Large-Scale Modeling of Smart Cities Considering the Mutual Impact of Transportation and Communication Systems.

Ahmed A. Elbery

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Computer Science and Applications

Madhav V. Marathe, Co-chair
Hesham A. Rakha, Co-chair
Lenwood S. Heath
Narendran Ramakrishnan
Mustafa Y. El-Nainay

April, 16 2018
Blacksburg, Virginia

Keywords: VANET, Communication, ITS, Large-scale, Transportation, Modeling, Eco-routing

Large-Scale Modeling of Smart Cities Considering the Mutual Impact of Transportation and Communication Systems.

Ahmed A. Elbery

(ABSTRACT)

Intelligent Transportation Systems (ITSs) are key components of the transportation systems within future smart cities, in which information and communication technologies interact to enhance the transportation system. By collecting and analyzing real-time data, and applying advanced data analytics techniques, ITSs can make better-informed decisions, that are sent back to the network actuators (cars, drivers, traffic signals, travelers, etc.) to solve or at least mitigate the ongoing transportation problems. In such feedback systems, the communication network is a major component that interacts with the transportation applications. Consequently, it is imperative to study the mutual interactions and effects between the communication and the transportation networks.

The key enabler for such studies is the large-scale modeling of communication and transportation systems. However, developing such models is challenging, not only because of the intricate interdependency between the communication and transportation systems but also because of the scale of these systems that usually covers a city-level network with hundreds of thousands of travelers concurrently moving and communicating in the network. Consequently, in our research, we are interested in studying the mutual impact of the communication and transportation systems in large-scale networks while focusing on eco-routing navigation applications that attempt to minimize the transportation network carbon footprint. Our objectives are: 1) to enable the large-scale modeling of transportation systems in smart cities including both transportation and communication systems and 2) to study the mutual interactions between the communication and the transportation systems in real-world networks. Under this umbrella, we introduced two simulation frameworks to realistically model the communication in vehicular systems. Subsequently, we used them to study the mutual influence of the communication and transportation system. Moreover, we designed, developed, and tested a multi-modal agent-based simulation platform which can simulate large-scale transportation systems. The results show that, in congested road networks, the communication performance has a significant impact on the transportation system performance. Moreover, they show that there is a negative mutual impact loop that may lead to a degrading performance of both systems. Thus, it is important to consider this impact when deploying new ITS technologies that utilize vehicular wireless communication.

Large-Scale Modeling of Smart Cities Considering the Mutual Impact of Transportation and Communication Systems.

Ahmed A. Elbery

(GENERAL AUDIENCE ABSTRACT)

In future smart cities, communication network is a major component that interacts with the transportation applications. Consequently, it is imperative to study the mutual interactions and effects between the communication and the transportation networks. Studying these systems is challenging, not only because of the intricate interdependency between the communication and transportation but also because of the scale of these systems that usually covers a city-level network with hundreds of thousands of travelers concurrently moving and communicating in the network. Therefore, in this dissertation, our main objectives are: 1) to enable the large-scale modeling of transportation systems in smart cities including both transportation and communication systems and 2) to study the mutual interactions between the communication and the transportation systems in real-world networks. Under this umbrella, we introduced two simulation frameworks to realistically model the communication in vehicular systems. Subsequently, we used them to study the mutual influence of the communication and transportation system. Moreover, we designed, developed, and tested a multi-modal agent-based simulation platform which can simulate large-scale transportation systems. The results show that, in congested road networks, the communication performance has a significant impact on the transportation system performance. Moreover, they show that there is a negative mutual impact loop that may lead to a degrading performance of both systems. Thus, it is important to consider this impact when deploying new ITS technologies that utilize vehicular wireless communication.

To my dearest wife Amany Kandeel, my great sons and daughter: Saleh, Nour, and Moemen. To my parents, my brothers, and my sisters.

Acknowledgments

Firstly, I would like to express my sincere gratitude to my advisor Prof. Hesham Rakha and Prof. Madhav Marathe for the continuous support of my Ph.D study and related research, for their patience, motivation, and immense knowledge. Their guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisors and mentors for my Ph.D study.

Besides my advisors, I would like to thank the rest of my thesis committee: Prof. Lenwood Heath, Prof. Narendran Ramakrishnan, and Prof. Mustafa Y. El-Nainay, for their insightful comments and encouragement, but also for the hard question which incented me to widen my research from various perspectives.

Contents

List of Figures

List of Tables

Terminology Clarification

In this dissertation, we cover two different fields: transportation and communication, which use the same terminologies to refer to different entities or meanings. For example, the terms, *node* and *link* are used in both fields to refer to different entities.

The meaning of these terminologies should be clear from the context. However, to avoid any confusion with these terminologies, throughout this dissertation we will use the term *node* and *link* to refer to transportation entities, where node refers to transportation network node such as intersection or turning point, and *link* refers to a road link that connects between two nodes. additionally, the term *routing* is used to refer to process of assigning routes to vehicles.

To refer to communication entities, we will use the term *communication node* that refers to communication enabled vehicle or a roadside unit, and the term *communication link* to refer to links in communication networks. Sometime, the terms *vehicle* or *station* are used also to refer to communication enabled vehicle. To refer to routing in communication networks we use the terms *data routing*, *packet routing* or *routing protocols*.

Chapter 1

Introduction

The population of cities is continuously increasing, especially because of the continuously increasing migration toward cities. The United Nations (UN) reported in [1] that 54% of the population of the world lives in cities, and the forecasted population will be 66% or higher in 2050. Due to this tremendous increase in city population, many problems arise in urban cities including scarcity of resources and increase in urban traffic congestion. The air pollution resulting from this traffic congestion problem will also increase. Moreover, these traffic problems are anticipated to be tangled up in the infrastructure aging and deterioration [2]. Thus, traffic problems and their consequences are major challenges facing cities and mega-cities of the future.

Nowadays, cities utilize digital technologies across different functions to overcome these challenges. Such cities are known as smart cities. The main objectives of smart cities include, but are not limited to [3]:

- Efficient use of resources
- Environmental sustainability
- Efficient and sustainable transportation
- Better urban planninglivable cities
- Sustainable homes and buildings

In a typical smart city, among the most important objectives is the efficient and sustainable transportation. Efficient transportation does not only improve the transportation and mobility, it also

helps improve the environmental sustainability. For example, many researchers showed that vehicle emissions have become the dominant source of air pollutants, including carbon monoxide (CO), carbon dioxide (CO_2), nitrogen oxides (NOx), and particulate matter (PM) [4]. It is well known that increasing the severity and the duration of traffic congestion significantly increases the pollutant emissions and degrades the air quality.

Traffic congestion does not only affect health, it also has important economic impacts. In [5], the authors expected that by utilizing smart traffic management, in the United States alone, the elimination of traffic congestion may result in reducing the commuting hours by over 5 billion hours for drivers, which will save over $121 billion in total delay and fuel. To achieve this smart traffic management, we have to utilize the information and communication technologies to collect and process the system information in real time to make the appropriate decisions that can improve the system.

Therefore, in smart cities, solving transportation problems and improving transportation system performance does not solely mean building new roads or repairing aging infrastructure or even using new innovative designs. In addition to all of those, the future of the transportation system relies on integrating information and communication technologies and applying new computation methodologies into transportation system to operate the system more efficiently and to maximize its benefits by better managing its resources. In smart cities, this integration builds what is known as intelligent transportation systems (ITSs) [2], in which communication network is playing a critical role that can affect overall system performance. The packet drops and delays in the communication system can lead to incorrect representation of the transportation system state that may result in inappropriate decisions and can, consequently, affect the ITS performance.

To show the importance of communication in smart cities, the following sections provide an overview of the concept of smart cities, smart city architecture, and how this architecture applies to ITSs as a main component of a smart city.

1.1 Smart Cities: Definition and Architecture

The defining and conceptualizing of smart cities are still in progress. So, in literature there are many definitions for smart cities [6, 7]:

- A city well performing in a forward-looking way in economy, people, governance, mobility, environment, and living, built on the smart combination of endowments and activities of

self-decisive, independent, and aware citizens [8].

- A city connecting the physical infrastructure, the IT infrastructure, the social infrastructure, and the business infrastructure to leverage the collective intelligence of the city [9].
- The use of smart computing technologies to make the critical infrastructure components and services of a city, which include city administration, education, healthcare, public safety, real estate, transportation, and utilities more intelligent, interconnected, and efficient [10].
- A smart city brings together technology, government and society to enable the following characteristics: smart cities, smart economy, smart mobility, smart environment, smart people, smart living, and smart governance [11].

In all the definitions of smart cities, mobility, transportation, and environment are important components of smart cities.

From the architectural point of view, because it integrates computational, networking, and physical processes, the smart city is considered a large-scale cyber-physical system (CPS). Thus, the generic architecture of smart cities is similar to that of CPSs, shown in Figure 1.1 [12], which is known as the 5C architecture. Through a sequential workflow manner, the 5C architecture defines how to construct a CPS from the initial data acquisition to data analytics and final decisions.

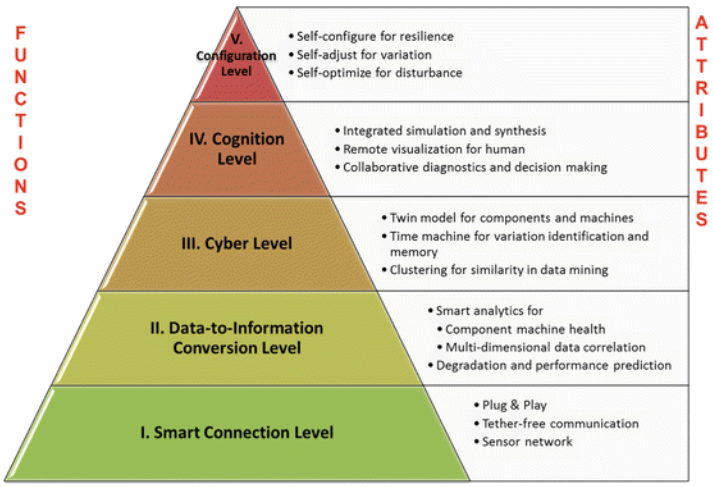


Figure 1.1: Cyber-physical systems architecture

In the 5C architecture, the smart connection level is responsible for the first function in the CPS, which reliably and accurately acquires data from different components or sensors. Then, meaningful information is inferred from the collected data in the conversion level.

The third level, the cyber level, is considered the central information hub in the CPS architecture. It collects information from different networked components. With the massive amount of information it gathers, specific analytics are used to extract additional information that provides better insight on the status of individual components as well as over the overall system. Based on this information and the historical data, the future can be predicted.

In the cognition level, the correct maintenance and operational decisions are made. To make these correct decisions, it is sometimes necessary to present the acquired knowledge to expert users through efficient human-machine interfaces and use of artificial cognitive systems (that are capable to interact with entities at the lower cyber level) for evaluating possible strategies and action plans.

The configuration level takes the decisions made by the lower cognition level and sends them as feedback to the physical system components. This feedback acts as a control system to apply the corrective and preventive decisions to the monitored system.

ITS is one of the important sub-systems of smart cities. It is also large-scale CPS. However, its main objective is to improve transportation and mobility and to mitigate their negative impacts. The following subsection provides an overview of ITS and its applications.

1.2 Intelligent Transportation Systems

ITS is a core component of smart cities. ITS integrates people, roads, and vehicles by applying and integrating communications, computers, and other technologies to enhance the transportation system performance and mitigate the negative impacts of the transportation system on human live. Building on the new communication and information technologies, data analysis and processing, ITS builds large, real-time, and efficient transportation management systems.

In an ITS, a network of sensors, microchips, and communication devices work together to collect, process, and disseminate information about the transportation system. Consequently, the ITS can make better-informed decisions about either transportation mode selection, departure time selection, route selection, traffic signal timing optimization, or even building new roadways.

Figure 1.2 [2] shows the main steps and some technologies that are being used for collecting, processing, and disseminating data in ITSs. The data can be collected using many technologies including, but not limited to, surveillance helicopters, surveillance cameras, loop detectors, or vehicle probes. Subsequently, the collected real-time data can be sent to the traffic management center for analyzing and making systematic decisions. These decisions are then sent to interested

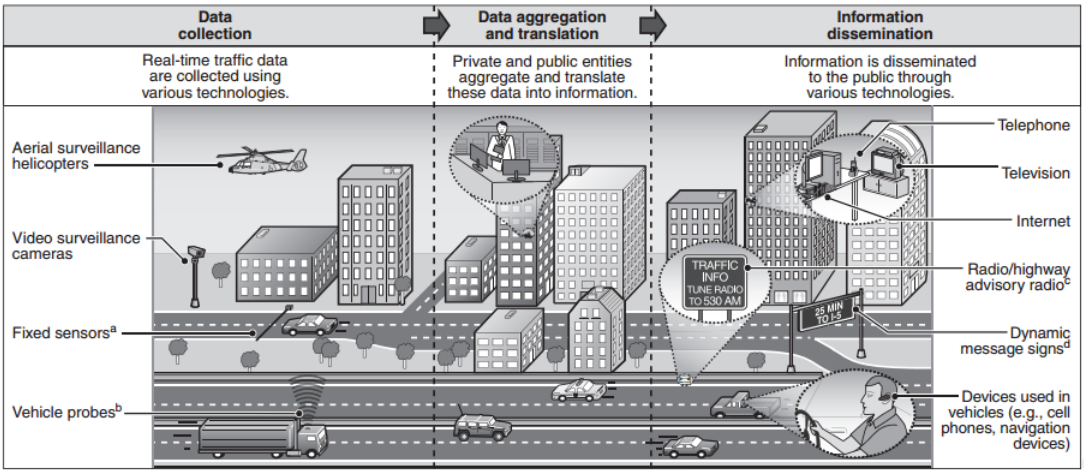


Figure 1.2: ITS Operation

drivers through either phone, the Internet, guidance and navigation systems, dynamic message signs, or broadcast to the public on television or radio.

ITS has many applications that are usually arranged in five systems based on their primary functional intent, which are:

- ***Advanced traveler information systems [3]*** are the most recognized systems. An ATIS provides drivers with real-time travel and traffic information, such as congestion information, accidents, weather conditions, and road repair work. An ATIS can also provide the driver with some route guidance based on the network conditions, driver preferences, and destination. In addition, it can provide some information about the availability of parking in the destination area.

- ***Advanced transportation management systems (ATMS)*** are concerned with traffic control devices (i.e., traffic signals, ramp metering, and the dynamic message signs). For example, adaptive traffic signal control applies intelligent techniques to dynamically change signal timings based on traffic demands to achieve better network flow and lower delay. Ramp metering is the process of regulating traffic entering the freeway using simple traffic signals. Ramp metering is expected to reduce congestion and, thus, increase traffic speed and volume on freeways by reducing the demand and by breaking up platoons of cars.

- ***ITS-enabled transportation pricing systems*** include electronic toll collection (ETC), through which drivers can pay tolls automatically via a DSRC-enabled onboard device or tag placed on the windshield (such as E-Z Pass in the United States). One of the applications in this

system is congestion pricing where some cities charge vehicles on entering the urban areas in some peak hours. This charge will help reduce congestion. The HOV lanes are some mechanisms being used to reduce the congestion by dedicating some lanes to buses and high occupancy vehicles (HOV). Single occupancy vehicles are also allowed to use these lanes upon payment.

- ***Advanced public transportation systems (APTS)*** aim to help make public transportation a more attractive option for commuters. To achieve this, the system visually provides commuters with the locations and the real-time status (expected arrival and departure times) of buses and trains. This application utilizes the automatic vehicle location (AVL) application, through which public transportation reports its location to the traffic operation center. Through the AVL, the operators and managers can have a better real-time view of all their vehicles. APTS also include electronic fare payment, which enables travelers to electronically pay for the trips either by smart cards or mobile phones.

- ***Vehicle-to-infrastructure (V2I) and vehicle-to-vehicle (V2V)*** integration enable vehicles to communicate with roadside units as well as communicating with each other. This category has many applications such as Cooperative Intersection Collision Avoidance System (CICAS), cooperative driving, and incident reporting. V2I and V2V communication also represent the main infrastructure for collecting and disseminating data to and from vehicles.

1.3 Motivation

From the previous section, it is clear that communication technology is a major component of ITS, which is responsible for transferring the data from/to different sensors in the network to/from the management center. The vehicular Ad-hoc network (VANET) [13, 14] is a promising communication technology that is expected to form the communication infrastructure for many ITS applications. Consequently, when studying ITS in smart cities, it is essential to study the impact of communication system parameters (packet delay and drop rate) on the performance of the transportation sector.

This impact of communication and transportation will be more significant in large smart cities. This is because in these cities, the communication networks are expected to have a higher data traffic load, which can result in lower packet delivery rate and longer packet delay, and consequently, a higher impact on the transportation system performance.

Moreover, this impact of communication is application dependent, which means that some applications may be significantly affected by packet delay and/or packet drops, while other applications may not. For example, safety-related applications such as collision warning applications are very sensitive to delay and drops. On the other hand, some other applications may not be so sensitive to packet drops and delays. These include congestion notification applications that may work properly even in case of long communication delays.

Motivated by the above-mentioned importance of communication networks in smart cities, the main objective of this dissertation is to study the impact of communication network performance on the performance of large-scale ITS in smart cities, more specifically, we study the impact of communication network on sustainability of transportation in smart cities.

Since these impacts are application dependent, we selected the feedback-based eco-routing navigation application. It is a promising navigation technique that attempts to minimize fuel consumption and emissions by routing vehicles through the most environment-friendly routes. We focus on the eco-routing application and the mutual impact of vehicular communication and eco-routing in large-scale road networks.

1.3.1 Why Eco-routing?

The interactivity between transportation and communication is not that simple, it is more intricate. The reason is that there is a complicated mutual influence between communication and transportation systems, especially in the ITS applications that can affect vehicular mobility. Feedback-based routing applications are an example for this category of applications. In the feedback-based routing applications, TMC collects real-time information about the costs of the roads in the road-map. Subsequently, it updates its routing trees. Then, whenever a vehicle needs to find a route or needs to update its current route, TMC finds the best route based on the most recent routing information it has. In these applications, if the packet drop rate is too high or the packet delay is too long, then TMC routing information will not be correctly representing the current network state. Consequently, the routes it calculates may not be the best routes. In such cases, these routes can result in increasing travel time or increasing congestion on some roadways, and thereby degrading transportation system performance.

The mutual impact does not stop at this point. If this scenario continues for enough time, the congestion will result in increasing the vehicle density, consequently, the competition on the wireless medium in VANET medium access control (MAC) layer will increase, which will result in

degradation in the performance of the communication network. Such a scenario can result in a performance degradation loop, where the communication performance degrades the transportation performance, and the lowered mobility performance results in lower communication performance. This is the first reason why we selected eco-routing navigation in this study.

The second reason behind choosing the feedback-based eco-routing application is the anticipated economical and environmental benefits of eco-routing navigation techniques. Previous research showed that one major problem facing smart cities is the environmental and economical impacts of the transportation sector. The importance of this problem has been studied in many research efforts in the literature. For example, in 2008, the U.S. Department of Energy mentioned in [15] that approximately 30% of the fuel consumption in the U.S. is consumed by vehicles moving on the roadways. In addition, about one-third of the U.S. carbon dioxide (CO_2) emissions comes from vehicles. The 2011 McKinsey Global Institute report estimated savings of about $600 billion annually by 2020 in terms of fuel and time saved by helping vehicles avoid congestion and reduce idling at red lights or left turns.

Figure 1.3 shows the total energy consumption by sector from 1950 to 2014 in the US. It shows that the transportation sector is the second largest consumer according to the U.S. Energy Information Administration (EIA) [16]. The EIA reported in 2015 that the transportation sector consumes about 28% of the total energy consumed in the USA.

The U.S. Environmental Protection Agency (EPA) reported that approximately 59% of the energy consumed by the transportation sector is consumed by light-duty vehicles, as shown in Figure 1.4 [17]. It also reported that more than one-quarter of the total U.S. greenhouse gas emissions comes from the transportation sector, as shown in Figure 1.5 [17].

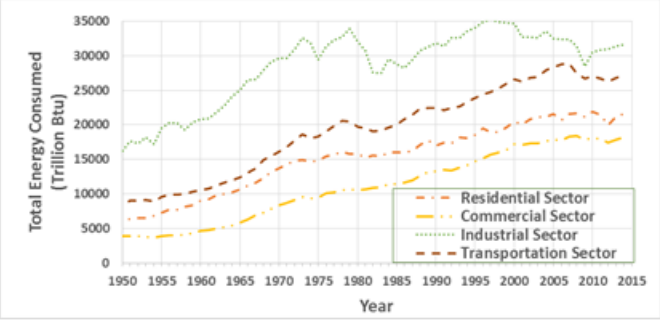


Figure 1.3: Total energy consumption by sector in the USA

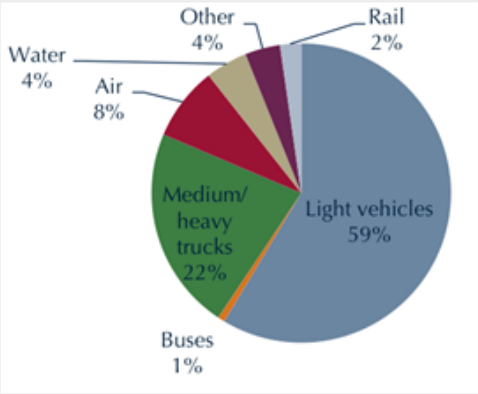


Figure 1.4: Transportation energy use by mode

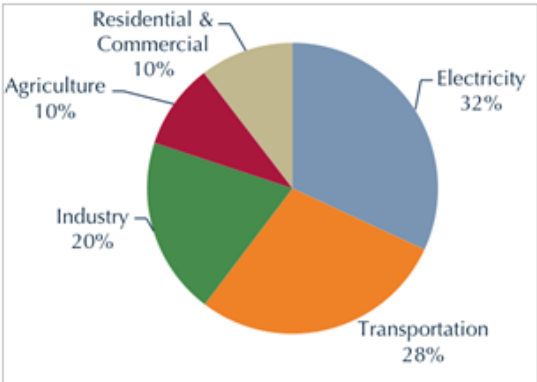


Figure 1.5: Gas emissions by sector

1.3.2 The Research Focus

When developing ITS applications in smart cities, it is essential to study and quantify the impact of communication on the performance of those applications. It is also imperative to consider the sensitivity of these applications to the communication parameters that include the communication network type, the network connection speed, the average density of communication nodes (vehicles), the average packet size, and the background data traffic rate in addition to many others. An important aspect that should be considered as well is the data transport protocols and data routing protocols that are being used in communication for each application. Moreover, the communication paradigm (i.e., V2V, V2I, or Hybrid) should be considered.

In fact, considering large-scale modeling of both transportation and communication systems as well as all the aforementioned communication parameters is too broad a research effort for a single dissertation study, especially since this is the first study that addresses this impact in a large-scale system.

Consequently, in our study, we focus on large-scale modeling of smart cities and consider only the V2I communication paradigm. Compared to V2V and Hybrid VANET communication paradigms, V2I does not need to consider packet routing and its impact on communication performance. Thus, the most important part we should consider in this research is the Medium Access Control (MAC) protocol in VANET. The importance of modeling the MAC in VANET is reasoned to three reasons. Firstly, the MAC performance is significantly affected by the mobility parameters of the vehicles such as vehicle speed and vehicle density. Secondly, the MAC layer is the gateway through which packets exit and enter the communication nodes (vehicles) and since the MAC layer accounts for these packets in its processing, it can capture some phenomena of the upper layer protocols.

Thirdly, the MAC layer performance has significant impacts on all upper layer protocols. Thereby, considering these upper layer protocols without considering the MAC performance parameters will be an inaccurate representation of these protocols that can lead to incorrect conclusions.

1.4 Research Contribution

In this research, we focus on eco-routing navigation and studying the mutual impact of communication and eco-routing in large-scale real-world networks. Our main objectives are 1) to enable the large-scale modeling of vehicular networks in smart cities including both transportation and communication systems and 2) to study the mutual interactions between the communication and the transportation systems in large-scale real-world networks.

To achieve these objectives, we started by developing the **V**ehicular **Net**work **Int**egrated **Sim**ulator (*VNetIntSim*) which supports small and medium size networks. It shows limited scalability, so it can not simulate large-scale networks with tens of thousands of cars in the network. Because of this limitations, we developed and tested a new model that can support large-scale scenarios. We used both models to study the mutual interactivity of the communication and transportation systems in large-scale real-world road network.

Moreover, we developed a system that utilizes parallel simulation to model large smart cities and used it to simulated the city of Los Angles in its morning peak hours.

In addition to these main contributions, we introduced two modifications to eco-routing navigation technique.

The next subsections give a brief overview of each of these contributions.

1.4.1 VNetIntSim

The first platform, which is published in [18, 19], we call VNetIntSim. This platform integrates two simulation softwares, namely, INTEGRATION software for transportation modeling and OPNET for communication modeling. This coupling forms a full-fledged system that is capable of simulating realistic vehicle mobility as well as vehicular communication in a very low level of details in both transportation and communication systems. In transportation and vehicle mobility area, this model tracks every individual vehicle every decisecond. It also can model different mobility phenomena such as lane changing, gap acceptance, and car-following behavior. In addition, it is

able to model different traffic control strategies including traffic signals, variable message signs on the road, yield signs, and stop signs. So, it can realistically capture the second-by-second vehicle speed, acceleration, fuel consumption, and emission levels. It inherits these features from the INTEGRATION software that we use in this model. In the communication area, OPNET can model and simulate wireless communication and can capture low-level details about packet drop, packet delay, and data traffic congestion, in addition to detailed statistics in different network layers.

After developing this model, we used it to study the impact of mobility on communication system performance at different road speeds. We studied the impact of speed on different network applications, which include File Transfer Protocol (FTP) and Voice over IP (VOIP). This work is published in [18].

Then, within VNetIntSim, we built an eco-routing application that is composed of two processes to realistically model the feedback eco-routing navigation system. Compared to the previous eco-routing implementations, our implementation simulates the mutual interaction of the communication and the eco-routing, so, it can capture these mutual interactions and model their impacts on the system-wide fuel consumption and emissions. We utilized this platform to study the impact of communication on eco-routing by quantifying the fuel consumption in the cases of different communication settings. This work was published in [19].

The work we have done in this part of our research is the first research work that studies and realistically models the mutual interactions between communication and ITS applications. It is also the first work that quantifies the communication impact on the fuel consumption in transportation systems.

Our scalability study of the VNetIntSim showed that it is not scalable because of its detailed simulation of communication events which uses a discrete event simulation technique in OPNET. This technique simulates every single event that happens in every single vehicle, which results in long simulation time and high memory usage.

1.4.2 Large-Scale Model for Vehicular Communication

Since we are interested in large-scale modeling of communication in vehicular environment, this scalability issue in VNetIntSim motivated us to develop a new scalable platform to enable the modeling of large-scale vehicular system. We decided to replace the communication modeling software, OPNET, by an analytical model for the VANET communication. So, we developed a new analytical model for the MAC technique in VANET. This communication model utilizes

Markov chain and queuing theory to estimate the packet drop probability and packet delay based on the surrounding vehicle density and the average packet generation rate. Then, we validated this analytical model against benchmark simulation data. The validation showed an accurate estimation of both the delay and drop probability at different network settings, including different packet generation rates, different vehicle densities, and different transmission techniques.

Subsequently, to realistically model the communication in vehicular networks, we incorporated this analytical mode within the INTEGRATION software, which produces our second simulation platform. A brief description of this model and a preliminarily results for a test network was presented in the conference on Traffic and Granular Flow (TGF 2017), which will be published by Springer in the book Traffic and Granular Flow 17". Another paper about the model has been accepted for publication in the 2018 IEEE International Conference on Communications Workshop (ICC2018) and will be published in the conference proceedings.

Then, we used this scalable simulation platform to study the mutual interaction of communication and transportation systems in a real large-scale transportation network, which is the downtown area of Los Angles (LA). To achieve realistic modeling, we built the road network and used a calibrated traffic demand for the morning peak hours in the city of LA. The traffic is calibrated based on data collected from different sources. The traffic calibration is out of the scope of this dissertation, interested readers can refer to [20] for more details. The simulation network includes the road network, traffic signals, and lane striping, in addition to other types of traffic control such as stop and yield signs. Using this network, we studied the impact of the communication on the mobility sustainability parameters including fuel consumption, emissions, travel time, and delay. We also observed the impact of communication on the traffic congestion level at different vehicular traffic demand rates, and studied the impact of the transportation traffic demand rate on the performance of communication by quantifying the packet drop rates and delays in the case of different vehicular traffic demand levels. This work is submitted to the IEEE Transactions on Intelligent Transportation Systems.

This simulation platform is the first vehicular communication modeler that supports that large-scale vehicular networks in terms of spatial area and vehicle count, where we use it to model the downtown area of Los Angles city. The area we modeled is about $133\,Km^2$ and the number of vehicles modeled is greater than $560,000$ cars. This is also the first research work that studies and quantifies the impact of vehicular communication on the eco-routing navigation system at this scale.

1.4.3 Modeling of Large-scale Agent-Based Multi-Modal Transportation System

The last main contribution is our proposal, implementation, and testing of a new system that allows the modeling of agent-based large-scale multi-modal transportation systems. The developed simulation platform combines parallel computing with two transportation modeling methodologies. In this simulation platform, we used both microscopic and mesoscopic traffic simulation techniques in such a way enable us to take advantage of the strengths of each approach. We use microscopic simulation when we have to, to maintain the accuracy, and use the mesoscopic simulation, where we can, to reduce the computational load. The microscopic simulator provides high accuracy by tracking the individual vehicles every second. The mesoscopic simulator enables high scalability in the model by computing average parameters for vehicles. The road links are statically categorized and assigned to either the microscopic or the mesoscopic simulator based on link importance in the network, which is basically related to their expected traffic load and the variation in the link parameters. For example, links that are expected to have high speed variation have to be simulated microscopically, and links with minimal variations are assigned to the mesoscopic simulator. Consequently, the network is divided into two main layers, one for micro-links and one for meso-links. In addition to these two layers, the system supports different optional transportation modes such as trains, bikes, and pedestrians. Each simulator runs separately, and all the simulators are managed by the simulation controller which is responsible for managing, synchronizing, and connecting the different simulators. We used this system to simulate the whole city of LA. This system model publication has been published in the 4th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS 2018) [21].

1.4.4 System Optimum Eco-routing

In addition to these main contributions, and based on our study for the eco-routing, we developed a novel eco-routing navigation system: linear programming stochastic eco-routing (LPS_ECO), which is the first system-optimum model for eco-routing navigation. This model uses linear programming and stochastic traffic assignment to minimize the system-wide fuel consumption. Compared to all previous eco-routing models that only consider the route fuel cost, the developed model accounts also for two other factors in addition to the route fuel cost. It considers the traffic load on each route to avoid overloading the low cost routes. It also considers the current traffic demand level to load balance this traffic demand over a set of routes. The proposed model also uses stochastic route assignment instead of utilizing the shortest path. This model has been published

in the 97^{th} Transportation Research Board meeting (TRB 2017) in [22].

1.4.5 Ant Colony-based Eco-routing

Finally, based on our study of the current implementations of the eco-routing, we realized some shortcomings that affect its performance in some scenarios. The main reason behind these shortcomings is the lag in the eco-routing feedback system. Thus, we proposed a new eco-routing system that solves these drawbacks by utilizing Ant Colony Optimization. By enabling vehicles to share periodic real-time information and applying Ant Colony, the performance of eco-routing is some network scenarios was improved. This work has been published in the [23].

1.5 Thesis Organization

The thesis is organized as follow:

Chapter 2 is a literature review on the related research areas including the large-scale modeling of transportation systems, the modeling of communication systems within vehicular networks, and the related work to the eco-routing navigation system.

In Chapter 3, we introduce the VNetIntSim, which is a simulation platform for vehicular systems that can be used to study and analyze both transportation and communication phenomenon in small-size and medium-size networks. Subsequently, Chapter 3 uses this proposed platform to study the impact of transportation parameters on the vehicular communication, where a simple intersection road network is used to study the impact of vehicle speed and density on the file transfer application and voice over IP application.

Then, Chapter 4 builds upon the VNetIntSim framework by using it to study the impact of the communication performance on eco-routing performance. More specifically, Chapter 4 studies the impact of packet drop, packet delay, and RSU allocation on the performance of the eco-routing application and quantifies the fuel consumption in the case of different communication settings that include communication transmission power and allocation of the roadside units.

Chapter 5 introduces a new scalable simulation platform. In this chapter, we develop an analytical model that can estimate the communication performance parameters based on the surrounding network conditions including vehicles count and distribution. Then, this model is utilized to study the mutual impact of the communication and the transportation in a large-scale network, which is

the downtown area in the city of Los Angles.

Chapter 6 introduces a scalable framework to simulate multi-modal transportation system in large-scale transportation system, and applies this system to the overall network of LA. Then, Chapter 7 describes the proposed system optimum eco-routing system and compare its performance to a shortest path based technique.

Chapter 8 introduces the Ant Colony-based eco-routing, which is a new eco-routing technique that attempts to improve the performance of eco-routing systems by utilizing Ant Colony optimization technique.

Finally, the conclusions, findings and future work are presented in Chapter 9.

Chapter 2

Background and Literature Review

Modeling of full-fledged ITS-connected vehicular systems is a multidisciplinary research area that covers two basic areas, namely, transportation and communication areas. This chapter gives an overview of these two areas and the previous work related to modeling of full-fledged vehicular system. Then, it describes the eco-routing navigation technique in details, its implementation in the integration software and its related literature, as well as the previous work in modeling large-scale transportation systems.

2.1 Transportation and Communication Inter-dependency

Combining transportation and communication systems together raises many challenges. Among these challenges, the most important one is considering the mutual impact of the two systems and how these impacts affect the performance of both of them.

In literature, there are many studies that consider the effect of mobility parameters (vehicle speed and density) on communication performance such as [24, 25]. However, most of these studies assume an average vehicle speed and stationary spatio-temporal distribution of the vehicles, which are not realistic assumptions. Assuming an average speed cannot capture the vehicle's acceleration/deceleration events in transportation systems, which have an important impact on many applications such as safety applications [26], which are sensitive to such events. From the transportation side, the acceleration/deceleration events are imperative to correctly estimate the carbon footprint of the vehicles. Ignoring these events cannot accurately capture the emissions and fuel consumption of the vehicles.

Moreover, on the communication side, assuming a stationary spatial distribution of the vehicles ignores the effects of the vehicles' mobility on the vehicle spatial distribution and its temporal changes. The vehicle distribution and mobility have significant impacts on communication performance, assuming stationary spatial distribution can mislead to deceptive conclusions, because, under this assumption, we cannot realistically capture these important impacts.

Communication performance can also affect transportation performance, such as vehicle routing application, cooperative driving, and intersection control algorithms, especially for applications that affect or control the vehicle mobility. For instance, in intersection control algorithms, in order to increase the traffic flow at the intersection and minimize the vehicle travel time, vehicles exchange information to find the optimum vehicles' trajectories to cross the intersection. In the case of long communication delay or high packet drop rate, vehicles may not get the required information in time or may miss some information. As a result, they may make inaccurate decisions that can result in congestion in the intersections.

This means that there is an intricate interaction between communication and mobility in the vehicular environment, and we need to capture this interactivity to achieve the correct conclusion. These interactions depend on the applications, their requirements, in addition to the real-time network conditions. In this study, we are interested in the eco-routing application and its mutual interactivity with the vehicular communication. Therefore, the next two sections describe the vehicular communication and the eco-routing navigation systems, respectively.

2.2 VANET Communication

Recent advances in wireless networks have led to a new type of network, the Vehicular Ad Hoc Network (VANET), which is a form of Mobile Ad Hoc Network (MANET). VANETs are developed as the communication infrastructure of ITS to significantly improve transportation systems performance. So, VANET communication is a key enabler for developing new ITS systems that enhance drivers' and passengers' safety and comfort.

By utilizing VANET communication, ITS intends to improve safety on the roads and to reduce traffic congestion, waiting times, fuel consumption, and emissions. For example, warning messages sent by vehicles involved in an accident enhance traffic safety by helping approaching drivers to take proper decisions before entering the dangerous crash zone [27, 28]. In addition to safety-related applications, VANET can also support other non-safety applications. For instance, information about current transportation conditions facilitates driving by taking new routes in case of

congestion, thereby saving time and fuel and reducing emissions [29, 30]. Moreover, VANETs are designed to support Quality of Service (QoS) for applications that are time-sensitive, such as audio and video applications. Thus, it enables commuters to have video and/or audio streaming while commuting.

Vehicular networks allow the communication enabled vehicles, roadside units, and other mobile devices to communicate in an ad hoc manner. Therefore, using VANET, moving vehicles can communicate with other vehicles (vehicle-to-vehicle [V2V]), roadside units (RSUs) (vehicle-to-infrastructure [V2I]), or even hand-held mobile devices (vehicle-to-device [V2D]) using a Dedicated Short Range Communication (DSRC) system. DSRC is an enhanced version of the WiFi technology suitable for VANET environments. The DSRC is designed to support communication in vehicular environments, which is characterized by its high mobility that results in rapid topology changes. DSRC has been standardized by IEEE 802.11p in the 1609 family of standards known as Wireless Access in Vehicular Environments (WAVE) [31].

2.2.1 VANET Physical Specifications

The DSRC spectrum is 75 MHz assigned by the U.S. Federal Communication Commission to vehicle-to-X (V2X) communications. Counterpart agencies in other countries have made similar allocations. This spectrum is divided into seven 10-MHz wide channels as shown in Figure 2.1. The frequency band 5.850 GHz to 5.855 GHz is reserved. The middle channel (channel 178) is the control channel (CCH). Channels 172, 174, 176, 180, 182, and 184 are service channels (SCH) that are intended for general purpose data transfer applications.

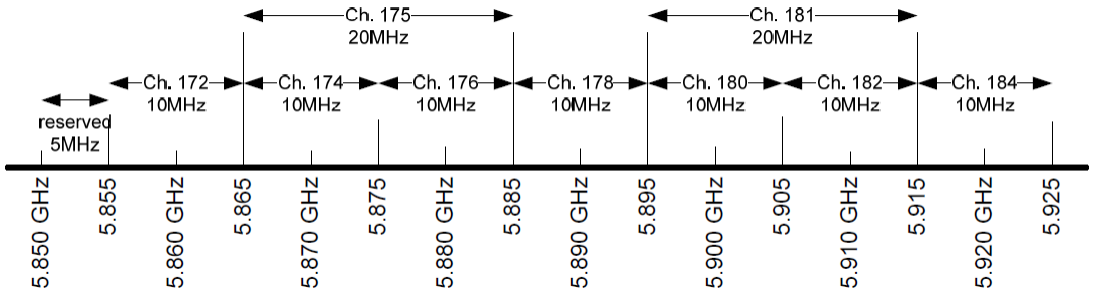


Figure 2.1: DSRC channels in the U.S.

While the physical layer of IEEE 802.11p is similar to the IEEE 802.11a standard [32, 33], IEEE 802.11p uses an Enhanced Distributed Channel Access (EDCA) media access control (MAC) sublayer protocol that is based on the IEEE 802.11e standard [34], with some modifications to the

transmission parameters. IEEE 802.11p supports transmission rates up to 54 Mb/s within ranges up to 1,000 m. In IEEE 802.11p, the EDCA provides quality-of-service (QoS) by supporting four different traffic Access Categories (ACs), namely, Background (BK), Best-effort (BE), Video (VI), and Voice (VO). Each AC has different medium access parameters such as Arbitration Inter-Frame Space (AIFS) and Contention Window (CW) limits.

2.2.2 WAVE Medium Access Technique

The basic MAC technique used by VANET is the EDCA, which is similar to the Distributed Channel Access (DCA) with a support to quality of service. EDCA supports four traffic access categories. Every access category works as a stand-alone virtual station that has its own queue and its own access parameters.

When an AC has a packet to send, it initializes its back-off counter to a random value within a given range called the contention window (CW). Then, it senses the medium until it becomes idle. If the medium continues to be idle for a specific time period called the Arbitration Inter-Frame Space (AIFS), it counts down its back-off counter. When this counter becomes zero, the AC can send its frame. Within the same station, two ACs can start transmitting at the same time, this situation is known as an internal collision. In this case, the higher priority AC will be granted the transmission, while the lower priority AC will double its CW range and re-initialize the back-off counter and back-off again.

If two stations start sending at the same time, the collision of the two signals will destroy both the frames. So, when a station sends a frame, it has to wait for an acknowledgment (ACK). If an ACK was not received within a specific time period, a collision is assumed, and the station must double its CW range and tries to retransmit the frame.

More details about the WAVE MAC will be discussed in Chapter 5, where we develop an analytical model for this technique using Markov chains.

2.2.3 VANET Characteristics and Challenges

Compared to other wireless networks, VANETs are characterized by unique characteristics that distinguish them.

1. **High mobility:** cars, which are the VANET communication nodes, are characterized by

high speed compared to other network types. This high speed makes VANET a dynamic environment, which brings many challenges to VANET such as frequent communication link drops and their impact on communication performance.

2. **Predictable and restricted mobility patterns:** in VANET, node movements are restricted by the road network and its properties such as the speed limits and road controls (traffic signals and stop signs). Moreover, node movements are governed by traffic flow theory rules that relate the speed, density, and flow rate to each other. All these rules and constraints make the vehicles mobility trajectories predictable, at least in the short term.

3. **Rapid topology changes:** the high speeds of vehicles lead to frequent network topology changes. Consequently, it increases the communication overhead due to higher packet retransmission rates. Additionally, in the case of multi-hop communication, to adapt to frequent topology changes, the routing protocol overhead will be increased to enable faster convergence. This increase in network overhead results in higher background packet traffic demand that significantly affects communication performance.

4. **High power availability:** in VANETs, vehicles are equipped with batteries that can be considered as power supplies for all communications and computation tasks. Considering the low communication power required by the vehicle communication devices and the batteries energy capacity, these batteries can be considered an infinite power supply for these communication devices because a battery can supply power for weeks without depleting its energy.

5. **Large-scale network size:** VANET networks can cover a city level road network, which is a very large-scale network. Moreover, high vehicle density makes this network size larger even in the case of small road networks.

Due to these unique VANET characteristics, many challenges need to be addressed to improve VANET performance. For example, high mobility in the VANET environment results in shorter communication link lifetimes between moving vehicles, especially for the high relative speed scenarios, such as vehicles moving in opposite directions on highways. Therefore, the MAC mechanism in VANETs is designed to support fast link establishment. Considering the time constraints required by some application such as safety application, the MAC mechanism in VANET is designed to support low latency communications to ensure the service reliability for safety applications even under high data traffic demand conditions. This feature is enabled in the VANET MAC by adding the QoS features to the MAC mechanism as described earlier.

An example of these challenges is the frequent neighborhood change due to high mobility. The frequent change must be considered, for example, in the routing algorithms, which should be designed to ensure the quality and continuity of services in such high dynamic communication environments.

One important challenge in VANET is high vehicle density, which results in high contention in the shared wireless medium. These challenges significantly increase when considering the larger coverage area allowed in VANET compared to other wireless LAN standards. Consequently, the medium access control algorithms in VANET must account for this high vehicle density and must be designed to support the intended service for different applications. The high vehicle density in VANT brings another challenge to routing because the higher the number of communication nodes (vehicles), the higher the data routing update volume and, consequently, the higher the routing overhead.

These challenges must be also considered when designing, developing, and deploying ITS services over VANET communication networks. For instance, time-sensitive ITS applications have to use the proper AC in the MAC layer such as audio or video ACs to satisfy the required time constraints. Moreover, ITS services design should consider the network scenarios in which these services are intended to work. For example, efficient navigation services are important in high congested road networks, in which vehicle density is expected to be very high, and here, communication performance is most probably low, including high packet drop rate and longer delay. Thus, such navigation services must be designed to work efficiently over this low communication performance.

2.2.4 Eco-routing Application Requirements and Communication Performance

Current network protocols are able to handle both failures and congestions. For example, some of the link layer protocol provides reliable services by supporting frame retransmission in the case of transmission error or collisions. In the higher level, routing protocols are able to handle communication link failures by finding alternative routes. One important failure recovery and congestion control is in the reliable transport service provided by the transmission control protocol (TCP) [35, 36] in the transport layer, which can handle any problems with the end-to-end packet delivery by identifying and retransmitting any dropped packets. Additionally, TCP is able to handle congestions in the communication network by utilizing congestion control mechanisms. However, the way the network handles failures and congestions does not completely satisfy the requirements

of traffic and services in the ITS applications [37]. According to the applications and services requirements published in [37], the vehicle routing application to private users can work properly with information delivery delay between 5 to 15 seconds at normal and low vehicular traffic congestion levels. In emergency conditions, this delay can be 25 to 40 seconds. Different applications have different time constraints and requirements, especially safety-related applications.

In our work, we focus on the feedback-based eco-routing navigation system as an ITS application. Our study in the next chapters shows that, in some scenarios, the eco-routing packet delivery delay can exceed these time requirements to hundreds of seconds. The reason is that in the case of V2I, vehicles may not be covered by RSUs. And in the case of congestion, the speed is very low. Considering these two reasons together, one can realize that vehicles may need a longer time to become connected. During these periods of disconnectivity, vehicles cannot send or receive navigation updates, which can result in longer information delivery delay to or from vehicles. This delay can significantly affect the performance of the eco-routing navigation system. On the other hand, by using V2V and hybrid communication techniques, this delay can be much shorter, because vehicles themselves work as relays for other vehicles. Thus, most of the time, vehicles will have connectivity either directly (through a direct connection to RSUs) or through other vehicles using multi-hop routes. However, in these cases, because vehicles relay information to/from other vehicles, the background data traffic overhead will be higher. In addition to this, the required routing protocols will also increase the background data traffic overhead in order to exchange network topology information. This high background packet generation rate directly affects communication performance by increasing the packet delay and drop rate.

Such degradation in communication performance can significantly influence the performance of the eco-routing navigation system, and consequently, the transportation system-wide performance.

2.3 Modeling of Connected Vehicles: Literature Review

The necessity of integrating a full-fledged traffic simulator with a wireless network simulator to model the cooperative ITS systems built on V2X communication platform has been recognized for a decade. A number of attempts have been made within recent years to develop an integrated traffic simulation platform that allows vehicles mobility conditions to be dynamically adapt to the wirelessly received messages. Two different approaches have been considered by researchers to facilitate this interoperability.

One common approach was to embed well-known vehicular mobility models into established net-

Table 2.1: Integrated Simulators Summary

Traffic Sim.	Network Sim.	Integrated Simulator
VISSIM	NS-2	MSIE [41]
SUMO	NS-2	TraNS [42]
SUMO	OMNET++	VEINS [43]
SUMO	NS-3	OVNIS [44]
SUMO	NS-3	iTETRIS [45]

work simulators. These features are sometimes combined with the original simulator as separate functional modules or APIs. For example, Choffnes et. al. [38] integrated the Street Random Waypoint (STRAW) model into the Java-built scalable communication network simulator SWANS, which allowed parsing of real street map data and modeling of complex intersection management strategies. A collection of application-aware SWANS modules, named ASH, were developed to incorporate the car-following and lane-changing models providing a platform for evaluating inter-vehicle Geo-cast protocols for ITS applications [38, 39]. Following a similar approach, the communication network simulator NCTUns extended its features to include road network construction and microscopic mobility models [40]. More recently, NS-3 has been engineered to incorporate real-time interaction between a wireless communications module and vehicular mobility models using a fast feedback loop.

A different approach is to integrate two standalone simulators - a traffic simulator coupled with a wireless network simulator. The choice of traffic simulators considered by the community for coupling in this manner included CORSIM, VISSIM, and SUMO, whereas network simulators ranged over NS-2, NS-3, QUALNET, and OMNET++. Table 2.1 summarizes some of these integration attempts.

CORSIM is a commercial traffic simulator that does not provide dynamic routing capabilities, while VISSIM does provide some dynamic routing capabilities. These are limited compared to the INTEGRATION software, which provides a total of ten different routing strategies ranging from feedback to predictive dynamic routing. Consequently, both CORSIM and VISSIM do not provide sufficient routing algorithms for testing in a connected vehicle environment. The first attempt of integrating two independent open source traffic and wireless simulators was TraNS (Traffic and Network Simulation Environment) [42], which combined SUMO and NS-2. Later, VEINS [43] also adopted the open source approach of TraNS by combining the network simulator OMNET++ with SUMO. VEINS allowed for interaction between the two simulators by implementing an interface module inside OMNET++ that sends traffic mobility updating commands to SUMO. For example, VEINS could impose a given driving behavior to a particular vehicle upon receiving

wireless messages from another vehicle. Most recently, the Online Vehicular Network Integrated Simulation (OVNIS) [44] platform was developed, which coupled SUMO and NS-3 together and included an NS-3 module for incorporating user-defined cooperative ITS applications. OVNIS extends NS-3 as a ”traffic aware network manager” to control the relative interactions between the connected blocks during the simulation process. Last but not least, iTETRIS [45] moves one step beyond the state-of-the-art solutions and overcomes one limitation that is present in Trans, VEINS, and OVNIS by providing a generic central control system named iCS to connect an open-source traffic simulator with a network simulator, without having to modify the internal modules of the interconnected simulation platforms.

In this area, we developed two simulation platforms. The first, VNetIntSim, is capable of modeling the vehicular communication in small and medium size. It can capture low-level details in both communication and transportation systems. This model will be presented in Chapter 3 and then developed to include the eco-routing application to realistically model the impact of the communication on the eco-routing performance in Chapter 4. The second platform utilizes an analytical model to estimate the communication parameters in the vehicular network based on the surrounding network conditions, which is presented and used in Chapter 5.

2.4 Eco-Routing

Previous studies have shown that standard navigation systems can provide travelers accurate minimum path calculations based on either the shortest distance or the shortest travel time and that, by using these systems, we can achieve some fuel savings [46]. However, previous studies have also shown many cases where the shortest distance route can produce higher fuel consumption and emission levels due to road grade or higher congestion levels. Similarly, using the shortest travel time routes may also result in higher fuel consumption and emission levels. An example for this case was demonstrated in Ahn and Rakha [47], where significant savings in fuel consumption and emission levels were produced by using longer-time and shorter-distance arterial routes.

In the last two decades, the environmental and economical impacts of the transportation sector attracted the attention of scholar community. So, the research community devoted much research effort to sustainable mobility to save fuel consumption and emissions. Thus, eco-routing navigation techniques were introduced to minimize fuel consumption and emissions in transportation systems by utilizing the route fuel cost as a metric, based on which the most environmentally friendly route will be selected.

Developing and deploying eco-routing navigation techniques is very challenging. One major challenge is the estimation of the route fuel cost. This challenge comes from the fact that the route fuel cost is a function of many parameters, including route characteristics (i.e., length, maximum speed, grade), vehicle characteristics (e.g., weight, shape, engine, and power), and driving behavior. It has been proven that it is too difficult to combine all these parameters in one model, especially because many of these parameters are stochastic and there is a complex dependency among all of them. Therefore, the best way to calculate the route cost is to use feedback from the vehicles moving on these routes. These data can be collected in real-time and fused with historical data to estimate the route fuel cost and, consequently, calculate the best route for the vehicles traveling in real time [48, 49]. This feedback system is simple and accurately estimates the route cost. But, on the other hand, it requires a communication infrastructure through which this information can be exchanged. In addition to communication, vehicles should be capable of quantifying the fuel consumption for each road link.

Eco-routing was proposed in [17] to select the route with the lowest total fuel consumption and thus the lowest total CO_2 emissions. The authors in [17] used the street network in the city of Lund in Sweden. They classified the consumption factor into 22 street classes for peak and none peak hours and used 3 vehicles classes. This results in a 4% average saving in fuel consumption. Ahn and Rakha in [50] showed the importance of route selection on fuel and environment, by demonstrating through field tests that an emission and energy optimized traffic assignment based on speed profiles can reduce CO_2 emissions by 14% to 18% and fuel consumption by 17% to 25% over the standard user equilibrium and system optimum assignment. In [46] is an attempt to minimize the vehicle fuel consumption and emission levels; the authors proposed a new set of cost functions that include fuel consumption and emission levels for road links. In [48], the authors introduced a stochastic, multi-class, dynamic traffic assignment framework for simulating eco-routing using the INTEGRATION software [51]. They demonstrated fuel savings of approximately 15% using two scenarios. In [49], the authors developed an eco-routing navigation system that uses both the historical and real-time traffic information to calculate the link fuel consumption levels and then to select the fuel-optimum route. However, the authors did not consider the communication network and its influence on system performance.

2.4.1 Eco-routing User-Equilibrium Model

All previous eco-routing models use shortest path techniques, so they are basically user-equilibrium models, which try to minimize individual vehicles fuel consumption. The user-equilibrium model

for eco-routing can be defined as follows. Given a road network directed connected graph $G(N,L,C)$, where $N = \{1,2,...,n\}$ is a set of n road network nodes, $L = \{l_{ij} : i,j \in N\}$ is a set of l directed road links, and the road link costs $C = \{C_{ij} : L_{ij} \in L\}$ is a positive real-valued cost function $C : L \rightarrow \mathbb{R}^{+}$. Let a vehicle trip from source node $s \in N$ to destination node $d \in N$. Then, the user-equilibrium eco-routing computes the path P, which is a sequence of road links $P \subset L$, for this individual vehicle that minimizes the vehicle fuel consumption i.e., *minimize* $\sum_{l_{ij} \in P} C_{ij}$. This shortest path problem can be easily solved using the Dijkstra's algorithm [52].

This problem can be also solved using integer linear programming as shown in equation 2.1. The variable x_{ij} is either 0 or 1 that identifies whether link l_{ij} will be included in the shortest path.

$$\begin{aligned} &\text{minimize} \quad \sum_{ij \in N} x_{ij} C_{ij} \\ &\text{subject to :} \quad \sum_{j} x_{ij} - \sum_{j} x_{ji} = \begin{cases} 1 & \text{if } i = s \\ -1 & \text{if } i = d \\ 0 & otherwise \end{cases} \quad \forall\, i \in N;\ k \in F, \end{aligned} \tag{2.1}$$

This integer linear program can only work for single traffic demand, i.e., all vehicles move from the same source to the same destination.

All previous eco-routing implementations use shortest path algorithm and focus on how to compute the cost function, C. One technique, which is used in [49, 53, 54], is to develop a mathematical model based on road link parameters (vehicle density and vehicle speeds). However, developing mathematical model for fuel cost that captures all transportation phenomena and stochasticity is too hard, as we mentioned earlier. Thus, one accurate and simple technique is to use real-time feedback from vehicles currently on the network. This feedback technique is enabled and facilitated by the vehicular communication technology, which enables vehicles to communicate to each other or to centralized entities.

A general disadvantage of the user-equilibrium models is that it can result in overloading the best routes, which increases the system-wide fuel consumption. These user-equilibrium models try to overcome this problem by periodically updating the link costs and recompute the routes, which result in changing the best routes, consequently reducing the load on the previous best routes. But on the other hand, these periodic updates produce the vehicle route oscillation problem, that we will describe in detail in section 2.5.4.

In Chapter 7, we utilize the feedback technique to develop a system-optimum eco-routing model,

which considers the vehicle traffic load on each road link and current traffic demand levels in addition to the fuel consumption costs. The model we developed supports multiple traffic demands from different source to different destinations. Thus, we developed it as a linear program and used stochastic route assignment to build the vehicle routes.

2.4.2 Eco-routing: Mobility and Communication

Feedback-based eco-routing is an example of ITS applications that have both directions of interaction between mobility and communication network. In one direction, the routes of the vehicles that are generated by the TMC determine the vehicles' mobility trajectories and the vehicle spatial distribution on the road network, which are important mobility parameters that can affect the communication performance. In the other direction, these routing decisions are made based on the data collected from the moving vehicles. The correctness and the accuracy of these routing decisions are determined by the completeness and accuracy of the data, which are affected by packet drop rate and delay in the communication network.

Thus, in this thesis, we are interested in feedback-based eco-routing and the mutual impact of the communication network performance and eco-routing performance. To accurately capture this, we need a high fidelity model especially for the eco-routing parameters, namely the link fuel consumption cost. So, to achieve high fidelity levels, we decided to use a microscopic traffic modeling tool, INTEGRATION software [48], which uses the VT-Micro model [55] for computing the fuel consumption and emissions based on the vehicles instantaneous speed and acceleration. The second-by-second instantaneous speed and acceleration enable the model to capture events that can affect the fuel consumption, thereby, accurately estimating the link costs.

The next section gives an overview of the eco-routing implementation in the INTEGRATION software and the fuel consumption model it uses.

2.5 Eco-routing in INTEGRATION software

In the INTEGRATION framework, eco-routing is developed as a feedback system. It assumes that the vehicles are equipped with GPS systems and communication systems. Moreover, vehicles are assumed to be capable of calculating the fuel consumption for each road link it traverses and communicating this information to the TMC. In INTEGRATION, the fuel consumption rate and emission rates of each vehicle are calculated every second based on the instantaneous speed and

acceleration. As shown in Figure 2.2, every decisecond, the speed and the acceleration of each vehicle are calculated as well as the vehicle's fuel consumption rate. The fuel consumption is accumulated for each road link. Then, whenever the vehicle finishes that link, it updates the link cost in the TMC directly.

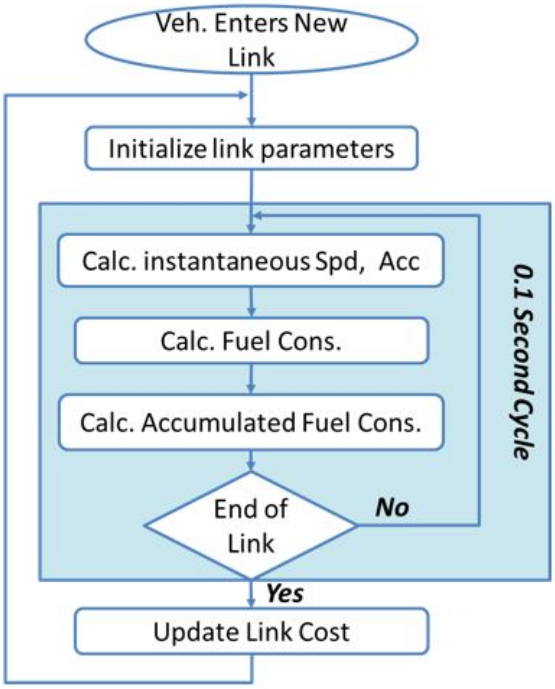


Figure 2.2: Eco-routing without the Communication

2.5.1 Estimating Vehicle Fuel Consumption

The INTEGRATION modeler computes a number of measures every decisecond including the fuel consumed, vehicle emissions of carbon dioxide (CO_2), carbon monoxide (CO), hydrocarbons (HC), oxides of nitrogen (NOx), and particulate matter (PM) [48]. The granularity of decisecond computations permits the steady-state fuel consumption rate for each vehicle to be computed each second on the basis of its current instantaneous speed and acceleration level. INTEGRATION computes the fuel consumption and emission levels using the VT-Micro model [56] which was developed as a statistical model from experimentation with numerous polynomial combinations of

speed and acceleration levels to construct a dual-regime model, as demonstrated in Equation 2.2.

$$F(t) = \begin{cases} exp\left(\sum_{i=1}^{3}\sum_{j=1}^{3} L_{i,j} v^i a^j\right) & if \quad a \geqslant 0 \\ exp\left(\sum_{i=1}^{3}\sum_{j=1}^{3} M_{i,j} v^i a^j\right) & if \quad a < 0 \end{cases} \tag{2.2}$$

Where $L_{i,j}$ are model regression coefficients at speed exponent i and acceleration exponent j, $M_{i,j}$ are model regression coefficients at speed exponent i and acceleration exponent j, v is the instantaneous vehicle speed in (km/h), and a is the instantaneous vehicle acceleration (km/h/s) [56].

To calculate the vehicle speed, the INTEGRATION software uses the Van Aerde car-following model [57] which considers the headway between the subject vehicle and the leading one, while considering the speed of this leading vehicle as well. Based on the current and new speed, the required acceleration can be computed. However, INTEGRATION ensures a realistic estimation of the vehicle acceleration by using a variable power vehicle dynamics model to estimate the vehicle's tractive force. Consequently, it implicitly accounts for gear-shifting on vehicle acceleration. More specifically, the model computes the vehicle's tractive effort, aerodynamic, rolling, and grade-resistance forces, as described in details in the literature [58], [59]. More details about this car following model will be introduced in Chapter 6

2.5.2 Initial Route Assignment

Initially, when the network is empty, the fuel consumption is computed assuming the vehicle travels at the roadway free-flow speed. The rationale is that the routes are initially empty and the vehicles will use the maximum speed. INTEGRATION also uses the route grades such that $a_{l_m} = gG_{l_m}$, where g is the gravitational acceleration $(9.8067 m/s^2)$ and G_{l_m} is the grade along link l_m.

This assignment will be then updated based on the information received from the individual vehicles, where each vehicle is assumed to be able to measure its fuel consumption level along each link and able to send this link fuel-cost to the TMC.

2.5.3 Updating the Route Cost

If route R_n consists of k links, the total route fuel consumption level F_{R_n} is the summation of the fuel consumption of the constituting links as expressed in Equation 2.3

$$F_{R_n}(t) = \sum_{m=1}^{k} F_{l_m}(t) \tag{2.3}$$

where F_{l_m} is the fuel cost for the link l_m at the current time.

As mentioned before, F_{l_m} is based on the free flow speed cost. Then this value is updated whenever a vehicle exits link l_m. INTEGRATION uses a smoothing factor α to update link fuel consumption cost. So, whenever the TMC receives a new cost C_m for link l_m it uses α to update the links cost as:

$$F_{l_m}(t+1) = (1-\alpha)\, F_{l_m}(t) + \alpha\, C_m \tag{2.4}$$

A typical value of α in INTEGRATION is 0.2.

2.5.4 Route Oscillation

INTEGRATION periodically updates the routing trees. This updating period is a user-specified parameter. During this updating interval, the link costs are continuously updated based on the cost information received from the en-route vehicles.

It is important to mention that the eco-routing implemented in INTEGRATION is a user equilibrium algorithm. That is, every vehicle tries to minimize its fuel consumption by using the best available route from its current location to it destination based on the latest routing tree. Experimental results showed that using the same link cost F_{l_m} for all the traffic demands in this user equilibrium technique results in overloading the best route during the tree updating interval while leaving the other route underutilized. This over-utilization of the best route produces higher congestion level on this best route, resulting in lower speed and higher fuel consumption on the links constituting it. So, in the next updating interval, this route will be of a high cost, consequently, vehicles will prefer another optimum route. This process produces route temporal oscillation among a set of route choices.

To avoid this temporal oscillation across route choices, the user can introduce a white noise error function into the link cost function. More specifically, a user-defined error factor $\mho$ is used as a coefficient of variation to add some error to the fuel cost of the links, subsequently, to the tree building and the route selection algorithms. The error value added to the link cost is a randomly selected point from the standard normal distribution $N(0,\sigma)$, where σ is the standard deviation and $\sigma = \mho\, F_{l_m}$.

If the cost along alternative routes are very close, this white noise allows vehicles to select slightly sub-optimum routes, and, thus, distribute the traffic across a set of alternative sub-optimum routes instead of using a single optimal route. Simulation results showed that in most cases using $\mho$ values between 0.05 and 0.2 always give lower fuel consumption than using the exact link costs.

2.6 Modeling of Large-Scale Transportation Networks

The benefits of modeling large-scale transportation networks have attracted the scholars' attention over the last three decades. They use many techniques to model transportation systems. These techniques can be categorized into three basic simulation techniques; macroscopic, microscopic and microscopic modeling algorithms.

2.6.1 Traffic Modeling Techniques

Macroscopic modeling techniques compute the average number of vehicles traversing each road links based on which the average link parameters can be computed, including speed, travel time, fuel consumption, and emissions. Because of this simplicity, macroscopic models are characterized by high scalability and high simulation speed. However, the high scalability and speed come at the cost of the accuracy because by using average link parameters, many transportation phenomena cannot be captured by the model such as acceleration/deceleration events, which is the predominant factor affecting the fuel consumption and emission levels. Macroscopic systems also cannot model the interactivity between vehicles such as lane changing, gap acceptance techniques, etc. Therefore, macroscopic models lack the needed accuracy.

On the other extreme, the microscopic models achieve the highest possible accuracy at the cost of the computation and scalability. To achieve the highest accuracy, microscopic models track every individual vehicle on a second-by-second basis. Microscopic models can model and capture all the events that happen in the transportation network, as well as model the interactivity between

the vehicles on the road as well as between vehicles and control systems such as traffic signals. However, the computation and memory requirements for this level of accuracy limit the scalability and reduces the simulation speed.

Between these two extremes, the mesoscopic models are located, by which each road link is represented by a queue. These mesoscopic models can estimate a time-varying average for the link parameters based on the number of vehicles in the queue. By the help of the queuing theory, mesoscopic models can compute different travel times for different vehicles on the same link at the same time. Therefore, mesoscopic models are more accurate than the macroscopic models. However, because mesoscopic models are still using average parameters, they are still of lower accuracy compared to microscopic models.

2.6.2 Literature of Modeling of Large-scale Transportation Systems

In the last three decades, there are many attempts to build systems that are capable of simulating large-scale transportation systems. In 1997, the TRANSIMS simulation tool [60] was used to simulate the traffic in large areas for traffic planning purposes. The research work in [60] uses discrete space modeling for the traffic micro-simulation based on the cellular automaton approach [61], where the road is separated into cells (of length 7.5 meters) which are either empty or occupied by one car. It uses a simple algorithm for car following and lane changing. The use of cellular automaton makes this system fast, however, it cannot accurately capture observed transportation phenomena including car following, lane changing, and gap acceptance. In 2002, TRANSIMS was updated to better include the impact of the congestion on the system performance and it was run on a parallel cluster for fifty iterations to achieve better trip planning [62]. TRANSIMS has been used to model the Switzerland network in the morning peak hours using parallel computation [63], [64]. Then, in 2012, TRANSIMS was used in [65] to evaluate the performance of the transportation network of the Buffalo-Niagara metropolitan area during significant snow events. However, the authors mentioned that extensive efforts are required to make the simulated network realistic in terms of network configuration, lane connectivity, pocket lane and signal locations. In [66] the same modeler was used to evaluate the impact of dynamic routing on the fuel consumption.

A hybrid traffic modeler was presented in [67], [68], [69], [70] to model large-scale traffic networks. The hybrid modeler simulates different network links with different fidelity levels (microscopic, or mesoscopic levels), where microscopic simulation was applied to areas of specific interest while simulating a large surrounding network in lesser detail with a mesoscopic model. In this way, it can provide a customized performance and simulation speed.

In [71], [72], the authors used the INTEGRATION software to fully microscopically model the dynamic routing on the fuel consumption in the downtown Cleveland and Columbus, Ohio, USA, in the case of different system market penetration rates and congestion levels. The network has about 3,000 links with a traffic demand of 65,000 vehicles per hour during the morning peak hour. Our proposed framework uses parallel INTEGRATION instances enabling our system to capture the morning commute of 1.2M vehicles. In 2015, the authors of [73] proposed the Scalable Electro-Mobility Simulation (SEMSim), an architecture for a cloud-based platform, as a proof of concept to use the cloud for simulation of large-scale transportation systems. The authors used this model to simulate the network of Singapore that has about 500,000 private owned vehicle. However, the model uses simple vehicle characteristics (e.g., kinematic model) and driving behavior models.

MATSim [74] is a transportation simulation software designed for large-scale scenarios, it adopts a queue-based approach which is computationally efficient. But the queue-based approach lacks the accuracy because the car-following effects are not captured. To increase the computation efficiency, MATSim combines the waiting-queue approach with an event-based approach. However, this event based technique still lacks the accuracy because it does not consider the car following. For example, links do not have to be processed while agents traverse them.

Under the umbrella of modeling of large-scale transportation systems, we developed a novel multi-modal agent-based framework that is capable of modeling city level system. This model uses a hybrid simulation technique and will be presented in Chapter 6.

Chapter 3

An Integrated Architecture for Simulation and Modeling of Small- and Medium-sized Transportation and Communication Networks

In this chapter, the vehicular network integrated simulator (VNetIntSim) is introduced as a new transportation network and VANET simulation tool by integrating transportation and VANET modeling. Specifically, it integrates the OPNET software, a communication network simulator, and the INTEGRATION software, which is a microscopic traffic simulator. The INTEGRATION software simulates the movement of travelers and vehicles, while the OPNET software models the data exchange through the communication system. Information is exchanged between the two simulators as needed. The chapter describes the implementation and the operation details of the VNetIntSim as well as the features it supports such as multi-class support and vehicle-object reuse. Subsequently, VNetIntSim is used to quantify the impact of mobility parameters (vehicular traffic stream speed and density) on the communication system performance considering Transmission Control Protocol (TCP) [35, 36] and User Datagram Protocol (UDP) [75] applications. The routing performance (packet drops and route discovery time), IP processing delay in case of a file transfer protocol (FTP) application, and jitter in case of a Voice over Internet Protocol (VoIP) application are evaluated.

3.1 Introduction

Vehicular Ad Hoc Networks (VANETs) and intelligent transportation systems (ITSs) have a wide spectrum of applications, algorithms, and protocols that are important for the public, commercial, environmental, and scientific communities. From the communication perspective, these applications range from on-road-content-sharing (Li, et al. [76]), entertainment-based, and location-based services [77]. From the transportation perspective, these applications include safety applications [26], cooperative driving and warning applications [78], traffic control and management [79], fuel consumption and carbon emission minimization applications [80], speed harmonization [81], road traffic congestion detection and management [82], and taxi/transit services [83]. This wide application spectrum demonstrates the importance of these systems.

On the other hand, evaluating these systems is challenging, not only because of the cost needed to implement these systems due to the need for a large number of vehicles equipped with communication devices, the required communication infrastructure, and the required signal controllers, but also because of the need for road networks to run the required experiments. A third reason is that some applications/algorithms work in special conditions of either weather and/or traffic congestion, which are not easily provided. Fourthly, and most importantly, the failures in some of these applications may result in loss of lives of the participants.

Thus, currently, the best approach to study these systems is to use simulation tools. However, simulating ITS and VANET systems is challenging. The reason is that these systems cover two fields, namely, the transportation field and the communication field. The transportation field includes the modeling of vehicle mobility applications including traffic routing, car-following, lane-changing, vehicle dynamics, driver behavior, and traffic signal control, in both macroscopic and microscopic modeling scales. The other main field is the data and communication network modeling that includes data packet flow, vehicle-to-vehicle (V2V) communication as well as vehicle-to-infrastructure (V2I) communication, wireless media access, data transportation, data security, and other components. These two fields are not distinct or isolated, but, instead, they are interdependent and influence one another. For example, vehicle mobility, speeds, and density affect the communication links between vehicles [26] as well as the data routes and hence the communication quality (i.e., reliability, throughput and delay) [84] which was demonstrated in [85], in which the authors used archived global positioning system (GPS) traces to model the multi-hop V2V connectivity in urban vehicular networks, and they revealed many interesting characteristics of network partitioning, end-to-end delay, and reachability of time-critical V2V messages. In the opposite direction, the number of packet losses between vehicles and the delivery delay will af-

fect the accuracy of the data collected and hence the correctness of the decisions made by the ITS system. Taking into consideration the complexity of each system (transportation and communication), in addition to the high and complex interdependency level between them, we can see how challenging the modeling and simulation of VANET and ITS are.

In Chapter 2, Section 2.3, we gave an overview of the literature research in modeling of the connected vehicles. In addition to these models presented in Section 2.3, there are other approaches; most of them are based on using fixed mobility trajectories that are fed to the communication network simulator. These trajectories may be generated off-line using a traffic simulator platform or extracted from empirical data sets. This simulation paradigm is useful for single directional influence (i.e., studying the effect of mobility on the network and data communication) such as data dissemination in VANET. However, this approach cannot capture the opposite direction of influence (i.e., the effect of the communication network performance on the transportation system) such as vehicle speed control in the vicinity of traffic signals, where vehicles and the signal controllers exchange information to compute and optimize vehicle trajectory. These interactions have to be run in real-time to accurately model the various component interactions.

In this chapter, we introduce a new framework for modeling and simulating an integrated VANET and ITS platform. This new framework has the capability of simulating the full VANET/ITS system with full interdependence between the communication and transportation systems and hence allows for the analysis of VANET and/or ITS applications and algorithms with any level of interaction or interdependence between them. This framework integrates two simulators, namely, the INTEGRATION [51] as microscopic traffic simulator and the OPNET modeler [86] as the data and communication simulator by establishing a two-way communication channel between the models. Through this communication channel, the two simulators can interact to fully model any VANET/ITS application. Subsequently, the developed framework is used to study the effect of different traffic characteristics (traffic stream speed and density) on V2V and V2I communication performance.

The chapter is organized as follows. Section 3.2 provides a brief description of the OPNET and INTEGRATION softwares. Subsequently, the VNetIntSim operation and how the two simulators interact is described in Section 3.3. The architecture of the VNetIntSim and the implementation of the proposed framework is presented in Section 3.4. A simulation case study is presented and discussed in Section 3.5, in which the VNetIntSim is used to study the effect of various traffic mobility measures on communication performance. Finally, conclusions of the study and future research directions are presented in Section 3.6.

3.2 VNetIntSim Components

VNetIntSim uses the concept of separation between the internal simulation modules and the new modules that were added to support the model integration. This feature is actually inherited from the two simulators we selected for the VNetIntSim. INTEGRATION is fully built in a modular fashion with a master module that manages and controls all the modules. And OPNET supports full modularity in its different levels (Network, Node, and process). The interaction between different modules is modeled using interfaces between them. Consequently, updating any module will not affect the others as long as this interface does not change.

OPNET and INTEGRATION have their unique features compared to the other simulators. The following subsections give an overview on both of them.

3.2.1 OPNET Modeler

The OPNET Modeler is a powerful simulation tool for specification, simulation, and analysis of data and communication networks [86]. OPNET combines finite state machines with analytical models. The modeling in OPNET uses Hierarchical Modeling, which has a set of editors (Network, Node, and Process editors), all of which support model level reuse. The most important OPNET characteristic is that its implementation for many standard protocols have been tested and validated before publishing. OPNET is built in a hierarchical modular fashion at all its levels (network, nodes, links, and processes). The network consists of a set of nodes and links. Each node consists of a set of process modules. The process modules interact through interrupts and the associated Interface Control Information (ICI). The modules that are added to the simulators in this research effort maintain the same concept, so that updating the simulators does not affect the integration between them.

Compared to NS-2 and NS-3, OPNET has three features; 1) a well-engineered user interface that allows for easy building and managing of different simulation scenarios, 2) the OPNET modeler provides its powerful debugging capabilities, and 3) OPNET supports a visualization tool that allows for tracking data packets within the nodes. OMNET++ is a simulation framework that does not have modules. However, there are many open source frameworks based on OMNET++ that implement different modules such as VEINS [43]. In VEINS, the update interval is 1 second, which is a long interval from the communication perspective. For example, if the speed of the vehicle is 60 km/h (37.28 mi/h), which is a common speed in cities, this update interval corresponds to 16.6 m, which is a long distance that can affect the communication between vehicles. In our

model, the OPNET simulator runs totally using discrete event simulation, which has a very small time granularity, and the INTEGRATION has time step of 1 decisecond, which corresponds to 1.6 meters longitudinal step at 60 km/h vehicle speed.

3.2.2 INTEGRATION Software

The INTEGRATION software is agent-based microscopic traffic assignment and simulation software [51]. INTEGRATION is capable of simulating large-scale networks up to 10,000 road links and 500,000 vehicle departures with a time granularity of 0.1 seconds. This granularity allows detailed analyses of acceleration, deceleration, lane-changing movements, and car following behavior. It also permits considerable flexibility in representing spatial and temporal variations in traffic conditions. These are very important characteristics needed when studying the vehicular communication in the dynamic vehicular networks.

The INTEGRATION software computes a number of measures of performance including vehicle delay, stops, fuel consumption, hydrocarbon, carbon monoxide, carbon dioxide, and nitrous oxides emissions, and the crash risk for 14 crash types [51].

INTEGRATION supports many features, such as dynamic vehicle routing and dynamic eco-routing [17], eco-drive systems, eco-cruise control systems, vehicle dynamics, and other features that are not supported by other traffic simulation softwares. The INTEGRATION model has been developed over three decades and has been extensively tested and validated against empirical data and traffic flow theory. Furthermore, the INTEGRATION software is the only software that models vehicle dynamics and estimates mobility, energy, environmental, and safety measures of effectiveness. The model also includes various connected vehicle applications including cooperative adaptive cruise control systems, dynamic vehicle routing, speed harmonization, and eco-cooperative cruise control systems.

3.3 VNetIntSim Operation

To integrate the two simulators together, they should be consistently initialized and synchronized. Then some information must be exchanged to reflect the communication to the transportation modeler, meanwhile to reflect the mobility to the communication simulator. Thus, this section describes the detailed operation of the VNetIntSim.

3.3.1 Integrating OPNET and INTEGRATION

The main idea behind VNetIntSim is to use the advantages of both the INTEGRATION and OPNET platforms by establishing a two-way communication channel between them. Through this channel, the required information is exchanged. The basic and necessary information that should be exchanged periodically is vehicle locations. The locations of vehicles are calculated in INTEGRATION every decisecond and transmitted to the OPNET modeler, which updates the vehicle locations while they are communicating.

In this version of VNetIntSim, the communication channel between OPNET and INTEGRATION is established by using shared memory, as we will explain in the next section. The shared memory supports the required speed and communication reliability between the two simulators.

3.3.2 Initialization and Synchronization

When starting the simulators, and before starting the simulation process, the two simulators should initialize the communication channel using two-way Hello Messages. After establishing the connection, the two simulators synchronize the simulation parameters: simulation duration, network map size, location update interval, maximum number of concurrent running vehicles and number of signals. In this synchronization phase, INTEGRATION serves as a master and OPNET serves as a slave, i.e., values of these parameters in OPNET should match those calculated in INTEGRATION. Mismatching in some of these parameters (such as simulation duration, number of fixed signal controllers and the maximum number of concurrent running vehicles) will result in stopping the simulators. In this case, the OPNET software sends a Synchronization Error message to the INTEGRATION software. This behavior guarantees the consistency of the operation and the results collected in both systems. Additional parameters allow some tolerances. For example, the map size in OPNET should be greater than or equal to that in INTEGRATION, the rationale behind this is the road-map is configured in the INTEGRATION, consequently, the mobility and the vehicle locations are determined and controlled by the INTEGRATION software.

After successful synchronization, the simulation process should start by exchanging the simulation start message sent from OPNET. OPNET starts the simulation by initializing its scenario components and initializing the vehicles locations and status. The component initialization takes place by sending a start simulation interrupt to each module in each component in the scenario (i.e., routers, hosts, and vehicles). The purpose of this interrupt is to read the configuration parameters, initialize the module state variables, and invoke the appropriate processes based on the configuration. After

Table 3.1: Message Codes

Code	Function
01	Initialization; Hello Message
02	Initialization : Connection Refused
10	Parameter Synchronization
11	Synchronization Error
30	Signal Locations Request
31	Signal Locations Updates
40	Start Simulation
50	Locations Information Request
51	Locations Information Updates
60	Speed Information Request
61	Speed Information Updates
99	Termination Notification

this initialization, OPNET finds the vehicle nodes in the scenario and maps each one to a vehicle ID in the INTEGRATION software. Using this mapping, each vehicle in OPNET corresponds to only one vehicle in INTEGRATION. However, this behavior can be overridden as described in the next section. Then, OPNET disables all the vehicles, which means that all the vehicles will be inactive. After that, OPNET enables vehicles based on the information it receives from INTEGRATION. The vehicle in OPNET is a mobile node that we customized by adding new attributes such as speed, acceleration, and movement direction. Also, we added some modules to this vehicle node to represent some vehicular applications such as the eco-routing module that implements the eco-routing algorithm [17], whose detailed implementation and operation will be described in Chapter 4.

During the simulation phases, there are many types of messages that can be exchanged between the two simulators. Each message type has its unique code. Based on the code, the message fields are determined. Table 3.1 shows the different message codes. The gaps between the code values allow for the addition of new functionalities in the future.

3.3.3 Vehicles Location Updating

During the simulation, the INTEGRATION software computes the new vehicle's coordinates. To calculate the vehicle's new location, INTEGRATION uses a vehicle dynamics model that uses the vehicle characteristics such as weight, maximum speed, maximum acceleration, and maximum engine power. The vehicle dynamics model also accounts for the different forces affecting the ve-

hicle's mobility, including the vehicles tractive effort, aerodynamic, rolling, and grade resistance forces. After computing the new vehicle locations for all the vehicles, INTEGRATION sends them to the OPNET software, which updates the location of each vehicle, as shown in Figure 3.1. This cycle is repeated each update_interval, which is typically 0.1 seconds. The time synchronization during location updating is achieved in two ways, 1) using two semaphores (intgrat_made_update and opnet_made_update) one for each simulator, 2) at each update time step the INTEGRATION software sends the current simulation time to OPNET. If it does not match the OPNET time, OPNET will take the proper action to resolve this inconsistency. Figure 3.2 shows the flow chart for the basic location update process. In each location update cycle, the INTEGRATION software computes the updated vehicle locations. Subsequently, it checks whether the last update has been copied (intgrat_made_update = 0). If so, it writes the new update to the shared memory and sets the intgrat_made_update flag to 1. Then, it waits for new updates.

When it receives a new update, if the received time equals to its current time, the Driver process in OPNET will copy the locations, set the intgrat_made_update flag to 0, and then move the vehicles to the new locations. If the received time is greater than the OPNET current time, it schedules the process to be executed again at the received time. In this case, OPNET does not change the control flags ($intgrat\_made\_update or opnet\_made\_update$), which causes INTEGRATION to wait until the OPNET finishes processing, reaches this time step, and allows the simulation to proceed. If the received time is less than the current time, OPNET discards this update.

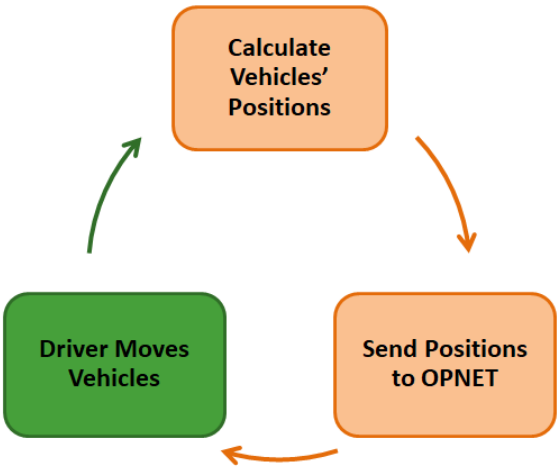


Figure 3.1: Location update cycle

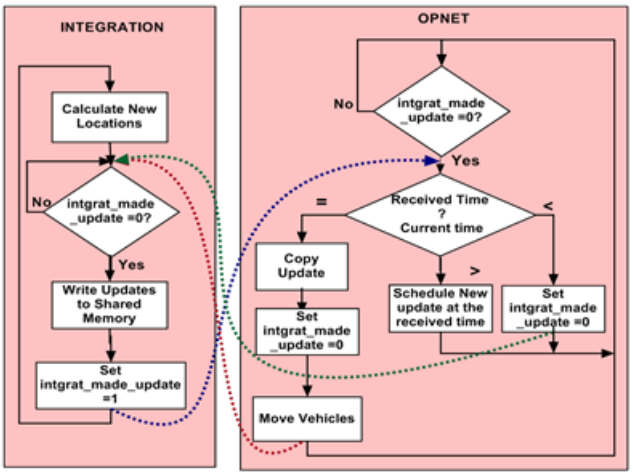


Figure 3.2: VNetIntSim basic operation

3.3.4 Application Communication

The basic operation described above is only for updating locations, which is the core of the VNetIntSim platform. However, ITS applications need the exchange of other types of information that reflect the communication results to INTEGRATION. The type of this information and how and/or when it should be exchanged depend mainly on the application itself. Thus, the application specifications should define what other information should be exchanged, as well as how and when it should be exchanged.

The applications will use the established communication channel to exchange the required information. VNetIntSim supports simultaneous multi-applications, where each application can use one or more codes to support its functionalities. Figure 3.3 shows the complete communication cycle when running an application.

For example, in variable speed control systems, the integration will move the vehicles. Then, in OPNET, the vehicles and signals communicate the speed information. Based on the exchanged information, each vehicle finds its new speed. These new speeds should be sent to the INTEGRATION software, which computes the updated parameters (i.e. acceleration or deceleration) and then computes the updated vehicle locations.

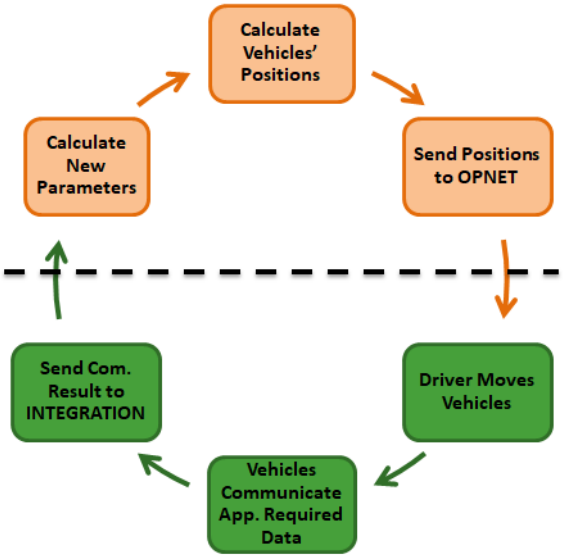


Figure 3.3: Complete communication cycle

3.4 Architecture, Implementation, and Features of VNetIntSim

This section describes the architecture and the detailed implementation of the modeler and then the supported features in the current version.

3.4.1 Architecture and Implementation

Figure 3.4 shows the VNetIntSim architecture and the modules that were added to each simulator (dashed boxes). Within INTEGRATION, the Configuration Reader Module reads the input files. Based on the configuration, it generates an XML topology file for OPNET. This topology file contains the vehicle specifications and signal controller locations as well as the application and profile specifications. This file is used by OPNET to generate its network scenarios.

Inter-process Communication

The first issue that arises during implementation entails identifying the inter-process communication mechanism that should be used to connect the simulators. In VNetIntSim, two methods were selected, namely, TCP sockets and shared memory. Each of these methods has its advantages over other methods. The shared memory approach supports very high-speed communication, which is

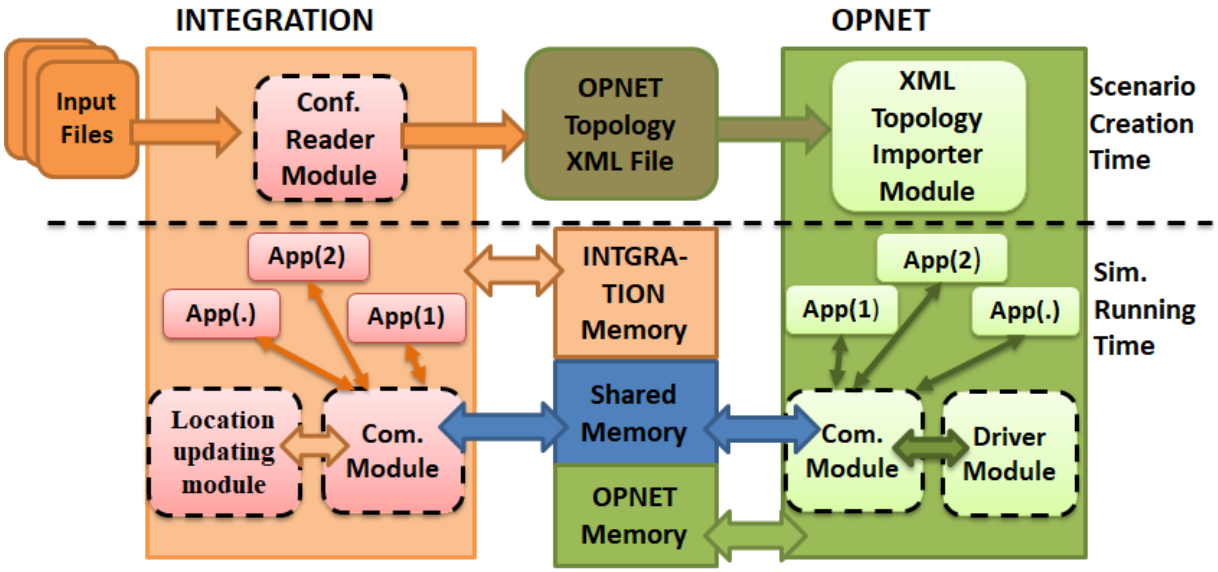


Figure 3.4: Architecture for integration OPNET and INTEGRATION

needed when modeling large simulation networks. In addition, the operating system manages the mutual execution of this shared memory so this does not need to be considered.

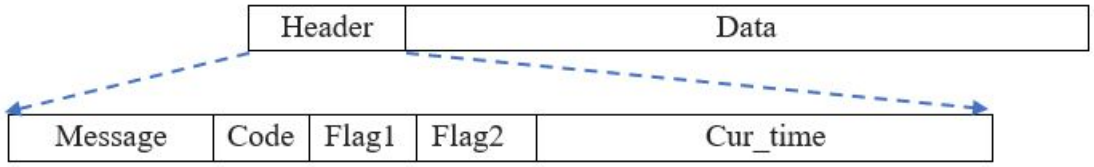
However, it is limited by the machine capabilities in terms of processing speed and memory size. On the other hand, TCP sockets provide more flexibility so that the INTEGRATION software can be connected to any other simulator on a different OS/machine, in addition to the processing capabilities that will be gained from the other machine. However, TCP sockets introduce network dynamics and delay problems to the simulation process, which may result in some communication delay. Consequently, the approach used in this chapter is the shared memory approach. In future, we plan to implement the TCP socket communication.

The Communication Module

In each of the two simulators, a Communication module is created. These two modules are responsible for 1) establishing the communication channel by creating a shared memory, 2) exchanging information between the two simulators through the shared memory, 3) addressing the applications using message codes shown in Table 2, based on the received code, the Communication module forwards the data to the appropriate application, 4) synchronizing the communication, and 5) control the data integrity against damages or losses by using intgrat_made_update and opnet_made_update semaphores, one for each direction.

The Communication module is an abstraction for the communication functionality in both simulation sides. It provides an interface that enables other functionalities to send/receive information to/from the other simulator. The Communication module has direct access to the shared memory space, while all the other modules (i.e., applications) use this communication interface to communicate with the other side.

Since the Communication module provides the communication services for different functions, the data that are transfered by a Communication module is dependent on these functions in terms of size and content. The Communication module provides a unified interface to all other modules to send their data. This interface includes a data structure shown in Figure 3.5, which consists of header and data fields. The header field is accessible by the Communication module, while the data field is accessible by the sender and receiver end modules, such as Driver and Application modules. The header field has a fixed format of five fields: the test message that we use for debugging purpose, the *Code* is used to identify the end side application or function as shown in Table 3.1, the two $Flag$ semaphores ($intgrat\_made\_update$ and $opnet\_made\_update$) are used to control the data integrity against the data damages or losses, as will be shown later, and the Cur_time field is used by the communication for the synchronization, as described earlier.



Field	Type	Size (Bytes)
Message	String	128
Code	Integer	1
Flag1	Integer	1
Flag2	Real	8
Data	Structure	Variable

Figure 3.5: Data format for the communication interface

The *Data* field is a data structure that depends on the sender/receiver functions. Thus,each module that uses this communication service should define its own *Data* format. For example, the location updating has its own format that will be described later.

The Communication module has read/write access to the $Header$ field. This access enables it to identify the sender and the receiver applications using the *Code* field. Thus, it can pass the data to the corresponding module to handle it.

Shared Memory and Data Integrity

Utilizing shared memory supports the required speed for the data transfer between the two simulators. However, it introduces new challenges to the implementation, including the concurrent memory access and controlling data integrity. Concurrent memory access happens when the two simulators tries to access the memory at the same time. Fortunately, the operating system transparently handles this issue. So, we do not need to consider the concurrent access to the shared memory space. However, because the operating system manage the concurrent access based on memory blocks that may be smaller than the shared memory space, we have to consider such situation to correctly and completely transfer the data.

The data integrity is essential to transfer the data between the two simulation sides completely and correctly. For example, a simulator sent data to the other side, and before the other side reads this data in, one of the simulators overwrite it, this scenario results in data damage or losses. In this case the required information will not be correctly transfered to the other side. This situation can also result in missing the system synchronization and crashing the simulation. Thus, to control the data integrity, we utilize two flag semaphores in the communication header (shown in Figure 3.5). These tow flags are used by the simulator to control the read and write processes on the shared memory space. The $intgrat\_made\_update$ flag tells the OPNET that INTEGRATION has pushed some data to the memory, thus, OPNET cannot overwrite it before reading this data in. additionally, it tells INTEGRATION, itself, that there is data pushed to the shared memory. So, if the Communication module in INTEGRATION wants to send other data it has to wait for this flag to clear. The $opnet\_made\_update$ flag has similar meaning.

Using these two flags, if the Communication module in one side has data to send, it must wait until both flags are cleared. If, at the same time, both sides have data to send and the two flags are clear then both sides will try to write their data, which will result in concurrent access and the operating system will grant access to the first access attempt.

To correctly control data integrity and achieve system synchronization, the flags should be changed in the correct time to the correct values. This is how it works. When reading in data from the shared memory, the flags will be changed after reading the data. While, when writing to the shared memory, the flags must be the first fields to be changed and, then, changed again at the end of writing process. When the Communication module reads in data from the shared memory, it keeps the other side flag set to 1 until it finishes reading the data. keeping the other side flag unchanged will prevent any change attempts from the other simulator over the whole shared memory space until reading the complete data structure. At the end of the reading process, the Communication

module should clear the the other simulator flag. When the Communication module writes data to the shared memory, it starts writing by setting the other side flag to 1 and clearing its own flag. These values of the flags will prevent any access (read or write access) by the other simulator over the entire shared space. Then, when it finishes writing its data to the memory, the Communication module should inverse the values of the two flags as a signal to the other side that it can read the new data that was pushed to the memory.

Location Updating Module and Driver Module

The Location Updating module in INTEGRATION and the Driver module in OPNET are responsible for synchronously moving vehicles in both simulation sides.

The Location Updating module in INTEGRATION is responsible for calculating the location of each vehicle and sending them along with the other parameters to the Driver module in OPNET.

The Location Updating module *data* format is shown in Figure 3.6 and its functions are described in Subsection 3.3.3. *NCV* is the number of vehicles concurrently in the network. *NFV* is the number of vehicles finished their trip in the previous time period. Car_Link_ids is an integer array that has the car identification numbers and road link identification numbers. Finally the X_Y_Spd is a real valued array that has the location and speed for each car. The cars that has finished their trips are appended to the Car_Link_ids with corresponding links and coordinates set to -1

NCV	NFV	Car_Link_ids	X_Y_Spd

Field	Type	Unit Size (Bytes)
NCV	Integer	4
NFV	Integer	4
Car_ Link_ids	Integer array (n x 2)	4
X_Y_Spd	Real array (n x 3)	8

Figure 3.6: Location updating data field format

To send its data,Location Updating module creates its data field and pass it to the Communication module to send it.

The Driver module in OPNET receives the location updating messages (code 51) from the Communication module and then it 1) checks the received simulation time from the other side, and, in case of time mismatch, it takes the appropriate decision (as shown in Figure 3.2) to overcome

this mismatch, 2) updates the location for the moving vehicles, 3) activates any required new vehicles, and 4) deactivates the vehicles which finished their trips. Using the number of moving vehicles and the activation/deactivation mechanism that will be described later drastically reduces the processing time in OPNET, especially for large scenarios. That is because OPNET cannot dynamically create or delete communication nodes (vehicles) during runtime, and all the vehicles must be created before running the scenario.

Performance Improvements

In this subsection we describe some implementation problems that we experienced and some of the techniques we utilized to improve the simulation speed and reduce the memory usage. We faced many challenges in the implementation. The first one is that INTEGRATION is built using FORTRAN, which has a limited support for the inter-process communication mechanisms compared to C/C++. To overcome this problem, we used mixed-language programming by building the Communication module using the C language and then compiling the C and FORTRAN codes into a single executable.

The second problem is that OPNET cannot dynamically create or delete communication nodes (vehicles) during runtime. This means that all the vehicles must be created and configured before running the scenario, i.e., if we have a 50,000 vehicle scenarios, then we have to create 50,000 communication nodes in OPNET at the design time. The problem is that this number of communication nodes in OPNET will result in a very slow simulation process. So, we used the activation/deactivation mechanism for communication nodes. This mechanism starts by deactivating all the communication nodes. Then, when receiving location updates, it activates the required nodes. When INTEGRATION sends a notification about a vehicle that completes its trip, the mechanism deactivates that vehicle. This mechanism drastically reduces the number of active vehicles in OPNET and thus enhances the simulation speed.

During the implementation phase, we realized that the OPNET software is the computational bottleneck that slows down the simulation speed because it model every single event that happens in the network. Based on this, we decided to do computations in OPNET, only, if it is necessary. Thus, most of the computations are made in the INTEGRATION software to minimize the load on the OPNET simulator. For example, one option was to send the vehicle speeds and directions to OPNET and have OPNET compute the new vehicle locations. But our analysis showed that OPNET runs much slower than INTEGRATION; the reason is that OPNET processes a huge number of events that occur in the network. Thus, we decided to make all computations in the

INTEGRATION side and send vehicle coordinates and speed to the OPNET.

One more solution we used to improve the system efficiency is the vehicle-object reuse feature, which will be described later. This feature reduces the number of communication node objects required in the system. Consequently, it reduces the computational load and improves the simulation speed.

3.4.2 Modeler Features

VNetIntSim has some features that were added to achieve different objectives, as described in this section.

Vehicle Reuse

One of the main issues when simulating the vehicular network is scalability, which is mainly affected by the number of vehicles traveling along the network. As mentioned in the previous subsection, OPNET cannot create vehicles at runtime. Consequently, we have to create all the required vehicles in the design phase. In the case of large network scenarios, the large number of vehicles will result in a very long initialization time when starting the simulation and also results in large memory usage. Subsequently, this limits the model scalability. To overcome this limitation, VNetIntSim can make reuse of the same vehicle as a communication node to represent multiple moving vehicles, obviously in different time slots. In this way, the required number of vehicles in OPNET can be reduced from the total number of vehicles or trips (which may be thousands of vehicles) to the maximum number of concurrent vehicles, which is significantly smaller than the total number of vehicles or trips. The vehicle reuse feature can significantly increase the scalability by reducing the number of vehicles simulated in OPNET, consequently, decreases the memory requirements and the execution time. This feature can be safely used when we are interested in studying the global system behavior. However, it is not suitable when studying the communication behavior of an individual vehicle or connection.

Vehicles Multi-Class Support

This capability is inherited from INTEGRATION, which supports up to five classes of vehicles. Each class can be configured to run in a different way and use a different algorithm. We extend this feature to OPNET, where the class information is associated with the vehicle and transferred from

INTEGRATION to OPNET. So, the user can implement communication protocols or configure them to work differently for different classes of vehicles. For example, in data dissemination in VANET, the user can choose to send the data only to a specific vehicle class (i.e., Trucks). Using this feature, the data routing protocol can prioritize the next hop based on its class (i.e., vehicles of the same class move in similar speeds, thus their relative speeds (if they move in the same direction) are very low. Another application of this feature is the penetration ratio of a specified technology where we want to check the effect of the penetration ratio of some new technologies (i.e., cooperative driving).

Customizable Updating Interval

The location updating interval determines how frequently the location information is sent from INTEGRATION to OPNET. The shorter the updating interval, the higher the accuracy of the mobility. However, the shorter the updating interval the more the processing and, thus, the longer the execution time. VNetIntSim enables the user to change this interval based on the network requirements. Its default is 0.1 seconds, which is also the minimum updating interval. It can be changed to any value that is a multiple of 0.1 seconds. Also, it is not necessary to be matched in the two sides of the VNetIntSim, because the INTEGRATION can overwrite the updating interval setting in OPNET.

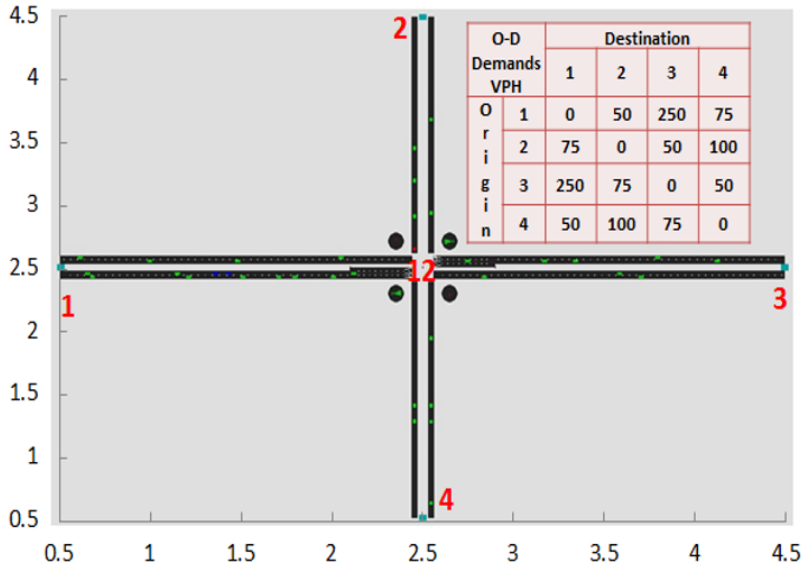
3.5 Case Study

Data packets routing is one of the important protocols that are sensitive to vehicle mobility and density parameters. In this section, the VNetIntSim is used to study the effect of mobility measures on the AODV routing [87], in the case of file transfer protocol (FTP) traffic. In addition, the effect of vehicle density on VOIP jitter is studied. Subsequently, the scalability of the VNetIntSim modeling tool is tested because scalability is a critical drawback in existing simulators, including VEINS and iTETRIS.

3.5.1 Simulation Setup

In this case study, the road network shown in Figure 3.7 is used. The road network consists of an intersection numbered 12, and four zones numbered 1, 2, 3, and 4. Each zone serves as a vehicle

origin and destination. Each road link is 2 kilometers in length. The vehicular traffic demand that was considered in the study is presented in Figure 3.7. For example, the traffic rate from zone 2 to 1 is 75 vehicles per hour. The vehicles speeds are determined using two speed parameters, namely, the free-flow speed and the speed-at-capacity [88]. Throughout the chapter, the notation Free/Capacity will be used to represent the values of free-flow speed and the speed-at-capacity. Two speed scenarios are considered, namely: 40/30 km/h and 80/50 km/h.



O-D Demands VPH		Destination			
		1	2	3	4
Origin	1	0	50	250	75
	2	75	0	50	100
	3	250	75	0	50
	4	50	100	75	0

Figure 3.7: Road network and O-D demands

For the application, we used FTP, in which we can control the connection time by deciding the file size. Also, in OPNET we can control the traffic rate of the FTP connections. The FTP server is located at the intersection. Starting at 250 seconds, the moving vehicles attempt to download a 100 Kbyte file from this server. The FTP clients re-establish a new connection every 20 seconds. The FTP server is spatially fixed and modeled as a roadside unit (RSU). The IEEE82.11g was employed at the wireless communication medium with a data rate of 24 Mbps. For data routing, the AODV is used as the routing protocol for both scenarios.

3.5.2 Number of Moving Vehicles in the Network

The vehicular traffic simulation included three phases: two transient phases and one steady-state phase. The loading and unloading phases are transient phases, which represent the two shoulders of the peak period, as illustrated in the graphs in Figure 3.8. In the loading phase, vehicles enter

the road network, while in the unloading phase vehicles exit. Between them, there is a steady-state phase in which some vehicles are entering the network, while others are exiting. In the steady-state phase, the change in the number of the vehicles in the network is not significant, while in the loading phase the network loading changes significantly. The length of these phases depends mainly on the speed setting, vehicle departure rates, and the roadmap. Figure 3.8 shows the number of vehicles in the network for different speed parameters (Free/Capacity). The importance of determining these phases is that during the transient phases the communication network may be spatially partitioned without data routes linking these partitions together, while in the steady-state phase vehicles almost cover the entire road network and most probably there is full connectivity between vehicles. Consequently, the network communication behavior during the transient phases is different from that during the steady-state phase.

By controlling the speed parameters and the departure rate distribution, we can control the network partitions during the simulation time. Using this methodology, we can model the delay tolerant communication networks (DTN) [89] and intermittently connected mobile networks [90].

Defining these phases gives us estimation of the vehicle density in the network at any time instant. This density significantly influences the communication performance as will be shown later.

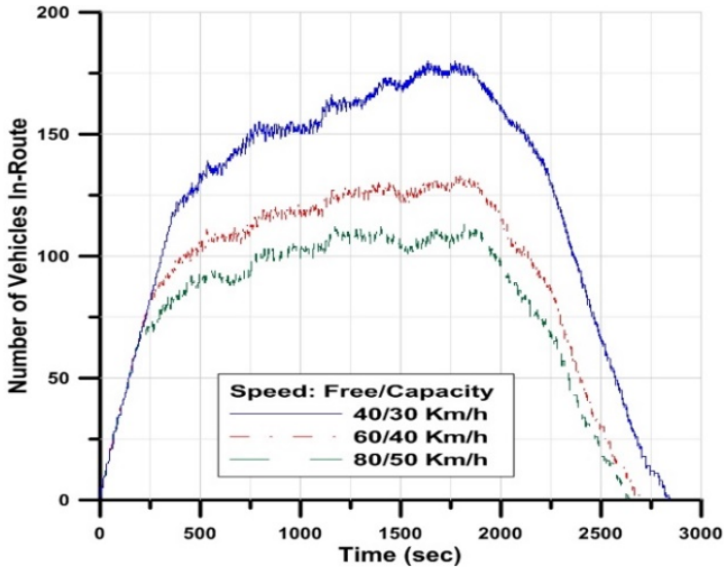


Figure 3.8: Number of vehicles in the network

3.5.3 FTP Connections and AODV

In this section, some results obtained from the FTP communication will be presented. As we described in the previous subsection, the vehicle density significantly affects the communication performance. Figure 3.9 shows the cumulative number of packets dropped by AODV across the entire network due to the loss of a route to the destination.

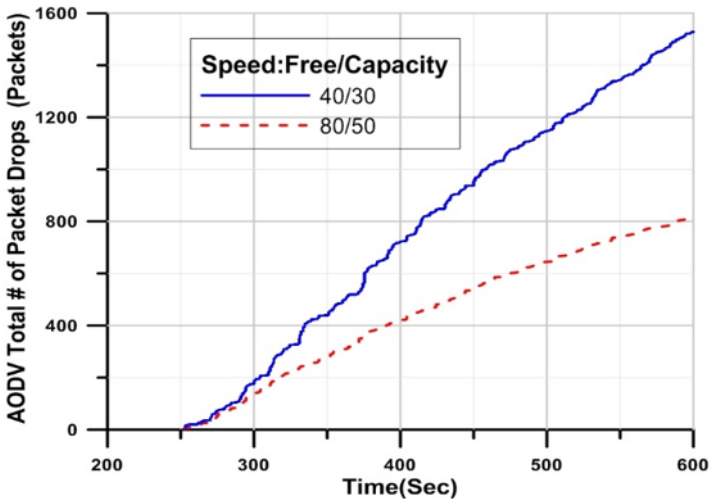


Figure 3.9: AODV total number of packet drops

The AODV packet drop can be caused by two main reasons. The first is the number of vehicles in the network; the larger the number of vehicles the larger the traffic, so any route missing will result in a larger number of drops. The second is the vehicle speeds; the higher the speed the faster the route changes and, consequently, the larger the number of packet drops.

In an attempt to identify which of the two factors is more influential on the routing, Figure 3.10 illustrates how the average number of drops varies across the network. It shows that around 300 seconds, both speeds have a similar average packet drop rate. During this interval, the number of vehicles for both scenarios is very similar, while, as the difference in vehicle density increases with time, the average number of drops also reflects the changes in traffic density.

The two figures demonstrate that, for the two scenarios, despite the fact that the vehicle density is related to the traffic stream speed, the vehicle density has a more significant impact on the performance of the communication system. Consequently, a change in the traffic stream density caused by other factors, such as traffic demand, has a more significant impact on the data routing than does the changes in the traffic stream speed.

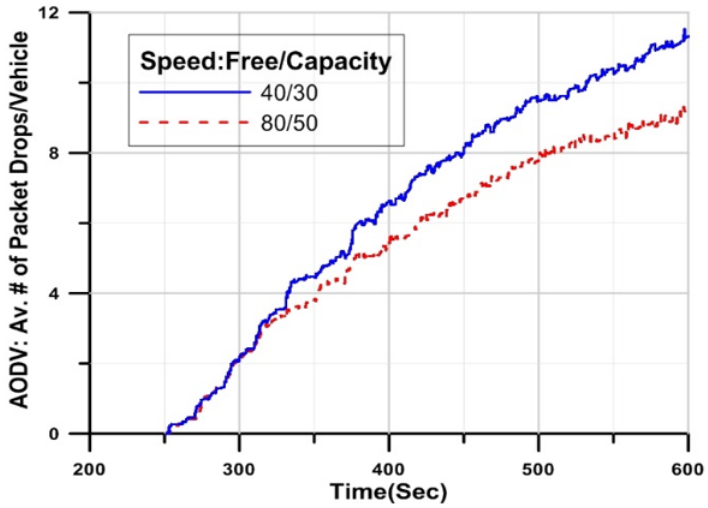


Figure 3.10: AODV Av. number of packet drops per vehicles

Another important parameter in routing efficiency is the route discovery time. Figure 3.11 shows the correlation between the route discovery time and the IP processing delay (processing plus queuing) in the vehicles. It shows that, after 250 seconds, each vehicle attempts to establish an FTP session with the server resulting in a flood of AODV route request packets. This flood increases the amount of IP packets being sent and processed at the IP layer in each vehicle and, thus, increases the IP processing (queuing + processing) delay, which is reflected on the route discovery time.

From Figure 3.11, it is clear that the long route discovery time when initiating the communication is mainly due to the IP queuing and processing delay in the higher density scenario. Subsequently, the TCP congestion control logic paces the packets based on the acknowledgments it receives. This pacing results in lower queuing and processing delay. Consequently, both the processing delay and route discovery time gradually decrease.

Figure 3.12 illustrates the effect of the speed and density on the number of active TCP connections on the FTP server. The figure demonstrates that, when initiating the FTP connections, there are 69 and 61 TCP connections for the 40/30 and 80/50 speeds, respectively. These numbers are proportional to the number of concurrent vehicles in the network for each scenario. The results also demonstrate that some of these connections were completed before the start of the second cycle (at 270 seconds). Similarly, the second cycle increases the number of connections. The results demonstrate that, later, the number of connections for the 80/50 scenario decreases significantly because some vehicles exit the network, so their connections are timed-out and dropped, while, in the 40/30 scenario, vehicles are still traveling on the network.

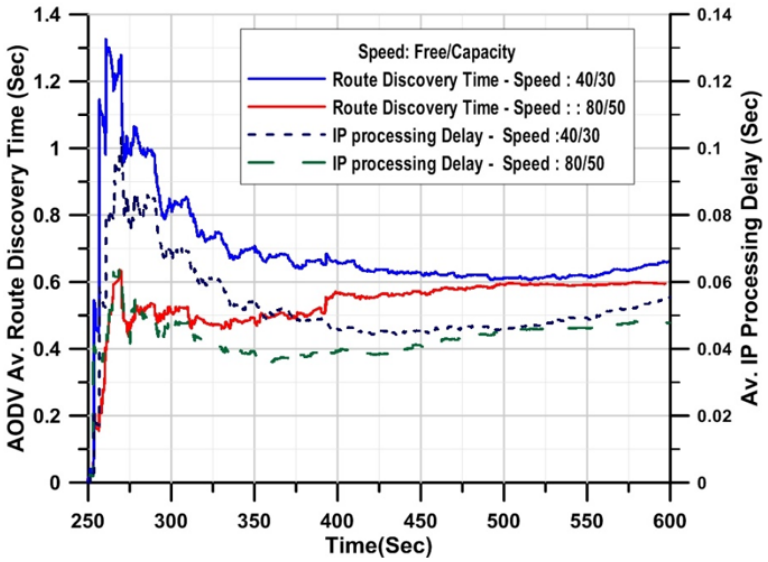


Figure 3.11: AODV Av. route discovery time and Av. IP processing delay

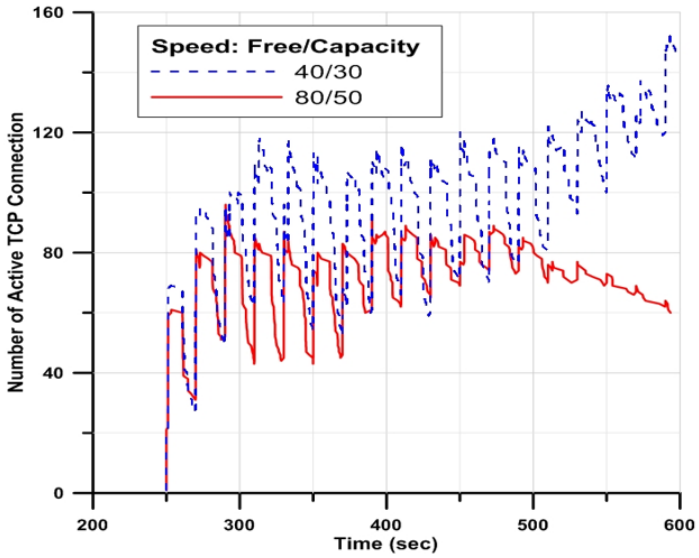


Figure 3.12: Number of TCP connections of the FTP server

The above results and analysis for the simple scenarios we used are realistic and consistent with the protocol behavior.

3.5.4 VOIP Jitter

This subsection focuses on the VOIP traffic and how the mobility parameters affect the performance of the voice application. The start time of the voice sessions is normally distributed with a mean and variance of 350 and 50 seconds, respectively. The session duration is 250 seconds. Figure 3.13 shows the average jitter across the entire network. Figure 3.13 shows that the jitter for the low speed is very high compared to the high speed.

The results show that, when the voice session starts around 350 seconds, the jitter in both scenarios is similar. Furthermore, as the number of sessions increases, the jitter increases gradually. For the 80/50 speeds, the jitter reaches the steady-state earlier because the network enters a steady-state (the change in number of vehicles is not significant), while, for the 40/30 scenario, the jitter continues to increase to unacceptable values because of the increase in the number of vehicles that increases the session, thus the competition in the wireless medium.

Figure 3.13 shows the importance of the vehicle density in the network and how influential it is on the VOIP connections. It shows that, as the vehicle density in the network increases, the overall jitter across the network becomes unacceptable. Although the routes in lower speed are relatively more stable, the jitter is higher due to the vehicle density.

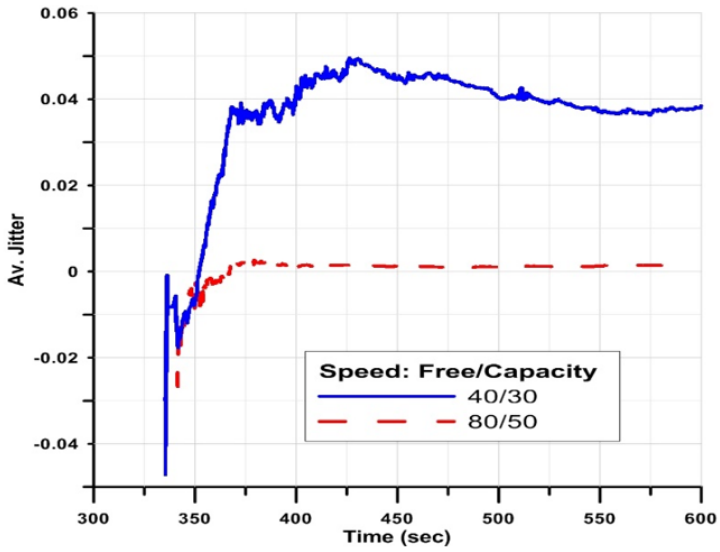


Figure 3.13: Average VOIP jitter

3.5.5 System Scalability

The scalability is the most critical drawback of existing platforms including the implemented platform. The two main scalability parameters are the memory usage and the execution time. The results show that the number of nodes and the data traffic rate per vehicle are key factors behind the scalability issue. Specifically, the results show that the memory usage grows exponentially with the number of vehicles in the network, as shown in Figure 3.14. The result shows also that the execution time is mainly dependent on the average traffic rate per vehicle. As shown in Figure 3.15.

Figure 3.14, shows that the memory utilization increases exponentially with the number of vehicles in the network. This poses a scalability limitation to the modeler. This scalability problem is due to the detailed implementation of the network simulation models. However, this detailed implementation is necessary when studying the behavior of individual vehicle, individual connection between two vehicles, or the detailed behavior of a specific protocol.

On the other hand, in the case of focusing on global analysis, where the individual detailed behavior is not of high importance, we can reduce the number of vehicles in the network by reuse the vehicles as described earlier. In this case, the total number of vehicles we need in the simulation network become the maximum concurrent number of vehicles.

Figure 3.14 also shows that for a specific number of nodes, increasing the traffic rate has no significant effects on the memory usage. We attribute that behavior to the ability of OPNET to destroy the packets after they arrived at the destination application, so it frees its memory.

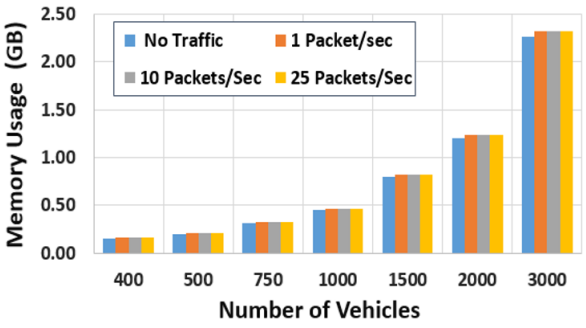


Figure 3.14: The memory usage (GB) vs. the number of nodes for different traffic rates

Figure 3.15 shows that the execution time is exponentially increasing with the number of vehicles, and increases also with the average traffic rate per vehicle. We can notice the abrupt increase in the execution time when increasing the traffic rate to only one packet. This large increment is reasoned to the broadcast nature of the AODV protocol that is used in this scenario, where any application

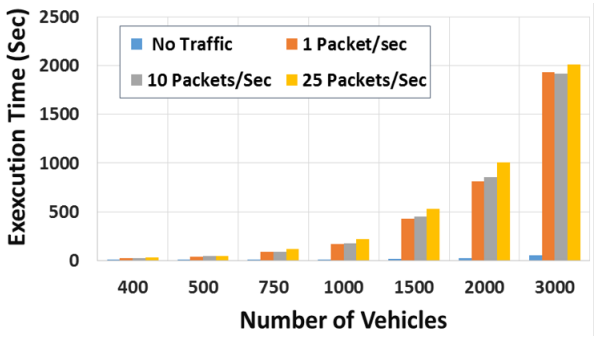


Figure 3.15: The execution time (Sec) vs. the number of nodes and the traffic rate per vehicle

packet to a new source triggers the AODV to broadcast a route request message to all its neighbors. Each of these neighbors receives and processes this message and might rebroadcast it, which results in a wave of broadcasts that spans the network, consequently, increasing the execution time.

We also notice that increasing the traffic rate per vehicle from 1 to 10 packets per vehicle does not result in such an increase in the execution time. That is because the first packet only initiates the broadcast waves in the network, and any other packets to the same destination needs only route maintenance.

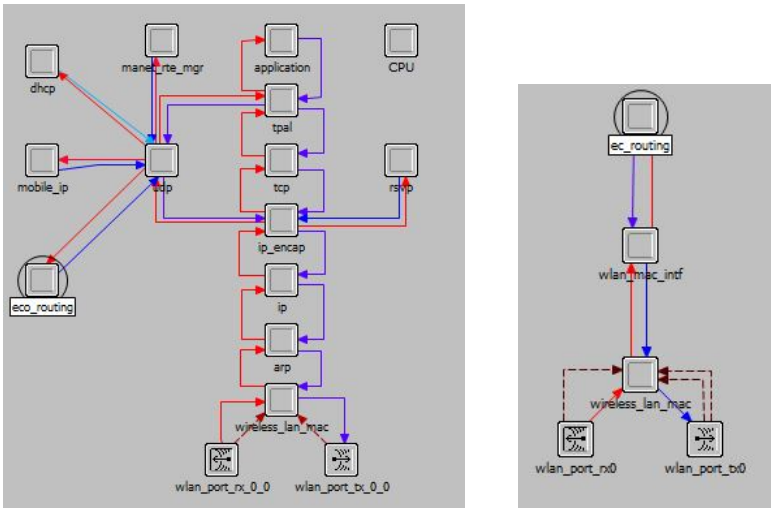
These results are obtained on a machine of Intel Core-i7 Quad-core processor, with 4 GB of memory, and running Windows 7 Ultimate.

Computational Bottleneck

To improve the scalability of the implemented model, we used two techniques; activation/deactivation and vehicle reuse, as mentioned before. One other idea, that was promising, to improve the model simulation scalability is to replace the full stack communication node object (vehicle object), which is shown in Figure 3.16-a, by a simple vehicle object that only has medium access layer and the application module with an interface between the application and the medium access layers as shown in Figure 3.16-b

We have already tried this way to reduce the computations in the vehicle objects. However, it did not result in the improvement we need to model the large-scale networks. For example, the simulation execution time in the 3000 vehicle scenario was reduced by about two minutes, which is about 6% of the original execution time (about 1900 seconds).

This performance results motivated us to look for the computational bottleneck in OPNET. We



(a) Complete vehicle object

(b) Simple vehicle object

Figure 3.16: Complete vehicle object vs simple vehicle object

realized that this bottleneck exists in the medium access layer and there are three reason for this computational bottleneck.

The first reason is that the medium access technique in this layer is the point where a vehicle (as a communication node) gets in touch with other vehicles, where the medium access layer senses the wireless medium to check if any other vehicles are sending. To imagine the computational load in this process, imagine a vehicle wants to send a frame. In this case, it will first sense the medium, which means, it will check if any other vehicles are sending at this time. Then for each sending vehicle, it must compute the signal strength based on the distance and the attenuation model it uses. Based on the combined strengths from all sending vehicles, the subject vehicle will determine whether the medium is idle. This process may be repeated many times in order to send a single frame, because, if the medium is not idle, the medium access technique will wait for a specified time interval and re-sense. In the case of large number of vehicles concurrently in the network, this process will be computationally very expensive because the subject vehicle has to compute the distance and the signal strength for each other sending vehicle many times.

Secondly, when the medium becomes idle and the vehicle has the opportunity to send its frame, this frame will trigger at least one event in every other vehicle to compute the signal to noise ratio for this frame. Based on this calculation, each other vehicle will decide whether it can receive this frame. Moreover, those vehicles, which can correctly receive this frame, will continue receiving the frame (or at least its header), because any of them maybe the frame destination. During this

receiving time, every receiving vehicles should also account for any other vehicles that may start sending before this frame ends. For example,frames transmitted from hidden terminals may result in destroying other frames that have already started before. This sensing of the medium, during receiving, will also increase the computational load in the medium access layer.

The third reason is the mobility, because vehicles need to recompute the signal strength as well as the connectivity parameters whenever it has a change in its location or whenever the sender has. For instance, whenever a sending vehicle changes it location, all the other communication enabled vehicles have to recompute the distance, the signal strength, and other connectivity parameters. Thus, mobility also increases the computational load in every single vehicle.

Compared to the MAC layer events, events in the upper layers are limited to their vehicles and do not trigger any events in other vehicles. So, the computational load of the upper layers events are limited.

These reasons together, means that the medium access layer is the bottleneck in the discrete event simulation. However, this layer is very important, because it is the gate through which all the data frames exit and enter the vehicle, do, it can affect all the upper layers. Thereby, it is important to accurately model the impacts of this layer in a computationally efficient method to achieve scalable modeling of the vehicular communication. Thus, in Chapter 5 we replace OPNET with an analytical model that simulates the medium access technique by estimating the packet drop rate and delays.

3.6 Conclusions and Future Work

VNetIntSim is presented as an integrated platform for simulating and modeling vehicular networks. VNetIntSim integrates a transportation simulator (INTEGRATION) with a data communication network simulator (OPNET modeler). Results obtained from the simulation scenarios are realistic and consistent with protocol behavior. VNetIntSim has the capability to fully simulate the two-way interdependency between the transportation and communication systems, which is necessary for many applications. In addition, it provides the power of both simulators to study global network parameters as well as low-level detailed parameters for each system at a microscopic level considering a 0.1-second granularity. Subsequently, the VNetIntSim modeler is used to quantify the effect of mobility parameters on the communication performance. The results show that the effect of vehicle density is of higher significance than that of the speed. More specifically, the higher speed results in a lower drop ratio and lower jitter due to the lower traffic density. Despite of its

limited scalability, VNetIntSim can be used to accurately model and study the mutual interaction of the communication and mobility in small and medium size networks. So, an important future work is to implement some ITS applications such as speed harmonization, eco-driving, congestion avoidance and vehicle routing. In the next chapter, we use VNetIntSim to study the impact of the communication on feedback based eco-routing navigation.

Chapter 4

Eco-Routing Using V2I Communication

In this chapter, we build upon the VNetIntSim platform, which is presented in the previous chapter, to study the impact of communication on ITS performance. Because this impact depends on the ITS application, we use the feedback-based eco-routing navigation as an example for this impact. Eco-routing is a technique proposed to optimize the fuel consumption in transportation networks. In the feedback-based eco-routing, the feedback from vehicles currently in the network is used to compute the routes fuel consumption level for other vehicles. More specifically, when a vehicle traverses a road link, it reports its fuel consumption on this link to a TMC, which updates the routing information. Subsequently, vehicles are assigned routes or rerouted based on the updated information. The vehicular communication network is responsible for transferring this information from vehicles to the TMC, where packets are subjected to delays and drops.

So, in this chapter, we study how the packet drops and delays can affect the route choices and, consequently, the network-wide fuel consumption in a small network that can be supported by VNetIntSim.

To the best of our knowledge, none of the previous research efforts on eco-routing studied the effect of the communication parameters including the drops and delay on the performance of eco-routing algorithms.

4.1 Introduction

Major existing challenges include global warming coupled with energy and fuel shortage and environmental pollution. An important source of these problems is the transportation system. As we mentioned in Chapter 1, the U.S. Department of Energy mentioned in [15] that approximately 30% of the fuel consumption in the U.S. is due to vehicles moving on the roadways. In addition, about one-third of the U.S. carbon dioxide (CO_2) emissions comes from vehicles. The 2011 McKinsey Global Institute report estimated savings of about $600 billion annually by 2020 in terms of fuel and time saved by helping vehicles avoid congestion and reduce idling at red lights or left turns. These estimations show the importance of addressing and optimizing the fuel consumption in transportation systems.

To achieve this optimization, ITSs employ new algorithms and technologies to improve the transportation fuel consumption levels and reduce vehicle emissions. In this context, eco-routing was initially proposed in [17] to select the routes that result in the lowest fuel consumption. Eco-routing relies on the ability of the vehicles to measure the fuel consumption on each road link and report this information to a TMC, which, in its turn, updates the routing information. Communication networks including V2V and V2I represent infrastructure that transfers the information reported from the vehicles to the TMC. Consequently, the communication performance and parameters such as end-to-end delay and packet drop ratio should be considered when studying the performance of these systems.

To the best of our knowledge, none of the previous research efforts on the eco-routing studied the effects of the vehicular communication on the performance of eco-routing systems. All the previous researches assume an ideal communication network, that is, no packet drop or delay, which is not a realistic assumption. Packet drop rate and/or delay can lead to the inaccurate representation of the current network state at the TMC, which can result in incorrect routing decisions that may produce higher fuel consumption and emission levels.

Consequently, this chapter addresses this research area by studying a realistic eco-routing system that includes both communication and transportation components. Building upon VNetIntSim platform that was introduced in Chapter 3, this chapter firstly develops a new module for the eco-routing application. This module is incorporated into the VNetIntSim to realistically simulate the eco-routing process. So, in this chapter, we describe the implementation and the operation of the eco-routing as an application in the VNetIntSim platform. Secondly, the updated VNetIntSim framework is used to study the effects of packet drops, packet end-to-end delay, and the number and locations of the roadside units on eco-routing system performance.

The remainder of this chapter is organized as follows. Section 4.2 provides a brief overview of eco-routing and Vehicular ad-hoc networks. A description of the methodology and the eco-routing implementation are presented in Section 4.3. The simulation results for different network scenarios are presented in Section 4.4 before the final conclusion.

4.2 Related Work

This chapter studies the impact of vehicular communication on the eco-routing application performance. Chapter 2 introduced some details about both the vehicular communications, eco-routing, and the related previous work. So, this section gives only a quick overview of each of these technologies.

4.2.1 Eco-Routing

In eco-routing, the road link fuel cost is utilized as a metric to evaluate the alternative routes to a given destination. In fact, the vehicles fuel consumption and the emission levels depend on many factors such as the route characteristics (i.e., length, speed, grade, and traffic congestion) and vehicle characteristics (e.g., weight, shape, engine type, and power) in addition to the driving behavior. All these factors are reflected in the direct measurements of vehicle fuel consumption levels while it is moving. Consequently, eco-routing algorithm that relies on such direct measurements will capture all possible effects in building optimum routes.

Several research efforts implemented eco-routing models such as [48, 49, 53, 54, 91], as introduced in Section 2.4. However, all of these implementations assume a perfect underlying communication network in terms of packet drop rate and end-to-end delay. Realistically, the network packets may be dropped (due to noise in the wireless medium, high competition over the wireless medium, and queuing in the different process queues) or delayed (due to the processing and queuing), which may affect the performance of the eco-routing application.

In this chapter, to realistically simulate the eco-routing and the effects of communication network parameters on it, we induce the eco-routing module and how it is incorporated into the VNetIntSim platform. In our implementation, the eco-routing module runs as a network application, it uses the user datagram protocol (UDP) [75] as a transport protocol(as we described in Section 2.2.4), and 802.11p [92] as the wireless medium access protocol.

4.2.2 Vehicular Ad-hoc Network

Using VANET, moving vehicles can communicate with other vehicles, RSUs, or even with hand-held mobile devices using the DSRC system, which was described in detailed in Section 2.2.

A variety of applications are based on VANET communications. The safety applications are the primary category of these applications, where safety information is disseminated for the purpose of accident prevention to save people's lives [26, 93, 94, 95]. In addition, post-accident investigation and traffic jam information can be exchanged to mitigate the effect of accidents. Other types of information can be exchanged over the VANET including non-safety information such as cooperative driving applications [78, 96] and road traffic congestion detection and management systems [82, 97, 98]. Eco-routing information (fuel consumption costs of route links) is an example of the information that can be exchanged over the VANET to improve the navigation in transportation systems. These costs are used as a route metric based on which vehicles are assigned the routes that minimize their fuel consumption.

4.3 Modeling Eco-routing in VNetIntSim platform

To accurately model the eco-routing logic, we use the VNetIntSim platform. In VNetIntSim, an application is modeled by two modules, one in each simulator. Consequently, we developed two eco-routing modules, one in INTEGRATION and one in OPNET. The application data is transferred between application modules through the communication channel by calling the communication modules. This section describes these two modules and how the eco-routing application works using realistic communication modeling.

4.3.1 Eco-routing Logic in VNetIntSim

In INTEGRATION software, to simulate the eco-routing, when a vehicle passes a road link, INTEGRATION calculates its fuel consumption along the link. Then the eco-routing module in INTEGRATION transfers this information (Vehicle ID, Link ID, Time, and Experienced Cost) to the eco-routing module in OPNET. OPNET models communicate this information to the TMC. Thus, OPNET creates a packet with these contents and sends it from the subject vehicle to the TMC. In this communication process, packets may be delayed or dropped based on the network conditions. If the packet is correctly delivered to the TMC, OPNET notifies INTEGRATION, which updates

the link fuel consumption cost based on the delivered information. In this way, the VNetIntSim can realistically simulation the eco-routing process including both the mobility and communication.

4.3.2 Estimating Vehicle Fuel Consumption

In INTEGRATION, when the vehicle enters a new link, the vehicle fuel consumption and emission levels are reset to zero for the new link. Subsequently, INTEGRATION periodically calculates the fuel consumption and emission rates for each vehicle using the VT-Micro model as described in detail in Section 2.5. For each vehicle, the estimated fuel consumption and emission rates are accumulated until the vehicle finishes the link. When a vehicle leaves a link, it submits its fuel consumption cost for this link to the TMC, which updates the link fuel consumption using some smoothing techniques. Subsequently, INTEGRATION periodically rebuilds the routes for each origin-destination pair at a frequency specified by the user. These updated routing trees are then used whenever any vehicle requests a route or needs to be rerouted.

4.3.3 Vehicles Eco-routing Models in OPNET

In OPNET, we developed two eco-routing sub-modules, one for the vehicles and one for the TMC. Both of them work over UDP protocol. The use of UDP avoids the overhead imposed by the TCP protocol as vehicles send only one message for each link they traverse. The finite state machine (FSM) of the vehicles eco-routing process sub-module is shown in Figure 4.1. It has two initialization phases. The first one (Init0) forces the process to wait until the lower layer protocols initialize. The need for this waiting interval comes from the fact that this process needs to read the IP address of the TMC (which will be used as the destination address fro the eco-routing updating packets) to initialize its state variables. Since this process works in the application layer and the IP address is in the lower layer, the process has to wait until the lower layers fully initialize. In the second initialization phase (Init), the process initializes the required state variables such as its own port number, the TMC IP address, the TMC port number, and the required statistics.

After initialization, the eco-routing process as well as all the other processes should be completely initialized and become ready to run. At this point, the eco-routing process moves directly to the Idle state where it waits for an interrupt from the Driver process (shown in Figure 3.4) in VNetIntSim to send the eco-routing message. The driver process is responsible for periodically communicating to the INTEGRATION software through the communication module (shown in Figure 3.4). When it receives an eco-routing command from INTEGRATION, it sends an interrupt to the eco-routing

module in the specified vehicle. This interrupt includes the Vehicle ID, link ID, the fuel consumption level on the link, and the time when the vehicle passed this link. Whenever the eco-routing process in a vehicle receives an eco-routing interrupt, it promptly attempts to send an eco-routing message to the TMC IP address. Then, the eco-routing process returns back to the Idle state. In this way, INTEGRATION will not update the routing information when a vehicle finishes a road link. Instead, it just triggers the corresponding vehicle in OPNET to send an eco-routing message to the TMC.

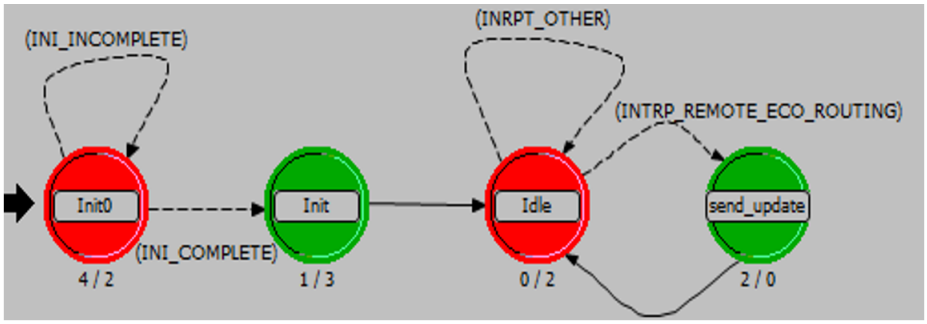


Figure 4.1: The vehicles eco-routing process FSM

The eco-routing message format is shown in Figure 4.2. The sequence number of the message is used by the TMC to avoid processing duplicated messages. The time field enables the TMC to neglect the obsolete information.

The TMC eco-routing process receives these messages, stores the received information, and sends this information every decisecond to the communication module that will pass it to the INTEGRATION, which in its turn updates the routing information and assigns the best routes to the vehicles.

In this modeler, we do not consider the communication from the TMC to the vehicles when sending routes to them. We assume that the drivers use a Web interface that shows the best routes and that the drivers follow the TMC recommendations. The Web interface uses a reliable communication transport service (TCP) that can detect and recover the dropped packets. Thus, it can work correctly even in the cases of high drop rates. However, a more important question is about the time constraints for the route request and the availability of this service, especially in the case of the Web application which uses the TCP. As mentioned before, according to the applications and services requirements published in [37], the routing information to private users can work properly with a delay between 5 to 15 seconds in normal conditions and 25 to 40 in emergency conditions. These time constraints can be satisfied by the Web TCP service. Moreover, the drivers need the routes to be updated when approaching intersections. At this these points, vehicles are almost con-

nected to the TMC because the RSUs are supposed to be installed at the intersections (as shown in Section 4.4). This means that drivers will receive the latest routing updates when they need to make decisions.

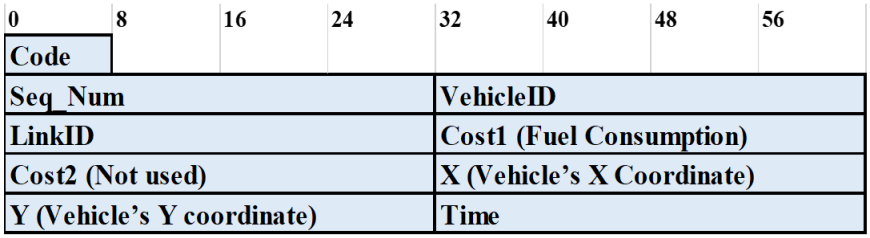


Figure 4.2: Eco-routing message format

4.4 Simulation Setup and Results

To study the impact of the communication on the eco-routing performance, we use the road network shown in Figure 4.3. The road network consists of one highway (center horizontal road) between zone 5 and zone 10, and two arterial roads (side roads). The network size is 3.5 km by 1.5 km. The free-flow speeds are 110 and 60 km/h for the highway and arterial roads, respectively. The traffic between the zones 5 and 10 is 1500 veh/h for each direction. For all other zone-pairs, it is 100 veh/h. These traffic rates continue for half an hour, resulting in 3700 vehicle trips. The simulations were run for a complete hour to guarantee that all vehicles reach their destinations. An incident is simulated on the highway (at location X in Figure 4.3) that reduces the road capacity for 15 minutes starting at 600 seconds. It blocks 1.5 lanes from three lanes on the highway (50% reduction in the road capacity) for 5 minutes. Then, the blockage is reduced to 0.75 lanes (25% of the highway capacity) for 10 minutes. The network has six traffic signals located at intersections A through F.

Regarding the communication setup, we used the infrastructure based VANET (V2I), where the RSUs are located at all signalized intersections, resulting in six RSUs. The RSUs were located at intersections for three reasons. First, these intersections are surrounded by link ends, therefore, it provides the best location to collect eco-routing messages. Second, since these intersections have signal controllers, they have the required infrastructure for connecting the RSUs to the traffic management center. Finally, the vehicles need to be connected to the TMC when approaching the intersections (decision points) to get the latest routing updates.

The RSUs are connected to a switch that connects them to the TMC server. The wireless interfaces

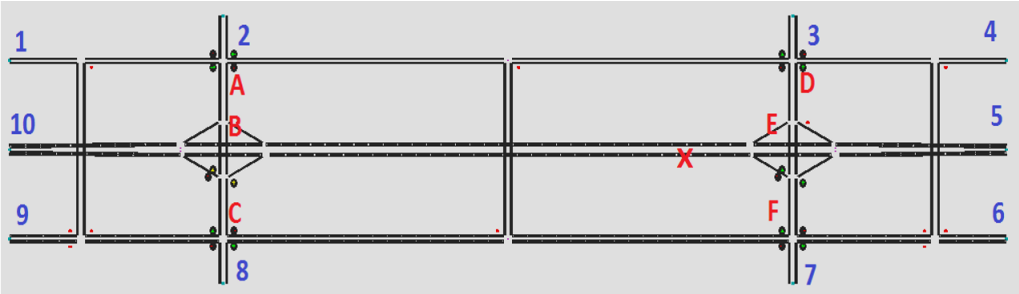


Figure 4.3: Simulated road network

are configured to use Orthogonal Frequency Division Multiplexing OFDM (802.11a), which is the base for the WAVE. We use channel 174 (5.86 GHz) with a 12 Mbps data rate. We use transmission power levels of 0.001, 0.005, 0.010, and 0.019 watts. These power levels cover the range from Wi-Fi to DSRC (from 200 meters up to 1000 meters).

The RSUs are typically access points. Consequently, when a vehicle has a message to send, it waits until it comes into the communication range of one of the RSUs. This means that the packets are being stored and forwarded. Thus, the packet end-to-end delay depends on the location at which it was generated. If it was generated near an RSU, then the delay will be very small and vice versa. This behavior is shown in Figure 4.4 for three transmission power levels. It shows that, as the transmission power increases, the average end-to-end delay decreases. The reason behind the multi-modal distribution is that the eco-routing packets are generated at a fixed set of locations (the ends of links). So, for low power scenarios, more packets are generated in locations that are out of the communication range. This is reflected in more peaks for lower power scenarios.

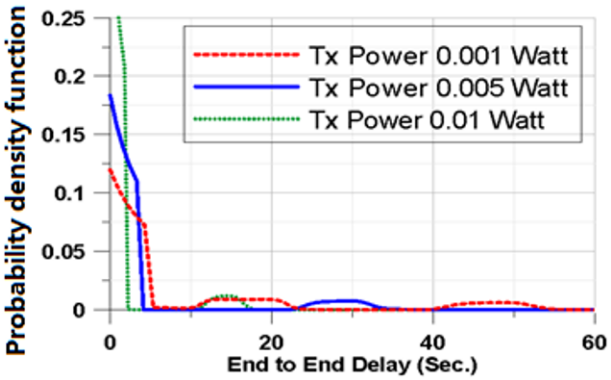


Figure 4.4: Probability density function (PDF) of the end-to-end delay

4.4.1 Effect of Packet Drops

Introducing the communication network into the eco-routing introduces two primary effects; the packet drops and the end-to-end delay. These two effects are expected to affect the performance of the applications. The packet drops will result in missing some information about the current links states, consequently inaccurate route fuel consumption calculations. Therefore, packet drops will result in errors in the route assignments. However, these errors do not necessarily lead to higher fuel consumption. It depends on which link information is dropped and the value of this information relative to the others.

Figure 4.5 compares the average vehicle fuel consumption (L/veh) for different transmission power levels, and for 0.0 and 0.2 coefficient of variation (error factors) in the eco-routing calculations, which is used to introduce white noise to the link costs, as described in Section 2.4. The "Ideal_Com ER" uses eco-routing assuming ideal communication performance (no packet drop and no delay), while the "Realistic_Com ER" scenarios are the communication-based eco-routing where the communication is introduced.

Figure 4.5 shows that the eco-routing produces network-wide fuel savings in the range of 8%. This saves about 120 liters for this one hour of simulation. It also shows a small difference between the Ideal_Com ER and Realistic_Com ER scenarios. The difference between them is generally inversely proportional to the transmission power, i.e., as the transmission power increases, the difference decreases. This is logical, because, as the transmission power increases, the number of packet losses decreases due to the wide coverage, as shown in Table 4.1, which shows the average packets sent, packets received, and packet drop ratio. Each row represents the averages of four simulation runs.

An important question arises here related to the significance of the changes introduced by the communication effects. Since the largest difference occurs for the lowest power, we compare the lowest power scenario for the Realistic_Com ER to the Ideal_Com ER using the analysis of variance (ANOVA) to check if these differences are significant. So, we first run each scenario ten times with different seeds. The only difference between these two scenarios is the modeling of the communication: the first scenario assumes ideal communication and the second one uses realistic communication. Then, ANOVA is used to compare the means, with the null hypothesis ($H_0 : \mu_1 = \mu_2$) and the alternate hypothesis $H_a : \mu_1 \neq \mu_2$. The result shows that the p-value is $0.06 > 0.05$. Thus, there is not enough evidence to reject the null hypothesis. Subsequently, the differences introduced by the communication network are not statistically significant in these cases.

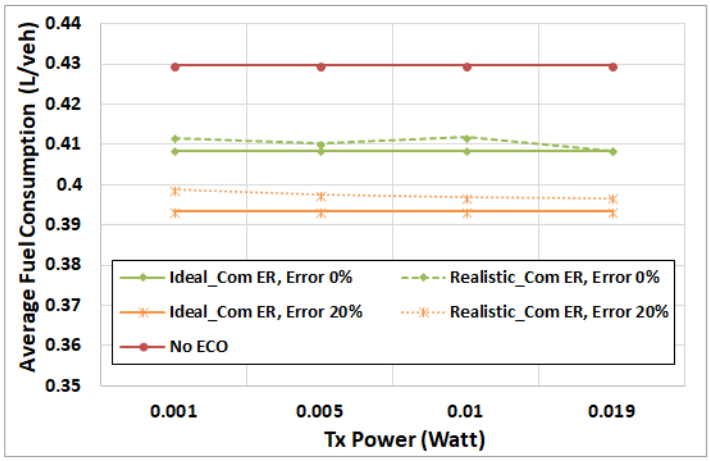


Figure 4.5: The effect of transmission power on the average Fuel consumption

Table 4.1: The drop ratio vs the transmission power

Tx Power	Packets Sent	Packets Received	Drop Ratio	Drop Ratio Variance
0.001	18,675	13,778	26.22	0.094
0.005	18,582	14,644	21.19	0.098
0.01	18,672	18,268	2.17	0.002
0.019	18,734	18,506	1.22	0.026

4.4.2 End-to-end Delay and its Effect

Figure 4.6 shows the packet end-to-end delay for the 0.001 watt power level for the two error factors. The end-to-end delay reflects the average speed in the network. The long end-to-end delay after 2000 seconds is due to the congestion on the highway that results in reducing the speed. Subsequently, the packets are stored for a longer time until the vehicles enter the RSU communication range.

In the previous scenarios, the TMC processes all the packets it receives without considering the packet delay. However, some packets are delayed for about 500 seconds such as shown in Figure 4.6. Consequently, the information it carries might be obsolete and does not describe the current state. To investigate the effect of the packet delay, we configured the TMC to discard the packets whose delay is greater than a specific time interval (allowable packet delay). We set this allowable delay to 60 seconds and ran the same scenarios for the different power levels. After discarding the delayed packets, the result shows an insignificant difference.

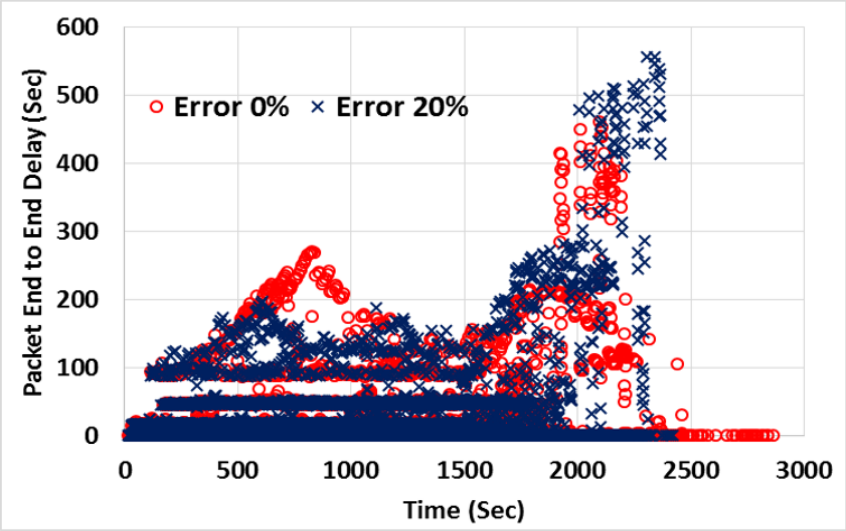


Figure 4.6: Packet end-to-end delay for Tx power 0.001 watt

4.4.3 Effect of RSU Locations

The location of the RSUs, in addition to the power levels, determine which link information will be received, which link information will be delayed, and which link information will not be received at all. To find the effect of the locations, we use the same network but with only two RSUs. We compare two 2-RSU scenarios. In the first (HW scenario), the two RSUs are located on the highway at intersections B and E shown in Figure 8.1, while in the second (AR scenario), the two RSUs are located on the arterial roads at intersections A and F.

The result shows that the locations of the RSUs result in significant changes in the case of lowest power level (0.001 watts) only. While for the higher power levels (0.005, 0.01, 0.19 watt), the effect of the locations is insignificant, as shown in Figure 4.7.

We again used ANOVA to find the significance of changes. For the lowest power level, the p-value is less than 0.0001, which provides strong evidence that the changes due to the RSU locations are significant, while for the 0.005 watts scenario, the p-value is 0.606, which indicates the non-significance of the locations in case of high power scenarios.

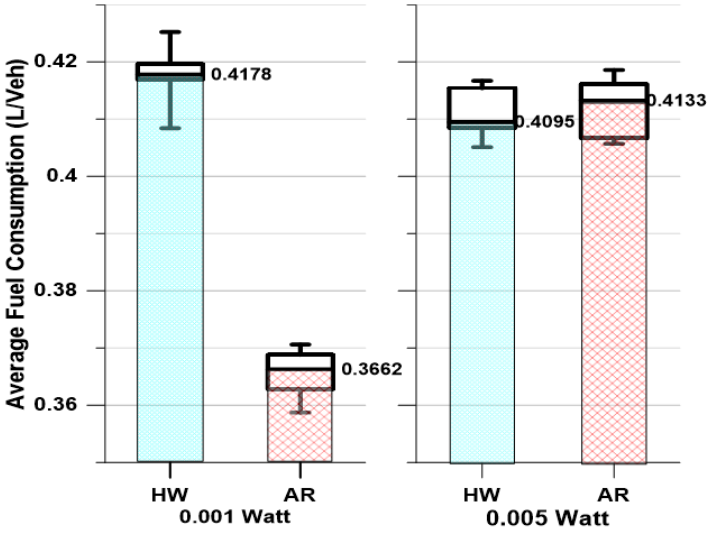


Figure 4.7: Fuel consumption Box-Whisker plots for 2RSU on the arterial roads scenarios for 0.001 and 0.005 power levels

4.5 Conclusions and Future Work

The simulation results show that the effect of the packet end-to-end delay and the packet drops are not significant on the eco-routing system performance. Consequently, the eco-routing system implemented in INTEGRATION is robust against both packet drops and the end-to-end delay within the communication system. These results are based on the small sample network used in this chapter. However, we can not generalize this conclusion. Thus, the next chapter studies these impacts in the case of real large-scale networks. But, because of the scalability issues of the VNetIntSim, in the next chapter we do not use the VNetIntSim, we introduce a new scalable platform for studying the mutual interaction of the communication and transportation systems.

The results also show that the location of RSUs combined with the communication range are critical. Because these parameters determine which link information will be collected as well as determining the delay imposed for each link information. Consequently, we suggest developing algorithms to identify the optimum location of RSUs.

Chapter 5

Modeling Vehicular Communication in Large-Scale Transportation Networks and its Interactivity with the ITS Performance

In the previous chapters, we introduced the the VNetIntSim platform to model communication in vehicular environment. VNetIntSim showed a limited scalability; it cannot be used to model large-scale road networks with high vehicular traffic demand. Our analysis showed that this scalability limitation is reasoned to the communication modeling utilized in OPNET, which uses event-driven simulation technique. Thus, in this chapter, we introduce a new scalable and computationally fast framework for large-scale modeling of communication in vehicular networks. This framework uses a realistic analytical model that we developed in this chapter for the medium access in VANET communication. We use this analytical model to replace the event-based modeling of the communication system in the VNetIntSim. The proposed communication model uses Markov chains and the M/M/1/K queuing model to estimate the packet drop rate and delay for each packet based on the communication network parameters setting such as the background packet generation rate and the average packet size. It also considers the surrounding road network conditions, which is represented basically by the vehicle traffic density. Then, after validating this model, we incorporated it into the INTEGRATION software, where the INTEGRATION behavior is changed to compute and adapt the communication performance parameters. This framework is then used to study the impact of communication on the performance of feedback-based eco-routing in a Los Angles network. Finally, in this chapter, we quantify the impact of the communication on the eco-routing at different penetration ratios and different vehicular traffic demand levels. We also study the mutual

impact of the communication and transportation by studying the impact of the vehicular traffic demand level on the communication performance.

5.1 Introduction

Intelligent Transportation Systems (ITSs) are expected to be the core of future transportation systems. Utilizing different sensors (cameras, communication enabled cars, ... etc.) and communication devices, an ITS can collect information about the transportation system. Processing these data enables the traffic management center (TMC) to better manage and improve the performance of the overall transportation system by making better-informed decisions. The correctness and the accuracy of these decisions depend on the accuracy of the data collected at the TMC. The communication network is responsible for exchanging data between the different sensors/actuators in the network and the TMC. Consequently, the communication network can affect the ITS application performance eco-routing navigation application.

The eco-routing application can be negatively affected by the communication network performance and setting as demonstrated in Chapter 4. This negative influence will be more noticeable in the cases of high vehicular traffic congestion, in which, the need for better eco-routing performance increases. On the other hand, mobility parameters can also influence communication performance. Since eco-routing should run on a city level road network, it is imperative to study these impacts in large-scale road network. However, it is apparent that it is too difficult to model all the communication details at this scale that may have tens of thousands of cars concurrently moving in the network. Thus, we decided replace the OPNET simulator, that we used in the previous chapters, by an analytical model that can represent the performance parameters in VANET communication. More specifically, we replaced the OPNET simulator by an analytical model for the medium access technique in the Wireless Access in Vehicular Environments (WAVE) [31].

Choosing the MAC layer is because of the following three reasons.

- The MAC protocol is a good representative of the mutual impact of communication and transportation systems.
- The performance of the upper layer protocols are highly dependent on the MAC performance.
- The MAC layer can capture some impacts of the upper layer protocols

Firstly, the WAVE MAC protocol has a direct mutual interaction with transportation network, where it can be significantly affected by the transportation network parameters. For example, in the case of highly congested road networks, and considering the large coverage area in VANET, in which the communication range is up to $1000m$, the large number of connected vehicles attempting to send can result in higher competition in the wireless medium. This competition will result in higher data collision probability, higher drop probability, and longer packet delivery delay. On the other hand, the drop rate and packet delay in the MAC protocol can influence the performance of ITS applications such as eco-routing application. Consequently, the MAC protocol is a good representative of the mutual impact of communication and transportation systems.

Secondly, the upper layer protocols are highly dependent on MAC performance. Thus, modeling any protocol in an upper layer without modeling the MAC layer will produce inaccurate results. For example, if we only consider the routing protocols and their impact on ITS performance, which is expected to be significant, without modeling the MAC layer, the results will capture some interesting phenomena such as communication links drops and establishments, route connectivity and its temporal changes. However, these results will not be able to capture many important phenomena. One of these important phenomena is the impacts of vehicle density and its distribution in the road network on the communication performance in terms of drops and delays, which can significantly affect the performance of routing protocol. Moreover, it will not be able to capture the impact of the routing protocol packet load overhead on the medium access. The routing protocol packet overhead results in increasing the collisions in the wireless medium, consequently, increases the delay and drop rate. Thus, to accurately model any upper layer protocols, it is essential to consider modeling of the MAC layer first.

The third reason is that the MAC layer can capture some impacts of the upper layer protocols. Because it is the gate through which all the traffic enters and exits the communication nodes, the MAC layer can captures the traffic load for all upper layer processes and model the impact of this traffic on the communication performance.

because of its impact on the communication performance and its dependency on the mobility. However, routing in VANET is still an open and wide area of research and the performance of the different routing techniques are still under investigation by the research community. Thus, in this research work, we do not consider the routing protocols and their impact on ITS applications. Consequently, we assume only a V2I communication. However, the developed simulation framework presented in this chapter can be easily modified to include the impact of multi-hop communication in the case of routing.

Based on the aforementioned reasons, we decided to focus on the MAC protocol in the WAVE stack and its mutual interactivity with the transportation system. So, in this chapter, we develop a new analytical model for the MAC and incorporate it into the INTEGRATION software to build a full-fledged platform for vehicular communication modeling. The proposed analytical model is capable of estimating the packet delay and drop probability considering the number of vehicles in the communication area and background packet generation rate in addition to other network parameter settings.

Using this developed analytical model enables the proposed simulation framework to overcome the scalability limitation, where the proposed platform supports tens of thousands of concurrent cars in the network. Under the umbrella of the proposed framework, in this chapter:

- Firstly, we develop a new model for the MAC technique with a finite buffer size in vehicular networks. This model utilizes both the Markov chain and the queuing theory to estimate the packet delay and drop probability.
- Secondly, this model is validated against a benchmark simulation data for different communication parameters including packet size, vehicle density, and transmission modes.
- Thirdly, to allow for studying the mutual impact of the communicating and transportation systems, this model is incorporated in a transportation microscopic traffic simulation software; INTEGRATION [48]. The traffic simulation software is modified to account for the packet drop and delay when updating the vehicles eco-routing information.
- Fourthly, the developed simulation framework is used to quantify the impact of communication on the eco-routing application using a real large-scale network, approximately 10 km x 15 km, which is the downtown area in the city of Los Angles (LA). The car traffic demand in this network is calibrated based on data collected from different sources.

To the best of our knowledge, this is the first work that considers and models the mutual interaction of communication and transportation systems in such real large network with real traffic (10 km x 15 km), which is bigger than many small cities. For example, the Washington DC area is about $177 Km^2$, which means the proposed platform can model the vehicular communication on the road network of the Washington DC area. In addition to that, this is also the first work that quantifies the impact of the communication on the ITS feedback based eco-routing application.

However, the resulting simulation framework is only a first step on a long road towards large-scale modeling of vehicular communication. The importance of this first step is that it paves the road for

the research community in computer science, communication, and traffic engineering to conduct more realistic research in these fields at a scale that covers city level vehicular network.

This chapter continues by providing an overview of the Markov chains in Section 5.2 because it is the tool used for the model development. Then, we introduce the proposed communication model for vehicular communication based on the IEEE 802.11p standard [13, 14] in Section 5.3. After validating the model, we describe how the communication model is integrated into microscopic traffic simulator in Section 5.4. Then, the simulation network and the simulation results are presented and discussed in Section 5.5 before the conclusion.

5.2 Markov Chains

Markov chains [99, 100] is the mathematical tool that we use in this chapter to develop the analytical model for the MAC in VANET. Thus, it is imperative to understand the basic concepts of Markov chain. This section also describes some Markov chains properties that are necessary for a Markov to have a unique solution.

5.2.1 Introduction to Markov Chains

Markov chains are named after Andrey Markov as a mathematical model for stochastic processes that satisfy the Markov property (known as memoryless property), which means that we can make predictions for the future of the process based only on its current state regardless of the process state history. In other words, the prediction based on the current state is the same as the prediction made based on the process's full history [99, 100].

Mathematically, in a Markov chain, the process (or the system represented by the Markov chain process) has a set of states called the state space $S = \{s_1, s_2, ..., s_k\}$. Within this state space, the system stochastically moves from one state to another. So, the system has a random variable X_t, which is the state of the system at time t. Using this notation, the Markov property can be written as:

$$P(X_t = s_i \,|\, X_0, X_1, ..., X_{t-1}) = P(X_t = s_i \,|\, X_{t-1}). \tag{5.1}$$

There are two types of Markov chain, continuous time Markov chain (CTMC) and discrete time Markov chains (DTMC). The main difference between the two types is whether the time space is continuous or discrete space.

In modeling the VANT medium access technique, we use the DTMC. So, the next subsection describes it in some details.

5.2.2 Discrete Time Markov Chains

In discrete time Markov chain, the process has a set of k states, $S = \{s_1, s_2, ..., s_k\}$, and the time-space is discretized into steps. So, the time is denoted by the step n rather than t. At $n = 0$, the process starts at one of the states (initial state) and moves successively from one state to another. Each move is called a step or transition. If the chain is currently in state s_i, then it can move to state s_j in the next step with a probability denoted by p_{ij} which is known as the transition probability. p_{ij} is time independent. That is, at any time, if the system is at state s_i it may move to state s_j with the same probability p_{ij}. The transition matrix P compiles all the transition probabilities from any state i to any state j as shown in Equation 5.2.

$$P = \begin{pmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,k} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,k} \\ \cdots & \cdots & \cdots & \cdots \\ p_{k,1} & p_{k,2} & \cdots & p_{k,k} \end{pmatrix} \tag{5.2}$$

In Markov chain, the state probability $\pi_n(i)$ is the probability that the system is in state i at the time step n. Consequently, the state probability distribution of the system at any time n, π_n, defines the state probabilities at that time as shown in 5.3.

$$\pi = \{\pi_n(i) \quad \forall\, i \in S\} = \{\pi_n(1), \pi_n(2), \pi_n(3), ..., \pi_n(k),\} \tag{5.3}$$

By definition, the summation of the state probabilities equals 1 as in Equation 5.4 because the system can be in only one state at any the time step:

$$\sum_{n=1}^{n=k} \pi_n(k) = \pi_n(1) + \pi_n(2) + \pi_n(3) + ..., + \pi_n(k) = 1. \tag{5.4}$$

Based on these definitions, if we know the state probability distribution π_n of the system at any time step n and the transition matrix P, then we can predict the probability distribution at next time step $n+1$, simply, by multiplying them together as shown in 5.5:

$$\pi_{n+1} = \pi_n P = \begin{pmatrix} \pi_n(1) \\ \pi_n(2) \\ \dots \\ \pi_n(k) \end{pmatrix}^T \begin{pmatrix} p_{1,1} & p_{1,2} & \dots & p_{1,k} \\ p_{2,1} & p_{2,2} & \dots & p_{2,k} \\ \dots & \dots & \dots & \dots \\ p_{k,1} & p_{k,2} & \dots & p_{k,k} \end{pmatrix}. \tag{5.5}$$

Using equation 5.5, we can compute the state probability distribution after any number of time steps:

$$\pi_n = \pi_0 P^n = \begin{pmatrix} \pi_n(1) \\ \pi_n(2) \\ \dots \\ \pi_n(k) \end{pmatrix}^T \begin{pmatrix} p_{1,1} & p_{1,2} & \dots & p_{1,k} \\ p_{2,1} & p_{2,2} & \dots & p_{2,k} \\ \dots & \dots & \dots & \dots \\ p_{k,1} & p_{k,2} & \dots & p_{k,k} \end{pmatrix}^n. \tag{5.6}$$

The Markov chain stationary distribution, which is also called the steady-state probability distribution, is defined as the state probability distribution as $n \to \infty$. If the process satisfies some conditions, then its Markov chain will have a unique stationary distribution. Markov chain and its states have many properties, we will briefly describe some of them, which will help us figure out whether the process has a unique stationary distribution. Subsequently, using these properties we can find whether the proposed Markov chain for the MAC in VANET has a unique stationary distribution.

Markov chain irreducibility

One of the important Markov chain properties is irreducibility, which means that regardless of the present state, the system can reach any other state in a finite number of steps. In other words, there is a path form any state to any other state. Mathematically, irreducibility can be written as:

$$\forall S_i, S_j \in S \; \exists m < \infty : Pr(X_{n+m} = s_j \,|\, X_n = s_i) > 0. \tag{5.7}$$

Transient states

In a Markov chain, a state s_i is said to be a transient state if the system may not be able to return to it again with a non-zero probability [101]. In other words, given that we start in state s_i, there is a non-zero probability that we will never return to it. Formally, if the random variable T_i is the first

return time to state s_i, then s_i is transient if

$$Pr(T_i < \infty) < 1. \tag{5.8}$$

Recurrent states

State recurrence is the reverse of transience, which means that state s_i is recurrent if it is not transient. In Markov chain, the system will return to the recurrent state in a finite time with probability equals 1. This time T_i is called the return time or hitting time. Formally, this relation can be written as:

$$Pr(T_i < \infty) = 1. \tag{5.9}$$

The mean recurrence time for state S_i, which is denoted by M_i, is the expected return or hitting time $M_i = E(T_i)$. The state is called positive recurrent if it has a finite mean hitting time.

State periodicity

A state s_i has period k if any return to s_i must occur in multiples of k time steps. In this case, k is calculated as the greatest common divisor (GCD) for the return time. Mathematically, the period of a state is defined as:

$$k = GCD\{n > 0 : Pr(X_n = s_i | X_0 = s_i) > 0\}. \tag{5.10}$$

If k is defined, then the state s_i is called a periodic state. Otherwise, it is aperiodic. A Markov chain is called aperiodic if every state in this chain is aperiodic.

An irreducible Markov chain only needs one aperiodic state to imply all states are aperiodic. Moreover, a Markov chain that is aperiodic and positive recurrent is known as ergodic.

Stationary distributions of Markov chains

Based on the above-mentioned definitions, many other properties of a Markov chain can be derived. Among them, the existence of stationary distribution and the uniqueness of this stationary

distributions are essential Markov chain properties.

The stationary distribution of a Markov chain is the probability distribution that remains unchanged in the Markov chain as the time progresses. Thus, by applying Equation 5.5 on the stationary distribution π, it satisfies Equation 5.11:

$$\pi = \pi P. \tag{5.11}$$

By solving Equation 5.11, we can easily compute the stationary π distribution for a Markov chain, given its transition matrix P.

The important question is under what conditions a Markov chain has a stationary distribution and whether it is unique. It was proven in [102] that for a finite ergodic Markov chain, there exists a unique stationary distribution.

This next section uses the Markov chain to represent the MAC process in VANET. The Markov chain we develop trivially satisfies the ergodicity condition, it is basically irreducible and ergodic. Thus, it has a unique stationary distribution that we utilize to find the packet delivery probability, drop probability, and delay.

5.3 Proposed Medium Access Model

In Chapter 2, we gave an overview of the VANET communication. This section proposes an analytical model for the MAC in VANET. So, this section firstly describes the MAC technique in VANET in more details. Then, it will discuss and validate the model assumptions. Subsequently, it will derive the new proposed analytical model.

5.3.1 WAVE Medium Access Technique

The main objective of the medium access technique is to enable several communication nodes connected to the same physical medium to share it. In VANET, the physical medium characteristics is described by the DSRC standard [31] as shown earlier in Chapter 2, Section 2.2.

The MAC in VANET supports four traffic access categories, Background (BK), Best-effort (BE), Video (VI), and Voice (VO). The required quality of services is granted to different ACs by the different medium access parameters (such as Arbitration Inter-Frame Space (AIFS) and Contention

Window (CW) limits) that are assigned to each access category. I addition to that, the access technique guarantees the access to the higher priority AC when there are multiple ACs attempting to transmit at the same time.

According to the IEEE 802.11p [31] standard, within the station, every AC acts as a stand-alone virtual station that has its own queue and its own access parameters. Whenever any AC has a frame to send, it initializes its back-off counter to a random value within the range of the initial *CW* (w_0). Then, when it detects that the medium is idle for a specific time length (usually equal to the AIFS), it counts down the back-off counter. If the medium is busy, the counter will be held. The station can send the frame only when the back-off counter becomes zero. When more than one EDCA count down their back-off timers to zero and attempt to transmit at the same time, a virtual collision takes place within the same station, which is referred to as internal collision. In such cases, the access to the medium will be granted to the highest priority AC. The lower priority colliding AC doubles its CW and backs-off again. In this chapter, we assume there is only one AC in every station, so we do not model the internal collision in our model.

If two stations start transmitting in the same time slot, an external collision will take place and both transmitted signals will be destroyed. Because the sender cannot detect the collision while transmitting, it has to wait for the acknowledgment (ACK). If an ACK was not received within a specified time, the sender assumes a collision, doubles its CW, and backs-off again. This process can be repeated until reaching a retransmission attempt threshold $(M+f)$. During these attempts, the CW can be increased up to the maximum *CW* (w_{max}). Consequently, the number of times the CW can be increased is $M = log_2(w_{max}/w_0)$, and f is the number of retransmission attempts allowed after reaching (w_{max}).

5.3.2 Model Assumption

The main purpose of this chapter is to enable large-scale modeling of communication in a vehicular environment. Chapter 3 showed that the discrete event simulations of communication in a large-scale network is computationally expensive in terms of both memory and simulation time, which are exponentially increasing with the number of vehicle in the network as shown in Chapter 3. Thus, in this chapter, instead of using the discrete event simulations for the communication, we develop a communication model that can estimate the average packet drop probability and the average packet delay. To allow for such large-scale systems, we simplified this model to assume a single AC instead of the four ACs in VANET. This assumption is based on a comparative simulation study between the single AC and multiple ACs in VANET. We ran a simulation using OPNET

simulator to compare the network performance in case of single AC and four ACs for different traffic rates. In the case of four ACs, each AC generates the same traffic rate λ. In the case of a single AC, a traffic rate of 4λ is generated and assigned to the best-effort AC, which is the default AC for the traffic. Then, the throughput of the single AC is divided by 4 and compared to the traffic throughput in the case of the four ACs.

Figure 5.1 shows the comparison of the traffic in the case of four ACs and $\frac{1}{4}$ the throughput of the single AC. Figure 5.1 shows that the throughput of the BE AC is very close to the approximated BE using a single AC. The comparison shows that the error is less than 11%. Based on this analysis and the results in Figure 5.1, we can use the single AC to represent the BE AC in the full-fledged model.

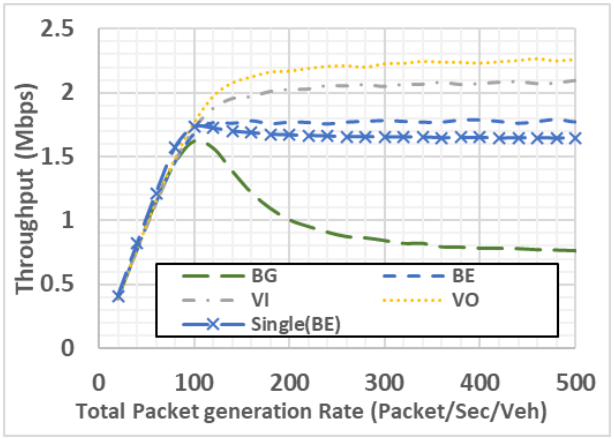


Figure 5.1: Comparison between the BE traffic using single AC and 4 ACs.

5.3.3 The Proposed Model versus Previous Models

Compared to the previous models in the literature, our proposed model is characterized by combining the following features:

- It considers the MAC layer queue size, so the model is capable of estimating both the average processing delay and queuing delay for the packet.
- It works for both saturated and unsaturated cases.
- It limits the number of retransmission attempts. When a packet reaches this limit, it will be dropped.

First, our proposed model considers the MAC layer queue size and its impact on communication performance. Most of the previous models that were developed for both IEEE 802.11a, IEEE 802.11e, and IEEE 802.11p do not consider the queue size, the queuing process, and their impact on the performance of IEEE 802.11. For example, the Bianchis model [103], which is the base for our proposed model that was proposed in 2000 and then refined in 2010 [104], as well as models proposed 2016 in [105, 106, 107], do not consider queuing at all. Many other models also do not take queuing into consideration at all, like [108, 109, 110, 111]. Some models consider queuing but assume an infinite queue size. Engelstad et al. [112] modeled the EDCA in IEEE 802.11e with an infinite queue, meaning the packet always finds a buffer to be stored in, which is not a realistic assumption. Moreover, the queue size can have a significant impact on the performance of IEEE 802.11 communication. For instance, the smaller the queue size, the lower the number of packets can be queued. In the case of high packet traffic rates, many of the packets will be rejected by the queue, which will increase the packet drop ratio. On the other hand, larger queue size will result in increasing the queuing delay to very long delays. In contrast to these models, our model assumes a finite queue size of length K in the MAC layer, which makes it more realistic. To model the queuing process, in the proposed model, we use the M/M/1/K queuing model [113], which is a model for a queue of a finite length (K) in a system having a single server and arrivals of packets are determined by a Poisson process and packet service times have an exponential distribution . This queuing model is incorporated with the MAC protocol so that the back-off technique and the queue interact with each other. Consequently, with the help of this queuing model, we were able to compute both the queuing and processing delay. In addition, the queuing model parameters were used to estimate the throughput and the packet drop rate.

Secondly, compared to most of the previous research, our proposed model supports both saturated and unsaturated data traffic conditions. For example, the models in [103, 104, 106, 107, 108, 109, 110, 111] assume saturated conditions in the modeling of the DCA/ECDA using a Markov chain. A model was developed by Engelstad et al. [112] for both unsaturated and saturated traffic conditions, but it assumes an infinite queue. Thus, it cannot realistically estimate the delay and throughput.

Thirdly, many of the models in the literature do not limit the number of retransmissions for each individual packet. This is basically to simplify the mathematical derivation of the model. Our proposed model assumes a limited number of retransmission attempts, after which the packet will be dropped. These three characteristics are illustrated in Figure 5.2 and the model shall be described in Subsection 5.3.4.

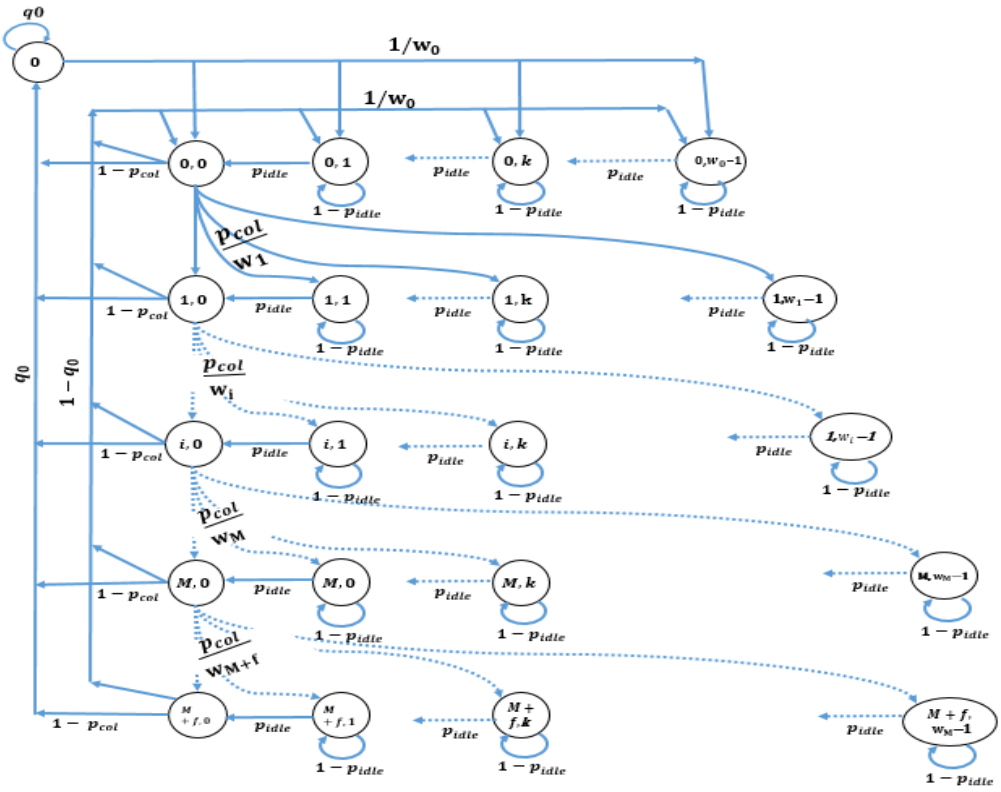


Figure 5.2: Markov chain model for the medium access.

5.3.4 Model Derivation

To build a model for the MAC technique, a two-dimensional Markov chain, shown in Figure 5.2, is utilized. The state 0 in the model represents the system-empty state when both the system and the queue are empty. This is a basic difference between our model and many of the previous models that only support saturated cases such as the Bianchis model [103, 104], which is the base for many other models. Bianchis model assumes saturated systems, which means the system will always have packets in its queue. In these models, after successful transmissions or exceeding the retransmission attempts, the process returns to the first stage directly to process the next packet. To enable our model to handle both saturated and non-saturated cases we added state 0, system-empty state, to the process. If the queue is empty, the process will return back to this system-empty state and will stay in this state until a new packet arrives. Packet arrival process is assumed to have exponential distribution for the inter-arrival times, which is modeled by the queuing model.

Each of the other states is defined by (i, j), where i and j are the back-off stage and back-off

counter value, respectively. Table 5.1 shows the symbols used for this model.

Table 5.1: The model parameters

Symbol	***Description***
i	The back-off stage number
j	The back-off counter
M	The maximum number of times of increasing the CW
f	The maximum number of times of retransmission without increasing the CW
w_i	The CW range for stage i
w_0	The initial value for the maximum CW i
α	The CW increasing factor, where $w_i = w_0\alpha^i$. Its typical value is 2.
$p_{idle_{slot}}$	The probability that a medium is idle in any time slot
p_{idle}	The probability that the medium is idle
q_0	The probability that the system is empty (no packet in the system)
p_{suc}	The probability that a medium is occupied with a successful transmission
p_{fail}	The probability that a medium is occupied with a failure transmission
p_{tran}	The probability that a station starts transmission in any time slot
p_{col}	The probability that the packet collides
$P(i,j)$	The probability that the system is in state (i,j)
λ	The packet arrival rate
μ	The packet service rate
T_{serv}	The packet service time
T_q	The packet queuing time
N	The number of vehicles (stations) in the communication range
T_{slot}	The length of the time slot (sec)
T_s	The transmission time of a successful frame transmission
T_f	The transmission time of a failure frame transmission
$AIFS$	The number of time slots for the Arbitration Inter Frame Space
ρ	The traffic intensity for the queuing model
K	The size of the buffer
Q_n	The probability that the queue has n packets
λ_{eff}	The effective packet generation rate

To solve this model, we need to compute the stationary state probability distribution for each state in the model. We use the notation $P(i,j)$ and $P(0)$ to represent stationary state probability of state (i,j) and state 0 respectively. Thus, we derive all the state probabilities $P(i,j)$ in addition to $P(0)$ as functions of $P(0,0)$. Then, according to Equation 5.4, the summation of all these state probabilities should equal 1. A detailed derivation for the equations is in Appendix A.

From the Markov chain in Figure 5.2, we can find that the probability that the system is in state 0 as:

$$P(0) = \frac{q_0}{1-q_0}\left(P(M+f-1,0) + (1-p_{col})\sum_{i=0}^{M+f-2} P(i,0)\right) \tag{5.12}$$

And $P(0,j)$ can be expressed as:

$$P(0,j) = \frac{w_0 - j}{w_0}\,\frac{1-q_0}{p_{idle}}\left(P(0) + P(M+f-1,0) + (1-p_{col})\sum_{i=0}^{M+f-2} P(i,0)\right) \quad ,j = 1,2,...,w_0-1 \tag{5.13}$$

and $P(0,0)$ is:

$$P(0,0) = (1-q_0)\left(P(0) + P(M+f-1,0) + (1-p_{col})\sum_{i=0}^{M+f-2} P(i,0)\right) \tag{5.14}$$

From Equations 5.13 and 5.14, we can drive $P(0,j)$ as:

$$P(0,j) = \frac{w_0 - j}{w_0}\,\frac{1}{p_{idle}}P(0,0) \quad ,j = 1,2,...,w_0-1 \tag{5.15}$$

Consequently, $P(i,0)$ can be calculated as:

$$P(i,0) = p_{col}^{i}\,P(0,0) \quad ,i = 0,1,....,M+f-1 \tag{5.16}$$

and

$$\begin{aligned} P(i,j) &= \frac{w_i - j}{w_i}\,\frac{p_{col}^{i}}{p_{idle}}P(0,0) \\ &,i = 1,2,...,M+f-1 \quad and \quad j = 1,2,...,w_i-1 \end{aligned} \tag{5.17}$$

From equations 5.12 and 5.16 we can find the relation between $P(0)$ and $P(0,0)$ as :

$$P(0) = \frac{q_0}{1-q_0}P(0,0) \tag{5.18}$$

Now we have all the state probabilities expressed in terms of $P(0,0)$. Since the summation of all

the probabilities equals 1, then

$$P(0)+\sum_{i=0}^{M+f-1}P(i,0)+\sum_{k=1}^{w_0-1}P(0,k)+\sum_{i=1}^{M+f-1}\sum_{k=1}^{w_{i-1}}P(i,k)=1 \tag{5.19}$$

Notice that the window exponential factor is α for $i \leq M$, i.e

$$w_i=\begin{cases} w_0\,\alpha^i & i\leq M \\ \\ w_0\,\alpha^M & i>M \end{cases} \tag{5.20}$$

From Equations 5.19 and 5.20, we can calculate $P(0,0)$ as shown in Equation 5.21.

$$\begin{aligned} P(0,0)=\Bigg(&\frac{q_0}{1-q_0}+\frac{1-{p_{col}}^{M+f}}{1-p_{col}}+\frac{w_0-1}{2\,p_{idle}}+\\ &\frac{1}{2\,p_{idle}}\Big[(\alpha^{M-1}\,w_0-1)\frac{{p_{col}}^{M-1}-{p_{col}}^{M+f}}{1-p_{col}}\\ &+w_0\frac{\alpha\,p_{col}-(\alpha\,p_{col})^{M-1}}{1-\alpha\,p_{col}}+\frac{p_{col}-(p_{col})^{M-1}}{1-p_{col}}\Big]\Bigg)^{-1} \end{aligned} \tag{5.21}$$

Now, to solve this model, we have to calculate the values of p_{col}, p_{idle}, and q_0. To do that, we need to find a relationship between these three parameters and the state probabilities.

A collision will happen when two or more stations start transmission in the same time slot. Let the probability that a station starts transmitting at a time slot be p_{trans}, then

$$p_{trans}=\sum_{i=0}^{M+f-1}P(i,0). \tag{5.22}$$

When a station sends a packet, the probability that this packet collides is

$$p_{col}=1-(1-p_{tran})^{N-1}. \tag{5.23}$$

So for the entire system, the medium will be idle at any time slot only when no station is sending:

$$p_{idle_{slot}}=(1-p_{tran})^N. \tag{5.24}$$

and the station decides that the medium is idle (p_{idle}) after $AIFS$ idle time slots in a row:

$$p_{idle} = {p_{idle_{slot}}}^{AIFS}. \tag{5.25}$$

For the entire system, the probability that a packet is successfully transmitted without collision is:

$$p_{suc} = \binom{N}{1} p_{tran}\,(1-p_{col}) = N\,p_{tran}\,(1-p_{col})$$

$$p_{suc} = N\,p_{tran}\,(1-p_{tran})^{N-1}. \tag{5.26}$$

Using Equations 5.22 through 5.26, we can calculate the two parameters p_{col} and p_{idle} in terms of state probabilities, therefore in terms of $P(0,0)$. The only missing part is finding a relation between q_0 and the state probabilities. To solve for q_0, we use the $M/M/1/K$ model, where we assume the packet inter-arrival time between packets is totally random (i.e. exponentially distributed) with an average rate λ. Assuming that the service rate is $\mu = \frac{1}{T_{serv}}$, the service time T_{serv} is the average time elapsed in processing the packet. It is the summation of the average time the packet stays in each stage. The packet can stay a time T_w in every state plus the average frame transmission time $T_{tr_{av}}$, which can be calculated as:

$$T_w = p_{fail}\,T_f + p_{suc}\,T_s + \frac{1}{p_{idle}} T_{slot} \tag{5.27}$$

$$T_{tr_{av}} = p_{col}\,T_f + (1-p_{col})\,T_s, \tag{5.28}$$

where $p_{fail} = 1 - p_{suc} - p_{idle_{slot}}$. T_s and T_f are the successful transmission time and the failure transmission time respectively. T_s and T_f depends on whether MAC uses the basic or advanced (RTS/CTS) access modes. In case of the basic access mode, the transmitting node directly sends its packet.

$$T_s = T_{AIFS} + T_{frame} + T_{pro} + T_{SIFS} + T_{ack} + T_{pro} \tag{5.29}$$

$$T_f = T_{AIFS} + T_{frame} + T_{pro}. \tag{5.30}$$

In the case of the RTS/CTS access mode, the transmitting node firstly sends a request to send a

packet and waits for the clear to send, after the successful handshaking, the sender can send its data frame. In this case, the successful and failed transmission times are calculated as:

$$\begin{aligned} T_s = {} & T_{AIFS} + T_{rts} + T_{pro} + T_{SIFS} + T_{cts} + T_{pro} \\ & + T_{SIFS} + T_{frame} + T_{pro} + T_{SIFS} + T_{ack} + T_{pro} \end{aligned} \tag{5.31}$$

$$T_f = T_{AIFS} + T_{rts} + T_{pro}. \tag{5.32}$$

The term T_w in Equation 5.27 is the time required by the station while sensing the medium to ensure it is idle. Consequently, the service time T_{serv} can be calculated as:

$$T_{serv} = T_{slot} \sum_{i=0}^{M+f-1} \left(\frac{T_w(w_i - 1)}{2} + T_{tr_{av}} \right) {p_{col}}^i. \tag{5.33}$$

Now we can calculate the traffic intensity $\rho = \frac{\lambda}{\mu}$, and subsequently q_0 as:

$$q_0 = \begin{cases} \frac{1-\rho}{1-\rho^{K+1}} & \lambda \neq \mu; \\ \\ \frac{1}{K+1} & \lambda = \mu. \end{cases} \tag{5.34}$$

Now we can solve this model and estimate the total communication throughput and delay. The total network throughput Thr can be calculated as:

$$Thr = N(1 - q_0)\big(1 - P(M + f - 1, 0)p_{col}\big)(1 - p_{fail}). \tag{5.35}$$

Thr is simply the multiplication of a set of terms: $(1 - q_0)$ represents the time ratio at which the system has at least a packet, $\big(1 - P(M + f - 1, 0)p_{col}\big)$ describes the probability that this packet was not dropped due to the maximum retransmission attempts constraint, and $(1 - p_{fail})$ is the portion of time that is not occupied by collisions.

5.3.5 Communication Model Validation

To validate this communication model, we ran extensive simulations for different traffic rates and number of communication stations considering V2I communication. We used the OPNET soft-

ware, which is known currently as the Riverbed modeler [86]. The OPNET Modeler is a powerful discrete event simulation tool for specification, simulation, and analysis of data and communication networks. The most important OPNET characteristic is that its results are trusted because its implementations of the standard protocols are tested and validated before publishing. Current versions of OPNET support WAVE as an extension of the IEEE 802.11 implementation. For each simulation scenario, we calculated the average network throughput and the average packet delay. The results show an accurate estimation of both throughput and delay of our model compared to the OPNET simulated results (Sim), as shown in Fig 5.3 and Fig 5.4.

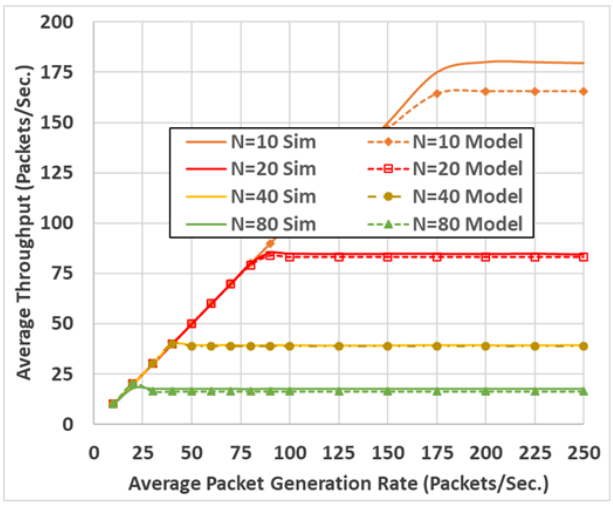
5.4 Transportation Traffic Modeling

The transportation network is the environment where the vehicular communication takes place. Thus, it is essential to integrate the proposed communication model into a scalable transportation simulation software. After validating the communication model, we implemented and incorporated it into the INTEGRATION software [48]. The INTEGRATION simulation model provides 10 basic user equilibrium traffic assignment/routing options [48]. Recently, we have added a system equilibrium routing model. One important feature of INTEGRATION is its support for eco-routing traffic assignment, which uses feedback from the en-route vehicles about the fuel consumption cost for the road links they traverse. By fusing this real-time cost feedback with the link history it can calculate the time-varying smoothed link costs. Using these costs, it tries to minimize the fuel consumption and emission levels by assigning vehicles the most environmentally friendly routes.

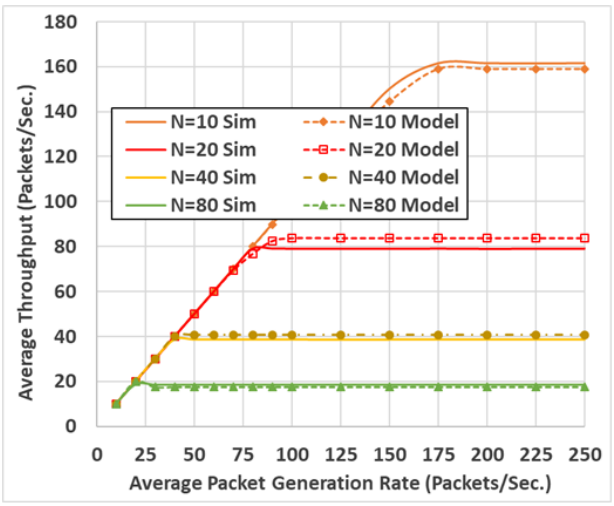
As all of the previous research work in this area, INTEGRATION assumes a perfect communication network with no packet drop or delay. By incorporating the proposed communication model into the INTEGRATION software, we develop a new scalable framework that is capable of modeling large-scale transportation network and capture the mutual impact of the communication and transportation systems. We use this framework to study the impact of communication on the eco-routing application as an example for the ITS applications.

5.4.1 Eco-routing with the Communication

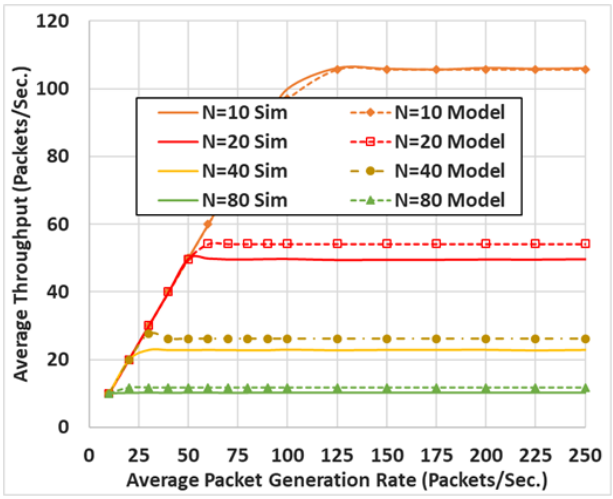
To the best of our knowledge, in all the previous implementations of the eco-routing, a perfect communication network is assumed, which means the drops and delay are assumed to be zero. The INTEGRATION also makes this assumption as described in details in Section 2.4.



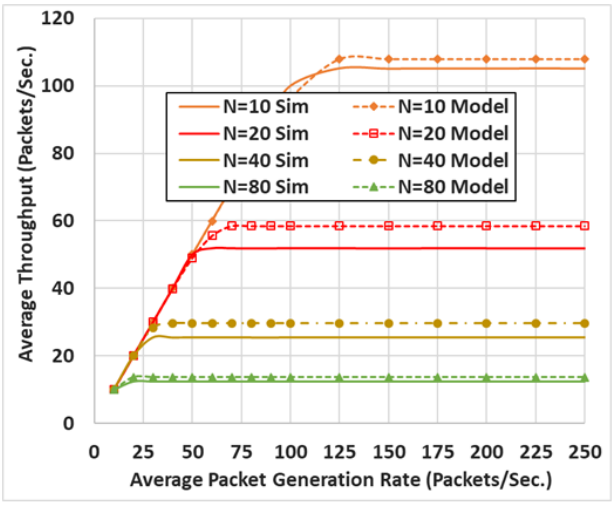
(a) Basic Access Mode, Packet Size = 500 Bytes



(b) RTS/CTS Access Mode, Packet Size = 500 Bytes



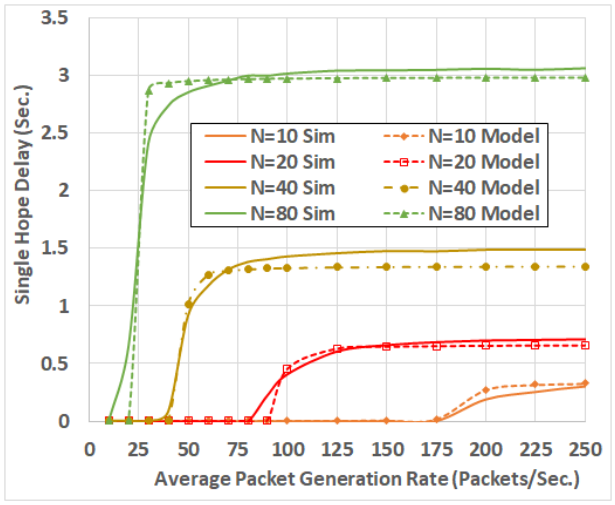
(c) Basic Access Mode, Packet Size = 1000 Bytes



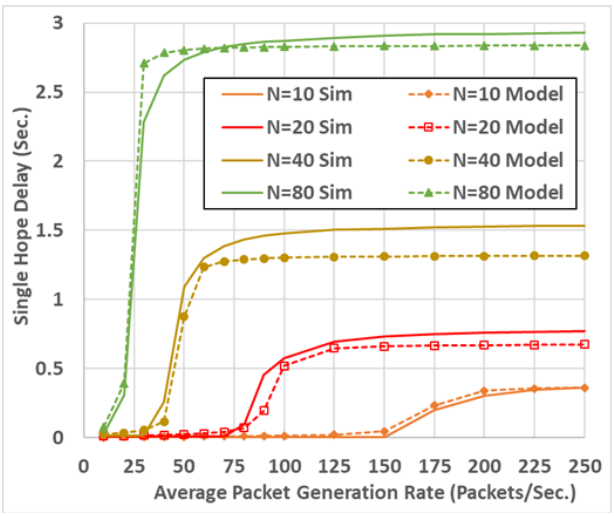
(d) RTS/CTS Access Mode, Packet Size = 1000 Bytes

Figure 5.3: Average throughput per vehicle (Packets/Second) versus packet generation rate (Packets/Second), comparing the model to the simulation for different number of vehicles

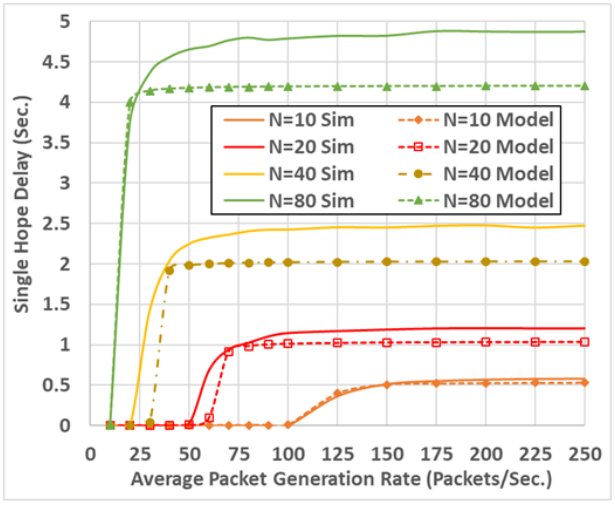
Thus, to build a realistic ITS simulation framework, we have changed the INTEGRATION behavior to adopt the communication process and its impact on the ITS applications including the eco-routing. The new behavior is illustrated in Figure 5.5. When a vehicle finishes a road link, instead of directly updating the link cost in the TMC as shown in Figure 2.2, the vehicle sends this information to the communication module by adding it to the transmission queue. Then, while the vehicle moves, the communication module checks for the connectivity. If the vehicle is not connected to an RSU (i.e., there is no RSU in its communication range), the queue will not be



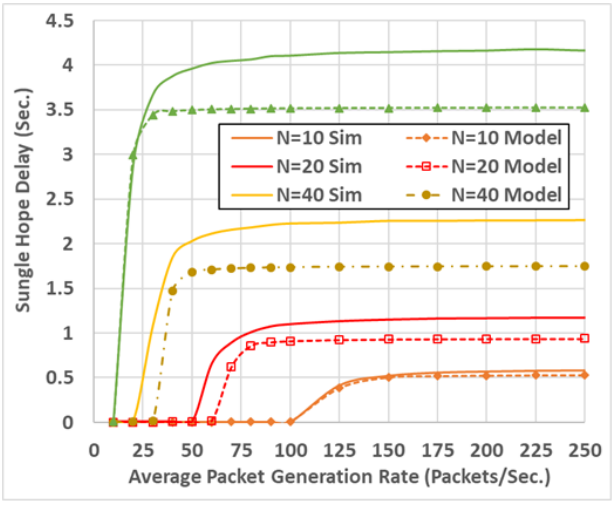
(a) Basic Access Mode, Packet Size = 500 Bytes



(b) RTS/CTS Access Mode, Packet Size = 500 Bytes



(c) Basic Access Mode, Packet Size = 1000 Bytes



(d) RTS/CTS Access Mode, Packet Size = 1000 Bytes

Figure 5.4: Average single hop delay, model versus simulation

processed and the packets in the queue will be held. Whenever it gets connected to an RSU, it will process the packets in the queue.

For each packet in the transmission queue, the communication module first calculates its drop probability as described later in Equation 5.38. If the packet should be delivered, the communication module calculates its average total delay and inserts it into a time-based ordered queue. So, it will be processed by the updating module in its time of arrival.

To calculate the packet drop probability and total delay, the communication model parameters

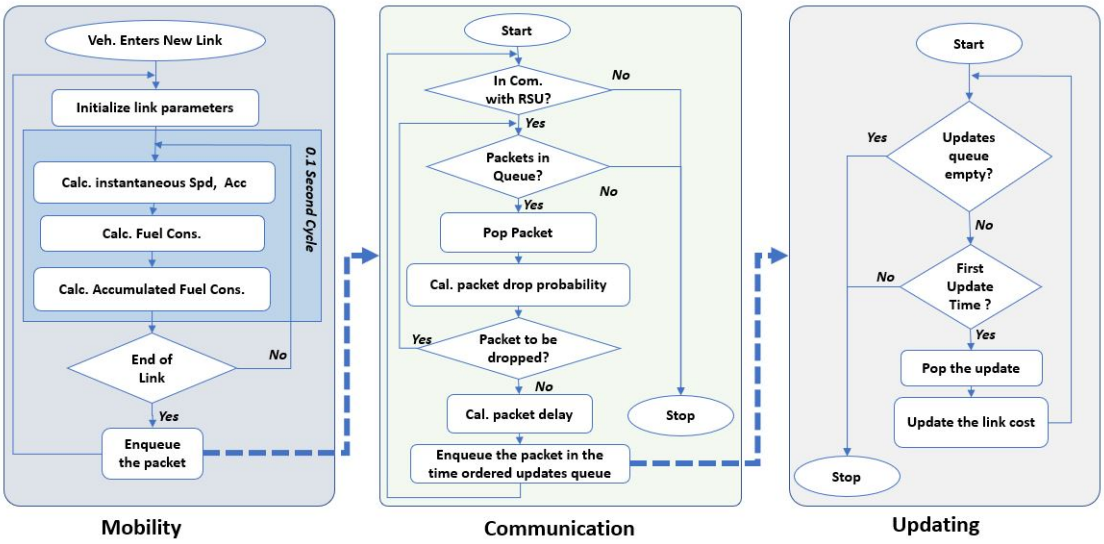


Figure 5.5: Eco-routing with the communication.

introduced in section 5.3.4 are used.

In regards to the packet drop probability, the packet can be dropped in two cases: 1) it arrives at the MAC while the queue is full, and 2) when its maximum retransmission attempts are reached.

When the queue is full, the packet will be rejected by the queue. We call this the rejection probability P_{rej}. In other words, P_{rej} is the probability that the queue has K packets. If Q_n is the probability that the queue has n packets, then according to the $M/M/1/K$ model [113], Q_n can be calculated as:

$$Q_n = \begin{cases} \rho^n \frac{1-\rho}{1-\rho^{K+1}} & \lambda \neq \mu; \\ \\ \frac{1}{K+1} & \lambda = \mu. \end{cases} \tag{5.36}$$

Consequently, the $P_{rej} = Q_K$, that is

$$P_{rej} = \begin{cases} \rho^K \frac{1-\rho}{1-\rho^{K+1}} & \lambda \neq \mu; \\ \\ \frac{1}{K+1} & \lambda = \mu. \end{cases} \tag{5.37}$$

For the second case, a packet in the queue might be dropped if it experienced a collision in its last retransmission attempt in the last stage whose probability is $P(M+f-1,0)\ p_{col}$.

If neither of these cases happened, then the packet will be correctly delivered. Thus, the packet

drop probability can be calculated as:

$$P_{drop} = 1 - (1 - P_{rej})\Big(1 - P(M + f - 1, 0)\ p_{col}\Big). \tag{5.38}$$

The total delay of the packet T_{delay} is the summation of both service time T_{serv} and the queuing delay T_q as shown in Equation 5.39.

$$T_{delay} = T_{serv} + T_q. \tag{5.39}$$

Where the average service time T_{serv} is calculated by Equation 5.33, and the average queuing delay T_q can be calculated as:

$$T_q = \frac{1}{\mu - \lambda_{eff}}. \tag{5.40}$$

In Equation 5.40, λ_{eff} is the effective packet generation rate which is the actual number of packets that can enter the queue per unit time. I.e., λ_{eff} is the packets that arrive when the queue is not full. If the Q_n is the probability that the queue has n packets, then λ_{eff} can be calculated as:

$$\lambda_{eff} = \lambda(1 - P_{rej}) = \lambda(1 - Q_K). \tag{5.41}$$

Using Equations 5.37 and 5.39 through 5.41, the average total packet delay can be calculated.

5.5 Simulation and Results

We used the developed model to study the mutual impacts of the communication system and the ITS feedback-based eco-routing application [72, 80]. These impacts include how the traffic congestion level and the traffic flow are affected by the communication errors. We also quantify these impacts of communication errors on the fuel consumptions, the travel time, the average vehicle speed, and the average emission levels. This section also shows how the vehicle demand level influences the communication performance in terms of packet drop rate and delay. In this study we use the V2I communication paradigm with $1000m$ communication range and 50 $Packets/second$ background packet generation rate (λ). The average packet size is set to 1000 $Bytes$ and the queue size is set to 64 $Packets$. In this study, we use two main communication scenarios; the ideal communication scenario assumes perfect communication (no drops nor delay), and the realistic communication case where the packets can be dropped and/or delayed based on the surrounding

network conditions.

The network shown in Figure 5.6 is used for the simulation analysis. This network is the downtown area in the city of Los Angles (LA). The modeled road network is shown by the red polygon in Figure 5.6 which is about $133Km^2$. It has 1625 nodes, 3561 links, and 459 traffic signals. In regards to the vehicular traffic demand, we use a calibrated traffic demand. The traffic calibration, which is based on the data collected from multiple sources, is described in detail in [20]. This traffic demand represents the morning peak hours in downtown area of the city of LA, which continues for 3 hours from 7:00 am to 10:00 am. We added one hour for traffic pre-loading. So, the demand runs for four hours. However, we run the simulation for 30,000 seconds to give the vehicles enough time to finish their trips. In order to study the impact of different traffic origin-destination demand (OD) levels, the calibrated traffic rates are multiplied by scaling factors (ODSFs) 0.1 through 1 at step 0.1. So, we have 10 traffic demand levels. The total number vehicles that should be simulated in each of these scenarios is shown in Table 5.2.

5.5.1 RSU Allocation

Since we focus on the V2I communication, we have to allocate the RSUs in the network. In Chapter 4, RSU allocation is shown to be critical to the performance of the communication network and consequently the performance of the eco-routing application. The most economical method is to install the RSUs at the traffic signals locations. The reason is that these traffic signals are already equipped with the required network connections and power sources. Consequently, the only requirement will be the RSU units and their antennas. In the road network of downtown LA, there are 459 traffic signals. So, the question now is which traffic signals should be selected to install the RSUs in order to achieve the best coverage with the minimum cost. This is a min-max coverage problem. To achieve this objective, we use a greedy algorithm shown in Algorithm 1.

Assuming that the distance between the traffic signals S_i and S_j is $D_{i,j}$, and C_i is the set of traffic signals covered by S_i. I.e., $C_i = \{S_j : D_{i,j} < R_{Com}\}$, where R_{Com} is the communication range. The algorithm starts with S includes all the traffic signals and empty set G of selected signals. It calculates the coverage for each signal in S. Then, it selects the traffic signal that covers the maximum number of uncovered signals, adds it to the selected signals G and removes it, along with all the signals it covers, from S. Steps 5 to 8 are repeated until S becomes empty. This algorithm does not guarantee to cover the whole network. But it covers the maximum signalized intersections with the minimum cost (minimum number of RSUs). Figure 5.6 shows the coverage map in the cases of 1000m communication range.

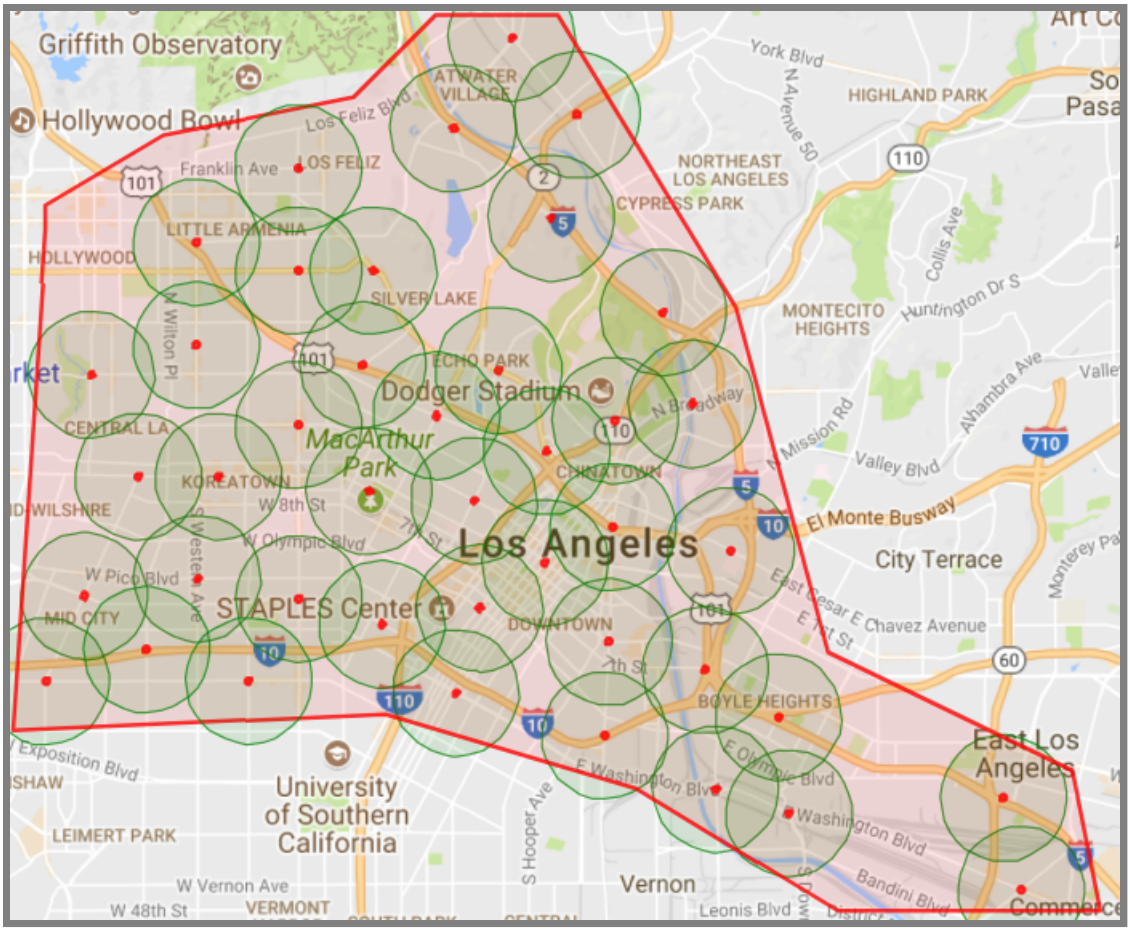


Figure 5.6: The LA downtown area and the coverage map for 1000m communication ranges.

Algorithm 1 Select traffic signals to install RSUs

1: **procedure** SELECT TRAFFIC SIGNALS ▷ Select the minimum number of traffic signals to install RSUs in such a way that maximizes the coverage
2: $S \leftarrow \{S_i : i = 1, 2....\}$
3: $G \leftarrow \phi$ ▷ The initial solution
4: **while** $S \neq \phi$ **do** ▷ There are uncovered signals
5: $C_i = \{S_j \in S : D_{i,j} < R_{Com}\}$ ▷ Recalculate the coverage
6: *Select* $S_i \in S$ *that maximizes* $\|C_i\|$
7: $G \leftarrow G \cup S_i$
8: $S \leftarrow S \setminus C_i$
9: **return** G ▷ The selected signals

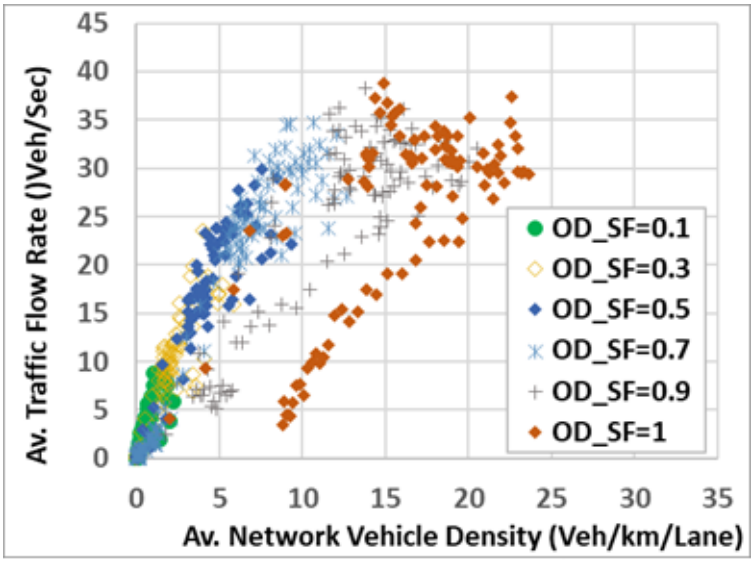
5.5.2 The Communication Impact on the Traffic Flow and Network Congestion

First, we run the network with different traffic demands levels in both ideal communication and realistic communication cases. To find the congestion level, we calculate the fundamental diagram [114] in each case, as shown in Figure 5.7.

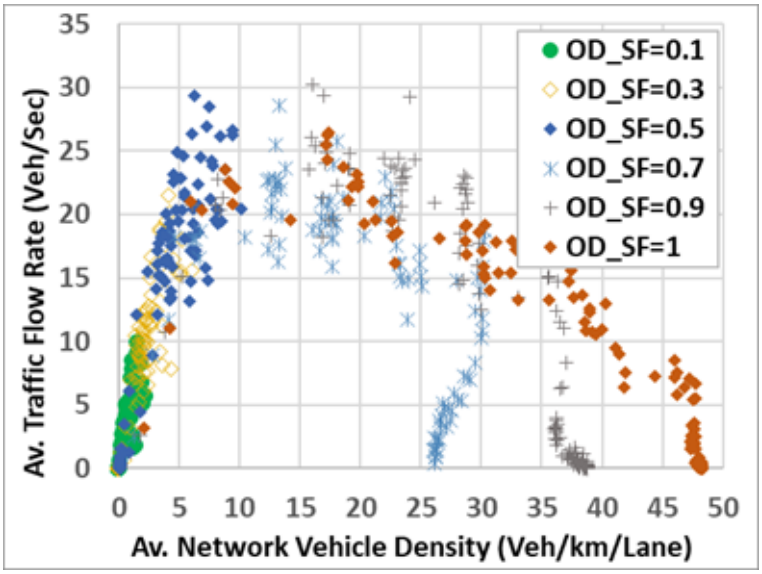
The network fundamental diagrams show that in the ideal communication case, the network reaches the congested regime only at $ODSF = 0.9$ and $ODSF = 1$. However, in the realistic communication case the congested regime takes place at lower traffic demand level at $ODSF = 0.7$. It also shows that there is a deadlock in the network, that results in a large number of vehicles being unable to exit the network at ODSFs of 0.7 through 1 in the realistic communication case. Moreover, Figure 5.7 demonstrates that at the full demand level, the maximum average vehicle density is less than 25 $vehicles/Km/lane$ in the case of ideal communication compared to about 47 $vehicles/Km/lane$ in the realistic communication case, which means that the congestion levels in the case of realistic communication is approximately twice that in the case of ideal communication.

Table 5.2 shows that in the case of ideal communication only 0.63% and 2.59% of the vehicles did not finish their trips in the two highest traffic demand levels, respectively. It also shows that 0.04% and 1.02% were not able to enter the network, because there are no available spots in their entrance links. However, in the realistic communication case at traffic scale of 0.7, about 16.88% of the vehicles were not able to finish their trips, and 16.45% were not able to enter the network. At the full traffic demand, it shows that 22.41% of the vehicles did not finish and 36.39% had no chance to enter the network because of the congestion. This higher congestion is combined with lower traffic rate exiting the network in the realistic communication case. This high congestion level hinders some vehicles from entering the network at the scheduled time. It also prevents some vehicles from finishing their trips.

From Table 5.2 and Figure 5.7, we can conclude that the realistic communication results in packet drops and delays that lead to incorrect routing decisions that cause the network to be highly congested at a lower traffic demand compared to ideal communication cases.



(a)



(b)

Figure 5.7: The Network Fundamental diagrams (a) With Ideal Communication and (b) With Realistic Communication

Table 5.2: Vehicles count comparison for different traffic scaling factors

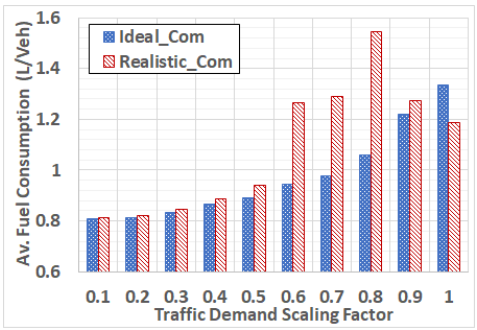
OD SF	Total No. of vehicles	%Vehicles started and finished		% Vehicles entered but didn't finish		% Vehicles deferred	
		Ideal	Realistic	Ideal	Realistic	Ideal	Realistic
0.1	50273	100	100	0	0	0	0
0.2	107047	100	100	0	0	0	0
0.3	164499	100	100	0	0	0	0
0.4	222326	100	100	0	0	0	0
0.5	277973	100	100	0	0	0	0
0.6	338366	100	100	0	0	0	0
0.7	394313	100	74.67	0	16.88	0	8.45
0.8	450670	100	66.87	0	21.05	0	12.08
0.9	507427	99.33	60.83	0.63	20.01	0.04	19.16
1	563626	96.39	41.2	2.59	22.41	1.02	36.39

5.5.3 The Communication Impact on Mobility Sustainability

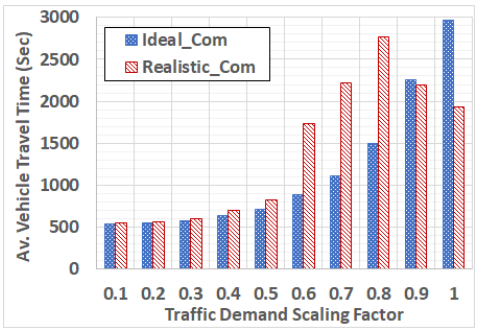
In the realistic communication scenarios, the incorrect routing and high congestion levels resulting from the packet drops and delays are expected to result in higher fuel consumption levels, longer travel times, and longer delays compared to the ideal communication case. Figure 5.8 compares mobility parameters in the two cases for the different traffic levels.

Figure 5.8-a shows that :- (1) at the low traffic demand levels, $OSDF = 0.1, 0, 2$ and 0.3, the fuel consumption per vehicles has insignificant differences between the two cases, (2) as the OSDF increases, the fuel consumption per vehicle in the case of realistic communication case becomes significantly higher than that in the ideal case, and (3) at the two highest traffic demands, $OSDF = 0.9$ and $OSDF = 1$, the fuel consumption per vehicles in the realistic communication case becomes lower. The average travel time and delay have the same behavior as shown Figure 5.8-b and Figure 5.8-c. The emission levels in Figure 5.8-d, Figure 5.8-e, and Figure 5.8-f have similar trends.

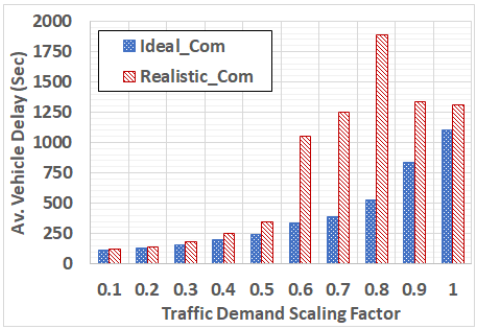
The third note does not align with increasing the OD traffic demand. The higher traffic demand should result in higher vehicle density, higher packet drop rate, and longer delay, which leads to incorrect routes, higher congestion levels, and higher fuel consumption levels. This behavior of the output is due to two reasons. First, these high vehicular demand levels produce high congestion levels, as shown in Figure 5.7, consequently, a larger number of vehicles will not be able to finish their trips as shown in Table 5.2. These vehicles are not accounted for in the fuel consumption, because the simulation software only considers the vehicles that finished their trips. This means, the estimated fuel counts only for $60.83\%(308667 vehicles)$ and $41.2\%(232213 vehicles)$ of the



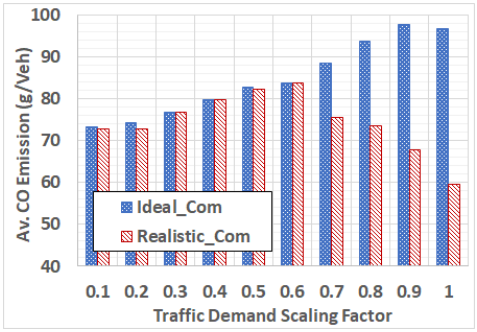
(a) Average fuel consumption per vehicle



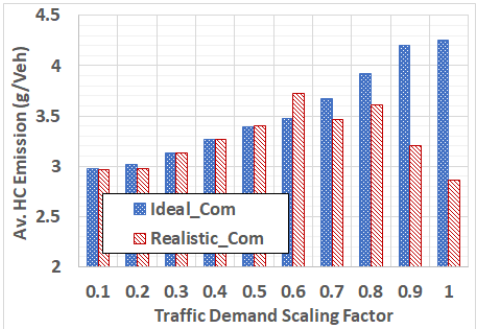
(b) Average vehicle travel time



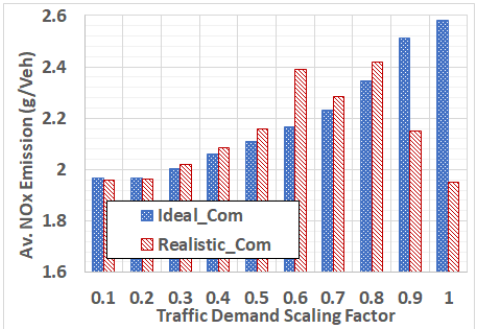
(c) Average vehicle delay



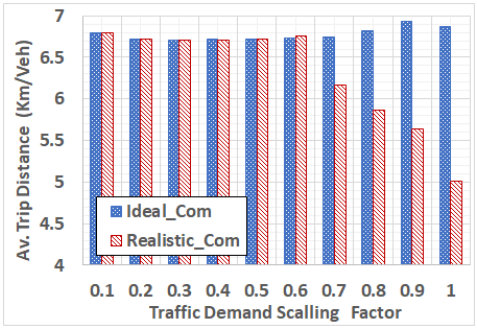
(d) Average CO emission per vehicle



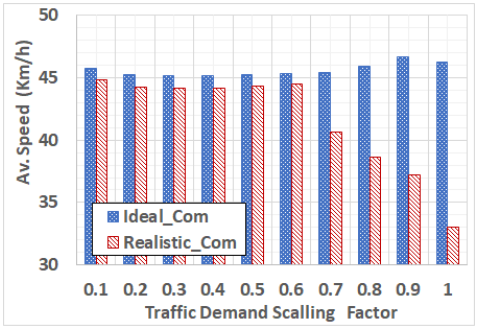
(e) Average HC emission per vehicle



(f) Average NOx emission per vehicle



(g) Average trip distance



(h) Average traveling speed

Figure 5.8: The outputs for the ideal communication versus the realistic communication

vehicles in the last two scenarios respectively, in the realistic communication case, compared to about $99.33\%(504027 vehicles)$ and $96.39\%(543279 vehicles)$ in the ideal communication case.

The second reason is that most of the trips that were finished are short trips, because the short trips do not pass through the middle of the road network, which experiences high congestion as shown in Figure 5.9. Figure 5.8-g shows that the average distance per trip significantly decreased in higher traffic demand rates in the realistic communication scenarios. This shorter distance justifies the lower fuel consumption, emissions, and travel times in the highest traffic demand levels.

Figure 5.8-h shows the average speed in the network for the different traffic demand levels in both the ideal and the realistic communication cases. It shows that the average speed at the highest traffic level is reduced from $46 Km/h$ to about $33 Km/h$, which also conforms with the high congestion level occurred in realistic communication cases.

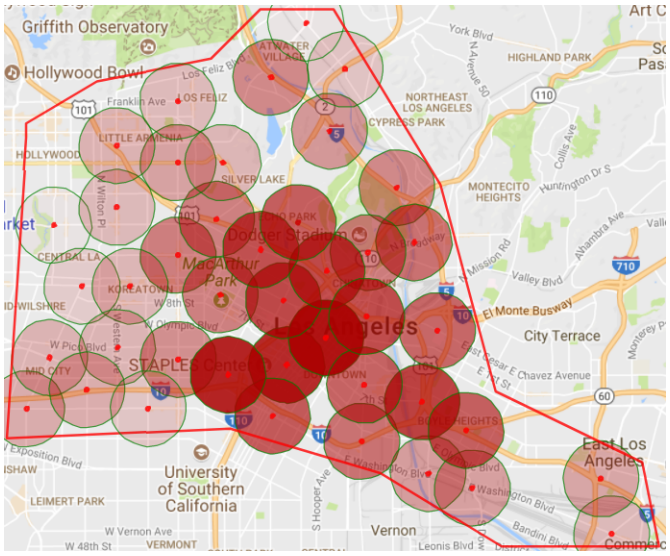
5.5.4 The Traffic Congestion Level Impact on the Communication Performance

The packet drop rate and delay are sensitive to the vehicle density. The higher the vehicular traffic demand, the higher the packet drop rates and the longer the delay.

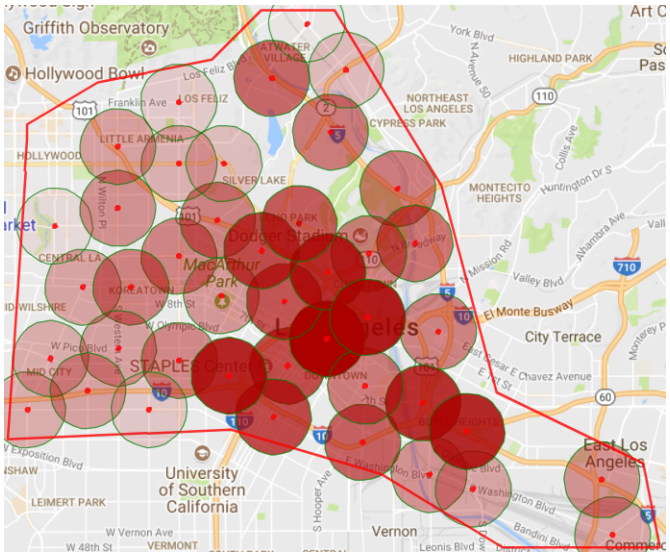
The traffic congestion level impact on the communication drop rate

Figure 5.10-a shows the distribution of the drop probability for different traffic demand levels. It shows that the packet drop probability is exponentially increasing with the traffic level. Figure 5.10-b shows the mean drop probability for each traffic demand level. Combining Figure 5.10 and Figure 5.8, we can see that at OSDF of 0.3, the eco-routing is not significantly affected by the high drop rate, which reaches about 93% of the transmitted packets.

This means that the eco-routing can work properly even at high packet drop rate. The reason is that a high packet drop rate is usually accompanied by a high vehicular traffic rate. This high vehicular traffic produces high redundant cost reporting packets from the vehicles for each road link. Thus, dropping these redundant packets will not significantly affect eco-routing decisions.



(a) ODSF=0.8

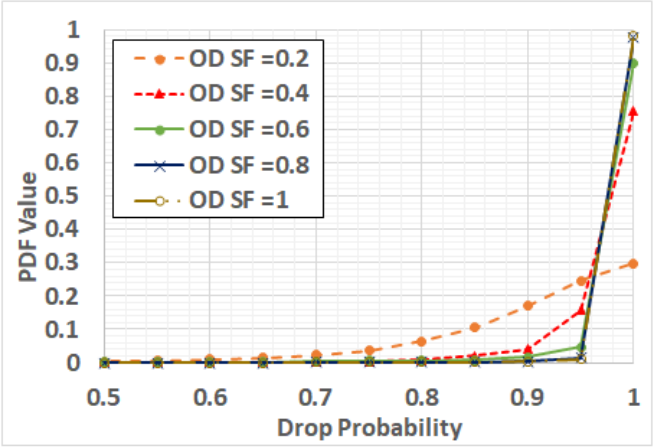


(b) ODSF=1

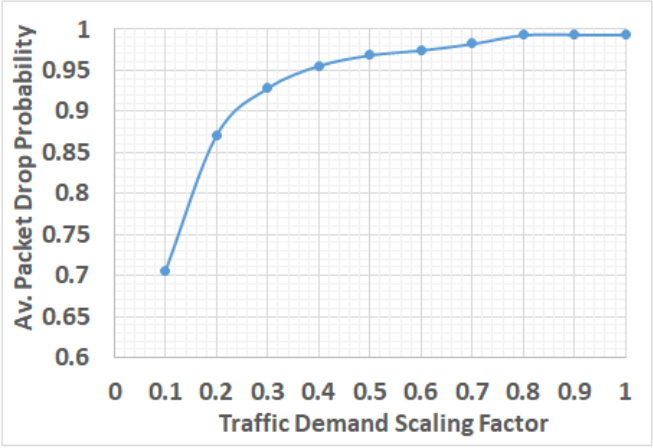
Figure 5.9: The vehicle density at the RUS locations shown by the darkness of the red color

The traffic congestion level impact on the packet delay

In a vehicular environment, due to the intermittent connectivity, the packet queuing delay does not include only the waiting for processing, but it also includes the waiting for connectivity, for example, a packet is generated while the vehicle is not connected to an RSU. This connectivity-waiting queuing delay is dependent on the average vehicle speed, which is inversely proportional



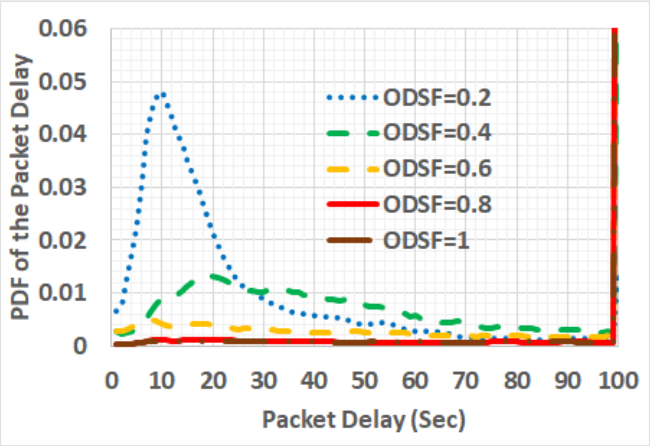
(a)



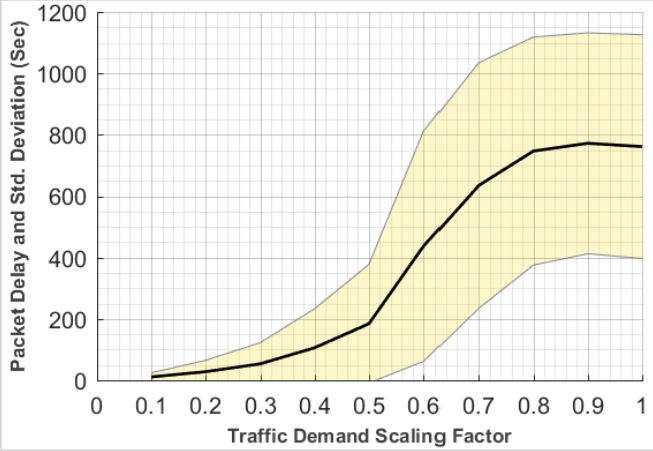
(b)

Figure 5.10: The packet drop probability: (a) the probability density function (pdf) of the packet drop for different traffic demand rates, (b) the average drop probability versus traffic demand rate

to the vehicle density. So, the higher the speed (or the lower the vehicle density) the shorter the time needed by the vehicle to reach a covered area, consequently, the shorter the packet queuing delay. Figure 5.11-a shows the delay distribution for different ODSF. It shows a shorter packet delay for the low ODSF values. Figure 5.11-b shows the average delay and the standard deviation area. It shows that at the high congestion levels, the mean packet delay is about 770 *seconds* and the standard deviation is about 360 *second*, which means that about 4% of the packets can be delayed for more than 1490*seconds*. This long delay can be reasoned to the high congestion levels and the low average speed of the vehicles in these scenarios.



(a)



(b)

Figure 5.11: The packet delay (Sec): (a) the probability density function (pdf) of the packet delay, (b) the average delay and the squandered deviation

5.5.5 System Scalability

Since we are interested in the large-scale modeling of VANET, it is important to evaluate the system scalability. To achieve this objective, we calculated the time required to simulate one second for different concurrent numbers of vehicles in the network, which is shown in Figure 5.12, which demonstrates that the simulation time is approximately linearly proportional to the number of vehicles in the network. Figure 5.12 also demonstrates that adding the communication modeling significantly increases the simulation time needed to simulate one second approximately by a factor of 2 when the number of vehicles in the network is 30000 vehicles. It is worth mentioning that

this simulation completely runs sequentially, there is no parallelization utilized. So, we believe that, by applying some parallel computation techniques, the simulation speed can be significantly decreased.

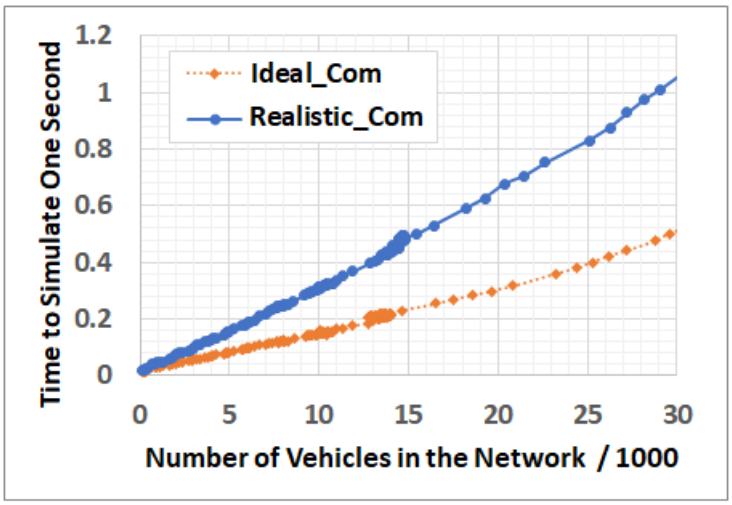


Figure 5.12: Simulation speed using Markov model.

To compare the simulation speed of the new model to that in VNetIntSIm, we calculated the time needed to simulate one second in VNetIntSim using the same way and compared it to the ideal communication case. Because the limited scalability of VNetIntSIm, we ran it on a small network with up to 1200 concurrent vehicle in the network as shown in Figure 5.13. The result shows that, at this small scale, using the discrete event simulation to model communication results in simulation speed that is about 32 times slower than that without modeling the communication (ideal communication case). Combining Figure 5.12 and Figure 5.13 together, we can see that the analytical model runs about 16 times faster than the VNetIntSIm at this small scale.

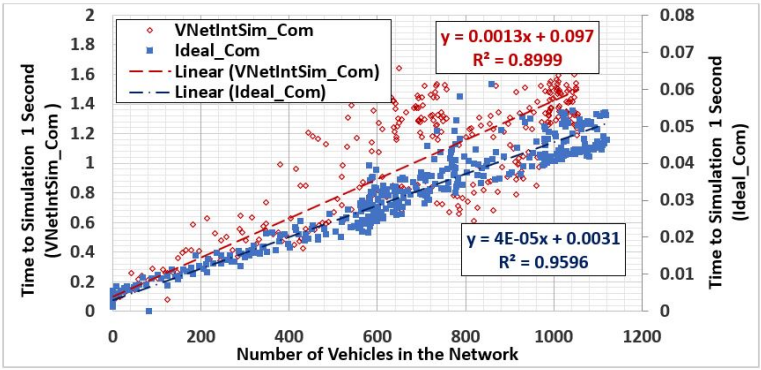


Figure 5.13: Simulation speed using discrete event simulation.

5.6 Conclusion

In this chapter, we developed a new scalable simulation and modeling framework for the vehicular networks. The developed framework integrates microscopic traffic modeling with a new VANET communication model to capture the mutual influence of communication and transportation systems. So, it can be used to model many ITS applications. The developed framework is used to quantify the impact of VANET communication performance on eco-routing ITS navigation system in downtown area of LA with a calibrated vehicular traffic demand. It shows that in the case of low vehicular traffic demand levels, the communication performance (in terms of packet drop and delay) does not have a significant impact on the eco-routing performance. However, as the vehicular traffic increases, this impact becomes significant. At a certain congestion level, it can result in routing gridlocks in the network due to incorrect routing decisions made by the TMC. Consequently, it is imperative to consider these mutual interactions of communication and transportation when deploying such systems, especially in the highly congested areas. The simulation results also show that the eco-routing system can work properly even at high packet drop rates that reach about 93% without significant effects. In other words, it can work properly if it receives only a small portion of the eco-routing updates correctly and in time. Consequently, it can be applied to low market penetration ratio of this eco-updating technology. In this Chapter, we assume V2I communication, so we did not consider the data routing in VANET which is an important component in VANET. Consequently, in the future, we plan to study the impact of routing on the performance of both systems. The proposed framework can also be utilized to develop novel routing protocols and study them on large-sale real-world scenarios.

Chapter 6

Large-Scale Agent-Based Multimodal Modeling of Transportation Networks

The performance of urban transportation systems can be improved if travelers make better-informed decisions using advanced modeling techniques. However, modeling city-level transportation systems is challenging not only because of the network scale but also because they encompass multiple transportation modes. This chapter introduces a novel simulation framework that efficiently supports large-scale agent-based multimodal transportation system modeling. Because INTEGRATION software is the main component in this system, we call it "INTEGRATION Ver. 3.0", or "INTGRAT3" for short. The INTGRAT3 framework utilizes both microscopic and mesoscopic modeling techniques to take advantage of the strengths of each modeling approach. In order to increase the model scalability, decrease the complexity, and achieve a reasonable simulation speed, the INTGRAT3 framework utilizes parallel simulation through two partitioning techniques: spatial partitioning by separating the network geographically and vertical partitioning by separating the network by transportation mode for modes that interact minimally. INTGRAT3 framework creates multimodal plans for each controlled trip and tracks the travelers trips on a second-by-second basis across the different modes. We instantiate this framework in a system model of Los Angeles (LA) supporting our study of the impact on transportation decisions over a 5 hour period of the morning commute (7am-12pm). The results show that by modifying travel choices of only 10% of the trips, significant reductions in traffic congestion are achievable with a significant reduction in total traveler delay.

6.1 Introduction

The performance of transportation systems is a critical factor that affects the human standard of life. The environmental impact of the transportation sector has major effects on human health [115]. Traffic congestion not only increases fuel consumption and emission levels but also wastes traveler times. Moreover, the congestion experienced by travelers increases the stress and affects individual social interactions [116]. As a result of all these economic, social, psychological and health impacts, the academic community has devoted significant research efforts to improving transportation system performance. While the majority of these studies use simulation [73, 117, 118], there are significant modeling challenges including scaling, calibrating, and validation issues that impact the accuracy of the results. In this chapter, we present a novel agent-based framework for modeling of large-scale transportation systems. The presented framework supports city-level networks with different modes of transportation (cars, buses, railways, walking, biking, and carpooling). The developed framework utilizes both microscopic and mesoscopic simulation to leverage their respective strengths of accuracy and scalability. The framework spatially partitions the network enabling distinct portions of the region to be micro-simulated in parallel and vertically partitions the network into layers representing loosely interacting modes. In this way, we can utilize the available processing resources either using single or multiple machines. The framework is capable of tracking individual travelers on a second-by-second basis from their origin to their destination across transportation modes.

To the best of our knowledge, INTGRAT3 is the first tool that supports an agent-based city-level transportation system, combining both microscopic with mesoscopic simulations, tracking individual travelers and vehicles on a second-by-second basis, and supporting multimodal mobility. We instantiate this framework into a system to study the impact of routing on travel time and fuel consumption in the Greater LA city from 7 am to 12 pm.

The chapter first introduces the related literature. Then, an overview of the system architecture, components, and the high-level operations will be presented. Subsequently, the last two sections demonstrate the case study on the Greater LA network along with the results.

6.2 INTGRAT3 Model versus Previous Models

In Chapter 2, we gave a literature review of existing modeling techniques for modeling large-scale transportation systems including TRANSIMS [60], MATSIM [74], and others.

Similar to TRANSIMS, our framework supports the parallel computation either on single multi-core or even multiple machines. However, the definition of the microscopic simulation in TRANSIMS is limited to the demand, where each trip is simulated individually as an agent. But, links and the mobility of vehicles on these links are mesoscopically modeled using a parallel queuing approach [119]. These queuing models are inaccurate in estimating the link travel time especially in congestion situations such as the LA morning commute. Furthermore, it cannot capture the acceleration/deceleration events of each vehicle that have a significant impact on the fuel consumption and emissions. In contrast to TRANSIMS, our framework uses continuous space model for the micro-simulation, which is the enabler to capture many of the mobility parameters.

Compared to the hybrid traffic modeler presented in [67], in INTGRAT3 system, we also utilize microscopic-mesoscopic hybrid modeling. However, in our model, we do not have that spatial separation between the microscopic and mesoscopic simulations. The two simulators are spatially overlapping but assigned different links. In INTGRAT3, links are assigned to the simulator based on their importance and their impact on the network.

The Scalable Electro-Mobility Simulation (SEMSim) system that was proposed in [73] uses a simple vehicle characteristics (e.g., kinematic model) and driving behavior models. In contrast to SEMSim, our framework is based on mature models that have been validated against observed transportation phenomena and supports travel across different transportation modes.

Compared to the MATSIM [74], which is considered the state of the art in simulating large-scale transportation system, the model we implemented is not only an agent-based simulation. In addition to that, it utilizes a hybrid simulation approach, it is also capable of microscopically simulating all the transportation aspects including demand, mobility, traffic signals, and road network aspects, as will be described in next section 6.3.

6.3 The INTGRAT3 Model

The INTGRAT3 model redefines the state-of-the-art of modeling and simulation of large-scale transportation systems by introducing a new framework that is capable of modeling large city level transportation systems. To achieve both the required accuracy and scalability, we utilize both the microscopic and mesoscopic modeling techniques.

The microscopic simulation defined in this chapter includes all the aspects of simulation that includes demand, mobility, and network. From the demand perspective, our framework can mi-

croscopically model each individual vehicle as an agent in the network that interacts with other vehicles as well as with the traffic signals and road control signs. It also provides dynamic demand modeling, that is, traffic demand changes throughout the simulation. From the mobility standpoint, the INTGRAT3 framework tracks every individual vehicle at a time resolution of decisecond (0.1 seconds). These features are gained basically from the microscopic nature of the INTEGRATION traffic simulator [48] utilized in the INTGRAT3 model. Based on this time resolution, it captures all the driving events by using validated models for car following, lane changing and gap acceptance. The model also can simulate different stochastic mobility phenomena such as stochasticity in speed calculation, route selection, and driver aggressiveness in acceleration/deceleration events. From the network standpoint, many network topological details such as link control methods (stop sign, yield sign, and traffic signals), lane striping, lane prohibition, and high occupancy vehicles (HOV) lanes are modeled in this framework.

To the best of our knowledge, none of the current traffic simulators support all these features for large-scale networks. The INTGRAT3 framework also incorporates other simulators for the modeling of the railway, pedestrian, and biking travel modes in addition to buses and carpooling. However, the details of these simulators are beyond the scope of this thesis.

An important advantage of the INTGRAT3 framework is its ability to track each trip on a second-by-second basis across different modes. Because of the computational cost required for the above-mentioned simulations, the implemented framework uses two partitioning techniques: vertical and spatial. Vertical partitioning combines mesoscopic and microscopic road vehicle simulation along with mode specific simulations for walking, biking, and trains. Spatial partitioning divides the microscopic network into smaller geographic regions. A simulation controller divides each trip into sub-trips to be simulated in different processes and monitors each sub-trip to ensure consistency. Before describing the model, the following subsection gives some definitions that will be used in the model.

6.3.1 Definitions

Global-network: The global road network includes all the road links in the area of interest. Each link is marked to be in the micro-network, the meso-network, or the train and pedestrian networks.

Meso-network: The meso-network is the connected subset of the links in the global-network that is simulated mesoscopically.

Micro-network: The micro-network is the connected subset of links in the global-network that is

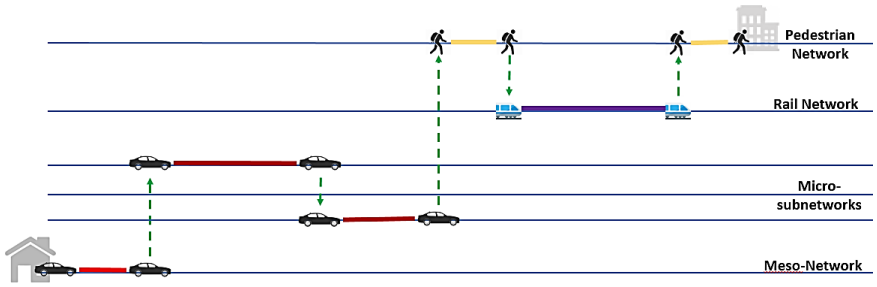


Figure 6.1: Multimode trip example.

simulated microscopically.

Subnetwork: A subnetwork is a spatial partition of the micro-network. Subnetworks are simulated microscopically using INTEGRATION software.

Zones: Nodes that can act as an origin or destination of the traffic. To have a fully connected network, each zone in the micro-network is mapped to a corresponding zone in the meso-network. However, some zones in the meso-network do not exist in the micro-networks.

Interconnection Zones (IZones): Interconnection zones are correspondences between zones in different networks. For example, for the micro-meso network connectivity, the zones exist in both micro and meso networks are IZones.

Trip: A trip is a traveler's planned path from origin zone to destination zone in the global-network. A single trip can go through multiple network layers (multimodal trips) and/or multiple subnetworks. For example, in the trip shown in Figure 6.1, a person can drive his/her car on the local road (in meso-network) from his/her home to the main road. Then, he/she continues driving on the main road in the micro-network (where he/she travels through two micro subnetworks). Then he/she parks his/her car and walks (on the pedestrian network layer) to the nearest railway station (rail network) from which he/she takes the train. Then he/she walks again to his/her work.

Trip local origin/destination: Local origin/destination is the origin or destination for the controlled trip in one of the simulator or subnetwork.

Trip origin/destination: These are the ultimate origin or destination for each controlled trip on the global network.

6.3.2 Network Partitioning

To increase the scalability, we employ two the network partitioning methods, spatial (or horizontal) and vertical partitioning.

Spatial Partitioning

Spatial partitioning divides the road network into subnetworks, such that each subnetwork is simulated in a separate process. All these processes are managed and connected using the simulation controller. Dividing the global network of interest into a set of subnetworks can significantly increase the simulation speed, because the simulation speed depends on the network size and the the number of vehicles in the network. Our analysis for the simulation speed of the INTEGRATION software shows that the time to simulate one second can be represented as a square function in the number of vehicles in the network as shown in Figure 6.2 and Figure Figure 6.3.

To get the results in these figures, we first ran the INTEGRATION software on a network with high traffic demand, and periodically we recorded the execution time and the number of vehicles in the network. Then we use the curve fitting to find the relation between the time and the vehicle count. The equation representing this relationship is shown in Figure in 6.2 with a fitting ration of 0.95. Using this relationship, we estimated the execution time for lager traffic demands as shown in Figure 6.3.

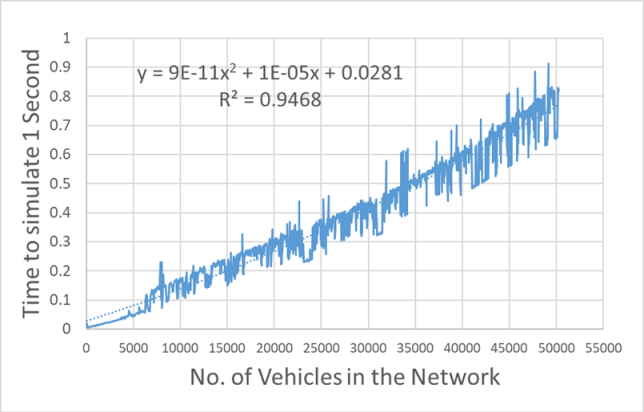


Figure 6.2: Simulation time versus number of vehicles in the network

Based on this relationship, in addition to the network information (size and traffic demand rates), we can decide how many subnetworks are needed to achieve the desired simulation speed.

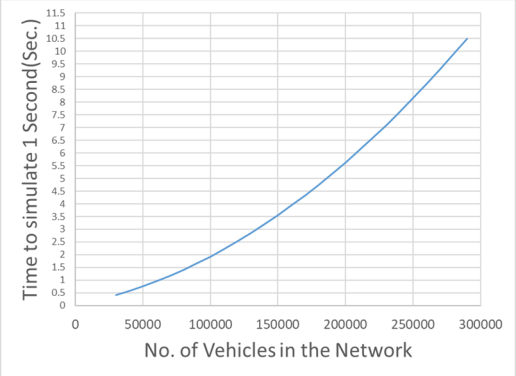


Figure 6.3: Expected simulation time versus number of vehicles in the network

Vertical Partitioning

In vertical partitioning, each network type (micro and meso road networks, trains network, biking network, and walking network) is represented by a different layer in the system. The connectivity between these layers is managed by the simulation controller. The multimodal trips across all these transportation modes are achieved by the trip planner.

The most important network layers are the micro-network and the meso-network. The reason is that all the vehicles are simulated in these two networks, while the other networks are working almost separately with very loss coupling and interaction between them.

Because both micro and meso networks represent portions of the road network, one important question is how to divide the road network into these two layers. The main objectives of combining micro and meso simulations together is a double folded reason: to achieve the highest possible accuracy while maintaining the system scalability.

To achieve these two goals (accuracy and scalability), we studied the different road link types and their impact on the network.

The main roads, the arterial roads, and highways are the most influential roadway segments of the city transportation network compared to the local links. These roads can significantly affect the network performance because of two reasons. Firstly, the high traffic demand they carry compared to the local links. Secondly, the traffic condition on the main roads, arterial roads and highways are characterized by its high variability from the free flow regime through the congested regime compared to the low traffic regime in the local roads as shown in Figure 6.4.

First, at moderate and high traffic demand levels, a vehicle's behavior can affect other vehicles

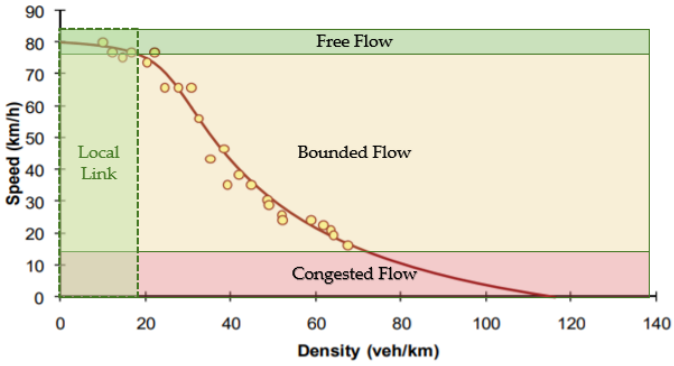


Figure 6.4: The speed-density fundamental diagram and the link types

because of the low distance headway. For example, if the leading vehicle decelerates, the following vehicle will have to decelerate also to maintain the safety conditions in order to avoid accidents. And when the headway increases, vehicles can accelerate towards the road free-flow speed. These acceleration/deceleration events have a noticeable impact on the network performance including fuel consumption, emission levels, and travel times. Using mesoscopic simulation for such links cannot capture all these events, which will reduce the system results fidelity. Thus, to capture all these effects, these road links (main roads, arterial roads, and highways) must be simulated microscopically.

On the other hand, the local roads that connect the main roads to residential areas are of low importance because of their low traffic flow rates. However, we cannot totally ignore these local roads, because they contribute to the travel time and fuel consumption of the vehicles. The low traffic demand on the local roads means that vehicles on these links will be moving, most probably, at the road speed limits with minimal variations as shown in Figure 6.4. So, by using the average speed of the vehicles on the local road links to estimate the fuel consumption, emission level and travel time, the accuracy will not be significantly reduced. Thus these links can be safely simulated mesoscopically without affecting the system accuracy.

Secondly, the temporal variation of the traffic flow rate on the main roads, arterial roads, and highways are expected to be very high. So, to accurately capture these temporal changes, the main roads, arterial roads, and highways must be modeled microscopically. While, for the local roads, it is very limited and assuming an average speed and average vehicle density is a valid assumption. Thus, the mesoscopic modeling is sufficient for the local road links.

Based on that, in the INTGRAT3 framework, the important links (main roads, arterial roads, and highways) are simulated microscopically which, gives the highest possible fidelity for this portion

of the network. Alternatively, the local roads are modeled mesoscopically to capture their impacts while reducing the modeling and computational requirements.

Consequently, the framework has two mandatory layers: the meso-network and the micro-network layers. In addition, the framework supports layers for other transportation modes such as railways and pedestrians. Figure 6.5 demonstrates the layering concept. A traveler uses more than one transportation mode means moving him/her from one network to another, consequently from one simulator to another. These interactions between different simulators are managed by a simulation controller (SC). Simulations notify the SC when a traveler finishes a sub-trip at an IZone. The SC finds the next sub-trip for this traveler and sends him/her to the appropriate simulator. The IZone must exist in the next network to guarantee the connectivity of the trip.

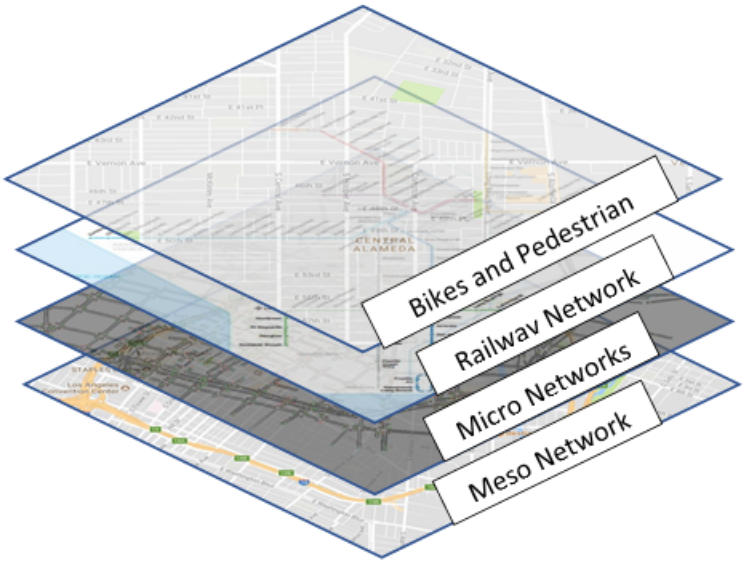


Figure 6.5: Partitioning the network into different layers.

Based on the scale of the area of interest, the micro-network can be divided into a set of subnetworks. Each subnetwork can run individually. In this way, we can utilize the available resources, and increase the simulation speed.

However, there is a trade-off between the simulation speed and the accuracy. The larger the subnetwork size, the slower the simulation speed, but the higher the accuracy of the simulation results. Within a subnetwork, the simulator models all the impacts between all connected road links (e.g., spilling the vehicles queue back from one link to its upstream link, and the interactions between vehicles in the intersection). Selecting the appropriate subnetwork sizes depends on the total network size, the traffic demand rates, the available computational resources, and the desired simulation

speed.

6.3.3 Traveler Types

In our framework, there are two types of travelers: background and controlled.

Background travelers: background travelers create the network traffic conditions in each network, such as congestion levels in the micro-network and meso-networks and vehicle loading on public transit vehicles. In the micro-network, INTEGRATION tracks every individual traveler (both background and controlled). In the other simulators, the travelers of the background traffic are not tracked individually, instead, they are used to estimate the network state (e.g., congestion levels and trainloads) in order to accurately calculate the travel time and fuel consumption.

Controlled travelers: Each controlled traveler represents a person traveling from an origin to a destination at a particular time. The planner creates a multimodal trip for each controlled traveler and submits it to the simulation controller, which in its turn ensures the traveler traverses the networks in the appropriate simulators. Each controlled trip is tracked on a second-by-second basis in all the transportation modes. Moreover, the controlled trip can be rerouted or replanned, while the person is traveling.

6.3.4 System Architecture and Components

Figure 6.6 shows the general architecture of the INTGRAT3 framework. A basic idea is separating the system software components from the hardware components. The communication layer is the enabler to transparently run this system on different infrastructures with minimal configuration changes. The communication layer utilizes the RabbitMQ implementation [120] of the Advanced Message Queuing Protocol (AMQP) [121].

The execution layer of the system consists of two plans: (1) the planning and simulation plan which is responsible for simulating trips and creating the multimodal routes for the controlled trips and (2) the control plan, which is responsible for controlling and managing the different system components. The framework has the components shown in Figure 6.7.

The input data repository contains all the required input data to be used by the system components. For example, the roadmaps for both the meso-network, micro-network and the subnetworks are stored in this database along with the OD inputs that represent the traffic demand, transit schedules,

energy models, and transit loading. The SC manages the different simulations. When started, each simulation module imports the corresponding input files from the input database then it initializes its environment and starts its internal synchronization procedure that communicates to the SC. Due to space limitation, we will give a brief overview of the operation for only the basic components including micro-simulator, meso-simulator, planner, and the SC focused on additions not reported in previous research.

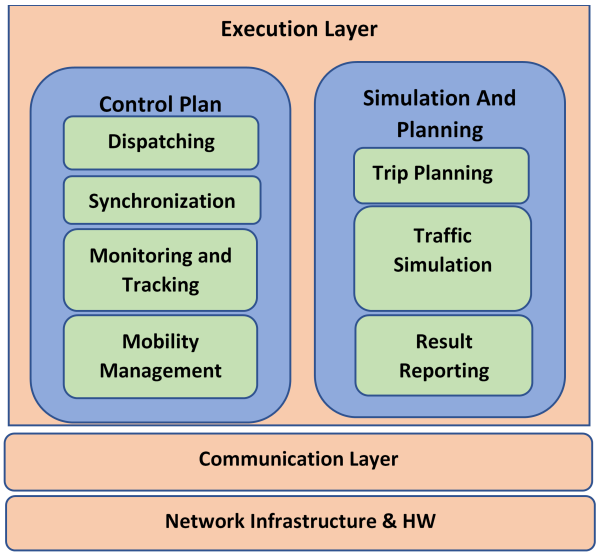


Figure 6.6: System architecture.

6.3.5 INTEGRATION and Micro-Models

Micro-simulation using INTEGRATION software is the focus of our framework. INTEGRATION is a discrete-time continuous-space trip-based microscopic traffic simulation and optimization model which is capable of modeling networks with thousands of cars. It is characterized by its accuracy that comes from its microscopic nature and its small-time granularity. INTEGRATION provides 10 traffic assignment/routing options with a full support of five vehicle classes, each class has its own parameters and routing trees.

INTEGRATION Car Following Model

INTEGRATION updates the vehicle speed and location every decisecond based on a user-specified steady-state speed-spacing relationship along with the speed differential between the subject ve-

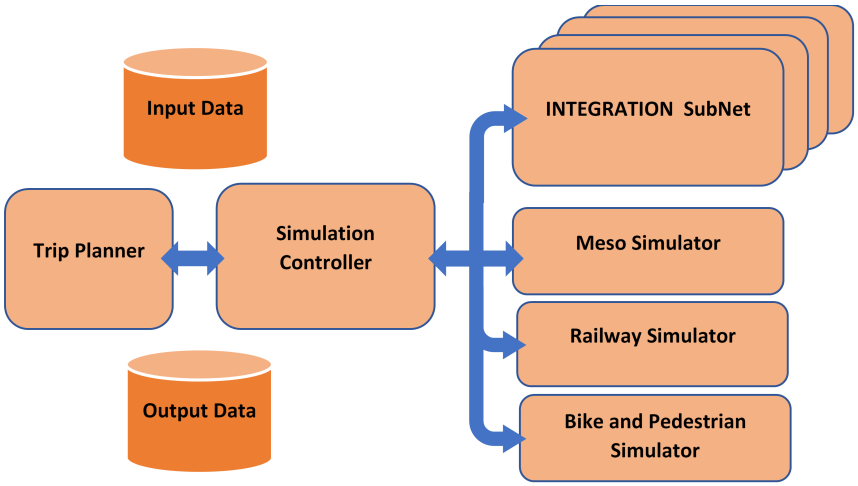


Figure 6.7: System components.

hicle and the heading vehicle. INTEGRATION uses the variable power vehicle dynamics model to estimate the vehicle's tractive force. Consequently, it implicitly accounts for gear-shifting on vehicle acceleration, which ensures a realistic estimation of the vehicle acceleration. More specifically, the model computes the vehicle's tractive effort, aerodynamic, rolling, and grade-resistance forces, as described in details in the literature [58, 59]. In INTEGRATION, the car-following model computes the speed $u_n(t+\Delta t)$ of the following vehicle (n) at the new time step $t+\Delta t$ as [48]:

$$u_n(t+\Delta t) = min\Bigg\{u_n(t)+a_n(t)\Delta t\ , \frac{-c_1+c_3u_f+\bar{S}_n(t+\Delta t)-\sqrt{A}}{2c_3}, \sqrt{u_{(n-1)}(t+\Delta t)^2+d_{max}(\bar{S}_n(t+\Delta t)-\frac{1}{k_j})}\Bigg\} \tag{6.1}$$

where

$$A = \big(c_1 - c_3\ u_f\ \bar{s}_n(t+\Delta t)\big)^2 - 4c_3\big(\bar{s}_n(t+\Delta t)u_f - c_1\ u_f - c_2\big) \tag{6.2}$$

and c_1, c_2, and c_3 are the model constants which are computed as:

$$c_1 = \frac{u_f}{k_j\ u_c^2}(2u_c - u_f) \tag{6.3}$$

$$c_2 = \frac{u_f}{k_j\ u_c^2}(u_f - u_c)^2 \tag{6.4}$$

$$c_3 = \frac{1}{q_c} - \frac{u_f}{k_j\, u_c^2} \tag{6.5}$$

and the vehicle $\bar{s}_n(t+\Delta t)$spacing is computed as:

$$s_n(t+\Delta t) = x_{(n-1)}(t) - x_n(t) + *u_{(n-1)}(t) - u_n(t)\Delta t + 0.5a_{(n-1)}(t+\Delta t)\Delta t^2 \tag{6.6}$$

Here $a_n(t)$ is the acceleration of the vehicle n; u_f is the free-flow speed of the roadway; u_c is the roadway speed-at-capacity; q_c is the roadway capacity; k_j is the roadway jam density; $x_n(t)$ and $x_{(n-1)}(t)$ are the positions of the subject vehicle the lead vehicle at time t; d_{max} is the maximum acceptable deceleration level (m/s^2).

Delay Computation

Within INTEGRATION, the delay D_n^l experienced by the vehicle n is computed for each traveled link l , as the difference between the vehicles simulated travel time and the free-flow speed travel time for this link [122]. And the total delay D_n experienced by the subject vehicles is computed as:

$$D_n = \sum_{(l \in \textit{the vehicle path})} D_n^l = \sum_{(l \in \textit{the vehicle path})} \int_{t_0^l}^{t_1^l} (u_f - \frac{u(t)}{u_f})dt \tag{6.7}$$

where t_0^l and t_1^l are the times at which the vehicle enters and exits the link l respectively.

Fuel Consumption and Emissions

Computing the fuel consumption and emission levels is important to capture the travel costs and environmental effects of transportation decisions. The INTEGRATION software is capable of computing the second-by-second fuel consumed, vehicle emissions of carbon dioxide (CO2), carbon monoxide (CO), hydrocarbons (HC), oxides of nitrogen (NOx), and particulate matter (PM). The micro-simulator uses the VT-Micro model [55] to calculate the second-by-second fuel consumption and emissions for each vehicle in the micro-network.

6.3.6 Meso-Simulator

The meso-simulator is implemented as a discrete event simulation [123]. The events represent the instant of reaching a network node through some link, at which moment a new event is generated for the next link and is added to the discrete event queue. The discrete event queue is an ascending sequence of events ordered by the time of their occurrence. The meso-simulator is given paths in the meso-network to be simulated together with the initial start time. The first event for each path then consists of the first node in the path and the start time, while all other events are generated as a consequence of the initial event. In the meso-simulator, each road link has its configuration parameters such as free-flow speed, speed-at-capacity, and jam density. In addition, each road link has state information, which includes the number of vehicles on this link and a queue that has these vehicles. This state information is updated by the events happening on the subject link, such as a vehicle enters the link or a vehicle exits the link. The average speed and travel time for each individual vehicle are calculated based on the current state of the link at the time the vehicle enters that link. The arrival of a vehicle to a given link triggers the meso-simulator to calculate its average speed and travel time, subsequently, to schedule another event at the time in which vehicle expected to exit that link. At exit time, the meso-simulator estimates the fuel consumption of the vehicle on this link and adds it up to the vehicles total fuel consumption.

6.3.7 Planner

Each controlled traveler has an origin, destination, and a travel window. The main task of the planner is the planning of these multimodal routes for the controlled trip. The planner also is responsible for updating or changing these routes whenever needed. During a window that begins 30 minutes before the earliest possible departure time for the controlled traveler, the planner starts planning the trip by using the up-to-date cost and timing information reported from each individual simulator. It also uses the connectivity information between the different subnetworks and/or layers in order to create the optimal route for the subject trip. The trip can be replanned or rerouted after the trip starts. For example, if the traveler can not catch the train at the scheduled time, or he/she can not board the scheduled bus because the bus is full, the responsible simulator notifies the SC which requests the planner to find an alternative route for the traveler.

In addition to the trip planning, routing is an important function in the system model. In the INTGRAT3 model, there are two routing levels: strategic routing and tactical routing

Strategic routing

Strategic routing is the high-level routing in the system. When the planner creates a trip, it should create is multimodal route based on the current network state. The objective function in the strategic routing level is to minimize the fuel consumption for each individual controlled trip considering different constraints including the travel time and distance in addition to the traveler preferences. It is easy to realize that using the fuel consumption as the only metric for the strategic routing can result in using the biking and walking only because both of them have zero energy consumption. Thus, the system uses other metrics as constraints when creating routes for the controlled trips. Among these metrics, the planner uses some user preferences such as the transportation mode preferences, the maximum walking distance, and the biking walking distance.

After creating the multimodal trip, the router finds the route for each mode in each network type, including the meso-network and the micro-network, the biking, and walking networks. Subsequently, this plan is sent to the SC to be executed in the network.

Tactical routing

The different simulators are responsible for simulating the trip in its different modes. The simulators for the micro-network and the meso-network have their own routing engines, which are responsible for routing the vehicles in each individual subnetwork based on its current state. Initially, when the micro-simulator receives a trip from the SC, it initializes the route for this trip based on the route it received from the SC. Then, periodically, it updates this route based on the latest routing information in this subnetwork. Updating the controlled trip route does not change its local destination zone in the current subnetwork.

In the tactical routing level, each simulator uses its local information about the road links in its own network. More specifically, it used the fuel consumption cost for the road links as the metric and applies Dijkstra's algorithm to find the shortest path for the subject trip from its current location to its local destination.

The fuel consumption cost for both micro-network links and miso-network links are periodically updated. In the micro-network, the INTEGRATION software uses the technique described earlier in Chapter 2. For the meso-network, the fuel cost for the road links is calculated based on the average speed of the vehicles on each road link. This speed can be computed using the fundamental diagram shown in Figure 6.4 by finding the vehicle density of on each road link. Because of the large size of the meso-network, its routing engine uses the A* algorithm [124, 125] instead of

Dijkstra's algorithm.

6.3.8 Simulation Controller

The SC is a core component of the model, which is responsible for:

- Initializing the simulation,
- Synchronizing the different simulators,
- Moving travelers between layers/subnetworks, and
- Tracking the individual controlled trips.

In the initialization process, the SC reads parameters such as the simulation duration and the locations of the input files for each simulation component. Then it reads in the network files, builds the required graphs for the networks, and checks for the appropriate connectivity among the different layers/subnetworks. It also builds a list of all the controlled trips. Then, it starts the different simulators (INTEGRATION, meso-simulator, bike and pedestrian simulator (BPSim), and railway simulator (RailSim)) and waits for all of them to initialize.

When a simulator starts and initializes its own environment, it must send the first synchronization request to the SC and wait for the simulation start messages from the SC. When all the simulators are ready, the SC allows them to start the simulation. During the simulation, all the simulators must be synchronized at pre-specified intervals. This period is defined as the maximum synchronization interval, which is a system-wide variable. Its default value is 1 second. After this time interval, the simulator cannot progress the simulation process until permitted by the SC. So, after finishing the maximum synchronization interval, each simulator sends a synchronization request to the SC and waits for a the response from the SC. When the SC identifies that all the simulators reached the same simulation time, it allows them to run the next interval.

During the simulation, the SC receives the state information about each controlled trip or subtrip from each simulator. Consequently, it can track every individual controlled trip in different networks/layers and is responsible for moving the traveler from one subnetwork/layer to another. By doing so, the SC establishes the connectivity between different subnetworks/layers. For example, when a driver finishes his/her subtrip on the meso-network (say, IZone1) and needs to be moved to the micro network, the meso simulator informs the SC to 1) update the trip information (travel

time, fuel consumption, and current location); 2) pull the trip information from its database and find the destination of the next sub-trip on the micro subnetwork (say, Z2); and 3) request the corresponding INTEGRATION instance to start a new sub-trip in its network from IZone1 to Z2 and passes the initial route for this sub-trip to INTEGRATION. In this case, INTEGRATION may defer the start time of this vehicle if the link to which the vehicle should enter is at jam density.

6.4 Case Study: LA Network

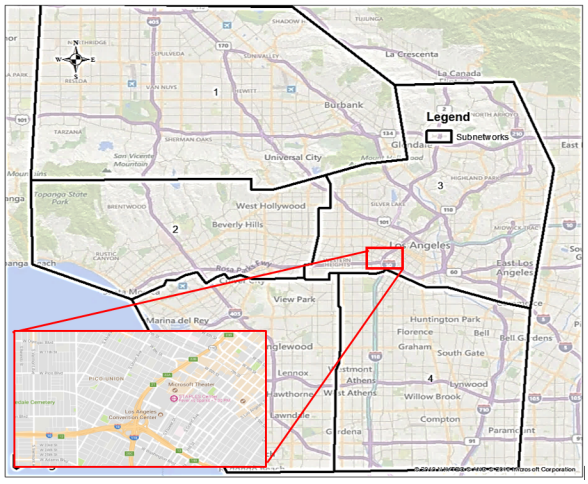
We use this system to model the overall city of LA in the peak hours. This section describes the network and the preliminary simulation results.

6.4.1 LA Networks

To build the micro-network and meso-network, we used three different data sources: (1) NavTeq is used for generating nodes and links, (2) OpenStreetMap is used for intersection traffic control information, and (3) Google Maps are used for validating road attributes including the number of lanes, one-way streets, speed limits, bus lane locations, etc. The global-network has 62,984 nodes and 181,840 links (Shown in Figure 6.8). The LA area is divided into five subnetworks shown in Figure 6.8. The walking and biking simulators use the meso-network as input. Our system model includes the largest operator of public rail and buses in LA, LA Metro. LA Metro bus service includes 170 lines, 15,967 bus stops, and 854,693 boardings/day. LA Metro rail service includes 6 passenger rail lines, 93 stations, and 359,861 boardings/day. Station level boarding data were provided by LA metro along with specifications of the vehicle fleet.

6.4.2 Traffic Calibration

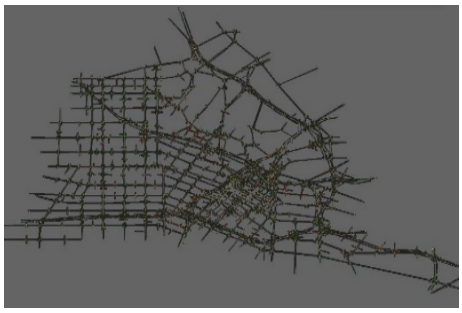
The traffic is created based on real data from Performance Measurement System (PEMS) database. The count and speed data from PEMS database are aggregated and the traffic demand between each Origin-Destination (OD) pair is estimated using the QueensOD [126] software, which utilizes the Maximum Likelihood Least Relative Error (LRE) approach. A portion of these trips is used as controlled travelers, while the remaining are modeled as the background. The background travelers are modeled in each network separately based on the calibrated traffic for each subnetwork as shown in Table 6.1. The vehicle count in Table 6.1 is the total traffic on each subnetwork that



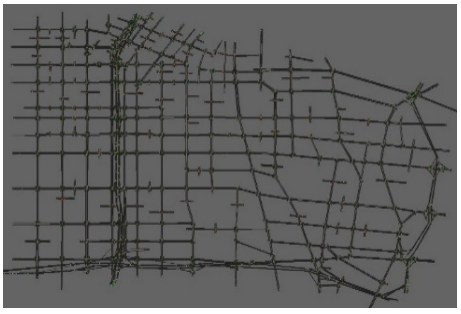
(a) LA network

(b) Subnetwork 1

(c) Subnetwork 2



(d) Subnetwork 3



(e) Subnetwork 4

(f) Subnetwork 5

Figure 6.8: LA total network and micro-subnetworks

includes both the controlled and the background traffic.

6.5 Simulation Results

We created the system model to enable assessing potential system-wide effects of individual transportation decisions across the LA region. We ran two scenarios for the LA area. In the base scenario, all the travelers make travel decisions by themselves. In the controlled scenario, the controlled travelers (10% of the driving population) are given directions regarding modes and routes by the planner. Our hypotheses are that the controlled case will result in an energy reduction and a reduced network congestion level. So, in the micro-network routing configuration, the subpopulation Feedback-based eco-routing (SPF-ECO) [72] is used for the controlled traffic, while the Time-Dependent Feedback Assignment (SFA) [48] is used for the background traffic. In the base scenario, since it does not have controlled traffic, only the SFA traffic assignment was run. All the other simulators use the energy as the routing metric. Furthermore, we expect a controlled travel mode distribution to be dominated by driving in the micro and meso-networks. The global network traffic calibration showed that there were approximately 1.3 million vehicle trips in Greater LA area. When calibrating these ODs for the micro-networks, it generated 2.25 million trips as shown in Table 1. The reason for this large difference is that a portion of the global trips passes through multiple subnetworks, being divided into multiple subtrips across the subnetworks. Table 6.2 shows the system-wide comparison for the traveled distance, we can notice that the traveled distance decreases for the micro-network, while it increases for other modes as the 10% of controlled trips are planned over multiple transportation modalities. Combining the results in Tables

Table 6.1: Subnetwork sizes.

Sub-net	Nodes	Links	Signals	Vehicles
1	743	1691	256	404,191
2	940	2251	361	447,948
3	1625	3561	459	592,343
4	741	1724	237	445,857
5	647	1507	203	362,415
Sum	4696	10734	1516	2,252,754

6.2 and 6.3 together, we can notice that in the micro-network the vehicle's average traveled distance is about 8 km and the vehicle average travel time is about 39 minutes in the base case. In the controlled case, the vehicle's average traveled distance remains approximately the same while the vehicle average travel time is reduced to approximately 23 minutes demonstrating that by controlling 10% of the traffic, the vehicles moving on the main roads and highways (micro-network) achieved a 40% saving in the total travel time. Table 3 also shows that those vehicles achieved about an 18% saving in the fuel consumption, and their average delay is reduced by about 46%. Shifting the controlled trips to the public transit system is energy efficient because the increases in the energy consumption by the buses and trains due to the extra passenger loads is less than the savings accrued as a result of a reduction in the traffic congestion. However, some of these savings come at the cost of an increment of energy consumption in other transportation modes. Specifically, the energy consumed in both the meso-network and the public transit increased by 110% from 270028 to 568737 KW-hr. We have to mention that the system-wide comparison of the fuel/energy is not possible in the current version, because some of energy/fuel consumption models have not been implemented in the various submodels.

6.6 Conclusion

The chapter proposes and describes a novel multimodal large-scale agent-based transportation network modeling system that has a wide spectrum of application. The INTGRAT3 system is capable of modeling large urban cities including different transportation modes of travel (driving, biking, walking, riding a bus, riding a train, and carpooling). This system is tested by modeling the Greater LA Area during the morning peak period. The preliminary results show that the network is cur-

Table 6.2: System wide traveled distance comparison

Transportation Mode	Traveled Distance (Km)	
	Base	Controlled
Walking	0	1,418.6
Cycling	0	147,246.8
Riding Bus	0	3,886.4
Riding Train	0	9,230.5
Driving on Micro	18,298,072.8	17,760,530.8
Driving on Mesok	616,105.2	791,424.0
Carpooling	0	356,549.1
Total	18,914,178.0	19,070,286.2

Table 6.3: Micro-network results for base and controlled scenarios

Subnet Number	Number of vehicle Trips		Total Travel Time (*s*)		Total Delay (*s*)		Fuel Consumed (*Liter*)	
	Base	Controlled	Base	Controlled	Base	Controlled	Base	Controlled
1	404,191	399,298	1,425,241,984	737,314,176	725,780,800	223,516,944	839,149	580,246
2	447,948	399,301	716,592,000	370,059,488	374,176,480	201,135,536	580,276	448,206
3	592,343	539,949	1,466,868,736	907,472,448	702,194,240	437,199,680	901,559	714,016
4	445,857	402,510	1,126,294,656	602,945,728	544,548,864	318,766,464	704,458	544,593
5	362,415	337,392	588,823,424	306,889,216	295,441,600	126,577,376	499,629	383,526
Sum	2,252,754	2,078,450	5,323,820,800	2,924,681,056	2,642,141,984	1,307,196,000	3,525,071	2,670,588
Subnetwork Average/veh			2363.25	1407.15	1172.85	628.93	1.56	1.28

rently very congested, with an average speed of approximately 12.3 km/hr. The results also show that by replanning 10% of the trips, the performance of the network can be significantly improved. An important future work is to improve the system to achieve faster simulation speed. Currently, the average simulation speed is approximately half real-time, i.e., every virtual second is simulated in 2 actual seconds. We also plan to improve the mesoscopic traffic simulator to achieve better estimates of energy consumption, delay, and travel time. It is also important to study the complexity of the system by quantifying its simulation speed and memory usage for different demand levels. An advantage of a detailed system model like the one developed in this chapter is that it enables modeling mode changes under different scenarios including traffic accidents, construction, and special events. We intend to explore potential savings that could result from informed decision-making by groups of travelers in these scenarios.

Chapter 7

A Novel Stochastic Linear Programming Feedback Eco-routing Traffic Assignment System

In previous chapter, we showed that eco-routing can produce fuel saving, even in congested network. It is interesting also that this fuel saving is also combined with travel time saving. Because of these benefits, eco-routing has attracted scholars' attention, where many techniques have been proposed to develop eco-routing. However, a common theme among all these proposals is the use of the user equilibrium techniques that typically attempt to minimize the fuel consumption for each individual vehicle separately by routing it through the most environment friendly route, the best route. This user equilibrium can result in overloading these best routes, producing high congestion and consequently higher fuel consumption and longer delays.

In this chapter, we developed a novel system equilibrium eco-routing technique (LPS-ECO) that utilizes fuel consumption cost feedback from vehicles and employs linear programming to assign traffic flows to road links. Subsequently, the system stochastically assigns routes to vehicles. The main idea behind the proposed technique is that instead of routing all vehicles through the same best routes, the traffic load for each individual flow will be divided among the available network resources toward the flow destination. In this way, the system can avoid potential congestion on the best route by routing some vehicles through non-optimal routes while maintaining the system wide fuel consumption minimized. The proposed system is compared to the shortest-path-based eco-routing that was described in Section 2.4. The comparison shows that for low traffic demands there is no significant difference between the two algorithms. However, as traffic demand increases,

LPS-ECO produces fuel consumption savings that reach 38% for the small test network we used in this chapter. LPS-ECO also produces savings in travel time in most cases. However, these savings incur a high computational cost due to the large size of the linear program, especially for large networks. Consequently, future research would entail developing heuristics to reduce the computational load.

7.1 Introduction

In the last decade, researche awareness of global climate changes and environmental problems has inspired them to find a solution for these challenges. The transportation sector is a major factor in global warming and environmental pollution. In addition, the huge amount of fuel consumed by the transportation sector has resulted in energy and fuel shortages, as well as increased travel costs in terms of both time and money.

To enhance the transportation system and mitigate its negative impacts on human life, ITSs integrates people, roads, and vehicles together and utilizes electronic, computer, communication, and information technologies. Building on these technologies, data analysis, and processing, ITS can build large, real-time, efficient transportation management centers (TMCs) to better manage the transportation system resources.

From the human perspective, drivers usually use their knowledge and experience of road network conditions to minimize their costs, such as travel time or travel distance by selecting the lowest-cost route or, more scientifically, the shortest path. However, minimizing the travel time or the distance does not necessarily result in minimizing or reducing fuel consumption or emission levels [46, 47].

On the other hand, researchers have developed many strategies to minimize the fuel consumption in transportation systems [72]. One of them is the feedback-based eco-routing navigation system [17], which is a promising tool that can reduce the network-wide fuel consumption.

However, a common technique used in previous researches is the user equilibrium, which uses the best or shortest path routing strategy. The user equilibrium model of traffic assignment is based on the fact that humans choose a route so as to minimize his/her travel cost. It has been supported also by the fact that there is no cooperation between the drivers on the road due to a lack of communication, thus information about the route costs. So, drivers use only their knowledge about the route costs.

The development of new communication technologies, especially vehicular communication, helps overcome this barrier. Through these communication networks, vehicles can report the surrounding condition and the road link cost to other vehicles as well as to the TMC in real-time. Consequently, better routing strategies can be adopted to optimize the performance for the system not for each individual vehicle. Building upon these new communication technologies, developing a system optimum routing models has become achievable.

However, in such system equilibrium navigation techniques, some drivers have to sacrifice for the sake of the network wide performance. This means that some drivers will have to take higher cost routes in order to avoid increasing the congestion levels on the lower cost routes, which means that these systems assume that drivers will follow the routing decisions made by the system even though these routes are of higher cost.

Based on these assumptions, in this chapter, we propose a new system optimum eco-routing navigation technique that utilizes linear programming and stochastic route assignment techniques to route the vehicles. The proposed navigation technique is a feedback-based eco-routing model that assumes the vehicles has two capabilities. First, it assumes that vehicles are equipped with road maps and are capable of quantifying the fuel consumption on each road link. It also utilizes the utilize the communication capabilities of the vehicles to report these costs to the TMC and receive the routing recommendations.

In the remainder of this chapter, we discuss the eco-routing navigation system and its main components, describe the proposed system, and present results and conclusions.

7.2 Eco-routing System Components

The eco-navigation system has two main components. The first calculates the route cost in terms of fuel consumption or emissions. The second component decides how to optimize the route selection to minimize this cost.

7.2.1 Estimating Route Cost

A common technique for estimating the route fuel consumption is to develop a mathematical model that can estimate the fuel cost for each link in the route. These models use some link parameters, such as vehicle density, vehicle speeds, and vehicle accelerations, on the links constituting the

route. These parameters can be calculated from historical real data such as vehicle trajectories, road grades, and traffic conditions. This technique is utilized in Minett, et al., Boriboonsomsin et al., and Nie and Li [49, 53, 54], where the authors used historical real data to model the fuel consumption cost for the road links.

Vehicles fuel consumption and the emission levels depend on many factors, such as route characteristics (e.g., length, speed, grade, and traffic congestion) and vehicle characteristics (e.g., weight, shape, and power) in addition to driving behavior. Moreover, the traffic control mechanisms used in the road network (signals, yield signs, and stop signs) and the traffic signal timing significantly impact the fuel consumption and emissions. It has been proven to be too difficult to combine all these parameters in one model, especially because many of these parameters are stochastic and there is a complex dependency among all of them. Consequently, each model makes some assumptions that lead to oversimplification of the system, and the estimated costs from the models differ significantly from real measurements [53]. In other words, these models lack accuracy due to the simplifications embedded in their assumptions.

Another technique that attempts to accurately calculate the link cost is to use real-time feedback from vehicles about the cost they experience on each road link. In this feedback system, eco-routing depends on the vehicles ability to quantify the fuel consumption cost for each road segment it traverses. It must also utilize the vehicles communication capabilities to report this information to TMC. When receiving these data, the TMC updates the routing information, rebuilds the routes, and sends the new routes to vehicles traversing the network. This technique is used by Rakha et al. and Ahn and Rakha [48, 72]. The main challenge facing this technique is the market penetration of these enabling technologies, which is still not sufficient to fully run these systems in real time. However, most newly produced vehicles have such communication and Global Positioning System (GPS) technologies embedded.

In this chapter, the proposed eco-routing technique uses feedback from vehicles to estimate the fuel consumption cost of the road links. Then, it plugs these costs into a linear program along with the road network parameters such as the route maximum capacity. The linear program also uses the current network conditions such as the current traffic flow rate on each road link. The objective of the linear program is to find the best traffic assignment for each road link as will be described later in details in Section 7.3.

Compared to standard minimum cost network flow techniques, the LPS-ECO is not just a minimum cost flow model, it also includes the vehicle routing problem. The minimum cost network flow techniques consider the source and destination separately, the only condition they account for is

the network total balance, which means that the total traffic enters the network equals to the total traffic exits the network, regardless of the traffic demands between individual origin-destination pairs. In other word, these techniques do not consider the origin-destination (OD) traffic demands. However, in real transportation system, the traffic is defined by an OD matrix, which includes the traffic rate between each OD pair in the network.

LPS-ECO assumes multiple vehicular traffic flows from different sources to different destinations, each flow has its OD rate. It accounts for each individual OD traffic demand in the transportation network and guarantees that each vehicle in each traffic demand starts at its origin and ends at its destination through a valid route. Consequently, it includes vehicle routing problem impeded in it. It also considers the traffic flow balance for each individual demand at each node in the road network to guarantee the route continuity for each individual vehicle, as descried earlier.

7.2.2 Route Selection

The second component of the eco-routing system is the optimization technique used to select the routes and assign them to vehicles. To the best of our knowledge, all eco-routing techniques in the literature use shortest path algorithm to select the route between the source and the destination, such as [17, 48, 49, 53, 54, 72, 91].

Using the shortest path can work very efficiently in the case of low traffic demand. However, for moderate and high traffic demands levels, using the shortest path algorithm may lead to severe congestion, because it will route all the vehicles through the same best route [50]. Consequently, in the case of high traffic demands, using the shortest path can increase the travel cost in terms of both travel time, fuel consumption, and emissions. To compensate for this impact, the eco-routing navigation systems uses a rerouting technique, where vehicle routes are recalculated periodically based on the latest road network information.

In this chapter, we address this point by employing a new route assignment method. The main idea behind the proposed technique is to avoid overloading the shortest routes, and thus avoiding congestion on those routes, by sharing the traffic demand among a set of low-cost routes in such a way that minimizes the network-wide fuel consumption and emission levels.

In the proposed eco-routing system, instead of routing all the vehicles along the same best routes, and then rerouting traffic on other routes once the cost function increases, the traffic load for every individual traffic flow will be distributed among the available routes toward the flow destination. In this way, the system can avoid potential congestion by routing some vehicles to higher-cost routes.

We build the linear programming problem to calculate the link-flow assignment in a way that minimizes network-wide fuel consumption. The linear program also avoids overloading the road links by considering the link capacities in its constraints. Subsequently, based on the calculated link-flow assignment, vehicles are assigned routes stochastically.

7.3 Linear Programming Stochastic Routing

This section introduces the proposed linear programming stochastic-based eco-routing technique (LPS-ECO) that tries to minimize the network-wide fuel consumption by using all the available network resources (roads) toward the destination.

The linear program should be formulated in a way that minimizes the fuel consumption and at the same time guarantees route continuity for each individual traffic flow from its source to its destination. The route continuity condition is achieved by enforcing the individual flow balance at each node. The flow balance at a given intermediate node means that the summation of the traffic entering that node equals to the summation of the traffic exiting it. The combination of the objective function and the individual flow balance at each node guarantees the route continuity.

Since we use individual flow balance, we define the link-flow assignment as the portion of each individual flow that should go through each link. Consequently, if the network has f flows and m links, the linear program calculates the f portion for each of the m links. Consequently, the total number of variables in the program will be mf. The number of variables is relatively large for large networks with a large number of traffic flows. The other way to guarantee the traffic balance at each node is to use the total flow balance instead of the individual flow balance. The total flow balance can reduce the number of variables in the linear program. However, it cannot guarantee route continuity. Therefore, we have to use the individual flow balance method, not the total flow balance.

Based on the calculated flow distribution, the vehicles are assigned routes stochastically considering their source and destination. The route assignment algorithm will be described later.

7.3.1 The Objective Function: Minimizing Total Cost

We formulated the linear program objective function as follows: Given the network directed graph $G(N,L)$, where $N = \{i : i = 1,2,...,n\}$ is a set of n nodes and L is a set of m directed links. The

network has a set F of f concurrent flows (origin-destination traffic demands), each of rate q^k (in vehicles per hour $[veh/h]$) where k is the flow identification number ($k = 1, 2, ..., f$). The flow is an origin-destination traffic demand rate originated at a given origin and designated to a given destination. We assume that each traffic flow can be routed through multiple routes. To construct these routes we need to compute the portion of this traffic flow, $q^k_{i,j}$, that traverses the road link $l_{i,j}$, we call $q^k_{i,j}$ a sub-flow. The total flow rate passing through the directed link $l_{i,j} \in L$ from node i to node j is $q_{i,j}$. For each network flow k whose rate is q^k, every directed $l_{i,j} \in L$ is assigned a portion (or a sub-flow) of this flow rate equals to $q^k_{i,j}$ such that $0 \leqslant q^k_{i,j} \leqslant q_{i,j}$ and $0 \leqslant q^k_{i,j} \leqslant q^k$.

The total flow rate $q_{i,j}$ on link $l_{i,j}$ is the summation of all the sub-flows passing from node i to node j as in Equation 7.1:

$$q_{i,j} = \sum_{k=1}^{f} q^k_{i,j}. \tag{7.1}$$

Assuming that the fuel consumption cost for a vehicle that passes link $l_{i,j}$ is $C_{i,j}$, then the cost of all vehicles passing this link in the unit time can be calculated as shown in equation 7.2:

$$C^{Total}_{i,j} = C_{i,j} \sum_{k=1}^{f} q^k_{i,j}. \tag{7.2}$$

The objective of the linear program is to minimize the total network cost during the unit time, which can be mathematically written as:

$$minimize \sum_{i=1}^{n} \sum_{j=1}^{n} C_{i,j} \sum_{k=1}^{f} q^k_{i,j}. \tag{7.3}$$

7.3.2 Constraints

The program constraints are built to satisfy two conditions; the first is the route continuity that we mentioned above, and the second is that the link capacity constraint, that is the total flow on each link should not exceed its capacity.

The Individual Flow balance at each Node and Route Continuity Constraints

The individual flow balance is formulated as follows. For each intermediate node i, and for each individual flow k, the summation of the sub-flows of the k^{th} flow entering this i^{th} node must be equal to the summation of the sub-flow of the k^{th} flow exiting this node, as shown in Equation 7.4:

$$\sum_{d=1}^{n} q_{i,d}^{k} - \sum_{s=1}^{n} q_{s,i}^{k} = 0. \tag{7.4}$$

Equation 7.4 applies to an intermediate node. If the node i is a source or a destination node, then Equation 7.4 should include the flow rate q^k. For the source node that generates the k^{th} flow with rate q^k, we assume there is a fictitious source sending q^k to it, and vice versa for the destination nodes. This type of constraint will build nf constraints as shown in Equation 7.5.

$$\sum_{d=1}^{n} q_{i,d}^{k} - \sum_{s=1}^{n} q_{s,i}^{k} = \begin{cases} q^k & \text{if } \ i \ \text{ is the source of flow } k; \\ -q^k & \text{if } i \text{ is the destination of flow } k \quad \forall\, i \in N;\ k \in F; \\ 0 & \text{if } i \ \text{ is an intermediate node.} \end{cases} \tag{7.5}$$

The Link Capacity Constraints

For each directed link $l_{i,j}$ with capacity $\zeta_{i,j}$ (veh/h), the total flow rate traversing it should not exceed the link capacity. The link capacity constraints will build m constraints, one for each link, as shown in Equation 7.6:

$$\sum_{k=1}^{f} q_{i,j}^{k} \leqslant \zeta_{i,j} \qquad \forall \quad l_{i,j} \quad \in L. \tag{7.6}$$

The total number of variables in the linear program is mf, and the total number of constraints is $nf + m$. The final linear program to solve the problem is:

$$\text{minimize} \qquad \sum_{i=1}^{n}\sum_{j=1}^{n} C_{i,j}\sum_{k=1}^{f} q_{i,j}^{k}$$

$$\text{subject to :} \qquad \sum_{d=1}^{n} q_{i,d}^{k} - \sum_{s=1}^{n} q_{s,i}^{k} = \begin{cases} q^k & \text{if } \ i \ \text{ is the source of flow } k \\ -q^k & \text{if } i \text{ is the destination of flow } k \\ 0 & \text{if } i \ \text{ is an intermediate node} \end{cases} \quad \forall\, i \in N;\ k \in F,$$

$$\sum_{k=1}^{f} q_{i,j}^{k} \leqslant \zeta_{i,j} \qquad \forall \quad l_{i,j} \quad \in L,$$

$$q_{i,j}^{k} \geq 0$$

Calculating Link Capacities

The link capacities are a major factor in the proposed algorithm. Using these capacities in the linear program constraint enables the system to minimize the network congestion, thus improving network-wide performance. However, calculating the exact link capacity is a challenging task because of the different link control types and the dynamic conditions of the opposing links. For instance, estimating the capacity of yield-sign-controlled links is very challenging. The reason for this is that the actual capacity of such a link does not depend only on the configured link capacity and the number of lanes, but also on the number of opposing links, the traffic rate, and distribution of the traffic rate on each of those opposing links, which are stochastic factors.

In this chapter, we simplify this problem by using the formula in Equation 7.7 for the link capacities:

$$\zeta_{i,j} = R_{i,j} \times s_{i,j} \ \times \ \delta - \widehat{q_{i,j}}. \tag{7.7}$$

where $s_{i,j}$ is the lane base saturation flow rate for link $l_{i,j}$, $R_{i,j}$ is the number of lanes on the link, $\widehat{q_{i,j}}$ is the current flow rate traversing it, and δ is a capacity reduction parameter that depends on the link control method. We use typical values for δ as follows:

- For free links (no traffic controls), $\delta = 1$.
- For signalized links (controlled by a traffic signal), $\delta = \frac{T_g}{T_c}$, which is the ratio between the signal green time T_g to the total cycle length T_c of the signal.

- For yield sign links, $\delta = 0.4$.
- For stop sign links, $\delta = 0.3$.

In this way, the linear program can consider the different link parameters that can affect the actual link capacity.

Vehicle Route Construction

After solving the linear program and finding the link-flow assignment, the system can build the vehicle routes. When a given vehicle enters the network, the algorithm constructs its route from the source to the destination stochastically as follows.

The system first finds the flow of this given vehicle, k. Starting with the vehicles source node, node i, and using it as the current node, it finds the links out of this node, the link set $\check{L} = \{l_{i,j} : j \in N\}$. If a link $l_{i,j} \in L$ was assigned a sub-flow of the flow k equal to $q^k_{i,j}$, then this link $l_{i,j}$ can be added to the route of the vehicle with the probability $p_j$which can be calculated as:

$$p_j = \frac{q^k_{i,j}}{\sum_{l_{i,d} \in \check{L}} q_{i,d}} \tag{7.8}$$

where $q^k_{i,j}$ is the sub-flow of the k^{th} flow assigned to the link $l_{i,j}$.

After adding the first link to the route, the system finds the end node of this link and then uses it as the current node. This process is repeated until reaching the vehicles destination node. In this way, vehicles of the same flow (traveling from the same source to the same destination) can take different routes while keeping the network-wide cost minimized.

Updating Routing Information

Every dynamic routing technique updates the routing information periodically based on a predetermined time interval called the route updating interval. Changing the value for this updating interval may significantly influence the network output. The LPS-ECO updates the link-flow assignment every updating interval by recalculating the link capacities and flow rates, and subsequently rebuilds and resolves the linear program and updates the link-flow assignment. In this chapter, we use an updating interval of 60 seconds.

7.4 Simulation and Results

To evaluate the proposed LPS-ECO algorithm, we developed it in INTEGRATION software. and using the developed model, We compared the performance of the proposed LPS-ECO algorithm to that of the shortest-path-based eco-routing used in Subpopulation Feedback Assignment ECO-routing (SFA-ECO) [48], which is already included in INTEGRATION. Using SFA-ECO, the routing engine in INTEGRATION divides the traffic demand in each traffic class into five sub-populations, each one has its own routing tree. The routing tree for each subpopulation is updated after the updating interval. To avoid routing all the vehicles using the same routing information, SFA-ECO divides the updating interval into five equal sub-intervals, each sub-intervals is used to update one of the subpopulation.

Both routing techniques use the VT-Micro model to calculate the link costs. The main differences between them are the optimization method and the route building techniques. SFA-ECO utilizes the shortest-path algorithm to minimize the fuel cost for each individual vehicle, while LPS-ECO uses linear programming as described earlier. Secondly, every updating interval (60 seconds in this example), SFA-ECO uses the deterministic route for each source-destination pair, which is the lowest-cost route, so all vehicles belonging to the same flow in the same subpopulation should follow the same route (if the link costs do not change). This contrasts with LPS-ECO, which uses the stochastic route building technique in which vehicles of the same flow can take different routes.

7.4.1 Simulation Network and Traffic Demands

The network shown in Figure 7.1 was used for comparing LPS-ECO and the SFA-ECO. The network consists of 10 zones, whose numbers are shown in Figure 7.1. It has a main highway (center horizontal road) between Zone 10 and Zone 5, and two arterial roads (side roads). The highway has three lanes in each direction; the lower side road has two lanes while the upper side road has only one lane. The network size is 3.5 km 1.5 km. The free-flow speeds are 110 and 60 km/h for the highway and arterial roads, respectively. The network has 32 nodes and 68 links.

Regarding the origin-destination traffic demands (O-D demands), we used six different scenarios, as shown in Table 7.1. In each scenario, the traffic demand has two types of traffic flows; namely, main and secondary traffic flows. The main traffic demand is highway traffic, which produces two flows between Node 5 and Node 10, one in each direction. The secondary traffic flows are between the node pairs $\{1,6\},\{2,7\},\{3,8\}$ *and* $\{4,9\}$ in each direction, which results in eight traffic flows. The total demand is 10 traffic flows. In the first scenario (scenario No. 1), the main flow rate is

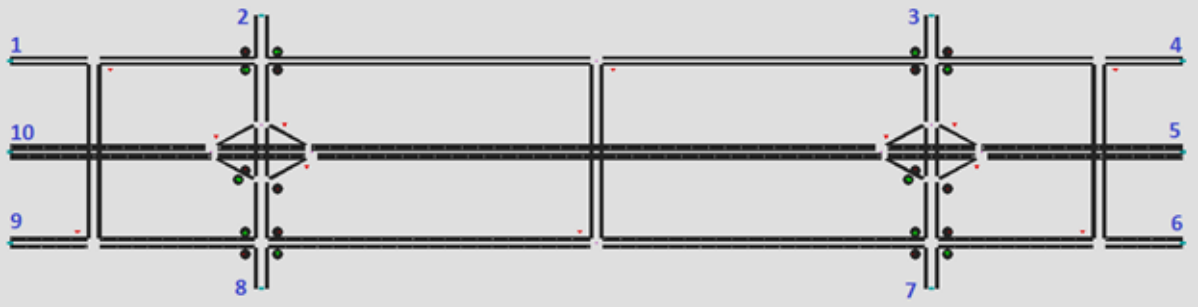


Figure 7.1: Simulation road network

750 vph in each direction, and the secondary flow rate is 125 vph, and each of the 10 flows runs for 1,800 seconds, resulting in a total demand of 1,250 vehicles. The flow rates for the other five simulation scenarios were generated by scaling up the first scenario by a scaling factor as shown in Table 7.1.

Table 7.1: O-D Traffic Demand Configuration

No.	Scale Factor	Main Demand (vehh)	Secondary Demand (vehh)	Total No. of Vehs
1	1	750	125	1250
2	2	1500	250	2500
3	4	3000	500	5000
4	6	4500	750	7500
5	8	6000	1000	10000
6	10	7500	1250	12500

In the simulation, the updating interval for both algorithms was set to 60 seconds. To have statistically accurate results, we ran each simulation scenario 20 times with different seeds. The results shown here are the averages of the 20 runs. In order to make sure that all the vehicles are cleared, and thus the result among the two algorithms are comparable for each scenario, the network is configured to run for 3600 seconds (1 hour), which gives it enough time to clear all the vehicles.

7.4.2 Simulation Results

Figure 7.2 shows the fuel consumption for both SFA-ECO and LPS-ECO in addition to the fuel savings achieved by the LPS-ECO relative to the SFA-ECO for the six traffic scenarios shown in Table 1. The figure shows that for low traffic demand (scaling factors 1, 2) the differences between the two algorithms are very small. As the traffic demand increases, LPS-ECO shows a significant

reduction in fuel consumption compared with SFA-ECO. LPS-ECO can save up to 38% of the fuel consumed in case of Scenario 5.

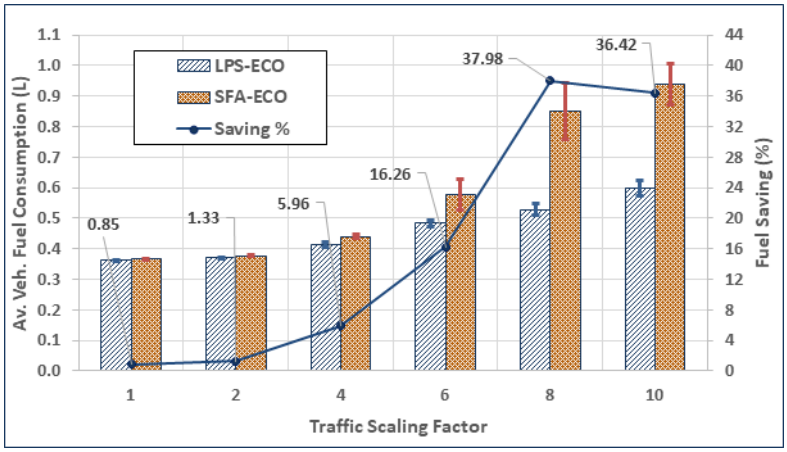


Figure 7.2: Fuel consumption for SFA-ECO and LPS-ECO for 60 s updating interval

More interestingly, the average travel time results show that the LPS-ECO significantly reduced the average vehicle travel time. Figure 7.3 shows up to a 62% reduction in the case of high traffic demand levels. For low-traffic scenarios, the differences are not significant.

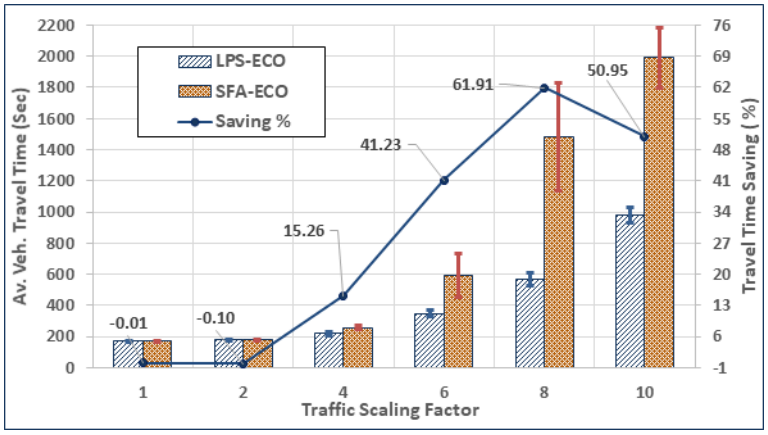


Figure 7.3: Average travel time for SFA-ECO and LPS-ECO for 60 s updating interval

Figure 7.4 shows the impact of the routing algorithms on the main traffic streams (highway traffic flows between Nodes 5 and 10). It shows the time-space diagram for the main traffic demand for a 1% uniformly selected sample of the vehicles in both the LPS-ECO and SFA-ECO cases. Figure 7.4 also shows that in the case of the shortest-path algorithm, many vehicles from the main traffic flow experienced congestion on the highway for a long time. In the case of LPS-ECO, there are small deviations from the highway free-flow speed, and vehicle trajectories show much smoother

mobility on the highway. It also shows that in the case of LPS-ECO all the vehicles in the main traffic demand finished their trips around 400 s earlier.

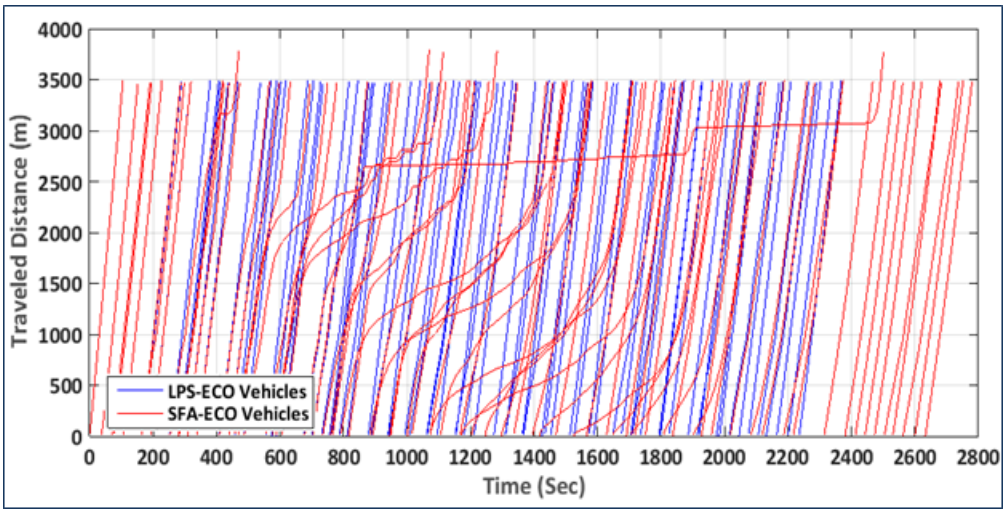


Figure 7.4: Time-space diagram for 1% sample of main traffic demand vehicles for both LPS-ECO and SFA-ECO

7.5 Discussion: The Optimality of the Results

An important question related to the optimality of this system is whether the resulting fuel cost is the system optimal. In fact, the system should produce the system optimal fuel consumption under two conditions: 1) the network conditions (link costs and traffic loads) do not change, and 2) the traffic flow for each OD is high enough.

First, the linear problem is based on an implicit assumption that the network conditions (link costs and traffic loads) do not change during the update interval, however, this is not the case. So, we used a short updating interval (60 seconds) to satisfy this assumption as possible. We assume that, within this short interval, the changes in the network conditions are negligible.

Second, the solution of the linear problem should produce the system optimal fuel consumption if it is applied directly to the continuous variable, q_{ij}^{k}. However, in the transportation network flow problem, the flow is discretized into vehicles. So, the stochastic process, which is used for constructing routes, converts the linear program solution into an integer program solution that fits the discretized network flow. This conversion deviates the solution from the optimal solution computed by the linear program. So, the final result is not exactly the system optimum, it is near

optimum. This deviation from the optimum depends on the number of vehicles in each traffic demand. If these numbers are high enough, then the discretization of the flow rate into vehicles will produce a close ratio (the number pf the vehicles from flow k that are assigned to l_{ij} divided by the total number of vehicles in this flow) to the q_{ij}^k, which makes the system closer to the optimum performance. And if the numbers of vehicles in the ODs are small, the deviation from the optimum will be higher. To show this, one can imagine a traffic demand k with a flow rate $q^k(veh/h)$, assume that the traffic portion q_{ij}^k assigned to link l_{ij} is 0.4, and the total number of vehicles N^k in flow k is only one vehicle ($N^k = 1$), in this case, the final system feasible solutions for q_{ij}^k are 0 or 1, which deviate significantly from 0.4. As N^k increases, the ratio of the number of vehicles assigned to l_{ij} will be closer to 0.4. For example, if the flow k generates 100 vehicles ($N^k = 100$), the stochastic route constructions process should ideally assign 40 vehicles to l_{ij}, but due to stochasticity, it may assign only 38, which is very close to 40 of 100 vehicles.

7.6 Conclusion and Future Work

This paper proposes a novel eco-routing technique based on linear programming optimization and stochastic route assignment (LPS-ECO). The proposed technique uses the feedback fuel consumption cost from vehicles along the road network. By considering the link capacity in the linear program constraints, the proposed algorithm is able to avoid congestion in the network. The proposed technique overcomes the shortcoming of the shortest-path-based routing techniques by assigning traffic demands on all possible routes to minimize the network-wide fuel consumption level. The simulation results show that the proposed LPS-ECO outperforms the shortest-path-based eco-routing techniques in the case of high traffic demands in terms of both fuel consumption and travel times. The performance of the algorithms, however are similar in the case of low traffic demands. The main challenge facing the proposed algorithm is the size of the linear program, which is relatively large. The total number of variables in the cost function equals $m \times f$, and the number of constraints equals $n \times f + m$, where n, m, and f are the number of nodes, links, and flows in the network, respectively. Consequently, for large networks the size of the linear program will be very large and thus heuristic algorithms would be needed to deal with the problem dimensionality.

Chapter 8

Eco-Routing: An Ant Colony Based Approach

Our study of the eco-routing feedback-based navigation technique showed that there are some limitations to its current implementations. Although the current implementations can produce significant fuel consumption savings, their performance degrades in some cases due to delayed updates or the lack of updates. Thus, in this chapter, we propose an ant colony-based eco-routing technique (ACO-ECO), which is a novel feedback eco-routing and cost updating algorithm to overcome these shortcomings. In the ACO-ECO algorithm, real-time performance measures on various roadway links are shared. Vehicles build their minimum path routes using the latest real-time information to minimize their fuel consumption and emission levels. ACO-ECO is also able to capture randomness in route selection, pheromone updating, and pheromone evaporation. The results show that the ACO-ECO algorithm and SPF-ECO have similar performances in normal cases. However, in the case of link blocking, the ACO-ECO algorithm reduces the network-wide fuel consumption and CO2 emission levels in the range of 2.3% to 6.0%. It also reduces the average trip time by approximately 3.6% to 14.0%.

8.1 Introduction

Drivers usually select routes that minimize their costs such as travel time or travel distance. However, the minimum time or distance routes do not necessarily minimize the fuel consumption or emission levels [46, 47]. There are many cases where the minimum time routes result in higher fuel

consumption levels such as high-speed routes; despite the time reduction that could be achieved, the higher speed routes may produce higher fuel consumption levels due to the higher vehicle speeds, route grades, or longer distance. Also, shorter distance routes can result in higher fuel consumption if the speed is too low or if the route has many intersections that result in numerous deceleration and acceleration maneuvers. Selecting the minimum time or minimum distance routes is simple compared to finding the minimum fuel consumption routes. The fuel consumption depends on many parameters such as distance, travel time, route grades, congestion level, vehicle characteristics, and the driving behavior.

Eco-routing was developed to select the route that minimizes vehicle fuel consumption levels between an origin and destination. In a feedback system, eco-routing depends on the vehicle and route characteristics as well as its ability to report this information to a traffic management center (TMC) that updates the routing information, rebuilds the routes, and sends the new routes to vehicles traversing the network. Eco-routing is a promising navigation technique because it results in a significant reduction in fuel consumption and emission levels. However, through some improvements, the eco-routing system can be further enhanced to produce additional fuel consumption and emission savings.

In this chapter, we first study eco-routing performance and show that in some cases its performance may not be the best we can achieve. Subsequently, based on this, we propose ant colony eco-routing (ACO-ECO) algorithm that employs ant colony optimization algorithms [127]. Due to the major differences between ant colony and the transportation network, ant colony algorithms are not directly applied to select the best routes; however, they are used to optimize the route selection process by optimizing the routing information updating and route selection mechanisms. Finally, we compare the proposed approach to the subpopulation feedback eco-routing algorithm (SPF-ECO) [128], which is implemented in the INTEGRATION software.

The remainder of this chapter is organized as follows. An overview of the subpopulation feedback assignment eco-routing (SPF-ECO) algorithm and its shortcomings is introduced. Subsequently, an overview of the ant colony optimization is presented. After that, the proposed approach (ACO-ECO) is described. Subsequently, the simulation results that compare the ACO-ECO to the SPF-ECO are presented and discussed. Finally, the study conclusions are presented together with recommendations for further research.

8.2 Subpopulation Feedback Eco-routing

In subpopulation feedback eco-routing implemented in INTEGRATION, when the vehicle enters a new link, the vehicles fuel consumption and emission levels are reset to zero for the new link. Subsequently, the SPF-ECO algorithm periodically calculates the fuel consumption and emissions for each vehicle using the VT-Micro model, as described in detail in Chapter 2.

For each vehicle, the estimated fuel consumption and emission levels are accumulated until the vehicle traverses the link. When a vehicle leaves a link, it submits its fuel consumption cost for this link to the traffic management center (TMC), which updates the link fuel consumption using some smoothing techniques. Subsequently, INTEGRATION periodically rebuilds the routes for each origin-destination pair at a frequency specified by the user. Subsequently, vehicles use the latest paths when looking identifying the next link along the route. This mechanism has three main shortcomings that are discussed in this section.

8.2.1 Fixed Cost for Empty Links

Assume that a link L_i was loaded with a high traffic flow that resulted in congestion on this link. This congestion will result in lower speeds and higher acceleration/deceleration noise, consequently, increasing the fuel consumption and emission levels on this link. At a certain time, the eco-routing system will re-route vehicles to another route with a lower cost. Since the vehicles on L_i have been exposed to the congestion, the link fuel consumption will be very high after these vehicles leave the link. As the system re-routes vehicles to other routes, the link will not be loaded by vehicles until the routing information changes. Consequently, the cost of L_i will continue to be high while it is actually decreasing. This lag in the system is typical of any feedback control system and will result in using suboptimal routes, consequently and, increasing the network-wide fuel consumption levels.

8.2.2 Fixed Cost for Blocked Links

A reverse situation can take place in the case of blocking a link (for example due to an incident). In this case, the vehicles that were not blocked will have a low fuel consumption level and will report it when leaving the link. The eco-routing will maintain a low cost for this link as long as the link is blocked since there are no vehicles leaving the link. Consequently, the SPF-ECO will continue to use this route and load more vehicles to this link resulting in higher fuel consumption

and emission levels.

8.2.3 Delayed Updates

The third point is that the updates are only sent when a vehicle leaves a link. For long links and/or low-speed links, the link travel time is relatively long. Consequently, the information used to update the SPF-ECO routing might be obsolete and may not reflect the current state of the link. This inaccurate routing information might result in incorrect routing decisions and hence increase the fuel consumption level.

In the proposed approach, we solve these problems by utilizing ant colony techniques to update the link cost function (the fuel consumption level in this application).

8.3 Ant Colony Optimization

Ant colony optimization [127] is a branch of the larger field of swarm intelligence, which studies the behavioral patterns of social insects such as bees, termites, and ants to simulate these processes. Ant colony optimization is a meta-heuristic iterative technique inspired by the foraging behavior of some ant species. In ant colony, ants walking to and from a food source deposit a substance called a pheromone on the ground. In this way, ants mark the path to be followed by other members of the colony. The shorter the path, the higher the pheromone on that route, consequently, the more preferable this route is. The other ant colony members perceive the presence of the pheromone and tend to follow paths where pheromone concentration is higher. Ant colony optimization exploits a similar mechanism for solving some optimization problems.

In this chapter, we use the same ant colony concept to improve the fuel consumption and emission cost for a transportation network. Vehicles are employed as artificial ants, while the pheromone is considered to be the inverse of the fuel consumption cost for each link. Each artificial ant periodically deposits the pheromone by updating the fuel consumption cost for the link it is traversing.

There are many variants of ant colony optimization. However, all of them share the same idea described earlier. The main steps in each iteration are: 1) construct the solutions, 2) conduct an optional local search step, and 3) update the pheromones. The ant colony system does not specify how these three steps are scheduled and synchronized; it leaves these decisions to the algorithm designer [129]. In the solution construction step, artificial ants construct a feasible solution and

add it to the solution space. The system starts with an empty solution space, the ants start at the nest, and each ant probabilistically chooses a solution e_i between a set of paths $\{e_1, e_2, ..., e_k\}$ to reach the food source. To choose between these paths, each ant uses the probability P_i computed in Equation 8.1:

$$P_i = \frac{\varphi}{\sum_{j=1}^{k} \varphi_j} \tag{8.1}$$

where φ_i is the amount of pheromone on path e_i. This probabilistic behavior for route selection guarantees the exploration of more feasible solutions and avoids converging to local ones. The pheromone updating takes place while the ants are moving, where they deposit the pheromone on their paths. Also, as time passes, the pheromone evaporates based on an evaporation factor ρ. Subsequently, after each iteration, the pheromone is updated according to Equation 8.2:

$$\varphi_i = (1-\rho)\varphi_i + \sum_{j=1}^{m} \Delta\varphi_j \tag{8.2}$$

where m is the number of ants that traverse a link, and $\Delta\varphi_j$ is the amount of pheromone deposited by ant j. After the solution construction and before the pheromone updating, a local search step can be carried out to improve the solution. This step is optional and problem specific.

In the proposed approach, we utilize these steps to achieve our objective of decreasing the fuel consumption and consequently the pollutant emissions.

8.4 Ant colony based Eco-routing (ACO-ECO)

This section presents the proposed approach (ACO-ECO) and describes its operation in detail. In ACO-ECO, ant colony techniques will be applied to optimize the fuel consumption and emissions in the transportation network. The vehicles are the artificial ants, and the pheromone is the inverse of the fuel consumption. Because of major differences between the ant colony system and the transportation network, we introduce some variations to ant colony techniques to tailor it to the specific application. The ACO-ECO uses a number of steps that are described here.

8.4.1 Initialization

This phase initializes the cost associated with the various links. Because, initially, the links are free, the cost of each link is initialized to the free flow speed fuel consumption as described in Chapter 2.

8.4.2 Route Construction

This phase starts directly after the initialization phase and is repeated periodically and was defined to be 60 simulation seconds in this application. In this phase, the ACO-ECO builds the minimum path based on the cost of each link. When the vehicle leaves a route link, it searches the tree to find its next link.

The probabilistic route selection (introduced by Equation 8.2) is an important mechanism in ant colony algorithms to search all the available routes. However, this mechanism as described in Equation 8.2 cannot be applied in vehicular route selection because it is not realistic. As mentioned earlier, drivers try to select routes that minimize their cost, while this probabilistic selection assigns a random route to each vehicle based on the routes pheromone level (route cost) relative to that for all other routes. Using this equation, and due to the randomness, a vehicle might be assigned a very high-cost route, which is not realistic and is not consistent with the driver behavior when selecting routes. Consequently, it will result in a higher fuel consumption level. So, we use another technique to introduce some limited randomness into the route selection mechanism while maintaining the error within a given predefined margin. An error factor is configured for the network. This error factor α is used to add some error to the cost of the links, subsequently to the tree building and the route selection algorithms. The error value added to the link cost is a randomly selected point from the standard normal distribution $N(0,\sigma)$, where σ is the standard deviation, which is computed as and $\sigma = \alpha.C_l$, where C_l is the link cost. In this way, we have a guarantee that 95.45% of the link costs are within $(1 \pm 2\alpha).C_l$, which means that by controlling the error factor we can control the randomness level within the route selection algorithm.

8.4.3 Pheromone Update

In this phase, two updating processes take place: pheromone deposition where ants deposit pheromone to indirectly communicate the route preference to the following ants and pheromone evaporation, where the pheromone level on each link decays with time.

8.4.4 Pheromone Deposition

In the SPF-ECO, the vehicles only submit the link cost when leaving the link. The advantage of this method is the small number of updates being sent on the network and consequently the low network overhead. But on the other hand, it results in delayed updates and fixed cost for empty or blocked links as mentioned earlier.

In contrast to the SPF-ECO, the ACO-ECO overcomes these issues by enabling vehicles to submit multiple updates while traveling the link. These updates can be sent periodically on either a time-basis or a distance-basis. Using time-based updating, the vehicles have a predefined maximum updating interval T. The vehicles should send their estimation of the link cost each T seconds. This cost updating method can control the number of updates that are sent over the network. However, it has an important drawback; for low-speed links or blocked links, the vehicles will send many unnecessary updates. Another drawback is that, for short length links and/or high-speed links, this time interval T may be longer than the link traversal time. Consequently, no updates would be sent for these links. This drawback can be overcome by setting T to a value that is shorter than the minimum link travel time in the network, however, this will result in many unnecessary updates for long links or low-speed links. Another way to submit the link cost updates is distance based updating, where vehicle submits an update every distance D it traverses the link. In contrast to the time-based updating, the distance based method limits the number of updates for each link. But on the other hand, for blocked links, the updates will not be sent and, consequently, the cost will be fixed for blocked links resulting in the same problem as the SPF-ECO algorithm.

Consequently, a compromise approach is utilized, which combines both the time- and distance-based updating to take advantage of the merits of each approach. Also, we used the end of the link updating where the vehicle sends an update when it leaves the link. To estimate the link fuel consumption, the ACO-ECO algorithm defines the maximum time interval T and the maximum distance D to report conditions. When any one of these conditions is met, the vehicle submits a new update quantifying its estimation for the overall link cost and then resets its time and distance counter. To calculate the fuel consumed, the ACO-ECO periodically estimates the fuel consumption rate using the VT-Micro model, and, then, uses Equation 8.3 to accumulate the total fuel consumed in the previous interval:

$$C = \sum_{t} F(t).\Delta t. \tag{8.3}$$

where $F(t)$ is the VT-Micro model instantaneous fuel consumption rate, and Δt is the fuel con-

sumption calculation interval which is typically 0.1 seconds in INTEGRATION. Whenever either T or D is reached, the ACO-ECO estimates the overall link fuel consumption C_l as shown in Equation 8.4:

$$C_l = \frac{C.L}{d}. \tag{8.4}$$

where d is the distance traveled in the previous period in meters $(d \leq D)$, and L is the link length in meters. This calculation assumes that the conditions on the remainder of the link will continue as was observed by the vehicle.

8.4.5 Pheromone Evaporation

To overcome the fixed cost problem for empty links, the cost of these links must be updated when the TMC has not received updates for a period of time. In ant colony optimization, if no pheromone is deposited for a long time, the link pheromone level will decay towards zero due to the evaporation; this is an indication of the low preference for that route. In a transportation network, not receiving an update about a link for a long time indicates that this link is empty. Consequently, the cost of this link must be updated toward the free flow speed cost (C_{ff_l}). So, in this case, the TMC updates the cost as follows. First, it finds the minimum updating interval (τ_l) for the link. This value is the minimum of three parameters; the updating interval (T), the link travel time at free-flow speed, and the updating interval in case of distance based updating. These parameters are shown in Equation 8.5. The rationale is that after receiving an update, the next vehicle will send an update in case of one of three situations: it reaches its updating interval T, it reaches its updating distance, or it ends the link. We have

$$\tau_l = min(T, \frac{L_l}{S_{ff_l}}, \frac{D_l}{S_{ff_l}}), \tag{8.5}$$

where T is the updating interval, D is the updating distance, L_l is the link length and S_{ff_l} is the free-flow speed of the link. Subsequently, the ACO-ECO algorithm estimates the overall link cost C_l as shown in Equation 8.6. This evaporation technique results in exponential increasing or decreasing in the link cost towards the free-flow speed cost.

$$C_l = C_l - \frac{\Delta t}{\tau_l}(C_l - C_{ff_l}), \tag{8.6}$$

where C_{ff_l} is the free-flow speed fuel consumption estimate for the link, and Δt is the evaporation interval after which the evaporation process should be performed for the link cost if no updates were received.

8.5 Simulation Results

In this section, we compare the proposed approach ACO-ECO to the eco-routing in INTEGRATION (SPF-ECO) for different traffic rates using the INTEGRATION software. The network shown in Figure 8.1 is used for comparing the two approaches. It is the same network used in Chapter 6.

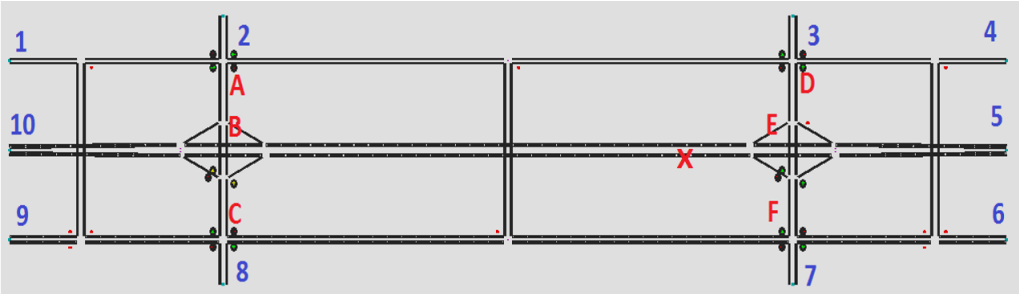


Figure 8.1: Simulated road network

The network consists of 10 zones with the main highway (center horizontal road) between zone 1 and zone 2, and two arterial roads (side roads). The network size is 3.5 km x 1.5 km. The free-flow speeds are 110 and 60 km/h for the highway and arterial roads, respectively. The highway has 3 lanes in each direction, while the other roads have only 2 lanes in each direction. Regarding the origin-destination traffic demands (O-D demands), we use 5 different scenarios, as shown in Table 8.1. The main traffic stream is the traffic between zone 1 and 2 for each direction, the side traffic streams are between each two other zone pairs. This traffic rate is generated for half an hour, and the simulation runs for 4500 seconds to ensure that all the vehicles complete their trips.

The comparison is done in two cases: the normal operation (no incident) case where there is no link blocking and in the case of blocking due to an incident (link blocking case). For each case, we run each traffic assignment technique (ACO-ECO, and SPF-ECO) 20 times with different seeds to consider the output variability due to randomization. This is repeated for each of the five O-D demand configurations. The error factor is set for both techniques to 1%. For the ACO-ECO parameters, the maximum update interval T is 180 seconds, and the maximum update distance D is 750 meters.

Table 8.1: Origin-Destination Traffic Demand Configuration

	Main Demand (Veh/h)	Secondary Demand (Veh/h)	Total No. of Vehicles
1	500	50	1600
2	1000	75	2650
3	1500	100	3700
4	2000	125	4750
5	2500	150	5800

8.5.1 Normal Operation Scenarios

For the normal operation scenarios, the results show no significant differences between the ACO-ECO and the SPF-ECO for average fuel consumption levels, as shown in Figure 8.2. The figure also shows that as the traffic demand increases, the average fuel consumption and the average trip time increases due to the higher congestion levels. Moreover, the results show the same behavior for the average trip time, the CO_2 and NO_x emissions levels, where ACO-ECO has no significant effect on any of them.

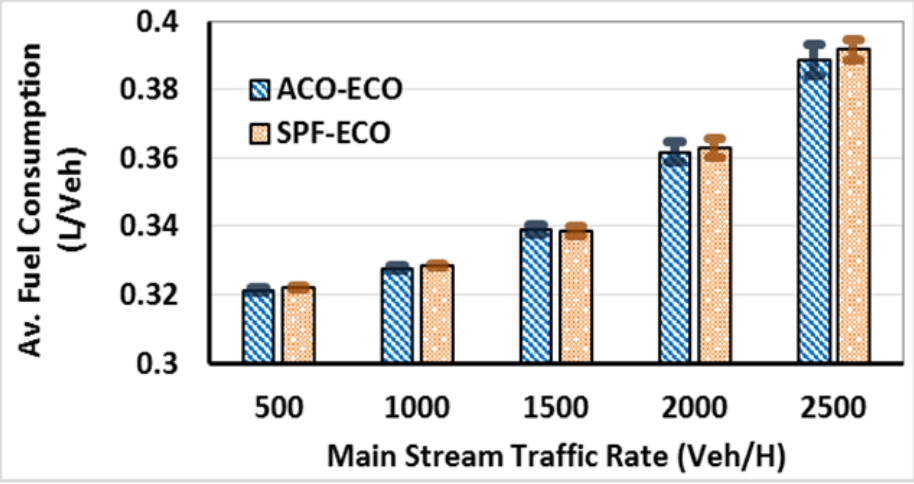


Figure 8.2: Average Fuel Consumption (L/Veh)

Regarding the *CO* emission, the ACO-ECO has a higher emission level as shown in Figure 8.3.

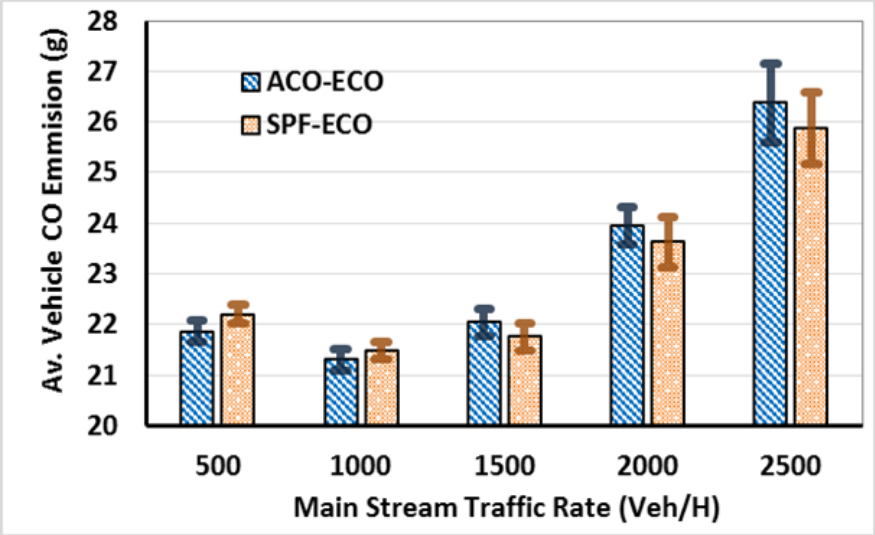


Figure 8.3: Average Vehicle CO Emission

8.5.2 Incident Scenarios

To simulate the link blocking in the network, we configured an incident on the highway from zone 1 and 2 at point (X) marked in Figure 8.1, the incident does not affect the other direction from zone 2 to zone 1. This incident occurs 10 minutes after starting the simulation and blocks 50% of the highway (1.5 lanes) for 5 minutes. Then the blocking is reduced to 25% of the highway for the next 10 minutes, then the incident is completely removed and the highway works at its full capacity.

Figure 8.4 shows the fuel consumption in case of an incident. The figure demonstrates that the ACO-ECO algorithm reduces the average fuel consumption level for all traffic demands. The reduction ranges between 2.3% to 6% compared to the SPF-ECO.

These results show the ability of ACO-ECO to reduce the fuel consumption level and the trip time in addition to all the time-related measurements. ACO-ECO also succeeds in reducing the pollutant emissions in most cases.

Table 8.2 shows the percentage reduction attributed to the ACO-ECO for both fuel consumption, different emissions, and different time-related measurements. For instance, the fuel consumption is reduced by 6% in the moderate traffic scenario, and this reduction ratio decreases as the traffic demand increases. This also applies to the CO_2 emissions and the time-related measurements. The reason is that as the traffic demand increases, the congestion increases and thus affects all the alternative routes, which limits the ACO-ECO ability to recover from the congestion.

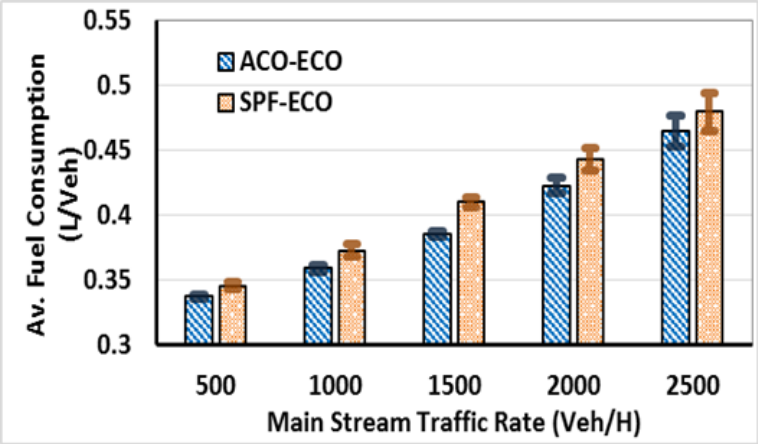


Figure 8.4: The Average Fuel for the Link Blocking Scenario

To find the significance of the reduction made by ACO-ECO, analysis of variance (ANOVA) is employed to compare means of ACO-ECO to that of SPF-ECO. The null hypothesis is that the means for both algorithms are equal $(H_0 : \mu_1 = \mu_2)$, and the alternate hypothesis is $(H_0 : \mu_1 \neq \mu_2)$.

We applied ANOVA for the fuel consumption results in the lowest traffic rate, given that this scenario has the lowest reduction in fuel consumption. The result shows that the p-value is less than 0.0001, which gives strong evidence to reject the null hypothesis, and shows the significance of the reduction made by the ACO-ECO. Since the lowest reduction level is significant, we can conclude that the higher levels for other configuration are also significant. Table 8.2 also shows some rare cases where some emissions increase due to the use of ACO-ECO. For instance, *CO* and NO_x emissions increased in the case of high traffic rates.

Table 8.2: Percent of reduction made by ACO-ECO over SPF-ECO in case of link blocking

Traffic rate	**Fuel**	CO_2	**CO**	**HC**	NO_X	**Trip Time**	**Stop Delay**	**Accel. Noise**	**Accel./Decel. Delay**
500	2.37	2.29	3.75	3.71	1.60	3.64	4.04	1.87	12.02
1000	3.72	3.86	1.05	1.73	0.91	8.83	19.04	4.90	21.97
1500	6.06	6.42	-1.51	0.38	0.24	14.98	27.68	5.28	25.43
2000	4.57	4.75	0.49	2.19	0.11	12.66	19.75	4.91	16.84
2500	3.09	3.32	-2.10	-0.58	-0.75	7.11	15.39	1.61	11.34

8.6 Conclusion

In this chapter, we developed the ACO-ECO traffic assignment technique that is inspired by the ant colony optimization algorithm. ACO-ECO attempts to enhance the SPF-ECO algorithm that is currently implemented in the INTEGRATION software. These enhancements include cases in which the links are blocked or no vehicles traverse the link. ACO-ECO employs the ant colony techniques to reduce the fuel consumption and emission levels. It uses route construction to build routes and assign them to vehicles, it also applies pheromone deposition and pheromone evaporation to update the route link costs. These ant colony techniques are customized to be suitable for transportation networks. In the case of normal operation, the ACO-ECO performance is similar to the SPF-ECO, while for link blocking scenarios, ACO-ECO reduces the fuel consumption, average trip time, stopped delay, and most of the emission levels. An important advantage of the ACO-ECO is its flexibility, where its parameters (error factor, maximum updating time, maximum updating distance, and evaporation interval) can be tuned to achieve better performance. The fine tuning and testing of these parameters are an important future extension of the work presented in this chapter.

Another future research is to study the effect of each of the new updating methods on the network traffic and studying the trade-off between the reduction in the fuel consumption and emission levels and the communication network traffic load. The market penetration rate is an effective and important parameter that should be studied. Also, it is important to study the effect of the communication network on the ACO-ECO performance.

Chapter 9

Findings, Conclusions, and Future Work

In this thesis, we focus on the study of the mutual interdependence of the communication and transportation systems in large-scale vehicular environments for the modeling of smart cities and ITS systems. Specifically, we focus on the eco-routing navigation technology as an example of new emerging ITS technologies. The reason behind selecting eco-routing is its expected benefits for human life, that is, the current statistics demonstrate the important economic and environmental impacts of the transportation sector on human life. So, the eco-routing is a promising technology that can mitigate these negative impacts by reducing the fuel consumption and emission levels. Another reason behind selecting eco-routing is that it is a good example of the mutual influence between the communication and transportation systems.

The study we conducted shows that this mutual interaction can significantly affect the performance of ITS applications. Moreover, it can affect the communication performance, especially for those ITS applications that control the vehicles' mobility trajectories such as routing applications. Therefore, it shows the importance of considering this mutual interaction and its impact when deploying new ITS technologies, where the need for accounting for the communication performance becomes essential in the cases of moderate and high vehicular traffic demand levels.

The research work we pursued in this dissertation paves the road for the research community in the areas of computer science, communication, and traffic engineering to realistically study communication and transportation systems and the mutual interaction between the two systems in large-scale vehicular environment. For example, from the computer science perspective, networks and systems are very active research areas, and our study in this dissertation is considered an important step towards large-scale modeling of communication in vehicular environment, where the models we developed in Chapter 3 and Chapter 5 enable the research community to study

communication and transportation in smart cities with realistic mobility that captures mobility interactivity between vehicles as well as between vehicles and different road network controls such as traffic signals. This realistic mobility enables computer and network researchers to have insight understandings of the communication at this scale. Thus, enables them to develop new algorithms that can improve both communication systems as well as applications running as services over this communication network.

Moreover, the models we developed in Chapter 6 enables researchers in traffic engineering and planing as well as computer science to model and study transportation systems of smart cities including different transportation modes at microscopic level.

9.1 Communication and Eco-routing

In this research, we studied the eco-routing navigation and its interactivity with the communication network. The modeling systems we developed Chapter 3 shows that, by coupling communication simulator with microscopic traffic simulator, we can capture low-level details related to both systems. Such low-level information, which comes from the communication discrete event simulation and transportation microscopic simulation, gives us insight understanding of the interdependency of communication and transportation systems, based on which, novel ITS technologies can be better developed, designed, and deployed.

Our study also showed that the modeling of large-scale vehicular systems including both mobility and communication is realizable by using realistic traffic simulators and an analytical model to represent the communication system instead of discrete event simulation. However, at this scale, it is too difficult to capture low-level communication details, because the analytical model gives us an estimation of communication performance parameters and cannot capture all the events that happen in communication network. However, data that can be collected from such large-scale vehicular system helps us to understand the interactivity of communication and transportation systems and, consequently, to anticipate network problems and to find proper solutions.

Impact of Mobility on Communication Performance

In more details, this study shows that considering transportation or communication systems separately from each other can be deceptive and may lead to incorrect conclusions. For example, our study of the impact of mobility on communication in VANET shows an interesting conclusion that contradicts with what was concluded in the literature. It shows that in vehicular networks, the

higher speed can result in better communication performance. The reason is that in real networks, the vehicle speed and density are inversely related to each other, i.e, higher speeds are always accompanied with lower vehicle density, which is a well-known concept in the traffic flow theory. This inverse relationship means that at higher speeds the medium access competition and collision will be lower (if we have all the communication parameters the same) and, consequently, the MAC layer will experience lower packet drop rate and shorter packet delay. This conclusion was inferred from different communication applications that include file transfer and voice over IP applications.

Mutual Impact Loop

Our study of the impact of communication on eco-routing, in Chapter 5, shows that the packet drop rate does not have significant impact on the eco-routing performance if the background data traffic is low. The reason is that, even at high congested networks, the eco-routing packets are generated at a low rate (a vehicle generates only one packet for each link), which means the communication performance will be acceptable.

However, at higher background packet generation rates, the results from real network and real traffic show that the communication performance can significantly influence the eco-routing performance. It also shows that the severity of this impact increases with vehicular traffic demand level and congestion level. The reason is three fold. First, at high demand levels, accurate and sufficient routing information becomes imperative to efficiently route vehicles through the congested network. Secondly, and contrary, this high congestion level increases the vehicle density in the network. Thus, it increases the medium access competition, subsequently, the packet drop rate and the packet delays, which directly affect the routing information accuracy and, consequently, the correctness of the routing decisions made by the traffic management center. If the errors in the routing decisions exceed certain levels, it may increase the traffic congestion level and result in higher fuel consumption and emission levels. The severity of this impact can reach an extent that incorrect routing decisions may result in network grid-locks. The third part of the reason is the mutual influence loop, where the resulting high congestion will increase the vehicle density that will negatively affect the communication performance. The system may circulate in this mutual influence loop until a grid-lock stops the vehicles in the network.

Fuel Consumption and Packet Drop Rate

The analysis of the relationships between fuel consumption and the packet drop rate in real network and real traffic, that we conducted in Chapter 5, shows that eco-routing can work properly even at higher packet drop rates as long as it receives sufficient updates about the cost of each road link. This conclusion was clear at moderate traffic demand levels in the real network we used,

where the packet drop rate exceeds 90% while the fuel consumption is not significantly affected. The reason is that, despite the high packet drop rate, at these moderate traffic conditions, the traffic flow of vehicles passing each road link is high enough that sufficient updates are correctly delivered to the traffic management center. As the traffic demand level increases, the congestion increases, producing lower vehicle speeds and lower flow rate of the vehicles on the road links. Consequently, the number of updates that are correctly delivered to the traffic management center becomes insufficient to correctly route the vehicles. If the network continues at this high demand level for enough time, these inaccurate routing decisions initiate the mutual influence loop.

Location of the Roadside Units

In our study, we focused on the Vehicle to Instructional communication. Thus, roadside unit allocation is an important communication setting. Therefore, we studied the impact of the roadside units allocation on the performance of eco-routing. The study shows that RSU allocation is an important factor that can affect eco-routing performance. The reason is that the locations of the RSUs determine which links will be updated immediately, updates of which links will be delayed, and which links will not be updated at all. Thus, when deploying eco-routing technology, if the network cannot be fully covered by RSUs, it is recommended that the coverage to be uniformly spatially distributed. This coverage spatial uniform distribution will enable the traffic management center to collect uniformly distributed information from overall the network that can represent the network state as better as possible within the coverage budget.

9.2 Utilizing New Computing Technologies

In our study, in Chapte 6 we developed a system for multimodal agent-based modeling of smart cities, INTGRAT3. Using this system to simulate the transportation system of the greater city of Los Angles in its morning peak hours shows that by utilizing new communication and computing technologies to route only 10% of the commuters, in addition to the using of eco-routing technology, can save about 9.7% of the total fuel consumed; moreover, it produces travel time saving compared to the time-based routing.

We also showed that by utilizing system optimum eco-routing techniques, the system we developed in Chapter 7 (LPS-ECO), the transportation network performance can be significantly improved. Compared to shortest path eco-routing, system optimum routing techniques can consider other network parameters such as road link capacities and current traffic demand rates on each road link. Therefore, these routing techniques can optimize the fuel consumption not only by selecting

the best route but also by routing some vehicles through non-best routes to avoid overloading the shortest routes. However, such system optimum routing techniques are computationally extensive. But, by utilizing parallel computing, these systems can be deployed and ran in real time.

Moreover, we demonstrated that by utilizing some heuristic techniques, we can overcome some shortcomings of the current eco-routing techniques, where the ant colony-based eco-routing we proposed improved the performance of the eco-routing system in the case of incident and route blockage.

9.3 Future Work

The study we conducted focused on vehicle-to-infrastructure communication. Thus, an important future extension of this work is to consider vehicle-to-vehicle communication and hybrid communication paradigms. This extension will impose the use of routing protocols, which present new challenges as well as opportunities to the feedback-based eco-routing navigation. Utilizing the vehicle-to-vehicle or hybrid communication paradigms eliminates or decreases the need for road-side units. Thus, it reduces the cost of deploying eco-routing technology. However, it imposes the routing protocols and their dynamics into the system, which can significantly affect system performance. An important question that should be answered in this context is how such a system will behave in the case of low penetration rate of connected vehicles.

Another future extension is to study the impact of other applications on the performance of eco-routing systems. In this context, different application types and their priorities in the VANET medium access technique must be considered. This can be easily done using VNetIntSim on the small scale networks, because VNetIntSim is capable of modeling these different applications and the medium access handling for their packets. However, for large-scale systems, we need to extend the analytical model to support the quality of service and the four access categories in the MAC layer. Thus, developing this model is one of our foreseen future works.

Using the platforms we developed, we can also study the impact of the communication system on other applications, especially, applications that are sensitive to the delay and the communication reliability. An important example for such applications in the safety applications, where the communication messages are exchanges to avoid accidents. Despite they use the highest priority access category, such delay sensitive applications may be significantly affected by the communication performance, especially in the high-density vehicular network with high packet generation rates.

It is also of interest to study the impact of the communication system on other ITS applications such as speed harmonization, eco-driving, congestion avoidance, and intersection control algorithms. Studying these applications mandates the study of the effect of quality of services on the performance of the transportation system and services offered for both users and vehicles.

Appendix A

The communication model derivation

This appendix shows the detailed derivation for the communication model in Chapter 5. To make it easier to follow up, we copy the Markov chain from Chapter 5 to here in Figure A.1. We use the notation in Table 5.1.

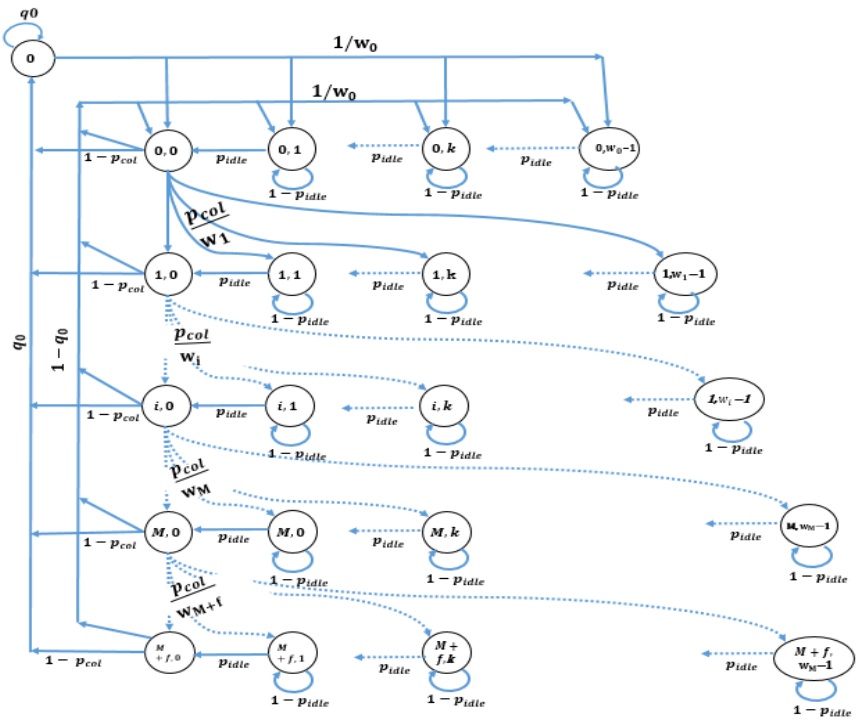


Figure A.1: Markov chain model for the medium access.

The general idea is to express all the state probabilities as function of $P(0,0)$ and then the summation of all these state probabilities must equal to 1.

From the Markov chain find the relationship between $P(0)$ and other states as:

$$P(0) = q_0\, P(M+f-1,0) \;+\; q_0\, P(0) + q_0(1-p_{col}) \sum_{i=0}^{M+f-2} P(i,0). \tag{A.1}$$

So, we can easily calculate $P(0)$ as:

$$P(0) = \frac{q_0}{1-q_0}\left(P(M+f-1,0) + (1-p_{col}) \sum_{i=0}^{M+f-2} P(i,0)\right). \tag{A.2}$$

Now, we start to find a relationship between $P(0)$ and the state probabilities in the first phase $P(0,j) where j = 0,1,2,...,w_0-1$. Stating form the last state in this phase where $j = w_0 - 1$, we will have the following relationship:

$$P(0,w_0-1) = \frac{1-q_0}{w_0}\left(P(0) + P(M+f-1,0) + (1-p_{col}) \sum_{i=0}^{M+f-2} P(i,0)\right) + (1-p_{idle})P(0,w_0-1). \tag{A.3}$$

So, $P(0,w_0-1)$ can be expressed as:

$$P(0,w_0-1) = \frac{1-q_0}{p_{idle}\, w_0}\left(P(0) + P(M+f-1,0) + (1-p_{col}) \sum_{i=0}^{M+f-2} P(i,0)\right). \tag{A.4}$$

For the next state in the first phase, $P(0,w_0-2)$, we can find the following relationship:

$$P(0,w_0-2) = \frac{1-q_0}{w_0}\left(P(0) + P(M+f-1,0) + (1-p_{col}) \sum_{i=0}^{M+f-2} P(i,0)\right) + P_{idle}P(0,w_0-1) + (1-p_{idle})P(0,w_0-2). \tag{A.5}$$

Then, we can derive the following Equation for $P(0,w_0-2)$:

$$
\begin{aligned}
P(0,w_0-2) &= \frac{1-q_0}{w_0}\left(P(0)+P(M+f-1,0)+(1-p_{col})\sum_{i=0}^{M+f-2}P(i,0)\right) \\
&\quad + p_{idle}\,P(0,w_0-1)+(1-p_{idle})\,P(0,w_0-2) \\
&= \frac{1-q_0}{w_0}\left(P(0)+P(M+f-1,0)+(1-p_{col})\sum_{i=0}^{M+f-2}P(i,0)\right) \\
&\quad + P_{idle}\,P(0,w_0-1)+(1-p_{idle})\,P(0,w_0-2) \\
&= \frac{1-q_0}{p_{idle}w_0}\left(P(0)+P(M+f-1,0)+(1-p_{col})\sum_{i=0}^{M+f-2}P(i,0)\right)+P(0,w_0-1).
\end{aligned}
\tag{A.6}
$$

Substituting $P(0,w_0-1)$ from Equation A.4 into Equation A.6, we can calculate $P(0,w_0-2)$ as:

$$
\begin{aligned}
P(0,w_0-2) &= 2P(0,w_0-1) \\
&= 2\frac{1-q_0}{p_{idle}w_0}\left(P(0)+P(M+f-1,0)+(1-p_{col})\sum_{i=0}^{M+f-2}P(i,0)\right).
\end{aligned}
\tag{A.7}
$$

Consequently, we can easily see that the state probability of any state $P(0,k)\,(k=1,2,...,w_0-1)$ in the first phase can be expressed as:

$$
P(0,k) = \frac{w_0-k}{w_0}\,\frac{1-q_0}{p_{idle}}\left(P(0)+P(M+f-1,0)+(1-p_{col})\sum_{i=0}^{M+f-2}P(i,0)\right) \quad \forall \;\; k=1,2,...,w_0-1. \tag{A.8}
$$

Now, we can find the probability of the first state in the first phase $P(0,0)$ as:

$$
P(0,0) = \frac{1-q_0}{w_0}\left(P(0)+P(M+f-1,0)+(1-p_{col})\sum_{i=0}^{M+f-2}P(i,0)\right)+P_{idle}\,P(0,1). \tag{A.9}
$$

By substituting the value of $P(0,1)$ from Equation A.8 into Equation A.10, we will have

$$
\begin{aligned}
P(0,0) = \frac{1-q_0}{w_0}\left(P(0)+P(M+f-1,0)+(1-p_{col})\sum_{i=0}^{M+f-2}P(i,0)\right)+ \\
\cancel{p_{idle}}\,\frac{w_0-1}{w_0}\,\frac{1-q_0}{\cancel{p_{idle}}}\left(P(0)+P(M+f-1,0)+(1-p_{col})\sum_{i=0}^{M+f-2}P(i,0)\right) \\
= (1-q_0)\left(P(0)+P(M+f-1,0)+(1-p_{col})\sum_{i=0}^{M+f-2}P(i,0)\right).
\end{aligned}
\tag{A.10}
$$

From Equations A.8 and A.10, $P(0,k)$ can be expressed in terms of $P(0,0)$ as:

$$
P(0,k) = \frac{w_0-k}{p_{idle}\,w_0}P(0,0) \tag{A.11}
$$

Now, we will calculate the state probabilities for the second phase, $P(1,k)\ \forall\ k = 0,1,2,...,w_1-1$ by finding a relationship between each of them and $P(0,0)$. Starting by the last state $P(1,w_1-1)$, we can derive the following relationship directly from the Markov chain

$$
\begin{aligned}
P(1,w_1-1) &= \frac{p_{col}}{w_1}P(0,0)+(1-p_{idle})P(1,w_1-1) \\
&= \frac{p_{col}}{p_{idle}\,w_1}P(0,0).
\end{aligned}
\tag{A.12}
$$

Then, the next state $P(1,w_1-2)$ can be calculated as:

$$
\begin{aligned}
P(1,w_1-2) &= \frac{p_{col}}{w_1}P(0,0)+p_{idle}P(1,w_1-1)+(1-p_{idle})P(1,w_1-2) \\
&= \frac{p_{col}}{p_{idle}w_1}P(0,0)+P(1,w_1-1) \\
&= 2\frac{p_{col}}{p_{idle}w_1}P(0,0).
\end{aligned}
\tag{A.13}
$$

By the same procedure we can find the following relationship for the sate probabilities $P(1,k)$ in the second phase:

$$P(1,k) = \frac{w_1 - k}{w_1} \frac{p_{col}}{p_{idle}} P(0,0) \qquad \forall\, k = 1,2,...,w_1 - 1. \tag{A.14}$$

Now, we can derive the probability of first state in the second phase $P(1,0)$ as follows:

$$\begin{aligned} P(1,0) &= \frac{p_{col}}{w_1} P(0,0) + p_{idle}\, P(1,1) \\ &= \frac{p_{col}}{w_1} P(0,0) + p_{idle} \frac{w_1 - 1}{w_1} \frac{p_{col}}{p_{idle}} P(0,0) \\ &= p_{col}\, P(0,0). \end{aligned} \tag{A.15}$$

Using the same procedure for the next phases $i = 2,3,...,M+f-1$, we can derive the following relationships:

$$P(i,0) = p_{col}^{i}\, P(0,0) \qquad i = 1,2,...,M+f-1 \tag{A.16}$$

and

$$P(i,k) = \frac{w_i - k}{w_i} \frac{p_{col}^{i}}{p_{idle}} P(0,0) \qquad \forall\, k = 1,2,...,w_i - 1 \quad \textit{and} \quad \forall\, i = 1,2,...,M+f-1. \tag{A.17}$$

From Equations A.2 and A.16 we can compute P(0) as:

$$
\begin{aligned}
P(0) &= \frac{q_0}{1-q_0}\left(P(M+f-1,0)+(1-p_{col})\sum_{i=0}^{M+f-2}P(i,0)\right)\\
&= \frac{q_0}{1-q_0}\left({p_{col}}^{M+f-1}\,P(0,0)+(1-p_{col})\sum_{i=0}^{M+f-2}{p_{col}}^{i}\,P(0,0)\right)\\
&= \frac{q_0}{1-q_0}\,P(0,0)\left({p_{col}}^{M+f-1}+(1-p_{col})\frac{(1-{p_{col}}^{M+f-1})}{(1-p_{col})}\right)\\
&= \frac{q_0}{1-q_0}\,P(0,0).
\end{aligned}
\tag{A.18}
$$

Now we have computed the probability of each state in the Markov chain in terms of $P(0,0)$. From the Markov chain properties we know that the summation of the state probabilities equals to 1.

$$
\begin{aligned}
1 &= P(0)+\sum_{i=0}^{M+f-1}P(i,0)+\sum_{k=1}^{w_0-1}P(0,k)+\sum_{i=1}^{M+f-1}\sum_{k=1}^{w_{i-1}}P(i,k)\\
&= \frac{q_0}{1-q_0}P(0,0)+P(0,0)\sum_{i=0}^{M+f-1}{p_{col}}^{i}+\frac{1}{p_{idle}}P(0,0)\sum_{k=1}^{w_0-1}\frac{w_0-k}{w_0}\\
&\quad +P(0,0)\sum_{i=1}^{M+f-1}\sum_{k=1}^{w_{i-1}}\frac{w_i-k}{w_i}\frac{{p_{col}}^{i}}{p_{idle}}\\
&= P(0,0)\left(\frac{q_0}{1-q_0}+\sum_{i=0}^{M+f-1}{p_{col}}^{i}+\frac{1}{p_{idle}}\sum_{k=1}^{w_0-1}\frac{w_0-k}{w_0}+\sum_{i=1}^{M+f-1}\sum_{k=1}^{w_{i-1}}\frac{w_i-k}{w_i}\frac{{p_{col}}^{i}}{p_{idle}}\right)\\
&= P(0,0)\left(\frac{q_0}{1-q_0}+\frac{1-{p_{col}}^{M+f}}{1-p_{col}}+\frac{w_0-1}{2p_{idle}}+\sum_{i=1}^{M+f-1}\left(\frac{{p_{col}}^{i}}{p_{idle}}\sum_{k=1}^{w_{i-1}}\frac{w_i-k}{w_i}\right)\right)\\
&= P(0,0)\left(\frac{q_0}{1-q_0}+\frac{1-{p_{col}}^{M+f}}{1-p_{col}}+\frac{w_0-1}{2p_{idle}}+\sum_{i=1}^{M+f-1}\left(\frac{{p_{col}}^{i}}{p_{idle}}\frac{w_i-1}{2}\right)\right).
\end{aligned}
\tag{A.19}
$$

Assuming the window exponential factor is , then $w_i = {w_0}^{i}$. This exponential increase takes place for $i <= M$

$$
w_i = \begin{cases} w_0\,\alpha^{i} & i \le M,\\ \\ w_0\,\alpha^{M} & i > M. \end{cases}
\tag{A.20}
$$

Based on Equation A.20, the last summation in Equation A.19 can be divided into two summations. Thus Equation A.19 can be written as:

$$
\begin{aligned}
1 = P(0,0)\Bigg(& \frac{q_0}{1-q_0} + \frac{1-{p_{col}}^{M+f}}{1-p_{col}} + \frac{w_0-1}{2p_{idle}} + \\
& \frac{1}{2p_{idle}}\Bigg(\sum_{i=1}^{M-2} {p_{col}}^{i}(w_0\,\alpha^{i}-1) + (w_0\,\alpha^{M-1}-1) \sum_{i=M-1}^{M+f-1} {p_{col}}^{i} \Bigg) \Bigg) \\
= P(0,0)\Bigg(& \frac{q_0}{1-q_0} + \frac{1-{p_{col}}^{M+f}}{1-p_{col}} + \frac{w_0-1}{2p_{idle}} + \\
& \frac{1}{2p_{idle}}\Bigg(w_0\,\frac{p_{col}\,\alpha-(p_{col}\,\alpha)^{M-1}}{1-p_{col}\,\alpha} - \frac{p_{col}-{p_{col}}^{M-1}}{1-p_{col}} \\
& + (w_0\,\alpha^{M-1}-1)\frac{{p_{col}}^{M-1}-{p_{col}}^{M+f}}{1-p_{col}} \Bigg) \Bigg).
\end{aligned}
\tag{A.21}
$$

Consequently, we can compute $P(0,0)$ as:

$$
\begin{aligned}
P(0,0) = \Bigg(& \frac{q_0}{1-q_0} + \frac{1-{p_{col}}^{M+f}}{1-p_{col}} + \frac{w_0-1}{2p_{idle}} + \frac{1}{2p_{idle}}\Bigg(w_0\,\frac{p_{col}\,\alpha-(p_{col}\,\alpha)^{M-1}}{1-p_{col}\,\alpha} - \\
& \frac{p_{col}-{p_{col}}^{M-1}}{1-p_{col}} + (w_0\,\alpha^{M-1}-1)\frac{{p_{col}}^{M-1}-{p_{col}}^{M+f}}{1-p_{col}} \Bigg) \Bigg)^{-1}.
\end{aligned}
\tag{A.22}
$$

Bibliography

[1] United Nations, "World's population increasingly urban with more than half living in urban areas." http://www.un.org/en/development/desa/news/population/world-urbanization-prospects-2014.html. Accessed: 2018-01-30.

[2] S. Ezell, "Explaining international IT application leadership: Intelligent transportation systems," *The Information Technology and Innovation Foundation (ITIF)*, vol. 1, 2010.

[3] A. Thaduri, A. K. Verma, and U. Kumar, *Analytics for Maintenance of Transportation in Smart Cities*, pp. 81–91. Singapore: Springer Singapore, 2018.

[4] K. Zhang and S. Batterman, "Air pollution and health risks due to vehicle traffic," *Science of the Total Environment*, vol. 450, pp. 307–316, 2013.

[5] M. Obaidat and P. Nicopolitidis, "Smart cities and homes key enabling technologies," in *Smart Cities and Homes, 1st edition*, ch. 1, Elsevier: Elsevier, 2016.

[6] R. G. Hollands, "Will the real smart city please stand up?," *City*, vol. 12, no. 3, pp. 303–320, 2008.

[7] T. Nam and T. A. Pardo, "Conceptualizing smart city with dimensions of technology, people, and institutions," in *Proceedings of the 12th Annual International Digital Government Research Conference: Digital Government Innovation in Challenging Times*, dg.o '11, (New York, NY, USA), pp. 282–291, ACM, 2011.

[8] R. Giffinger and N. Pichler-Milanović, *Smart cities: Ranking of European medium-sized cities*. Centre of Regional Science, Vienna University of Technology, 2007.

[9] C. Harrison, B. Eckman, R. Hamilton, P. Hartswick, J. Kalagnanam, J. Paraszczak, and P. Williams, "Foundations for smarter cities," *IBM Journal of Research and Development*, vol. 54, no. 4, pp. 1–16, 2010.

[10] D. Washburn, U. Sindhu, S. Balaouras, R. A. Dines, N. Hayes, and L. E. Nelson, "Helping CIOs understand smart city initiatives," *Growth*, vol. 17, no. 2, pp. 1–17, 2009.

[11] United Nations, "IEEE Smart Cities." http://www.un.org/en/development/desa/news/population/world-urbanization-prospects-2014.html. Accessed: 2018-01-30.

[12] J. Lee, B. Bagheri, and H.-A. Kao, "A cyber-physical systems architecture for industry 4.0-based manufacturing systems," *Manufacturing Letters*, vol. 3, pp. 18–23, 2015.

[13] "Approved IEEE standard for wireless access in vehicular environments (WAVE) - networking services," *IEEE Approved Std P1609.3/D23.pdf, Feb 2007*, 2007.

[14] "IEEE approved draft standard for wireless access in vehicular environments (WAVE) - networking services," *IEEE P1609.3v3/D6, November 2015*, pp. 1–162, Jan 2016.

[15] Dept. of Energy /Energy Information Administration, U.S., "Annual energy outlook 2008 with projections to 2030." https://www.procon.org/files/eia.outlook.pdf, 2008. Accessed: 2018-04-01.

[16] Energy Information Administration, "Energy Consumption by Sector." https://www.eia.gov/totalenergy/data/monthly/pdf/sec2.pdf, 2015. Accessed: 2018-04-01.

[17] E. Ericsson, H. Larsson, and K. Brundell-Freij, "Optimizing route choice for lowest fuel consumption-potential effects of a new driver support tool," *Transportation Research Part C: Emerging Technologies*, vol. 14, no. 6, pp. 369–383, 2006.

[18] A. Elbery, H. Rakha, M. Y. ElNainay, and M. A. Hoque, "An integrated architecture for simulation and modeling of small-and medium-sized transportation and communication networks," in *International Conference on Smart Cities and Green ICT Systems*, pp. 282–303, Springer, 2015.

[19] A. Elbery, H. Rakha, M. Elnainay, W. Drira, and F. Filali, "Eco-routing using v2i communication: System evaluation," in *Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on*, pp. 71–76, IEEE, 2015.

[20] J. Du, H. A. Rakha, A. Elbery, and M. Klenk, "Microscopic simulation and calibration of a large-scale metropolitan network: Issues and proposed solutions," in *Transportation Research Board 96th Annual Meeting, TRB2018*, 2018.

[21] A. Elbery, F. Dvorak, J. Du, H. A. Rakha, and M. Klenk, "Large-scale agent-based multimodal modeling of transportation networks - system model and preliminary results," in *Proceedings of the 4th International Conference on Vehicle Technology and Intelligent Transport Systems - Volume 1: VEHITS,*, pp. 103–112, INSTICC, SciTePress, 2018.

[22] A. Elbery and H. A. Rakha, "A novel stochastic linear programming feedback eco-routing traffic assignment system," in *Transportation Research Board 96th Annual Meeting, TRB2017*, 2017.

[23] A. Elbery, H. Rakha, M. Y. ElNainay, W. Drira, and F. Filali, "Eco-routing: An ant colony based approach.," in *VEHITS*, pp. 31–38, 2016.

[24] W. L. Jin and W. W. Recker, "An analytical model of multihop connectivity of inter-vehicle communication systems," *IEEE Transactions on Wireless Communications*, vol. 9, pp. 106–112, January 2010.

[25] M. Khabazian and M. K. M. Ali, "A performance modeling of connectivity in vehicular ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 57, pp. 2440–2450, July 2008.

[26] K. A. Hafeez, Z. Lian, L. Zaiyi, and B. N. Ma, "Impact of mobility on VANETs' safety applications," in *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, pp. 1–5.

[27] T. ElBatt, S. K. Goel, G. Holland, H. Krishnan, and J. Parikh, "Cooperative collision warning using dedicated short range wireless communications," in *Proceedings of the 3rd International Workshop on Vehicular Ad Hoc Networks*, VANET '06, (New York, NY, USA), pp. 1–9, ACM, 2006.

[28] Q. Xu, T. Mak, J. Ko, and R. Sengupta, "Medium access control protocol design for vehicle ndash;vehicle safety messages," *IEEE Transactions on Vehicular Technology*, vol. 56, pp. 499–518, March 2007.

[29] S. Dashtinezhad, T. Nadeem, B. Dorohonceanu, C. Borcea, P. Kang, and L. Iftode, "TrafficView: A driver assistant device for traffic monitoring based on car-to-car communication," in *2004 IEEE 59th Vehicular Technology Conference. VTC 2004-Spring (IEEE Cat. No.04CH37514)*, vol. 5, pp. 2946–2950 Vol.5, May 2004.

[30] T. Nadeem, S. Dashtinezhad, C. Liao, and L. Iftode, "TrafficView: Traffic data dissemination using car-to-car communication," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 8, no. 3, pp. 6–19, 2004.

[31] "IEEE Guide for Wireless Access in Vehicular Environments (WAVE) - Architecture," *IEEE Std 1609.0-2013*, pp. 1–78, March 2014.

[32] "IEEE standard for telecommunications and information exchange between systems - LAN/MAN specific requirements - part 11: Wireless medium access control (MAC) and physical layer (PHY) specifications: High speed physical layer in the 5 GHz band," *IEEE Std 802.11a-1999*, pp. 1–102, Dec 1999.

[33] "IEEE standard for information technology– local and metropolitan area networks– specific requirements– part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications amendment 6: Wireless access in vehicular environments," *IEEE Std 802.11p-2010 (Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008, IEEE Std 802.11r-2008, IEEE Std 802.11y-2008, IEEE Std 802.11n-2009, and IEEE Std 802.11w-2009)*, pp. 1–51, July 2010.

[34] "IEEE standard for information technology–local and metropolitan area networks–specific requirements–part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications - amendment 8: Medium access control (MAC) quality of service enhancements," *IEEE Std 802.11e-2005 (Amendment to IEEE Std 802.11, 1999 Edition (Reaff 2003)*, pp. 1–212, Nov 2005.

[35] K. R. Fall and W. R. Stevens, *TCP/IP illustrated, Volume 1: The protocols*. Addison-Wesley, 2011.

[36] "RFC0793: Transmission control protocol," *Internet Engineering Task Force (IETF), Tech. Rep*, 1981.

[37] M. Cello, C. Degano, M. Marchese, and F. Podda, "Chapter 14 - smart transportation systems (stss) in critical conditions," in *Smart Cities and Homes* (M. S. Obaidat and P. Nicopolitidis, eds.), pp. 291 – 322, Boston: Morgan Kaufmann, 2016.

[38] D. R. Choffnes and F. E. Bustamante, "An integrated mobility and traffic model for vehicular wireless networks," in *Proceedings of the 2nd ACM International Workshop on Vehicular ad hoc Networks*, pp. 69–78, ACM.

[39] K. Ibrahim and M. C. Weigle, "ASH: Application-aware swans with highway mobility," in *INFOCOM Workshops*, pp. 1–6.

[40] S. Y. Wang and C. L. Chou, "NCTUns tool for wireless vehicular communication network researches," *Simulation Modelling Practice and Theory*, vol. 17, no. 7, pp. 1211–1226, 2009.

[41] C. Lochert, A. Barthels, A. Cervantes, M. Mauve, and M. Caliskan, "Multiple simulator interlinking environment for IVC," in *Proceedings of the 2Nd ACM International Workshop on Vehicular Ad Hoc Networks*, VANET '05, (New York, NY, USA), pp. 87–88, ACM, 2005.

[42] M. Piorkowski, M. Raya, A. L. Lugo, P. Papadimitratos, M. Grossglauser, and J.-P. Hubaux, "TraNS: Realistic joint traffic and network simulator for VANETs," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 12, no. 1, pp. 31–33, 2008.

[43] C. Sommer, R. German, and F. Dressler, "Bidirectionally coupled network and road traffic simulation for improved IVC analysis," *Mobile Computing, IEEE Transactions on*, vol. 10, no. 1, pp. 3–15, 2011.

[44] Pigne, x, Y., G. Danoy, and P. Bouvry, "A platform for realistic online vehicular network management," in *IEEE GLOBECOM Workshops (GC Wkshps)*, pp. 595–599.

[45] M. Rondinone, J. Maneros, D. Krajzewicz, R. Bauza, P. Cataldi, F. Hrizi, J. Gozalvez, V. Kumar, M. Rckl, and L. Lin, "iTETRIS: A modular simulation platform for the large scale evaluation of cooperative its applications," *Simulation Modelling Practice and Theory*, vol. 34, pp. 99–125, 2013.

[46] M. Barth, K. Boriboonsomsin, and A. Vu, "Environmentally-friendly navigation," in *Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE*, pp. 684–689, IEEE, 2007.

[47] K. Ahn and H. Rakha, "The effects of route choice decisions on vehicle energy consumption and emissions," *Transportation Research Part D: Transport and Environment*, vol. 13, no. 3, pp. 151–167, 2008.

[48] H. A. Rakha, K. Ahn, and K. Moran, "INTEGRATION framework for modeling eco-routing strategies: Logic and preliminary results," *International Journal of Transportation Science and Technology*, vol. 1, no. 3, pp. 259–274, 2012.

[49] K. Boriboonsomsin, M. J. Barth, Z. Weihua, and A. Vu, "Eco-routing navigation system based on multisource historical and real-time traffic information," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 4, pp. 1694–1704, 2012.

[50] A. Kyoungho and H. Rakha, "Field evaluation of energy and environmental impacts of driver route choice decisions," in *Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE*, pp. 730–735.

[51] H. Rakha, "INTEGRATION Rel. 2.40 for Windows - User's Guide, Volume 1: Fundamental model features." https://sites.google.com/a/vt.edu/hrakha/software. Accessed: Dec. 2017.

[52] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms second edition*, ch. Chapter 24: Single-Source Shortest Paths, Section 24.3: Dijkstra's algorithm. The MIT Press, 2001.

[53] C. F. Minett, A. M. Salomons, W. Daamen, B. van Arem, and S. Kuijpers, "Eco-routing: Comparing the fuel consumption of different routes between an origin and destination using field test speed profiles and synthetic speed profiles," in *2011 IEEE Forum on Integrated and Sustainable Transportation Systems*, pp. 32–39, June 2011.

[54] Y. Nie and Q. Li, "An eco-routing model considering microscopic vehicle operating conditions," *Transportation Research Part B: Methodological*, vol. 55, no. 0, pp. 154–170, 2013.

[55] H. Rakha, K. Ahn, and A. Trani, "Development of VT-Micro model for estimating hot stabilized light duty vehicle and truck emissions," *Transportation Research Part D: Transport and Environment*, vol. 9, no. 1, pp. 49–74, 2004.

[56] H. Rakha, K. Ahn, I. El-Shawarby, and S. Jang, "Emission model development using in-vehicle on-road emission measurements," in *Annual Meeting of the Transportation Research Board, Washington, DC*, vol. 2, 2004.

[57] K. Ahn, H. Rakha, A. Trani, and M. Van Aerde, "Estimating vehicle fuel consumption and emissions based on instantaneous speed and acceleration levels," *Journal of Transportation Engineering*, vol. 128, no. 2, pp. 182–190, 2002.

[58] H. Rakha, I. Lucic, S. H. Demarchi, J. R. Setti, and M. V. Aerde, "Vehicle dynamics model for predicting maximum truck acceleration levels," *Journal of Transportation Engineering*, vol. 127, no. 5, pp. 418–425, 2001.

[59] H. Rakha and I. Lucic, "Variable power vehicle dynamics model for estimating truck accelerations," *Journal of Transportation Engineering*, vol. 128, no. 5, pp. 412–419, 2002.

[60] K. Nagel, M. Rickert, and C. L. Barrett, "Large scale traffic simulations," in *International Conference on Vector and Parallel Processing*, pp. 380–402, Springer, 1996.

[61] R. White and G. Engelen, "Cellular automata and fractal urban form: A cellular modelling approach to the evolution of urban land-use patterns," *Environment and Planning A*, vol. 25, no. 8, pp. 1175–1199, 1993.

[62] N. Cetin, K. Nagel, B. Raney, and A. Voellmy, "Large-scale multi-agent transportation simulations," *Computer Physics Communications*, vol. 147, no. 1-2, pp. 559–564, 2002.

[63] B. Raney, N. Cetin, A. Völlmy, M. Vrtic, K. Axhausen, and K. Nagel, "An agent-based microsimulation model of swiss travel: First results," *Networks and Spatial Economics*, vol. 3, no. 1, pp. 23–41, 2003.

[64] M. Balmer, K. Nagel, and B. Raney, "Large-scale multi-agent simulations for transportation applications," in *Intelligent Transportation Systems*, vol. 8, pp. 205–221, Taylor & Francis, 2004.

[65] Y. Zhao and A. W. Sadek, "Large-scale agent-based traffic micro-simulation: Experiences with model refinement, calibration, validation and application," *Procedia Computer Science*, vol. 10, pp. 815–820, 2012.

[66] L. Guo, S. Huang, and A. W. Sadek, "An evaluation of environmental benefits of time-dependent green routing in the greater buffalo–niagara region," *Journal of Intelligent Transportation Systems*, vol. 17, no. 1, pp. 18–30, 2013.

[67] W. Burghout, H. Koutsopoulos, and I. Andreasson, "Hybrid mesoscopic-microscopic traffic simulation," *Transportation Research Record: Journal of the Transportation Research Board*, no. 1934, pp. 218–255, 2005.

[68] W. Burghout and J. Wahlstedt, "Hybrid traffic simulation with adaptive signal control," *Transportation Research Record: Journal of the Transportation Research Board*, no. 1999, pp. 191–197, 2007.

[69] Q. Yang and D. Morgan, "Hybrid traffic simulation model," in *Transportation Research Board 85th Annual Meeting*, no. 06-2582, 2006.

[70] R. Balakrishna, D. Morgan, H. Slavin, and Q. Yang, "Large-scale traffic simulation tools for planning and operations management," *IFAC Proceedings Volumes*, vol. 42, no. 15, pp. 117–122, 2009.

[71] K. Ahn, H. A. Rakha, and K. Moran, "System-wide impacts of eco-routing strategies on large-scale networks," in *Transportation Research Board 91st Annual Meeting*, no. 12-1638, 2012.

[72] K. Ahn and H. A. Rakha, "Network-wide impacts of eco-routing strategies: a large-scale case study," *Transportation Research Part D: Transport and Environment*, vol. 25, pp. 119–130, 2013.

[73] D. Zehe, A. Knoll, W. Cai, and H. Aydt, "Semsim cloud service: Large-scale urban systems simulation in the cloud," *Simulation Modelling Practice and Theory*, vol. 58, pp. 157–171, 2015.

[74] M. Balmer, M. Rieser, K. Meister, D. Charypar, N. Lefebvre, and K. Nagel, "MATSim-T: Architecture and simulation times," in *Multi-Agent Systems for Traffic and Transportation Engineering*, pp. 57–78, IGI Global, 2009.

[75] "User datagram protocol," *Internet Engineering Task Force (IETF), Tech. Rep*, 1980.

[76] M. Li, Z. Yang, and W. Lou, "Codeon: Cooperative popular content distribution for vehicular networks using symbol level network coding," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 1, pp. 223–235, 2011.

[77] G. C. Bruner and A. Kumar, "Attitude toward location-based advertising," *Journal of interactive advertising*, vol. 7, no. 2, pp. 3–15, 2007.

[78] T. H. van den Broek, J. Ploeg, and B. D. Netten, "Advisory and autonomous cooperative driving systems," in *Consumer Electronics (ICCE), 2011 IEEE International Conference on*, pp. 279–280, IEEE, 2011.

[79] L. D. Baskar, B. De Schutter, and H. Hellendoorn, "Hierarchical traffic control and management with intelligent vehicles," in *Intelligent Vehicles Symposium, 2007 IEEE*, pp. 834–839, IEEE, 2007.

[80] Y. Xiao, Q. Zhao, I. Kaku, and Y. Xu, "Development of a fuel consumption optimization model for the capacitated vehicle routing problem," *Computers & Operations Research*, vol. 39, no. 7, pp. 1419–1431, 2012.

[81] A. Talebpour, H. Mahmassani, and S. Hamdar, "Speed harmonization: evaluation of effectiveness under congested conditions," *Transportation Research Record: Journal of the Transportation Research Board*, no. 2391, pp. 69–79, 2013.

[82] S. Roy, R. Sen, S. Kulkarni, P. Kulkarni, B. Raman, and L. K. Singh, "Wireless across road: RF based road traffic congestion detection," in *Communication Systems and Networks (COMSNETS), 2011 Third International Conference on*, pp. 1–6, IEEE, 2011.

[83] M. Hoque, X. Hong, and B. Dixon, "Innovative taxi hailing system using DSRC infrastructure," in *ITS America 22nd Annual Meeting & Exposition*, 2012.

[84] M. Alam, M. Sher, and S. A. Husain, "VANETs mobility model entities and its impact," in *4th International Conference on Emerging Technologies, ICET.*, pp. 132–137.

[85] M. A. Hoque, X. Hong, and B. Dixon, "Efficient multi-hop connectivity analysis in urban vehicular networks," *Vehicular Communications*, vol. 1, no. 2, pp. 78–90, 2014.

[86] Technology Riverbed Technology, "https://www.riverbed.com/products/steelcentral/opnet.html," Accessed Dec. 2017.

[87] C. Perkins, E. Belding-Royer, and S. Das, "RFC 3561: ad-hoc on-demand distance vector (AODV) routing," *Internet Engineering Task Force (IETF), Tech. Rep*, pp. 1–38, 2003.

[88] M. Van Aerde and H. Rakha, "Multivariate calibration of single regime speed-flow-density relationships," in *Proceedings of the 6th Vehicle Navigation and Information Systems Conference*, pp. 334–341.

[89] K. Fall, "A delay-tolerant network architecture for challenged internets," in *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, (863960), pp. 27–34, ACM.

[90] A. Lindgren, A. Doria, and O. Schel, "Probabilistic routing in intermittently connected networks," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 7, no. 3, pp. 19–20, 2003.

[91] C. Guo, B. Yang, O. Andersen, C. S. Jensen, and K. Torp, "EcoMark 2.0: Empowering eco-routing with vehicular environmental models and actual vehicle fuel consumption data," *GeoInformatica*, pp. 1–33, 2014.

[92] D. Jiang, V. Taliwal, A. Meier, W. Holfelder, and R. Herrtwich, "Design of 5.9 GHz DSRC-based vehicular safety communication," *Wireless Communications, IEEE*, vol. 13, no. 5, pp. 36–43, 2006.

[93] J. Yin, T. ElBatt, G. Yeung, B. Ryu, S. Habermas, H. Krishnan, and T. Talty, "Performance evaluation of safety applications over DSRC vehicular ad hoc networks," in *Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks*, pp. 1–9, ACM, 2004.

[94] K. A. Hafeez, Z. Lian, L. Zaiyi, and B. N. Ma, "A new broadcast protocol for vehicular ad hoc networks safety applications," in *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, pp. 1–5.

[95] K. A. Hafeez, L. Zhao, B. Ma, and J. W. Mark, "Performance analysis and enhancement of the DSRC for VANET's safety applications," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 7, pp. 3069–3083, 2013.

[96] G. K. Mitropoulos, I. S. Karanasiou, A. Hinsberger, F. Aguado-Agelet, H. Wieker, H.-J. Hilt, S. Mammar, and G. Noecker, "Wireless local danger warning: Cooperative foresighted driving using intervehicle communication," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 3, pp. 539–553, 2010.

[97] R. Bauza, J. Gozalvez, and J. Sanchez-Soriano, "Road traffic congestion detection through cooperative vehicle-to-vehicle communications," in *Local Computer Networks (LCN), 2010 IEEE 35th Conference on*, pp. 606–612, IEEE, 2010.

[98] A. Lakas and M. Cheqfah, "Detection and dissipation of road traffic congestion using vehicular communication," in *Microwave Symposium (MMS), 2009 Mediterrannean*, pp. 1–6, IEEE, 2009.

[99] P. A. Gagniuc, *Markov Chains: From Theory to Implementation and Experimentation*. John Wiley & Sons, 2017.

[100] P. Brémaud, *Markov Chains: Gibbs Fields, Monte Carlo Simulation, and Queues*, vol. 31. Springer Science & Business Media, 2013.

[101] E. Behrends, *Introduction to Markov Chains: With Special Emphasis on Rapid Mixing, Chapter 5: Transient states*, pp. 36–41. Wiesbaden: Vieweg+Teubner Verlag, 2000.

[102] E. Vigoda, "Markov chains, coupling, stationary distribution." https://www.cc.gatech.edu/~mihail/MC_basics_Vigoda_Lec6.pdf, 2003. Accessed: January 2108.

[103] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 535–547, 2000.

[104] I. Tinnirello, G. Bianchi, and Y. Xiao, "Refinements on ieee 802.11 distributed coordination function modeling approaches," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 3, pp. 1055–1067, 2010.

[105] J. Zheng and Q. Wu, "Performance modeling and analysis of the IEEE 802.11p EDCA mechanism for VANET," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 4, pp. 2673–2687, 2016.

[106] C. E. Weng and H. C. Chen, "The performance evaluation of IEEE 802.11 DCF using Markov chain model for wireless LANs," *Computer Standards & Interfaces*, vol. 44, pp. 144–149, 2016.

[107] N. Hajlaoui, I. Jabri, and M. B. Jemaa, "An accurate two dimensional Markov chain model for IEEE 802.11n DCF," *Wireless Networks*, pp. 1–13, 2016.

[108] P. Chatzimisios, A. C. Boucouvalas, and V. Vitsas, "Performance analysis of IEEE 802.11 DCF in presence of transmission errors," in *2004 IEEE International Conference on Communications*, vol. 7, pp. 3854–3858, IEEE, 2004.

[109] S. Eichler, "Performance evaluation of the IEEE 802.11p WAVE communication standard," in *Vehicular Technology Conference, 2007. VTC-2007 Fall. 2007 IEEE 66th*, pp. 2199–2203, IEEE, 2007.

[110] H. Chen, "Revisit of the markov model of IEEE 802.11 DCF for an error-prone channel," *IEEE Communications Letters*, vol. 15, no. 12, pp. 1278–1280, 2011.

[111] C. Han, M. Dianati, R. Tafazolli, R. Kernchen, and X. Shen, "Analytical study of the IEEE 802.11p MAC sublayer in vehicular networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 2, pp. 873–886, 2012.

[112] P. E. Engelstad and N. Olav, "Non-saturation and saturation analysis of IEEE 802.11e EDCA with starvation prediction," in *Proceedings of the 8th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pp. 224–233, ACM, 2005.

[113] L. Kleinrock, *Queuing Systems, Volume 2: Computer Applications*, vol. 66. wiley New York, 1976.

[114] D. Helbing, "Derivation of a fundamental diagram for urban traffic flow," *The European Physical Journal B-Condensed Matter and Complex Systems*, vol. 70, no. 2, pp. 229–241, 2009.

[115] J. I. Levy, J. J. Buonocore, and K. von Stackelberg, "Evaluation of the public health impacts of traffic congestion: A health risk assessment," *Environmental Health*, vol. 9, p. 65, Oct 2010.

[116] S. Boniface, R. Scantlebury, S. Watkins, and J. Mindell, "Health implications of transport: evidence of effects of transport on social interactions," *Journal of Transport & Health*, vol. 2, no. 3, pp. 441–446, 2015.

[117] C. Osorio and K. K. Selvam, "Solving large-scale urban transportation problems by combining the use of multiple traffic simulation models," *Transportation Research Procedia*, vol. 6, pp. 272–284, 2015.

[118] C. Zhang, C. Osorio, and G. Flötteröd, "Efficient calibration techniques for large-scale traffic simulators," *Transportation Research Part B: Methodological*, vol. 97, pp. 214–239, 2017.

[119] N. Cetin and K. Nagel, "Parallel queue model approach to traffic microsimulations," Proceedings of Swiss Transportation Research Conference, STRC 2002.

[120] J. W. Videla A., *RabbitMQ in action: Distributed messaging for everyone*. Manning Publications, 2012.

[121] J. L. Fernandes, I. C. Lopes, J. J. Rodrigues, and S. Ullah, "Performance evaluation of RESTful web services and AMQP protocol," in *Ubiquitous and Future Networks (ICUFN), 2013 Fifth International Conference on*, pp. 810–815, IEEE, 2013.

[122] F. Dion, H. Rakha, and Y.-S. Kang, "Comparison of delay estimates at under-saturated and over-saturated pre-timed signalized intersections," *Transportation Research Part B: Methodological*, vol. 38, no. 2, pp. 99–122, 2004.

[123] M. M.H., *Simulating Computer Systems: Techniques and Tools*. MIT press, 1989.

[124] J. Lerner, D. Wagner, and K. A. Zweig, eds., *Algorithmics of Large and Complex Networks: Design, Analysis, and Simulation*. Berlin, Heidelberg: Springer-Verlag, 2009.

[125] W. Zeng and R. L. Church, "Finding Shortest Paths on Real Road Networks: The Case for A*," *Int. J. Geogr. Inf. Sci.*, vol. 23, pp. 531–543, Apr. 2009.

[126] M. Aerde, H. Rakha, and H. Paramahamsan, "Estimation of origin-destination matrices: Relationship between practical and theoretical considerations," *Transportation Research Record: Journal of the Transportation Research Board*, no. 1831, pp. 122–130, 2003.

[127] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Computational Intelligence Magazine*, vol. 1, pp. 28–39, Nov 2006.

[128] H. Rakha, K. Ahn, and K. Moran, "Integration framework for modeling eco-routing strategies: Logic and preliminary results," *International Journal of Transportation Science and Technology*, vol. 1, no. 3, pp. 259–274, 2012.

[129] C. Blum, "Ant colony optimization: Introduction and recent trends," *Physics of Life Reviews*, vol. 2, no. 4, pp. 353 – 373, 2005.