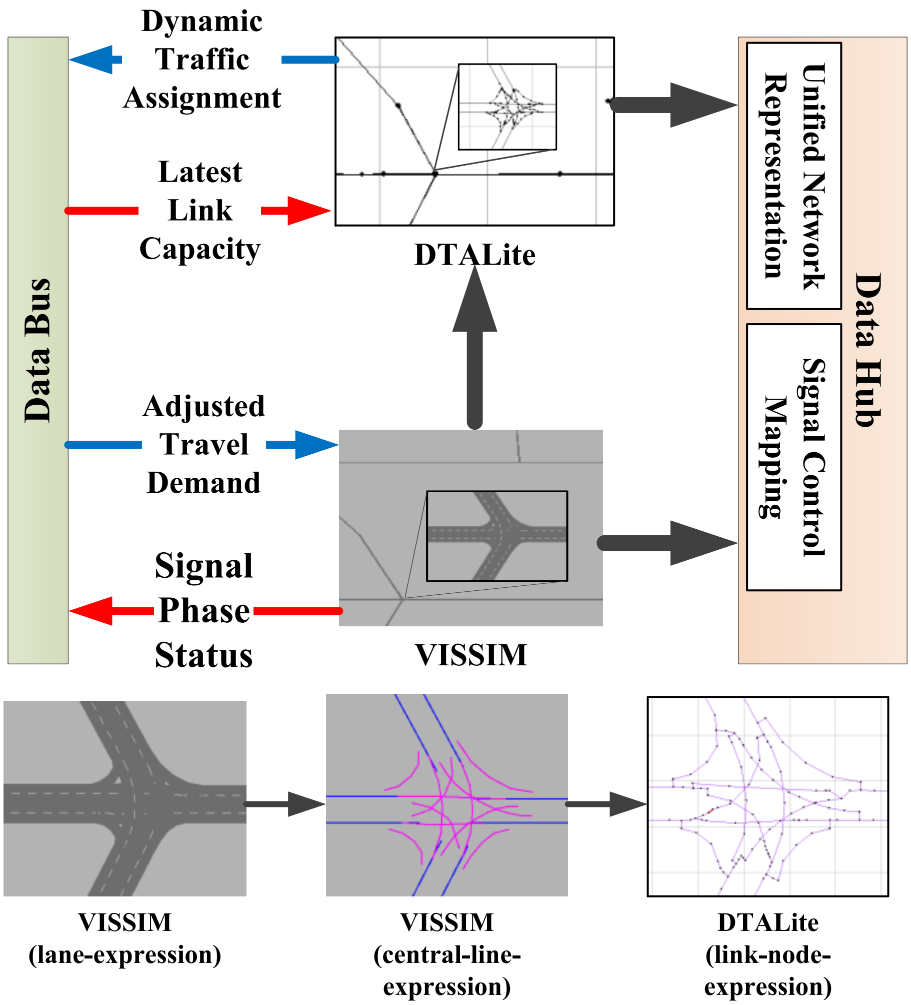


Figure 2 System Architecture of MetroSim and Network Representation

When MetroSim is running, both VISSIM and DTALite run continuously while their computing clocks are synchronized through a file exchange method (described later). Every six seconds, the VISSIM network sends the latest status of all the signal groups to the Data Bus. The latest signal status is then interpreted into the capacities of the corresponding links in DTALite. DTALite will calculate the link travel times and traffic dynamics to estimate and predict the time-varying traffic assignment. Every fifteen minutes, DTALite will send the latest traffic assignment to the Data Bus and it is then interpreted into the new travel demand in VISSIM. In the meantime, some en-route vehicles in VISSIM will also adjust their paths to respond to the latest local travel demand, reflecting a fact that drivers nowadays increasingly use certain mobile computing devices to monitor the latest traffic condition while driving.

The high-fidelity traffic signal system in VISSIM may respond and adjust their signal timings accordingly. As for the model calibration and validation, it can be carried out in multiple ways, depending upon the available data, project purposes. As examples, for the projects of signal optimization, after VISSIM model is calibrated, DTALite should use VISSIM's outputs (movement counts, link travel time) as the "observed data" to calibrate; for those projects of regional traffic control strategies, after DTALite is calibrated, VISSIM should be calibrated to have consistent link travel times, link capacities and vehicle counts with DTALite. The version of VISSIM is V5.40 with software-in-the-loop signal emulator for this project.

5. Challenges and Solutions

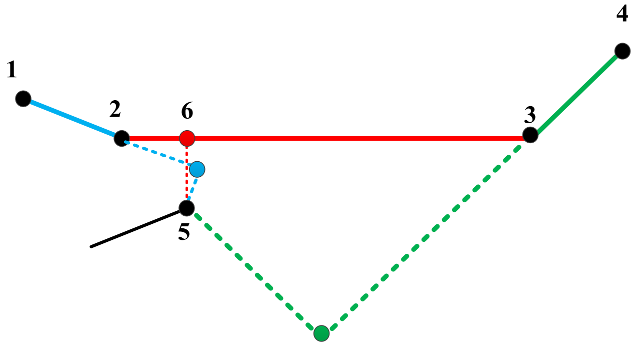
5.1 Development of Unified Road Networks for VISSIM to DTALite

It is not realistic to build two identical networks for VISSIM and DTALite manually and therefore an automated tool was developed, referred to as *VIS2DTA_NET*, to migrate the VISSIM network to DTALite network. There are two major challenges in this task: converting a lane-based network to a link-based network, creating link IDs in DTALite which can be mapped in VISSIM.

Links in VISSIM has width and can be curved while links in DTALite have no width and have to be lines. These features requires that one link in VISSIM has to be broken into several links to create curves and additional links must be created to guarantee the path connectivity (shown in FIG. 2). This task is critical in path finding and traffic assignment. Although the link-node-expression in DTALite is more complex than the usual mesoscopic simulator, it is necessary to break a link in VISSIM into several small links to DTALite to identify where a VISSIM connector should be connected to the regular links. Otherwise, newly added links to connect links may become too long to be realistic.

VIS2DTA_NET has two steps while converting a VISSIM network to a DTALite network:

- Step 1: Identify the coordinates of end and any intermediate points of VISSIM links, break VISSIM links into multiple smaller straight links to reflect the curves, and then reformat it following the DTALite's requirements for network construction;
- Step 2: Add additional links in DTALite. VIS2DTA_NET converts the VISSIM networks to DTALite networks according to the central lines of VISSIM links. As a result, ends of VISSIM links and connectors may not overlap. To address this issue, it is necessary to identify the end points of all connectors in VISSIM and create additional links to guarantee the connectivity. For each connector, VIS2DTA_NET will make vertical projection on the slope of each segment of the target link. If the intersecting point is within the scope of two end points of a link segment, then the connector is considered connecting to that link segment. FIG. 3 shows an example of DTALite network, the vertically projected Point 6 from Point 5 is within the scope of Point 2 and Point 3 (on link 1_2_3_4) and the additional link will be created from Point 5 to Point 6 in DTALite to ensure the proper connectivity. The lengths of such additional links are typically equal to one or two lane width (at most 7 meters) which can be ignored when DTALite is calculating the path travel times.



-

- Figure 3 Demonstration of how to connect VISSIM connectors to VISSIM links in DTALite

There is another important consideration for migrating a VISSIM network to a DTALite network: a rule for link numbering. Although DTALite is very flexible in numbering links and nodes, it is important to following certain rules so that the path volumes from the DTALite networks can be recognized and appropriately translated back to VISSIM. In general, a VISSIM link can be broken into at most 100 smaller links, while breaking a VISSIM link into multiple DTALite links, VIS2DTA_NET numbers the new DTALite links as:

$$Link\_ID_{DTALite} = 100 * Link\_ID_{VISSIM} + i \tag{1}$$

Where: *i* is the i^{th} road segment of the VISSIM link

In case when a new point is inserted between points (Point 6 in FIG. 3), the existing DTALite link ID is reserved for the first new segment (Point 2 to Point 6) and the second new road segment (Point 6 to Point 3) is numbered as $Link\_ID_{DTALite} + 50$.

Such a numbering rule will guarantee that whenever DTALite returns a path with the volume to VISSIM, it can be translated back to the correct path in VISSIM. For instance, if the returned DTALite path is “100001->100051->6701->5801”, then the corresponding path in VISSIM will be “1000->67->68”.

5.2 Signal Control Mapping Across Microscopic and Mesoscopic Simulator

Like many other mesoscopic traffic simulators, DTALite can only simulate simple signal control strategies. Adding a new dimension of advanced signal control system into DTALite will greatly increase the complexity of the simulation engine. On the other hands, many municipalities have installed advanced traffic signal control systems and they are playing important roles in urban areas. As such, ignoring the advanced features of traffic signal systems will make the simulation results questionable. MetroSim addresses this issue by mapping the link capacities in DTALite with the signal status in VISSIM. Such design will be able to bring any advanced signal control systems into DTALite without changing the core simulation engine.

Specifically, the VIS2DTA_NET will identify the link id for each signal phase in VISSIM and create a map like "***Control_phase***->***Link_id***". This information will be saved as a text file and DTALite will read this file into the memory while MetroSim is launched. While MetroSim is running, VISSIM sends the latest signal states periodically to DTALite and DTALite will adjust the corresponding links capacity as "full capacity" (if the signal is green), "half capacity" (if the phase is permissive green) or "zero" (if the signal is yellow or red). The traffic dynamics and traffic states in DTALite will be changed over time which in turn will change the next traffic re-assignment in DTALite.

5.3 Time-varying path volumes

Time-varying path volumes, or dynamic network loading, is the main purpose of most mesoscopic DTA simulators. In DTALite, drivers choose a route based on the traveling cost between a pair of origin and destination. This is based upon three components: travel time, value of time (VOT), and tolling/pricing, if any.

$$\text{Cost} = \text{Travel Time} \times \text{VOT} + \text{Toll} \qquad (2)$$

In DTALite, vehicles are generated according to the time-varying origin-destination matrix. Through some preliminary simulation runs, departing vehicles can choose their routes (e.g., time-dependent shortest path) according to the historical link travel costs as well as the link travel costs experienced in last fifteen minutes by the drivers. The link travel cost can be changed by traffic management measures, such as pricing, work zone or adjusted signal control systems. The traffic dynamics (e.g., shockwave or queue) is estimated following the triangle-shape fundamental diagram and linearized car-following model proposed by Newell *(6, 7)*. For more details, readers are suggested to read the related documents of DTALite*(8)*.

The general form of DTALite package will run continuously and generate the simulation outputs at the end of each simulation run. In MetroSim, the DTALite is customized to read the signal status from VISSIM every six seconds. While the simulation is moving forward, the changing link capacity will affect the shockwaves and queue lengths on that link and eventually the link/path travel times will be changed. Every fifteen minutes, DTALite will recalculate the traffic assignment using the time-dependent shortest path algorithms. This network-wide traffic re-assignment will also affect the path traffic volumes in VISSIM over time. Whenever DTALite has new re-assigned volumes, the values will be updated in VISSIM through Component Object Model (COM) technique without stopping the VISSIM simulation engine.

5.4 En-route Agent Path Changing Mechanism

Nowadays, increasing number of drivers are using mobile computing devices to monitor real-time traffic states while driving. In addition, road-side equipment, such as variable message system (VMS), can guide drivers to other routes if the road ahead is congested or has incidents. It is becoming common for drivers to change their original routes to minimize the travel cost while on the roads. Therefore it is necessary to enable drivers to adjust paths after the vehicles have been released into networks.

In DTALite, such a feature has been coded as a standard function of the agent-based simulation engine. Every minutes, the en-route drivers can re-calculate their time-dependent shortest path according to the latest traffic states and determine their new routes. In DTALite, the default value is about 10% en-route vehicles may use their mobile computing devices to adjust their paths.

On the VISSIM side, the 10% of vehicles with mobile computing devices can be configured through the traffic composition input. On the level of microscopic simulation, drivers' en-route route choice is a complex process in that different drivers may have different preferences of switching routes. Some so called "stubborn" drivers may stick to their original routes regardless the real-time information whereas the other drivers would possibly like to change routes even for a small saving of travel times. To reflect these facts, the well-known discrete choice model, the Logit model *(9)*, is chosen to simulate the process of switching routes.

For each en-route driver having the real-time traffic information, the en-route route switching is divided into two steps in VISSIM: (1) find the possible alternative routes from the current location to the destination; and (2) determine the route choice according to the latest route volumes and Logit Model.

Step One: Identify the alternative routes of each en-route vehicle with mobile computing devices

En-route vehicles' alternative routes are identified according to their origin, destination and their current locations. For example, there are five possible routes from A to B:

$$\begin{aligned}
&\text{Route 1: } 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 11 \rightarrow 13 \\
&\text{Route 2: } 1 \rightarrow 2 \rightarrow 6 \rightarrow 11 \rightarrow 12 \rightarrow 13 \\
&\text{Route 3: } 1 \rightarrow 5 \rightarrow 11 \rightarrow 12 \rightarrow 13 \\
&\text{Route 4: } 1 \rightarrow 7 \rightarrow 9 \rightarrow 10 \rightarrow 12 \rightarrow 13 \\
&\text{Route 5: } 1 \rightarrow 7 \rightarrow 8 \rightarrow 11 \rightarrow 12 \rightarrow 13
\end{aligned}$$

If a vehicle's current link is Link 2 and its current route is Route 1, then its possible alternate route is Route 2 and while the other three other routes are not possible.

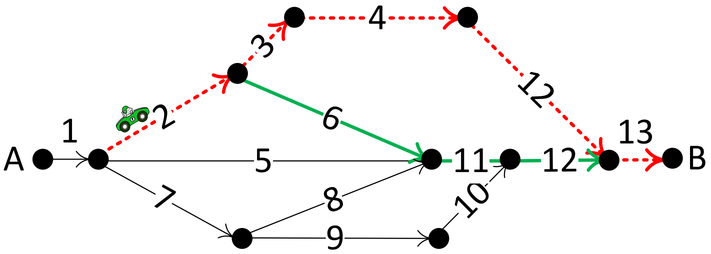


Figure 4 Demonstration of how to find the alternative routes for en-route vehicles

The algorithm of finding alternative routes is formulated as:

```
For each vehicle $i$ in all vehicles having real-time information
    1. Identify i's origin, destination and current link;
        2. Search all the routes (A) with the same origin and destination;
        3. For each route in A
            3.1 identify if the current link is one of their component
            3.2 If YES, this route is labeled as the "Alternate Route";
            3.3 If NO, go to next route;
        3.4 If reach the end of A, terminate;
```

Step Two: Estimate the probability for a vehicle to switch route

DTALite re-assigns traffic for the near future according to the latest travelling cost (e.g., total travel time), it will assign more volumes on those routes with less cost. When such future path volumes are translated into VISSIM, VISSIM should also re-assign the path volumes accordingly. Based on the Logit Model, the process for en-route drivers to switch routes is formulated as follows:

1. For an en-route vehicle having real-time traveler information, every 15 minutes, the driver will search possible alternative routes and examine the latest path volumes. If none of the alternative routes has the arbitrary 110% of the volume of the vehicle's current path, then the vehicle will stick to its current path. The use of 110% is to make sure the new routes must be more attractive enough for drivers to switch. Please note that vehicles in VISSIM are assumed to have no personal preferences and just following a general route-choice model.
2. If there are alternative routes with 110% or higher volumes than the current path, the top two routes with the highest volume will be chosen for further evaluation.
3. Normalize the top two volumes ($q_{\max 1}, q_{\max 2}$) and the current volume (q_c) as:

$$n_1 = \frac{q_{\max 1}}{q_{\max 1}} = 1, n_2 = \frac{q_{\max 2}}{q_{\max 1}}, n_c = \frac{q_c}{q_{\max 1}} \tag{3}$$

4. The probability of switching to other routes or sticking to the current routes is calculated as:

$$p_1 = \frac{e^{n_1}}{e^{n_1} + e^{n_2} + e^{n_c}}, p_2 = \frac{e^{n_2}}{e^{n_1} + e^{n_2} + e^{n_c}}, p_c = \frac{e^{n_c}}{e^{n_1} + e^{n_2} + e^{n_c}} \tag{4}$$

According to Eq. (3) ~ Eq. (4), the more volumes on the alternative routes than the current volume, the more likely those vehicles on the current path should be reassigned to a better alternative route. However, route switching will not occur for certain even if there are better routes. These configurations reflect the fact that some drivers would not like to change their routes once they depart.

5.5 Model Calibration and Convergence

While calibration of VISSIM can be fulfilled using commonly accepted indicators, such as turning movements, link travel times. We can also adjust the vehicles' headway, standstill distance, etc. in its underlying car-following model to change the road capacity and jam density to match with the corresponding settings in DTALite. Challenge of the MetroSim calibration and convergence is on the DTALite side. DTALite is a dynamic network loading (DNL) model based on Newell's kinematic wave theory to capture congestion phenomena and shock wave propagation in the process of solving dynamic traffic assignment problems. Given sensor data (i.e. observed link flows and derived densities according to link flows and speeds) and target (aggregated historical) OD demands, the proposed time-dependent path flow estimation model can be represented with a nonlinear programming problem using the path flows r(w, τ, p), ∀w, τ, p and least path travel times π = {π(w,τ), ∀ w, τ}, as the decision variables. Denote *c = {c (w,τ, p), ∀w, τ, p}, q = {q(l, t), ∀l, t} and k = {k(l, t), ∀l, t}.* Mathematically, the problem can be formulated as:

Min $$Z = \beta_d \sum_w \left[\sum_{\tau \in H_d} \sum_p r(w,\tau,p) - \bar{d}(w)\right]^2 + \sum_{l \in S} \sum_{t \in H_o} \left\{\square_q [q(l,t) - \bar{q}(l,t)]^2 + \square_k \left[k(l,t) - \bar{k}(l,t)\right]^2\right\} \qquad (5)$$

Subject to

$$(\boldsymbol{c}, \boldsymbol{q}, \boldsymbol{k}) = DNLF(\boldsymbol{r}), \qquad (6)$$

$$g(\boldsymbol{r}, \boldsymbol{\pi}) = \Sigma_w \Sigma_\tau \Sigma_p \{r(w,\tau, p)[c(w,\tau, p) - \pi(w,\tau)]\} = 0, \qquad (7)$$

$$c(w,\tau, p) - \pi(w,\tau) \geq 0,\ \forall w,\ \tau,\ p, \qquad (8)$$

$$\pi(w,\tau) \geq 0,\ \forall p \in \mathrm{P}(w,\ \tau),\ \forall w,\ \tau \qquad (9)$$

$$r(w,\tau, p) \geq 0,\ \forall w,\ \tau,\ p. \qquad (10)$$

where:

- DNLF: *Dynamic Network Loading Function*
- *A*: set of links
- *W*: set of OD pairs

- P: set of paths
- S: set of links with sensors, $S \subseteq A$
- H_d: set of discretized departure time intervals
- H_o: set of discretized observation time intervals
- t: index of simulation time intervals, $t = 0,\ldots,T$.
- τ: index of departure time intervals, $\tau \in H_d$
- w: index of OD pairs, $w \in W$
- p: index of paths for each OD pair, $p \in P$
- l: index of links, $l \in A$
- $\bar{q}(l,t)$: observed number of vehicles passing through an upstream detector on link l during observation interval t
- $\bar{k}(l,t)$: observed density on link l during observation interval t
- $\bar{d}(w)$: target demand, which is the total traffic demand for OD pair w over a planning horizon
- $r(w, \tau, p)$: estimated path flow on path p of OD pair w and departure time interval τ
- $c(w, \tau, p)$: estimated path travel time on path p of OD pair w and departure time interval τ
- $\pi(w, \tau)$: estimated least path travel time of OD pair w and departure time interval τ
- $\mathrm{P}(w, \tau)$: a set of paths as set of paths for OD pair w at departure time interval τ
- $q(l, t)$: estimated number of vehicles passing through an upstream detector on link l during observation interval t
- $k(l, t)$: estimated density on link l during observation interval t
- $d(w, \tau)$: estimated demand of OD pair w and departure time interval τ
- ▯: weighting factors

The solution to this problem in DTALite is a heuristic method based on Lagrangian relaxation and sub-gradient method. Through iterations, the low bound provided through the Lagrangian relaxation and subgradient method; the upper bound provided from the simulation will be closer and closer. The iteration will eventually stop either after an arbitrarily determined number of iterations is reached or a stopping criterion is satisfied. For more details, readers are suggested to refer to the literature*(8, 10)*.

5.6 Simulator Synchronization

In MetroSim, synchronization is achieved through the file exchange. In general, the method of file exchange is slower than those memory-based methods. However, the latest technology, such as the solid state hard drive, has already enabled the file exchange on the hard drive to be almost the same fast as in computer memories. Whenever a partner simulator generates new data for the other simulator, it will create a text file with the current simulation as the suffix. For instance, every six seconds, VISSIM will create a new file to report the latest signal states as "DEX_VIS_RT_xx.csv". (xx stands for 6, 12, 18,…) Whenever DTALite moves to the same time step, it will seek the corresponding "DEX_VIS_RT_xx.csv" to update the corresponding link capacity. Once DTALite finishes this action, it will create another file named as "DEX_DTA_RT_xx.csv". In the meantime, VISSIM also keeps seeking the "DEX_DTA_RT_xx.csv" file and will not move forward until that file is found. In this way, the clocks of DTALite and VISSIM is always the same.

5.7 Parallel Computing and Multi-threading to Increase the Computing Speed

Acceptable computing speed is one of the major challenges to all large-scale traffic simulators. To address this issue, two simulators apply different mechanisms. DTALite has the inherent capability of parallel computing to boost the computing speed. Unlike the other time-cell-based mesoscopic simulators, DTALite is event-driven. Vehicles' arriving and departing events at each node are scheduled according to the calculation of traffic dynamics, such as queue propagation, backward shockwaves, etc. and vehicles' movements on links are simplified as one atomic calculation (free-flow forward wave or constrained backward wave). In metropolitan areas, traffic dynamics are also determined by traffic signal control systems at intersections. Green or red signal statuses in VISSIM can be reflected by opening or closing links in terms of road capacities on the turning movements in DTALite. The event-driven simulating mechanism greatly reduce the overhead of parallel computing and avoid the necessity of network partition as in other simulator with parallel computing capability. As shown in FIG. 5, using the Intel OpenMP parallel computing technology *(8)*, each CPU thread will keep a copy of the whole network structure and the OpenMP will automatically split the computing tasks onto multiple cores and later combine the results. For more details, readers are suggested to read the relevant documents prepared by the DTALite developers*(8)*.

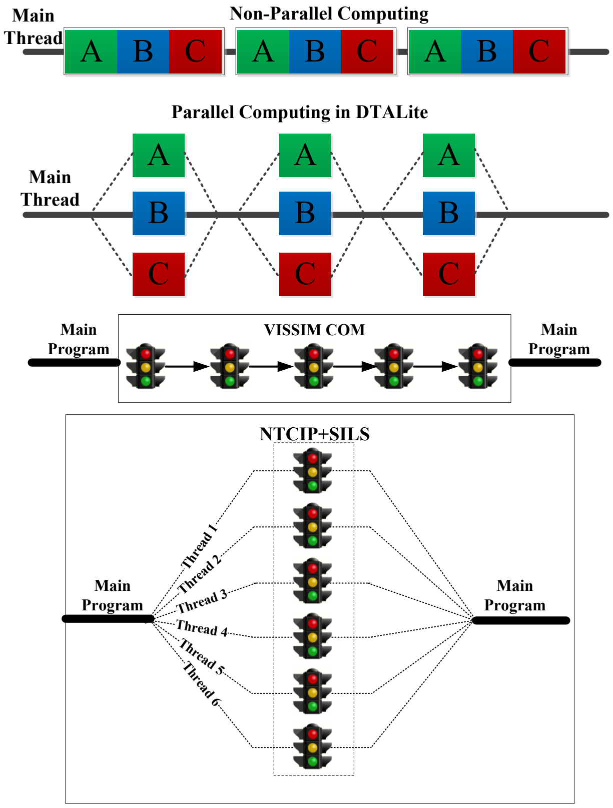


Figure 5 Demonstration of parallel computing and Multi-Threading in MetroSim

On the VISSIM side, given that VISSIM is a proprietary simulation package, it is impossible to conduct major revisions on the simulation engine independently. Fortunately, VISSIM also uses the multi-threading techniques on multiple CPU cores in order to improve the computing efficiency. The latest release of VISSIM contains advanced traffic signal emulator, software-in-the-loop signal emulator (SILS). Each SILS emulator will launch a server supporting the National Transportation Communications for ITS Protocol (NTCIP)*(11)* and therefore it is possible to interfere the signal timings from the outside. Each thread will independently poll one SILS controller and the results from all threads will be written into the signal status file every six seconds.

6. Case Study: Evaluating the Effectiveness of Signal Optimization Efforts in a Network Context

A hypothetical case study is conducted to demonstrate one of many potential applications of MetroSim. In practice, evaluating the effectiveness of traffic signal optimization is to compare the traffic mobility before and after the new signal timing applies. A long-standing question is that drivers will shortly recognize the improvement of traffic mobility and more likely choose this road. As a result, travel demand may be re-assigned among adjacent roads and the improved road may attract more traffic. If a static or arbitrarily adjusted travel demand is used, evaluating the signal control performance may be biased and it is not uncommon to witness this phenomenon in practice.

The top figure of FIG. 6 shows a hypothetical three-route networks of one mile long. During the morning peak hour (7:00 AM~9:00 AM). There are about 6,000 vehicles per hour From A to B. Route 1 and 3 have the 25 MPH speed limit and one lane. Route 2 has two lanes with 45 MPH speed limit and two intermediate intersections. Also, the approximate assignment by DTALite without signal constraints are also shown. Corresponding to the reality, Route 1 and 3 can stand for the residential roads while Route 2 can stand for an urban arterial. If the arterial is congested, some commuters will choose to use the adjacent residential routes to reach B.

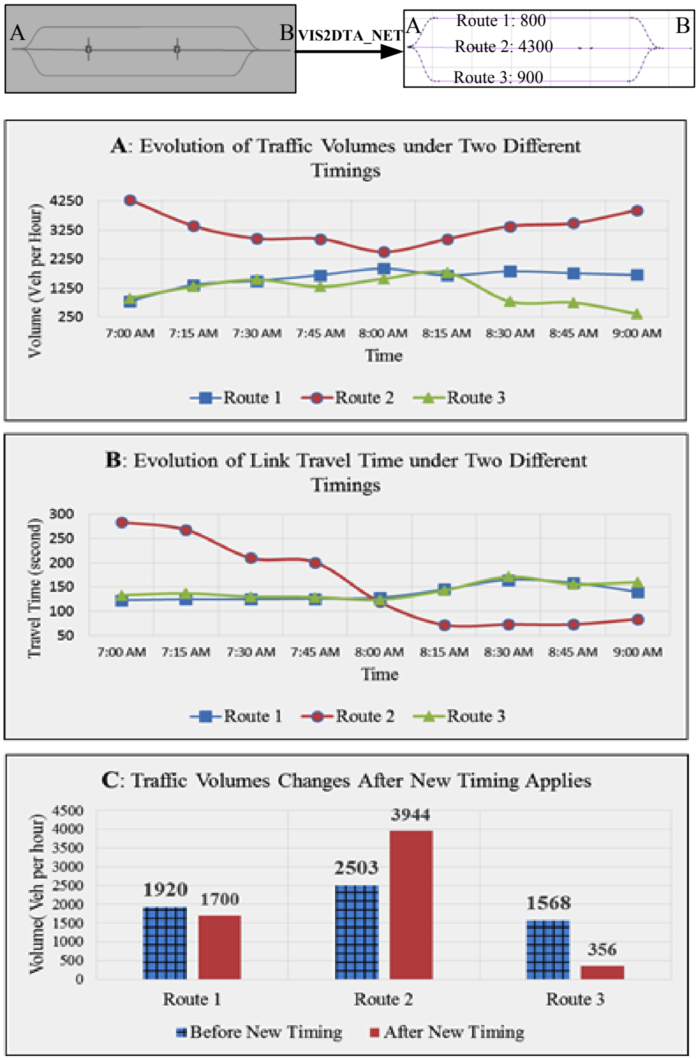


Figure 6 Simple network in MetroSim and the Evolution of Traffic Conditions with MetroSim after New Timing applies

During each simulation iteration, path volumes are first set as the numbers shown in FIG. 6. During the first hour, the signal timings on Route 2 are set as “existing” (bad) and only 20% of time is green. As a result, congested will be generated on Route 2 and eventually the travel time of Route 2 will increase. In the meantime, every 15 minutes, DTALite will examine the route travel time and reallocate the vehicles accordingly. As such, after the first hour, traffic volumes on Route 2 will be less than the initial 4,300 and Route 1 and Route 2 will receive additional traffic volumes. The updated traffic assignment on three routes after the first hour is considered the existing traffic counts. In the second hour, the signal timings at two intersections begin to favor Route 2 and 70% of time is green and so the traffic mobility on Route 2 are improved. VISSIM will report such changes to DTALite to adjust its network loading accordingly (No en-route route-changing agents in the case study for simplification). The additional traffic volumes on Route 2 is considered the newly attracted traffic volume. The performance of new signal timings is finally evaluated under the new traffic volumes (the second hour) rather than the static travel demand. MetroSim were run multiple times with different random seeds in both DTALite and VISSIM.

Evolution of traffic volumes on three links: the first step is to identify whether the traffic volumes are re-allocated as expected. This step is also considered part of validation of MetroSim. As shown in FIG. 6-A, it can be observed that the traffic on Route 2 keeps decreasing in the first hour due to the poor signal timing while more traffic is assigned to Route 2 by DTALite after the signal timing is improved. The results is consistent with our expectations. In addition, it is noticeable that the traffic on Route 1 is always higher than Route 3 even though their speed limits and road capacities are exactly the same. This also makes sense because vehicles on Route 3 will encounter two bottlenecks: when merging to Route 2 and when passing the merging areas of Route 1. By contrast, the vehicles on Route 3 will only encounter one bottleneck. As a result, the travel times on Route 3 is longer than Route 2 and consequently DTALite will assign more traffic on Route 1 than Route 3.

Evolution of link travel times: FIG. 6-B shows the evolution of link travel times on three routes. During the first hour, the link travel time on Route 2 keeps decreasing because of the traffic reduction on Route 2. After the new mainline-favored signal timing applies, the travel time of Route 2 shows a sudden drop at 8:15 AM and then travel time on Route 2 begins to increase due to newly attracted traffic. In the meantime, the travel time on Route 1 and Route 3 also show corresponding increases and decreases. According to the high-fidelity road network in VISSIM, the travel times on Route 1 and 3 are determined by two factors: traveling time on the link and waiting time for gaps while attempting to merge to the mainline Route 2. When the traffic on Route 2 increases, vehicles on Route 1 and 3 begin to have difficulty in finding gaps and merging into the mainline and therefore may have to wait longer. In the meantime, it is also possible that the travel times along the majority of Route 1 and 3 decrease due to the reduced traffic volumes. Therefore, the resulting travel times on Route 1 and Route 3 show a mixed pattern over time, depending which factor dominate.

Comparison between MetroSim and the conventional method in signal timing evaluation: FIG 6-C shows the comparison between MetroSim and the conventional method in this context. Without MetroSim, the (observed) traffic volume at 8:00 AM will be used as the background traffic. It is apparent the travel demand along Route 2 would otherwise be significantly underestimated. According to the MetroSim results, the true traffic volume along Route 2 will be 58% more than at 8:00 AM. By contrast, the new traffic volume along Route 2 would be usually predicted to be at most 20% more than the existing volume. As a result, the benefits of new signal timings in this case study would have been considerably overestimated without MetroSim.

Although we use a simplified three-route network to demonstrate, MetroSim have been preliminarily tested in medium-size networks. According to our investigation, VISSIM is always slower than DTALite and the computing bottleneck within one computer is on the real-time file exchange between VISSIM and DTALite and the real-time traffic status retrieving and archiving via a real industrial communication standard, NTCIP in VISSIM. To have one-to-one simulation speed using a solid-state-drive-enabled computer, up to 70 signal controlled intersections can be supported in MetroSim. It is usually sufficient for the downtown areas of cities.

7. Conclusions and Future Work

In this paper, a new multi-resolution traffic simulator for metropolitan area is presented, referred to as MetroSim. Compared to other similar traffic simulators, MetroSim is hierarchical and composed of two independent simulators: the mesoscopic level microscopic level. The advantages of such design is to combine the strength of mesoscopic simulation and microscopic simulation without adding additional complexity in either one. The open-source mesoscopic DTALite and microscopic VISSIM are selected as the components of MetroSim. Compared to other existing multi-resolution simulation package, MetroSim have several advantages: VISSIM provides a full-scale traffic signal emulator which enables fast retrieving and archiving real-time signal timings according to close-to-reality signal control strategies; DTALite is agent-based and so naturally capable of being customized by adjusting agent behaviors; DTALite also provides a scientific method, referred to as ODME, of calibration to utilize all available traffic data and it is event-driven rather than time-cell-based.

The majority of this paper describes the challenges in integrating two independent simulation software with acceptable accuracy and computing speed. Several solutions are also presented, such as synchronization, parallel computing, multi-threading technology, etc. Through a hypothetical case study, one of the advantages of MetroSim is demonstrated and it is expected that MetroSim will be able to solve many long-standing issues in traffic engineering as well as transportation planning.

The current form of MetroSim uses a historical origin-destination data, i.e., the static O-D matrix. The latest development in travel demand estimation has already allowed the time-varying origin-destination matrix. As such, it is our plan to introduce one more level to MetroSim in the future. The ultimate form of MetroSim will be composed of three levels: dynamic travel demand estimation (macroscopic level); dynamic traffic assignment (mesoscopic); microscopic traffic simulation in order to cover a full spectrum in transportation research and practice.

8. Acknowledgement

The research reported here was partially sponsored by the project: *AMS Testbed Development and Evaluation to Support Dynamic Mobility Applications (DMA) and Active Transportation and Demand Management (ATDM) Programs* (with Booz Allen Hamilton Inc. as prime contractor) funded by the U.S. Department of Transportation Federal Highway Administration; and partially sponsored by the National Science Foundation Grant #1239396: *A Cyber Physical System for Proactive Traffic Management to Enhance Mobility and Sustainability* and partially by USDOT University Transportation Centers Contract DTRT13-G-UTC55/Sub-award UNR-14-60. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the official views or policies of the above organizations, nor do the contents constitute a standard, specification, or regulation of these organizations.

References

1. Burghout, W. and J. Wahlstedt, *Hybrid Traffic Simulation with Adaptive Signal Control.* Transportation Research Record: Journal of the Transportation Research Board, 2007. **1999**: pp. 191-197.
2. Casas, J., J. Ferrer, D. Garcia, J. Perarnau, and A. Torday, *Traffic Simulation with Aimsun*, in *Fundamentals of Traffic Simulation*, J. Barceló, Editor. 2010, Springer New York. pp. 173-232.
3. Yang, Q. and D. Morgan. *Hybrid Traffic Simulation Model*. in *Transportation Research Board 85th Annual Meeting*. 2006.

4. Scherr, W., D. Adams, and T. Bauer, *An integrated model for planning and traffic engineering*, in *Nineth TRB Planning Methods Applications Conference*. 2003: Baton Rouge, LA.
5. Tokishi, J. and Y.-C. Chiu. *Evaluation and Improvement of Consistency of Hybrid and Multi-2 Resolution Traffic Simulation Models 3*. in *Transportation Research Board 92nd Annual Meeting*. 2013.
6. Newell, G.F., *A simplified theory of kinematic waves in highway traffic, part I: General theory.* Transportation Research Part B: Methodological, 1993. **27**(4): pp. 281-287.
7. Newell, G.F., *A simplified car-following theory: a lower order model.* Transportation Research Part B: Methodological, 2002. **36**(3): pp. 195-205.
8. Zhou, X. and J. Taylor, *A Queue-based mesoscopic traffic simulator for fast model evaluation and calibration.* Cogent Engineering (upcoming), 2013.
9. Ben-Akiva, M.E. and S.R. Lerman, *Discrete choice analysis: theory and application to travel demand*. Vol. 9. 1985: MIT press.
10. Lu, C.-C., X. Zhou, and K. Zhang, *Dynamic origin–destination demand flow estimation under congested traffic conditions.* Transportation Research Part C: Emerging Technologies, 2013. **34**(0): pp. 16-37.
11. Herrick, G.C., *The NTCIP Guide: National Transportation Communications for ITS Protocol (ver. 2)*. 1999.

LIST OF FIGURES: