

2014-07-11

# Connected Vehicle Extension and Integration of Traffic and Discrete Event Simulation Systems -- Applied to Evaluations Based on Dedicated Short Range Communication for Safety and Mobility Indices

Paikari, Elahe

---

Paikari, E. (2014). Connected Vehicle Extension and Integration of Traffic and Discrete Event Simulation Systems -- Applied to Evaluations Based on Dedicated Short Range Communication for Safety and Mobility Indices (Master's thesis, University of Calgary, Calgary, Canada).

Retrieved from <https://prism.ucalgary.ca>. doi:10.11575/PRISM/25417

<http://hdl.handle.net/11023/1625>

*Downloaded from PRISM Repository, University of Calgary*

UNIVERSITY OF CALGARY

Connected Vehicle Extension and Integration of Traffic and Discrete Event Simulation Systems

Applied to Evaluations Based on Dedicated Short Range Communication for Safety and Mobility Indices

by

Elahe Paikari

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE

DEGREE OF MASTER OF SCIENCE

GRADUATE PROGRAM IN ELECTRICAL AND COMPUTER ENGINEERING

CALGARY, ALBERTA

JULY, 2014

© Elahe Paikari 2014

## Abstract

This study is aimed at developing and modeling a specific extension of Connected Vehicles (CV) system and its applications in the Intelligent Transportation System (ITS) through the designated traffic and wireless simulation networks. A typical traffic micro-simulator and a discrete event simulator individually lack the ability of fully capturing the behavior of the CV system. In this research I investigate modeling the CV system, for Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communications based on Dedicated Short Range Communication (DSRC) by enabling the two simulators communicate sequentially. PARAMICS is selected as the traffic micro-simulator. OPNET is used as the discrete event simulator.

The contributions are: (1) designing the CV system as a Multiagent System (MAS) using MaSE methodology and implementing its outcomes as extensions for the PARAMICS using two distinctive APIs (Application Programming Interface); and (2) developing the integration of PARAMICS and OPNET for implementation and evaluation of DSRC-based vehicular communication protocols and their utilizations in the context of ITS.

The results are verified through experiments that demonstrate the overall effectiveness of the CV. Three case studies are presented: impacts of CV on improving (1) traffic safety and (2) mobility on a section of Deerfoot trail, Calgary, Alberta; (3) the optimum selection of DSRC communication range of Road Side Units (RSUs) and a definitive percentage of CVs in the network to have the least data loss and delay in V2V and V2I data transmission for weekdays' traffic counts.

## Publications

Part of materials, ideas, tables, and figures used in this thesis have appeared previously in one of the followings:

### Refereed Papers

- **E. Paikari**, L. Kattan, S. Tahmasseby, and B. H. Far, "Modeling and simulation of advisory speed and re-routing strategies in connected vehicles systems for crash risk and travel time reduction," in Electrical and Computer Engineering (CCECE), 2013 26th Annual IEEE Canadian Conference on, 2013, pp. 1-4.
- **E. Paikari**, S. Tahmasseby, and B. H. Far, "A Simulation-based Benefit Analysis of Deploying Connected Vehicles Using Dedicated Short Range Communication," in Intelligent Vehicles (IV) Symposium, 2014 IEEE Intelligent Transportation Systems Society, 2014, pp. 1-6.
- **E. Paikari** and B. H. Far, "Analysis, Design and Implementation of an Agent Based System for Simulating Connected Vehicles," in Software Engineering and Knowledge Engineering (SEKE), The 26th International Conference on, 2014, pp. 1-6.
- **E. Paikari**, M. Moshirpour, R. Alhajj, and B. H. Far, "Data Integration and Clustering for Real Time Crash Prediction," in Information Reuse and Integration (IRI), The 15th IEEE International Conference on, 2014, pp. 1-8.

### Report

- S. Tahmasseby, **E. Paikari**, L. Kattan, B. Far, A. Radmanesh, and M. Moussavi, "Modelling and assessing the benefits of Connected Vehicle initiative in a micro-simulation environment," Unpublished report for Encom Wireless Data Solutions Inc., NSERC Engage Grant, 2013.

## Acknowledgements

I would like to thank my supervisor, Dr. Behrouz H. Far, for his great supervision, insightful comments and continuous support during my master studies and research. His guidance made all of these possible. I also thank my co-supervisor, Dr. Lina Kattan, for her expert advices and comments on my thesis.

I thank Dr. Shahram Tahmasseby for his contribution in: implementing the crash risk model, designing the experiments, conducting the sensitivity and statistical analysis, and interpreting the results.

I would like to thank Dr. Mahmoud Moussavi and Dr. Ahmad Radmanesh for their assistances towards parts of this research. I also thank the members of my MSc committee Dr. Mahmoud Moussavi, Dr. Vassil Dimitrov and Dr. Mozart Batista de Castro Menezes for their valuable comments.

Last but not least, I express my gratitude to my sister, Elham Paikari, for her encouragement and guidance along every step of this way.

This research was partially funded by NSERC-Engage with ENCOM Wireless Data Solutions Inc., NSERC Discovery and AITF-AMA collaborative grants.

### **Contributions**

My original contributions in this thesis are:

1. Analysis and design of CV system as the MAS using MaSE methodology
2. Implementation of CV system using APIs :
  - Revising the existing codes and removing bugs (such as improving the speed of simulation runs, optimizing codes to remove several useless functions)
  - Adding codes to API #1 to create predefined incidents
  - Adding codes to API #2 to simulate infrastructures (RSU, VMS) and then developing V2I communication

- Programming new functions to randomly change the value of aggressiveness and awareness in a predefined threshold (adjusting driver's behaviors), to implement advisory speed and re-routing guidance for upstream and downstream of incident.
3. Integration of PARMICS and OPNET :
    - Importing vehicular mobility traces into wireless network simulator
    - Presentation of the optimum selection of DSRC communication range of RSUs and ultimate percentage of CVs in the network to have the least data loss and delay in V2V and V2I data transmission by simulating one scenario
  4. Evaluation of the impacts of V2V and V2I communications on improving safety and mobility applications by simulating two scenarios on Deerfoot trail, Calgary, Alberta (CCECE'2013 and IV'2014 papers): I was responsible for programming, developing, running examined simulations (150 runs for every paper), calculating results for IV'2014 paper (crash risks, travel time), and writing the draft and final versions of both papers.
  5. NSERC Engage report for ENCOM: I was responsible for programming, developing, running examined simulations (150 runs), calculating results (crash risks, travel time), and writing "simulation design and experiment design" section of the report which consists of "basic design", "incident creation" and "detailed design" subsections.

## Dedication

I dedicate this thesis to my parents, Sima and Mahmoud, who always inspire, support and love me. I could not have done it without you.

# Table of Contents

Abstract.....	ii
Publications.....	iii
Acknowledgements.....	iv
Dedication.....	vi
Table of Contents .....	vii
List of Tables.....	x
List of Figures.....	xi
List of Abbreviations.....	xiii
Chapter 1: Introduction.....	1
1.1 Motivation .....	1
1.2 Goal .....	2
1.3 Problems.....	2
1.3.1 Modeling CV system using traffic network.....	2
1.3.2 Modeling CV system using the integration of traffic and wireless networks.....	4
1.4 Methodology .....	4
1.5 Contributions.....	9
1.6 Structure of Thesis .....	9
Chapter 2: Background and Literature Review .....	11
2.1 Introduction .....	11
2.2 Connected Vehicles.....	12
2.3 DSRC .....	14
2.4 Simulation vs. Real World Experiment .....	16
2.5 Traffic Simulators .....	17
2.5.1 PARAMICS .....	22



2.6	Wireless Communication Simulators .....	27
2.6.1	OPNET .....	32
2.7	Overview of Previous Research on Integration of traffic and wireless communication simulators .....	36
2.7.1	Integration of PARAMICS with OPNET .....	39
2.8	Summary .....	40
Chapter 3:	Methodology .....	41
3.1	Introduction .....	41
3.2	MaSE Methodology .....	42
3.3	Analysis and Design of Connected Vehicles in PARAMICS .....	44
3.3.1	ANALYSIS .....	45
3.3.2	DESIGN .....	49
3.4	Implementation using Application Programming Interface .....	53
3.4.1	API #1: Incident creation .....	56
3.4.2	API #2 .....	57
3.5	Simulation Configuration in PARAMICS .....	61
3.6	Safety and Mobility applications.....	64
3.6.1	Safety measurements .....	64
3.6.2	Mobility measurements.....	67
3.6.3	ANOVA test.....	68
3.7	Integration of PARAMICS and OPNET .....	68
3.7.1	Mobility Traces .....	70
3.7.2	Simulation configuration in OPNET .....	73
3.8	Summery .....	76
Chapter 4:	Case Studies .....	78

4.1	Introduction .....	78
4.1.1	Study area and design of experiment .....	79
4.2	Scenario 1 .....	80
4.2.1	Scenario evaluation .....	82
4.2.2	Results .....	83
4.3	Scenario 2 .....	85
4.3.1	Study area and design of experiment .....	85
4.3.2	Scenario evaluation .....	87
4.3.3	Results .....	88
4.3.4	Interpretations of results .....	101
4.4	Scenario 3 .....	102
4.4.1	Study area and design of experiment .....	102
4.4.2	Results .....	104
4.5	Summery .....	108
Chapter 5:	Conclusion and Future works .....	110
5.1	Thesis Summery and Contributions .....	110
5.2	Future Works .....	112
References	.....	114

## List of Tables

Table 2.1 A few different types of traffic simulation models - their main features and capabilities .....	19
Table 2.2 Different comparative studies of simulation software .....	21
Table 2.3 A few Different types of wireless simulation models - their main features and drawbacks .....	29
Table 3.1 Trajectory file includes mobility traces in CSV file .....	72
Table 4.1 Three proposed scenarios accepted in referred conferences .....	79
Table 4.2 The overall ORCI for all 3 test cases [6, 36] .....	90
Table 4.3 ANOVA Test for 60% demand loading [6] .....	92
Table 4.4 ANOVA Test for 60% demand loading [6] .....	94
Table 4.5 ANOVA Test for 80% demand loading [6] .....	96
Table 4.6 ANOVA Test for 80% demand loading [6] .....	98
Table 4.7 ANOVA Test for 100% demand loading [6] .....	100
Table 4.8 The effects of DSRC range for every RSU .....	105
Table 4.9 The effects of percentage of CV on the network parameters .....	107

## List of Figures

Figure 1.1 The layered approach of integrated simulators .....	5
Figure 1.2 Inputs and outputs of integrated simulations - Proposed integration and future works	8
Figure 1.3 Thesis structure.....	10
Figure 2.1 The three-tiered OPNET hierarchy .....	33
Figure 3.1 MaSE phases [119].....	44
Figure 3.2 Goal Hierarchy Diagram .....	46
Figure 3.3 Sequence Diagram [36] .....	47
Figure 3.4 System Role Model .....	48
Figure 3.5 Concurrent Task Diagram .....	49
Figure 3.6 Agent Class Diagram.....	50
Figure 3.7 Create Connection conversation initiator and responder.....	51
Figure 3.8 System Architecture for agent class .....	52
Figure 3.9 System Deployment Diagram.....	53
Figure 3.10 Illustration of the CV simulation developed by PARAMICS APIs [4] .....	55
Figure 3.11 Implemented APIs using CV agents.....	56
Figure 3.12 Extended API for V2I implementation.....	60
Figure 3.13 Travel time indicator in Analyzer module.....	68
Figure 3.14 Converted mobility traces in XML file .....	73
Figure 3.15 Data modeling in two layers of simulation.....	75
Figure 4.1 The real vs. PARAMICS simulated network study area (source of real map: Google Maps)[6].....	80
Figure 4.2 Average travel time comparison between base case (no connected vehicle), 30% and 50% V2V (seconds) .....	83
Figure 4.3 Crash potential for the base case (No CV) and test cases (30%, 50% CV) .....	84
Figure 4.4 The average P2P travel time for all 3 test cases (sec) [6, 36].....	89
Figure 4.5 P2P journey time (sec) in tested network (60% demand load) [6].....	91
Figure 4.6 Crash likelihood prediction (60% demand load) [6] .....	93
Figure 4.7 P2P journey time (sec) (80% loading) [6].....	95
Figure 4.8 Crash likelihood prediction (80% loading) [6].....	97

Figure 4.9 P2P journey time (sec) in tested network (100% demand load) [6].....	99
Figure 4.10 Crash likelihood prediction in tested network (100% demand load) [6].....	101

## List of Abbreviations

ITS	Intelligent Transportation systems
RSU	Road Side Unit
V2I	Vehicle to Infrastructure
VII	Vehicle Infrastructure Integration
V2V	Vehicle to Vehicle
VMS	Variable Message Sign
DSRC	Dedicated Short- Range Communication
OSI	Open systems interconnection
MAC	Medium Access Control
PHY	Physical Layer of OSI model
WAVE	Wide Area Voice Exchange
SAE	System architecture evolution
GUI	Graphical User Interface
API	Application Programming Interface
MANET	Mobile Ad Hoc Network
VANET	Vehicular Ad Hoc Network
OFDM	Orthogonal Frequency Division Multiplexing
GPS	Global Positioning System
MaSE	Multi Agent System Engineering
MAS	Multi Agent System
WLAN	Wireless Local Area Network
WPAN	Wireless Personal Area Networks

OD	Origin Destination
P2P	Point to Point
ICT	Information and Communication Technologies
FCC	The U.S. Federal Communication Commission
ETC	Electronic Toll Collection
USDOT	The United States Department of Transportation
GIS	Geographical Information Systems
CAD	Computer Aided Design
IT	Information Technology
NICs	Network Interface Cards
DCF	Distributed Coordination Function
SINR	Signal to Interference plus Noise Ratio
PLCP	Physical Layer Convergence Procedure
TCL	Tool Command Language
ORCI	Overall Risk Change Index
PSM	Pedestrian Simulation Module
SDK	Software Development Kit
PDS	Probe Data Service
STD	state transition diagram
KP	Kernel Procedure
WAN	Wide Area Network
LAN	Local Area Network
ANOVA	Analysis of Variance

# **Chapter 1: Introduction**

## **1.1 Motivation**

Chain collisions can be potentially eluded, or their severity lessened, by reducing the delay between the time that an emergency event occurs and the time at which the vehicles behind are informed about it [1]. One way to give more time to drivers to react in emergency situations is to develop ITS (Intelligent Transportation System) applications to create CV (Connected Vehicles) systems using wireless communication technology. The primary benefit of such communication is to allow the emergency information to be propagated among vehicles much quicker than a traditional chain of drivers reacting to the brake lights of vehicles right ahead.

CV is a suite of technologies and applications that use varieties of wireless and/or cellular communications and sensor devices to provide faultless connectivity between the vehicles and/or road infrastructure. The main objective of CV systems is to improve safety, mobility and sustainability. The improvements help governments to reduce the number of injuries and fatalities. The Transport Canada's National Collision Database (NCDB) has reported the number of 2,006 for vehicle fatalities and 10,443 for serious injuries in 2011 [2].

CV communication systems are emerging type of vehicular communication networks in which vehicles and roadside units/equipment are the communicating points; providing information, such as traffic events, safety messages and general traffic information. These systems are mainly incorporated into ITS for safety and mobility improvement and traffic congestion mitigation. CV's communication system is categorized as V2V (Vehicle to Vehicle) and V2I (Vehicle to Infrastructure).

V2V is a main component of vehicular communication systems and allows detailed information to be exchanged among the individual vehicles. In V2V, the information related to road traffic and weather conditions at one location can be disseminated from a vehicle to another vehicle, benefits such as collision avoidance, travel time mitigation, and incident or congestion notification, could be achieved [3].



In V2I, vehicles exchange information with RSUs, which are fixed beacons. These beacons act as an interface between the vehicle network and external networks [4]. V2I takes an effective role in improving road safety and mobility through multiple in-vehicle and RSU technologies.

## **1.2 Goal**

In this study, our goals are to apply ITS applications, to develop, implement and demonstrate a DSRC (Dedicated Short Range Communication) based V2Vassisted V2I traffic information system to examine: (1) the effects of CV on reducing crash likelihood and travel time, improving safety and mobility indices, on examined traffic network by using a traffic microsimulator, (2) failure and time lag related to transmission of warning messages in V2V and V2I communications to propose the optimum DSRC range and percentage of CV using the integration of traffic and wireless communication simulators.

These two manners of demonstration cannot be done by using only traffic or wireless communication simulators due to their limitations of simulating CV individually. Wireless network simulators cannot simulate the large scale traffic network to replicate car-following, lane changing and other driver behaviours. Thus our first set of goals, which is improving the safety and mobility indices, urge the use of traffic simulator itself. However, traffic simulators do not have the ability to simulate different communication protocols and network parameters during the implementation of CV. Therefore, in order to reach our second set of goals, which is optimizing the V2V and V2I communications, the integration of these two simulators is created.

We implement and evaluate our deployed approaches using simulation experiments. In the rest, the problems we encountered while doing these two approaches and how we addressed the problems are explained with details. Note that the goal of the project is intended to help companies or city, and people are not the focus of getting result.

## **1.3 Problems**

### **1.3.1 Modeling CV system using traffic network**

In order to fulfil our first goal, which is implementing CV system to improve transportation system safety and mobility indices, we need to select a traffic microsimulator. The reason is that,

the field test on CV system cannot be conducted before the system is actually deployed [5]. Traffic simulation tools have been widely used to examine the impact of deploying various ITS applications on the traffic. These tools enable transport planners to assess different strategies without actual evaluating them in a real traffic network. Moreover, using simulation does not impact public driving and traffic safety [6].

Numbers of researchers investigated on comparing and assessing various traffic simulators [7-11]. From all of the studied simulators, the models AIMSUN [12], PARAMICS [13], and VISSIM [14] are found to be suitable for congested arterials and freeways, and potentially useful for simulating ITS applications [10]. While these packages have many similarities, each has its own specific characteristics that make it more or less suitable for particular modelling purposes. We selected PARAMICS as the traffic network microsimulator to implement CV systems. PARAMICS is a popular suite among universities and government agencies for traffic microsimulation and is capable to simulate and test advanced traffic systems. It is beneficial over other types of microsimulation packages as it lets users to examine their own traffic control scenarios; however, it does not support the simulation of CV and its application is limited when emerging systems such as CV need to be analysed in collision events. Therefore, in order to design our proposed system we encountered two problems:

1. Disability of PARAMICS microsimulator on modeling V2V and V2I connections as the simulation of CV.
2. Limitation of PARAMICS microsimulator on creating predefined incidents to study CV system under emergency situations.

We removed these two issues by using MaSE methodology to design the proposed system as the MAS, and then implemented the specific design as the extension to the PARAMICS by using two APIs. This extension is subsequently used to assess the effects of the CV system on improving safety and mobility applications by reducing incident risks and travel time on Highway 2 (i.e. Deerfoot trail) in Calgary as the examined network.

### **1.3.2 Modeling CV system using the integration of traffic and wireless networks**

Availability and quality of transmitted traffic information in V2V and V2I communications play an essential role on the performance and ability of CV in improving safety and mobility. Equipping vehicles and infrastructure with internet to implement CV system and subsequently providing V2V and V2I communications during the network properly will make these improvements possible. So in order to accomplish our second goal, which is providing vehicular internet access based on DSRC and then investigating the effects of various DSRC ranges and CV penetration rates on critical V2V and V2I network elements such as failure or delay, we need a wireless network simulator. The reason is that, traffic simulator does not have the ability to simulate vehicular internet access and different network protocols, but it is required during CV simulations to represent different traffic networks and their elements such as links, infrastructures and vehicles. Given that, we confronted the third problem:

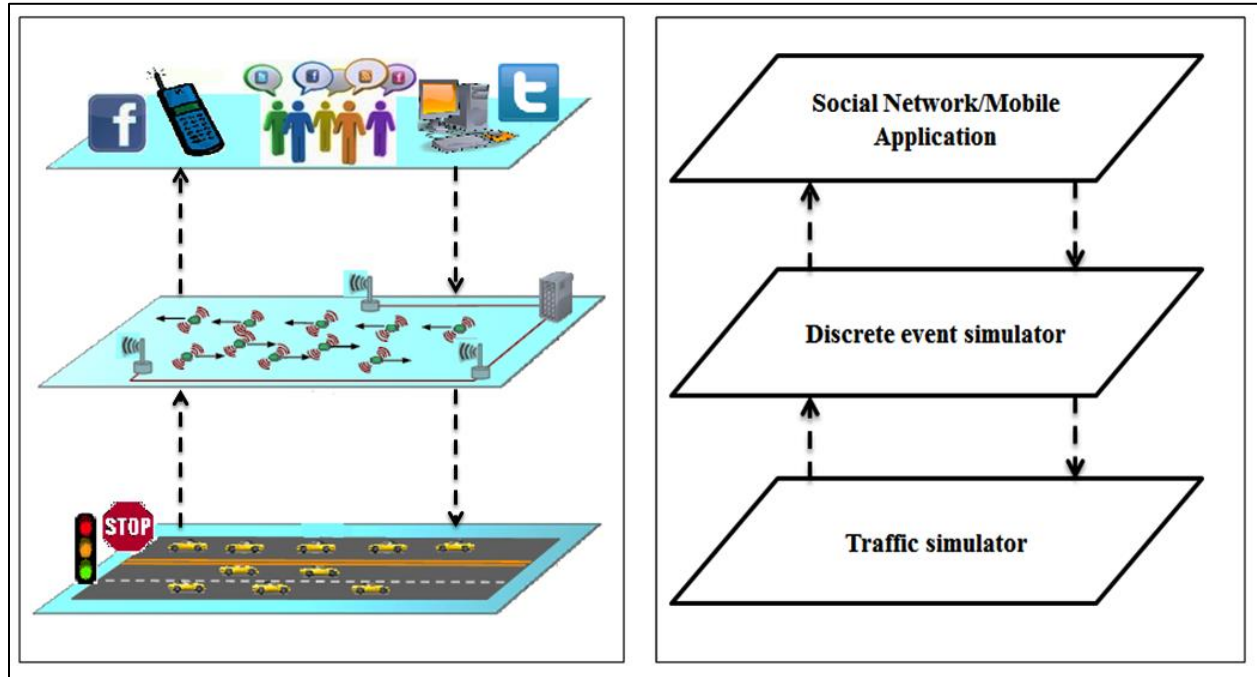
3. Limitation of traffic and wireless network simulators to solely simulate CV and satisfy the second goal.

The proposed CV implementation based on DSRC standard has to deal with several vehicles seeking to connect with RSUs or other vehicles, especially when an emergency event occurs and vehicles and infrastructure transmit several warning messages. This will lead to transmission delays and data losses related to the increased load and data transmission in the network. So in order to reach our second goal, which is the improvements of these elements by optimising the V2V and V2I communications, and surmount the third problem, we integrated traffic and wireless network simulators. We evaluated important issues concerning both network simulators.

## **1.4 Methodology**

In order to solve the proposed problems, we used the layered approach of integrated simulation environments, as shown in Figure 1.1. This study mainly focuses on the first two lower layers; however, by adding the third layer, social network and mobile applications can be used to directly deliver the important results of first two layers to the people. Such approach allows the specific simulation functions, which were developed and validated previously in each simulation layer, to be employed to model particular components of integrated simulators. For instance,

IEEE 802.11 have already implemented and tested in network simulators such as ns-2 and OPNET. Moreover, driver behavior models and CV system have been developed and evaluated in vehicle simulators. Therefore, by integrating these two simulators using layering method, complex procedures for addressing the limitations of these simulators can be produced.



**Figure 1.1 The layered approach of integrated simulators**

Based on that, to achieve the first goal and overcome the first two problems, we designed our proposed CV scenarios using MaSE methodology to view the CV modules as the MAS. CV consists of several intelligent agents which connect and transmit warning messages for estimating and disseminating traffic safety and mobility parameters, being hard for each of them to perform exclusively. Then, by using API, we developed the output design of MaSE method as the extension to the PARAMICS. Development mainly involved adding data structures to different PARAMICS objects and defining functions to process data and produce desired output statistics. Given that, CV system was implemented as extensions to PARAMICS by using two APIs. API #1 and API #2, which are considered separately in the simulation framework because of their various functions. API#1 is programmed to create incidents in the examined network in PARAMICS. API #2 then attempts to increase driver awareness of the upstream and downstream vehicles and thus diminish the effects of the generated incidents by disseminating information

through V2V and V2I communications. The simulated test network represents an 8-km southbound section of Deerfoot Trail (Highway 2) in Calgary, Alberta, Canada. This network is calibrated to replicate traffic counts obtained from Alberta Transportation [6].

Two scenarios have been made to evaluate the extended simulation system, to examine the implementation of advisory speed recommendation and re-routing guidance for urban freeways under emergency situations, and to recommend the optimum treatments and reduce rear-end and lane-change crash risks. We use CV strategies as a tool for safety and mobility improvements by using DSRC based V2V and V2I systems.

Safety and mobility applications were calculated by comparing the simulation results of following measures of CV scenarios with base case scenarios, where there is no CV:

4. Safety benefits measured as the ORCI [15] and crash likelihood [16] indices for rear-end and lane change accidents;
5. Mobility benefits measured as the average travel time (seconds) in a specific route.

The results of examined scenarios showed that the first goal was satisfied and deploying CV will improve the overall safety and mobility factors under various congestion levels and load conditions in the examined network. However the results were shown to be highly sensitive to the % penetration of CV. Mixed results are obtained at higher % of CV.

Although few attempts were made to model CV applications and assess their benefits in a microsimulation environment such as [17-20], this approach is the first which adds APIs to extend the abilities of PARAMICS to simulate CV to reach three sub-goals: first, evaluating both V2V and V2I communications together in examined scenarios, second, considering DSRC standard as the vehicular communication range, and third, investigating the impacts of advisory speed and re-routing guidance in CV system. All of these cases are required to be done together in the simulated network to get the optimum safety and mobility improvements. Additionally, despite of other studies, by implementing incident events, we assessed our proposed system by closely studying the effectiveness of CV, in incident situations, on reducing the duration of travel time and the amount of crash risks in a simulation environment.

In the next part, we combined the capabilities of a traffic network simulator together with wireless communication functionalities, as the discrete event simulator, to reach the second goal

and dispel the third problem. Wireless network experiments are not always feasible in real world especially when evaluations demand costly, temporary and non-scalable equipment to be installed. Simulation is an effective alternative in our study since we aim to detect pattern variations of crucial elements like message drops and latency in V2V and V2I communications, where different values for DSRC range and percentage of CV should be tested in the network.

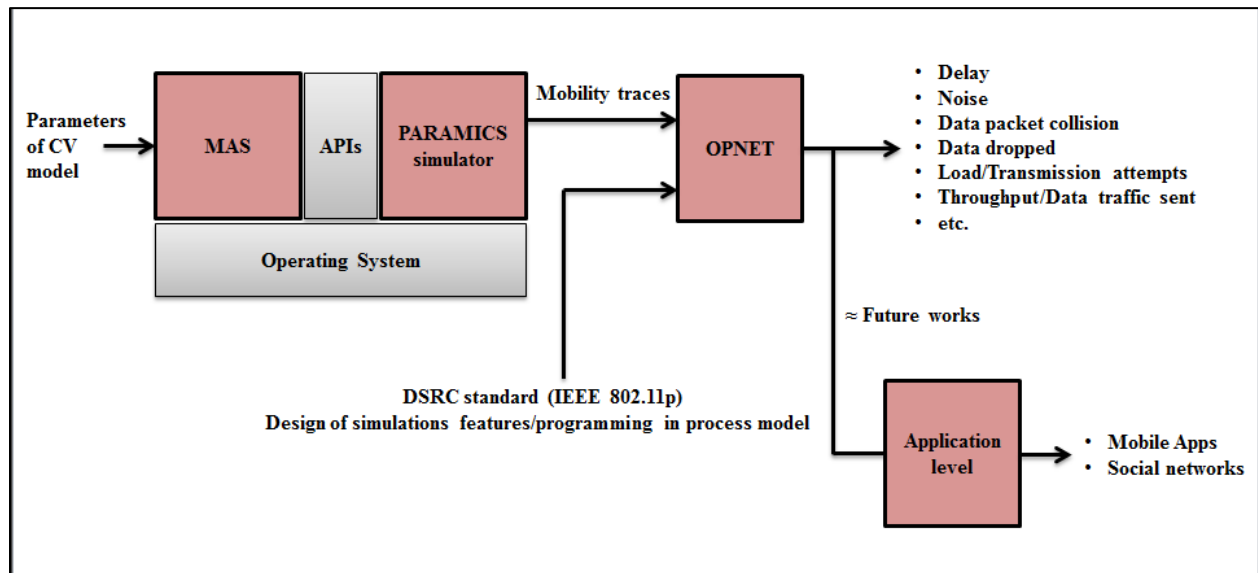
PARAMICS was used to realistically model the traffic flow of the selected test network and continuously collect traffic measurements and mobility traces during the simulation. The API #1 was also used to generate predefined incidents. On the other hand, generating CVs and non-CVs based on generated mobility traces from PARAMICS, modeling CV related infrastructures such as RSU, implementing the real-time V2V and V2I communications based on DSRC ranges, were modeled in the OPNET environment. OPNET, discrete event simulator, is one of the most popular network simulators which can model various protocols and devices and is easy to use to develop different complex models.

DSRC ranges were implemented by IEEE 802.11p standard adding wireless access in vehicular environments (WAVE). It is an approved correction of IEEE 802.11 standard which is the base of products marketed as WiFi [21]. Although cellular networks is considered as the primary means of vehicular internet access with wide coverage, they have low and variable data rates, high and variable latencies and occasional communication faults [22]. WiFi (802.11) and DSRC protocol, on the other hand, are becoming the optimum alternative for V2V and V2I communications. The U.S. Federal Communication Commission (FCC) allocated DSRC at 5.9 GHz to be used to provide the communication in CV systems [23]. In [24-26] authors stated that WiFi ensures continues and seamless connectivity in highly dynamic vehicular environment; also, it is suitable for applications with periodic connectivity due to short duration of connection. However, these studies only focused on the communication between vehicles and infrastructures and did not study the communications in vehicles themselves.

One test scenario has been conducted based on the proposed integration. The tested network is a section in Deerfoot trail, similar to previous scenarios, and the traffic counts were defined to replicate weekdays A.M peak hours. The results of this integration specified the best DSRC range and percentage of CVs in the network to have the least message drops and latency in V2V and V2I communications.

The demonstration that we proposed here has not been done by previous studies in two perspectives. First, this research is first to integrate PARAMICS with OPNET in order to simulate CV system and evaluate V2V and V2I communications based on DSRC standard. Second, previous studies which integrated other traffic and wireless simulators such as [27-29] did not evaluate V2V and V2I based DSRC communication elements all together under hazardous events. They did not include V2V, or DSRC ranges or emergency situations in their studies. Note that in emergency events, such as accidents, the number of message transmission in V2V and V2I will be increased leading to growth of overall load and throughput indices in the network. We enhanced the effectiveness of CV in these situations by presenting the optimum DSRC range and penetration rate of CV in the network to decrease these indices and yet to have the least message faults and delays. Smallest amount of data drops and latency leads to the transmission of warning messages at the right time and effectively improving safety and mobility applications in simulation of CV.

Figure 1.2 shows the overall picture of contributions of the designed system which will be described in details in the chapters 3 and 4. However, as shown in Figure 1.2, the application level is left for the future works.



**Figure 1.2 Inputs and outputs of integrated simulations - Proposed integration and future works**

## 1.5 Contributions

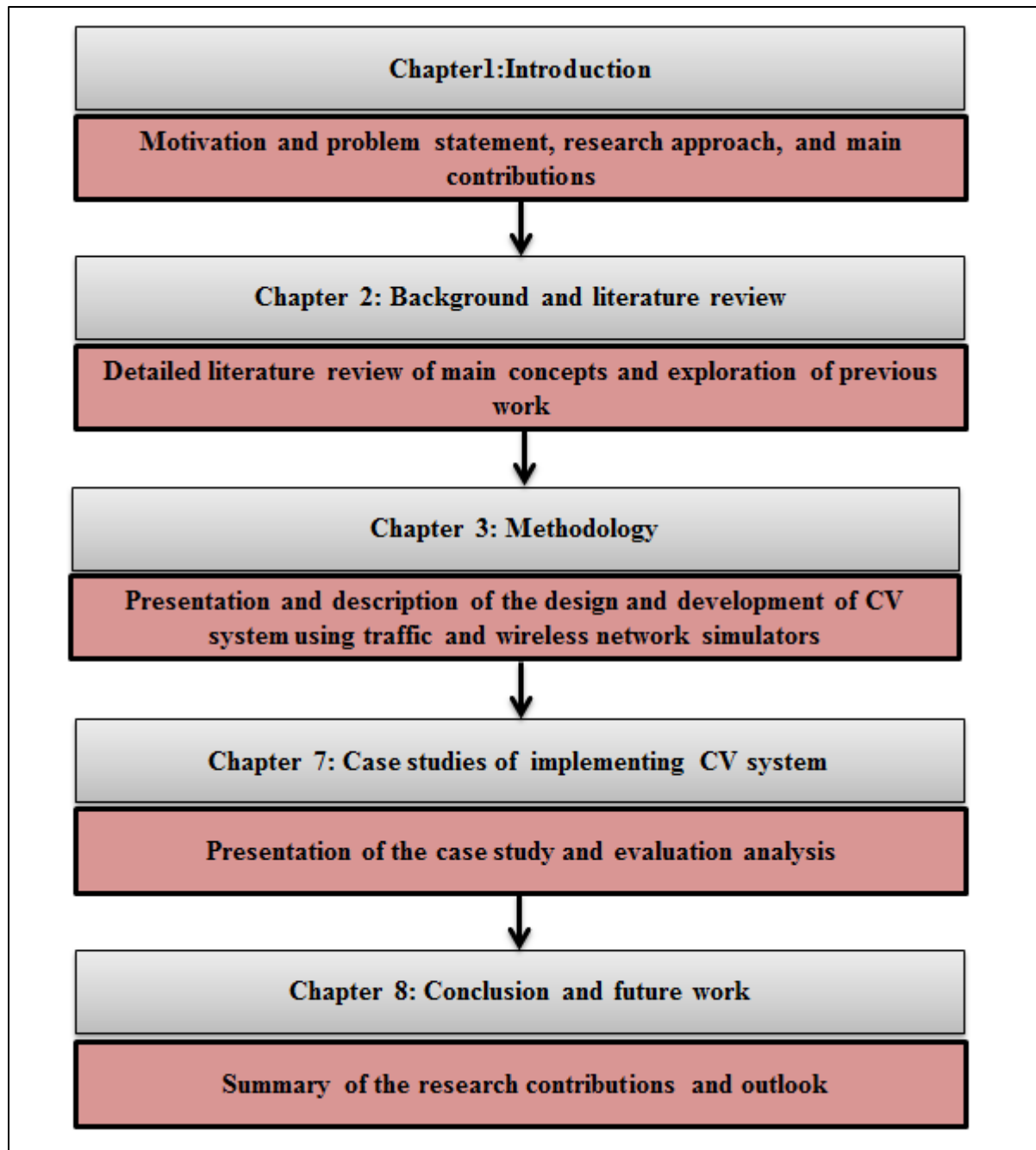
- Analysis and design of CV system as the MAS using MaSE methodology (Section 3.3)
- Implementation of CV system using APIs (Section 3.4)
- Integration of PARMICS and OPNET (Section 3.7)
- Importing vehicular mobility traces into wireless network simulator (Section 3.7.1)
- Evaluation of the impacts of V2V and V2I communications on improving safety and mobility applications by simulating two scenarios on Deerfoot trail, Calgary, Alberta (Sections 4.2 and 4.3).
- Presentation of the optimum selection of DSRC communication range of RSUs and ultimate percentage of CVs in the network to have the least data loss and delay in V2V and V2I data transmission by simulating one scenario (Section 4.4).

## 1.6 Structure of Thesis

The rest of thesis is organized as follows:

Chapter 2 gives a review of the related technologies and defines the terminologies used to fulfil our goals. It also summarizes the literature review on the thesis topic and highlights the motivation of having both traffic network and discrete event simulators. Then in chapter 3, the two approaches, proposed to accurately model CV system, are presented, followed by the explanations of all the methods used to attain the goals. Then the details of integrating two simulators are demonstrated in order to include traffic network modeling in wireless communication simulations. In chapter 4, test cases for assessing the effectiveness of CV and combined simulators are presented. Finally, Chapter 5 concludes the thesis and gives some recommendations for future works. Figure 1.3 represents the sequence of the material presented as the thesis structure.





**Figure 1.3 Thesis structure**

## **Chapter 2: Background and Literature Review**

### **2.1 Introduction**

In this study we develop, implement and demonstrate a DSRC based V2V assisted V2I traffic analysis system. It can be used for estimating and disseminating traffic safety and mobility information, as the focus of the ITS is to facilitate wireless communication between vehicles and infrastructure to exchange traffic data, such as safety information. Wireless communication is the transfer of data without using wires. A wireless network is a network that consists of two or more nodes transmitting information wirelessly. A node can be a device or a computer that has a wireless interface and is able to send and receive messages. A vehicular network is a type of wireless network in which the nodes are either vehicles communicating with each other or vehicles communicating with RSUs. There are plenty of implementations of wireless networks based on type and protocols used.

Computer simulation is a computerized version of a model which is run over time to investigate the concepts of the defined interactions. Simulations are generally developed by applying an iterative approach. Due to the interdisciplinary nature of CV systems, it is necessary to be able to model their behavior from several distinct points of view, including the communications perspective and the traffic perspective. In other words, we need to combine the performance of wireless system and traffic conditions to study CV system more precisely. Hence, we need a wireless network simulator, a traffic network simulator and, in addition, we need to integrate the two simulators to allow their intercommunication and exchange of data. However, first we should review the main components of CV systems, DSRC ranges, and the essential design of proposed system in the next sections.

## 2.2 Connected Vehicles

CV research initiative, formerly known as IntelliDrive, has the potential to improve safety, reduce congestion, benefit the environment and enhance traveler services by enabling vehicles to wirelessly communicate with roadside infrastructure, nearby vehicles, cell phones, and other mobile devices [30, 31]. IntelliDrive applications, attempt to combine advanced wireless technologies, onboard computer processing, advanced vehicle sensors, GPS navigation, smart infrastructures, and other technologies to improve the mobility and safety of urban and rural travel. Examples of CV applications include the use of vehicles as traffic probes, warning systems informing drivers about traffic slowdowns ahead, warning systems about cross-street vehicles that may potentially run through a red light, and systems notifying drivers about the roadway features, such as sharp curves. Authors in [32] further analyzed how probe vehicle data could be combined to traditional loop detector data to get real-time estimations about arterial travel times, while [33] assessed RSU coverage and the ability to obtain accurate speed estimates.

The basic concept of CV is the establishment of a networked environment between vehicles and roadway infrastructure (V2I) and among vehicles (V2V) through wireless communications. With CV system, V2V, V2I and other services are integrated to work together [34].

In V2V, vehicles exchange information with each other. It uses advanced information communication technologies to prevent road collisions and alert motorists. Moreover, V2V systems may lead to decreasing travel time and crash risk in traffic networks in case of a reasonable penetration rate, i.e. percentage of equipped vehicles [4]. Especially, when it comes to a restricted geographical area and slight budget; it was suggested that V2V systems could be comparable to costly traffic infrastructure development projects.

In V2I, vehicles transmit information with infrastructures (RSU). Its function varies from safety information, weather forecast, and traffic conditions transmission to vehicles, or opportunistic vehicular data collection. Vehicle can thus disseminate information (on their positions, speeds, travel times, road weather conditions, incidents, etc.) to the RSU. The information collected by these roadside components will be shared with the transportation infrastructure operators, such as control center, which will in turn adjust the operation of various

control devices (e.g. VMS) to maximize the efficiency of the transportation system and improve safety in response to traffic demand and road condition.

VMS is an electronic message board located close to a roadway. It represents a mechanism for disseminating information to drivers which their vehicles are unequipped to receive guidance about advisory speed or incidents ahead. Hence, such drivers can directly get this guidance through VMS messages. The VMS enables traffic controllers to inform drivers in real time about changing traffic conditions and is commonly used for parking guidance, safety warnings, and flow diversion [35].

In [19] a model for a traffic monitoring application of CV in a microsimulation environment is developed. The authors also examined the impact of the penetration rate of CVs on the quality of the data collected. Although, the study found a linear relationship between market penetration rates and the variance of speed collected from the network, it did not investigate on V2I and its impacts on results where it is implemented along V2V. In [17] various route guidance strategies with V2V communication using the VISSIM (Visual Traffic Simulation) micro simulation model were evaluated; however, same as [19] they did not consider V2I. The authors conducted sensitivity analysis to examine the impact of factors, such as the market penetration of CVs, congestion levels of a road network, update intervals of route guidance information and drivers' acceptance rates. The results of the study showed that CV technology along route guidance reduced travel time over the no guidance case. Re-routing means that vehicles which are CV equipped, after noticing that there is an accident ahead of them, will choose the alternative routes to reach their destination. They avoid encountering the present accident and creating further possible accidents. This will lead to decreasing the level of travel times and crash probability. In [20] a VMS control framework is proposed and evaluated that seeks diversion during incidents to enable a traffic system controller to favorably manage traffic conditions in real time. A hybrid framework is used to determine the information for the VMS. The results of the study showed that CV-based route guidance reduced travel time over the no guidance case.

The literature review on the implementation of CV in traffic simulators showed that most of the researchers investigated only on the V2V or V2I. The effectiveness of CV will become more increased when both V2V and V2I are considered [36]. Moreover, for the communication ranges, providing DSRC is important. The reason is that, DSRC is becoming the optimum choice

for vehicular internet access due to its several advantages, such as wide coverage, and it is accepted to be used in CV communications by FCC [37].

## **2.3 DSRC**

The USDOT's ITS [37] program focuses on integration of intelligent vehicles and infrastructure to improve traffic safety and mobility. For ITS applications, there has been considerable interest in DSRC wireless band. In 1999, FCC allocated 75 MHz of DSRC spectrum at 5.9 GHz to be used exclusively for V2V and V2I communications. It is a key enabling technology for the next generation of communication-based safety applications that reduce fatal injuries and improve traffic congestion.

DSRC standard (E2213-03 ASTM [38]) has permitted both safety and non-safety (commercial) applications and is implemented by IEEE 802.11p which improves IEEE 802.11 to deal with wireless access in the vehicular environment. It is expected that DSRC technology will be the predominant communication medium for safety applications relying on V2V and V2I communications, such as car accident avoidance, digital map update, ETC, lane-changing assistance, and emergency road event notification system in the form of warning messages including advisory speed and route choice recommendations [4]. The adaptability of DSRC increases the likelihood of its deployment by various industries and endorsement by customers.

The standard aims to provide wireless communications capabilities for transportation applications within a 1000m range at typical highway speeds. It provides seven channels at the 5.9 GHz licensed band for ITS applications, with different channels designated for different applications, including one specifically reserved for V2V communications.

DSRC technology has been tested for its propriety to handle safety messages in terms of reliability, high speed V2V information exchange, message propagation distance, time lag, security, channel congestion and other characteristics performed under the CV research program [39-41]. In [40] authors studied the communication level reliability of DSRC technology in terms of packet delivery ratio and distribution of following packet drops, and the application level reliability in terms of T-window metric using only three vehicles having DSRC capability and

GPS receivers. In [41] the IEEE 802.11p was studied, which provided the preliminary work for the DSRC standard, for vehicular communication in the context of propagation aspects.

DSRC is an ideal selection in comparison with cellular communications since it provides very high data transfer rates in circumstances where minimizing latency in the communication link and studying large communication zones are important. Thus, the USDOT currently holds the DSRC as the only short range wireless communication that provides desired qualities for vehicular communication as it is the only short-range wireless alternative that provides communication with low latency, high reliability, fast network acquisition, designated license bandwidth, priority for safety applications, interoperability, security and privacy [23]. In [42] several forms of a DSRC based V2V communication protocol were designed for safety messaging. Also, the performance of the protocols using reception reliability and channel usage was studied for different traffic flow conditions. A secure message protocol has been developed in [43] aimed at deploying security mechanism to secure network from possible abuse and having an efficient MAC for the purpose of safe and timely dissemination of safety messages.

In [44] authors gave an overview of DSRC applications and assessed the characteristics of the IEEE 802.11 MAC and PHY layers in this context. They stated that cellular networks can handle time sensitive communication between vehicles travelling at high speeds. However, this can be done with the help of base stations. These cellular base stations are significantly more expensive than their DSRC equivalent (i.e. 802.11 access points). Also, cellular handles only infrastructure to mobile communication.

Since DSRC is accepted as the vehicular internet access and based on the literature review it has more advantages comparing to other protocols, the implementation of CV system urge the use of DSRC to have the optimum effectiveness in V2V and V2I communications. However, before system can be implemented in real world, they must be evaluated completely to ensure their functionality and performance in a range of likely situations [5]. This evaluation can be done using simulations.

## 2.4 Simulation vs. Real World Experiment

Although a real network testbed allows maximum integrity for performance testing and prediction, evaluations in real world are not always possible, particularly when tests require costly, and often temporary and non-scalable, equipment to be deployed [5].

An efficient alternative to this problem is the use of simulation to conduct early performance evaluations before attempting any field deployment. Moreover, simulation can be performed in a very preliminary stage of the system design and can therefore be very helpful in the design process [5]. Simulation also has some desirable qualities that make it useful and it is cheaper than testing in real world in most cases. The actual cost of software and hardware does not compare to the cost of outfitting a road network, or the loss in confidence of the driving public on the driving environment which is always changing. Also, with simulations rarely occurring critical scenarios can be experimented, time can be sped up to predict future results and safety issues can be tested without the potential of hurting drivers. Computer simulation is a valuable tool especially for today's network with complex architectures and topologies and has two approaches: discrete and continuous. In discrete model, the state variables change only at a countable number of points in time. These points in time are the ones at which the event occurs or change in state. In continuous model, the state variables change in a continuous way, and not suddenly from one state to another (infinite number of states).

Detailed and realistic simulation of both traffic and communication interaction helps researchers in testing various fundamental designs, implemented algorithms, and parameter configurations removing the need for collecting field information after the implementation of a specific system.

The developed simulation models for this study are carefully calibrated and validated to realistically represent the real world [6], which should increase confidence in the study conclusions. The rest of the chapter outlines and compares different traffic and discrete event simulators. This helps us select the best ones to fulfil our goals.

## 2.5 Traffic Simulators

Increased computing power has contributed to precise modeling of the physical road and in simulation of specific elements of the transportation systems, such as the junctions. In this case, the integration of GIS and CAD systems, along with GUI, can play a significant role.

Recent advances in computer hardware and software technology have led to the increased use of traffic simulation models. Traffic modeling is a well-known research area in Civil Engineering and it is important to correctly model vehicular traffic during the design phase of new roads and intersections [45]. Depending on the required objective of the simulation, traffic models are divided into three models (macroscopic, mesoscopic, and microscopic models) and two main approaches (continuous and discrete), as explained in detail below.

Macroscopic models, like METACOR [46] and TransCAD [47], model traffic at a large scale. Its simulation model takes place on a section-by-section basis rather than by tracking individual vehicles. It models the description of traffic flow, and the measures of effectiveness, which are speed, flow, and density [48]. Macroscopic models have fewer demands on computer requirements than microscopic models since they do not have the ability to analyze the performance as much detail as the microscopic models.

Microscopic models continuously or discretely predict the state of individual vehicles and primarily focus on individual vehicle speeds and locations. Simulation time and memory requirements for microscopic models are high, usually limiting the network size and the number of simulation runs. Because of the high level of details required in a microscopic model, applications tend towards a relatively small geographical area in contrast to macroscopic models which are used in wide zones to contribute in transportation planning rather than traffic engineering [49]. By using the microscopic traffic simulators, the impacts of ITS applications can be evaluated.

Mesoscopic models, such as CONTRAM [50] and Dynameq [51], combine the properties of both microscopic and macroscopic simulation models. As in microscopic models, the mesoscopic' unit of traffic flow is a packet of vehicles. However, their movements follow the approach of the macroscopic model and are commanded by the average speed on the travel link, thus movements do not consider individual vehicle speed and volume relationships. The



Mesoscopic models fill the gap between the overall level approach of macroscopic models and the individual interactions of the microscopic ones by describing the traffic entities at a high level of detail, while their behavior and corporations are designed at a lower level of detail [52].

There is also another type of simulation model which is known as the nanoscopic model. It extends the capabilities of three basic components of microscopic simulation: vehicle modelling; vehicle movement modelling; and driver behaviour modelling. [8] There are also a few hybrid models consisting of any two of the three mentioned models (microscopic, mesoscopic, and macroscopic) in order to increase their strengths and eliminate the individual limitations.

Traffic simulation can be categorized into intersection, road section, and network levels. The simulation models can also be categorized by functionality, i.e. signal, freeway, or integrated [48]. The other categories might include traffic safety and the effects of advanced traffic information and control systems [53]. Simulation programs started with the modeling of specific elements and continued to model the whole network of the transportation system.

The advancements of IT have contributed to increased development of traffic simulation models. These include microscopic models and expanding the areas of applications ranging from the modeling of specific components of the transportation system to a whole network having different kinds of intersections and links. Microscopic traffic simulation models are becoming increasingly important tools in modelling complex transport networks and evaluating various traffic management alternatives in order to determine the optimum solution for traffic problems that cannot be studied by other analytical methods. In Table 2.1 and the subsequent paragraphs, the main features of a few microsimulation models are described and compared.

**Table 2.1 A few different types of traffic simulation models - their main features and capabilities**

Characteristic: Microscopic			
Name	Main features	Main capabilities	References
AIMSUN	Integration of traffic assignment models, a mesoscopic simulator, and a microsimulator in a single software application. It is developed based on car following, lane changing, and gap acceptance algorithms. Its features are favorable for creating large urban and regional networks.	It provides an additional option to practitioners to model dynamic aspects of very large networks and removes most of the calibration burden when compared to a micro-simulator.	[12]
CORSIM	Integration of microscopic, stochastic, link-node and periodic-scan, based traffic simulation program. The combination of arterial (TRAF-NETSIM) and freeway (FRESIM) simulation models makes CORSIM one of the analysis models available to traffic engineers that allow all of the individual components of the arterial and freeway system to be analyzed and simulated as a complete system.	It is designed for the analysis of freeways, urban streets, and corridors or networks. Stochastically determines the specific properties of each vehicle such as vehicle length, driver aggressiveness, acceleration rate, minimum acceptable gap, maximum free speed, and others. Perform the car-following and lane-changing logic to simulate vehicle movements on a second-by-second basis.	[54]
VISSIM	A discrete, stochastic, and time step based on a traffic flow model. Consists internally of the traffic simulator, a microscopic traffic flow simulation model, and signal control software. Its simulation systems consist of a traffic flow model and a signal control model. The model is developed based on these two researches [55, 56].	It considers driver-vehicle-units as single entities and contains a psycho-physical car following model for longitudinal vehicle movement and a rule-based algorithm for lateral movements. Sends detector values to the signal control program every second, and the signal control uses the detector values to decide the current signal aspects. Compiles information from the	[14], [57]

		traffic simulator on a discrete time step basis.	
PARAMICS	A suite of software models for microscopic stochastic traffic simulation. It is very comprehensive and has the potential for application to a wide set of freeway, arterial, and network situations.	It allows a unified approach to traffic modeling encompassing the whole spectrum of network sizes starting from single junctions up to national networks. It models the emerging ITS infrastructures.	[58]
ACTSIM	A dynamic micro-simulation model. Individual vehicle parameters, including vehicle speed, vehicle size, desired maximum speed, destination location, dwell time, and gap acceptance, are assigned by random variants derived from vehicle mode characteristics. It consists of a car following model, lane changing model, parking model, pedestrian crossing model, and passenger pickup/drop off model.	It simulates each vehicle independently, uses the distribution of population behavior, models changes in density, peaking in demand, curbside parking, and crosswalks. It provides a continual picture of the network. Statistical data gathered over a period provides an average view as well.	[59]

We attempt to summarize the comparative analysis of different microscopic simulation software in Table 2.2 to decide which one should be selected to fulfil our goals; however, it is not intended to provide a comprehensive list of comparative studies.

**Table 2.2 Different comparative studies of simulation software**

Software compared	Findings	References
VISSIM and CORSIM	CORSIM is suitable for the models which do not contain a transit element such as buses, LRTs, <i>etc.</i> On the other hand, VISSIM is recommended by users for its easy manipulation capability in the context of complex geometry, traffic control, and transit elements. There are three significant differences between the models. CORSIM uses a link-node structure while the network of VISSIM is built over a graphical map. The car-following modeling in CORSIM sets a desired amount of headway for individual drivers but VISSIM relies on the psycho-physical driver behavior model. VISSIM reports total delay by link and not for each turning movement, but CORSIM provides average control delay for each approach.	[7], [8]
CORSIM and AIMSUN	CORSIM can model more complex situations than AIMSUN and simulate the impacts of transit and parking on traffic operations. AIMSUN was found to operate acceptably well with outputs comparable to CORSIM, and it possesses features that would be useful for creating large urban and regional networks. They were evaluated using a variety of criteria, including hardware/software requirements; difficulty/ease of network coding; data requirements and appropriateness of defaults; and relevance/accuracy of performance measures reported in the output.	[8]
AIMSUN, PARAMICS and VISSIM	Findings showed lower error values for models implemented in AIMSUN and similar error values for the psychophysical spacing models used in VISSIM and PARAMICS.	[9]
AIMSUN and PARAMICS	With respect to the model building capacity, the input procedures of AIMSUN are easier and faster than in PARAMICS and some user claim it takes 30–50% less time to set up a model in AIMSUN than in PARAMICS.	[10]
CORSIM, VISSIM, PARAMICS	CORSIM outperformed others due to the least difficulty in coding and its ability to compute control delay for individual approaches. Three dimensional animations are available in both PARAMICS and VISSIM but not in CORSIM, which relies on two-dimensional animations. PARAMICS and VISSIM provided simulation results that better matched field-observed conditions, traffic engineering principles, and expectation or perception of reviewing agencies.	[7]
VISSIM and PARAMICS	Comparison in terms of “ease of use”, which consists of input data requirements, network coding/editing, and input/output review, claimed that network coding/editing and input/output review in PARAMICS are good, but data requirements need improvement.	[11]

From all of the studied simulators, the models AIMSUN, PARAMICS, and VISSIM [10] are found to be suitable for congested and integrated networks of freeways and streets. Also, these models are practically beneficial for ITS applications. While these packages have many resemblances, each has its own specific characteristics that make it somehow suitable for certain modelling purposes. There are a few other simulation models which are developed focusing on ITS such as MITSIMLab [60] and INTEGRATION [61].

In this research, PARAMICS is selected due to its scalability, capability, its use in previous works examining variable speed limits and real-time crash risk [5, 62, 63], proven background on freeways, urban roads and VMS concept, which is used in this study. PARAMICS is able to simulate ITS applications required for implementing and evaluating CV systems properly and allows users to extend and test their own traffic control strategies. Given that, with the use of API, PARAMICS satisfies the transmission and dissemination requirements of warning messages. Besides, the PARAMICS program can produce the vehicular movement information at every simulation interval as little as 0.1 second, which is an essential feature for setting up the detailed trajectory output file for the integration of the PARAMICS with wireless network simulator to reach our second goal. In the next section, PARAMICS microsimulator is described with more details and its modules are introduced.

### **2.5.1 PARAMICS**

PARAMICS (Quad stone PARAMICS) [58] is a software package that consists of several tools and help in designing, modeling, analyzing and testing a wide range of transportation networks. PARAMICS simulates the movements of individual vehicles based on car-following, lane-changing, gap-acceptance, and other driving behavior models. It is a stochastic simulation model, because many of the parameters defining the behavior of individual drivers are determined through probability distribution sampling.

PARAMICS is capable of representing many parts of the world's street maps and thus is a popular simulator among universities and government agencies. Its software is designed to handle scenarios ranging from a single intersection to a congested freeway, or the modeling of a complete traffic system. Moreover, the internal structure of the PARAMICS model, developed in

the early 1990's [64], consists of several components which together provide an integrated platform and are capable of modeling a wide variety of real world traffic and transportation problems:

**Modeller** - the Modeler module is responsible for building the network, simulating traffic conditions and providing statistical output through a GUI. Every aspect of the transportation network can potentially be investigated including integrated urban and freeway networks, advanced signal control, roundabouts, public transportation, car parking, incidents, truck or HOV lanes, or special lane usage. A PSM is also available as an external feature.

**Processor** - The processor module is responsible for extensive testing of a simulation. It allows the user to run a number of simulations automatically in batch mode and thus saves time and effort.

**Estimator** - The estimator module is used for OD matrix estimation.

**Analyzer** - The analyzer module is the post simulation statistics viewing tool whose primary function is to display and compile reports on statistical data output from Modeller.

The model output by Analyzer includes statistics at the network level (such as overall travel time, total travel distance, average speed), on a link-by-link basis (statistics such as traffic flows, queue lengths, delays speeds and densities) or at specific locations (such as immediate detector type of information).

**Programmer** - The programmer module is a development API that allows users to attach their plugins to the modeller simulations. It is a SDK for the PARAMICS suite which can be used for research of various aspects of ITS, real time connectivity and control, connectivity to real world hardware and software systems, and advanced model behaviors. PARAMICS Programmer allows users to customize some critical parts of the PARAMICS core models.

Through the use of an API, users can implement their own complementary traffic control strategies (such as signal optimization, adaptive ramp metering, incident detection, etc.).

**Designer** - The designer module is used with the Modeller module to design and edit 3D models and traffic networks. It is used mainly for adding visualizations and captured movies to presentations.

**Converter** - The converter module is used to convert geometric network data into a PARAMICS network. Data can be accepted from emme/2, Mapinfo, ESRI [65] and many other sources.

These components together enhance the capabilities of the platform and allow it to handle a wide range of scenarios from a single intersection to a congested freeway and up to the modeling of a complete traffic system.

In addition, PARAMICS can model a variety of real world traffic problems. The model input includes:

- Network characteristics: geometry, link description, signposting, lane restrictions, forced lane changes.
- Demand data: OD zone areas, level of OD demand, breakdown by time period, vehicle type and vehicle proportion.
- Assignment: link cost factors, coefficients of generalized cost equations, assignment techniques.
- General configuration: time step duration, speed memory, mean target headway, mean reaction time.

#### **2.5.1.1 Applications Programming Interface (API)**

Most existing commercial programs, such as PARAMICS and VISSIM, provide ways of simulating traffic models which usually require quite computer related knowledge and coding efforts. Since they are closed-source, they provide API functions through which underlying simulation logic can be changed. Thus, researchers can simulate traffic models other than built-in models through these API functions.

PARAMICS provides API which enables users to enhance its functions and communicate with the core models in order to query or set various parameters during the course of a simulation, add model functionalities, replace the default driver behavioral models, and let users test their own traffic control scenarios. To use the API, users need to write their source code in C or C++ and use the provided API library to access to the core models. Basically, PARAMICS

provides four classes of interface functions: getting functions, override functions, setting functions, and extending functions [66].

Although, PARAMICS is beneficial over other types of microsimulation packages, it lacks the ability of simulating CV systems and their applications in the designated traffic simulation network. Moreover, simulated vehicles in PARAMICS operate under ideal error-free driving world, and therefore zero incidents happen. To allow for testing CV applications, it was necessary to manipulate the model in a way that a predefined incident or various stochastic incidents such as running red light, rear end collision, and weather caused incidents can be reproduced in the simulation world. In earlier research project [4, 13], a model for traffic monitoring application of CV in a micro-simulation environment was developed using API to simulate the traffic information dissemination by individual vehicles to other vehicles or to RSUs. By adding two APIs to the PARAMICS, the connectivity between vehicles (V2V) and infrastructures (V2I) in the traffic network were implemented and evaluated

For transportation system operators, a particular interest is on the ability to use vehicles as traffic probes. By enabling information about traffic conditions to be collected from every vehicle on every link traveled, traffic data collection from road networks could be significantly expanded at a potentially low cost. Probe vehicles are to collect data at periodic intervals or when specific events occur. At a minimum, each vehicle is to record its position, speed, and heading. While generated data could be retrieved from vehicles using various technologies, it is currently envisioned that data will be stored onboard the vehicles until they come within range of RSU equipped with DSRC wireless devices. Authors in [27] simulated the testbed PDS, which consists of 52 deployed RSUs [67], using a vehicular simulator to study the impact of various parameters on performance, through PARAMICS' API. But they did not simulate the underlying wireless network and security protocols.

Through the use of API, PARAMICS and a wireless network simulator (e.g. OPNET) can be linked together to solve the proposed constraints model such as PARAMICS's incapability of simulating communication protocols. The trajectory file and specifications of CV models from PARAMICS is input into the OPNET to evaluate the connectivity between vehicles and infrastructures while the output of the OPNET such as delay can be captured by PARAMICS to assess the route choice behavior simulation.



By using API we develop and simulate CV system in the examined network. Since our first goal is to improve the safety and mobility benefits by implementing CV system; thus, measures should be selected to prove this enhancement. Next, we explain how literature review found these measures useful for evaluations.

### **2.5.1.2 Safety and Mobility improvements**

Traffic safety simulation is mainly based on the field of human centered simulation, where the reaction system of drivers, with all its weak characteristics, has to be described [53]. CV system is expected to result in improvement in road safety through the exchange of relevant information between vehicles. The information is either presented to the driver or used by automatic active safety system. Some examples are: cooperative forward collision warning, left/right turn assistant, lane changing warning, stop sign movement assistant and road-condition warning. By improving traffic safety, the probability of crash risk in the network would decrease.

In [5, 62, 63] statistical models to get a measure of real-time crash potential are developed. These models use the inputs loop detector data to gather information on average volume, occupancy, and coefficient of variation in travelled speed and assess the crash potential at a given location in real-time.

The objectives of minimising travel time which lead to mobility improvement allows the act on the true cost of trips. A cooperative traffic control method is presented in [68], where V2V and V2I communications are enabled by DSRC form ad hoc networking and computing grid.

CV communications allows the sharing of wireless channels for collision avoidance (improving traffic safety), reducing travel time (improving mobility index), improved route planning, and better control of traffic congestion [69].

In [4], the impact of considering advisory speed and re-routing guidance on safety and mobility applications were examined. The statistical models of PARAMICS are used to assess the effect of CV systems on travel time reduction for the simulated network. For the safety evaluation, the crash potential was calculated based on the model introduced in [5]. Although the study found the improvement in both mobility and safety applications, the DSRC range was not developed as a factor for distributing messages in V2I module and the focus was only on

upstream traffic. However, for enhancing traffic safety, the impacts of DSRC ranges also can be considered [42, 70, 71].

Next, in order to reach our second goal, as mentioned earlier, we need a wireless network simulator to simulate DSRC protocol for V2V and V2I communication and then evaluate the critical factors affecting both of them. Here, we introduce different wireless simulators and explain why we select OPNET.

## **2.6 Wireless Communication Simulators**

Vehicular communications has become one of the active domains of research in the wireless networking community. Vehicular networks (VANETs) present a challenging environment for protocol and application design because of their large scale.

The study of a complex system always requires a simulation package, which can compute the time that would be linked to real events in a real world situation. Network simulators allow researchers to study how the network would behave under different conditions. Users can then customize the simulator to satisfy their particular analysis needs. Functions and protocols are described either by finite-state machine, native programming code, or a combination of the two. A simulator typically comes with a set of predefined modules and user-friendly GUI.

Compared to the cost and time involved in setting up an entire testbed containing multiple networked computers, routers and data links, network simulators are relatively fast and inexpensive. Hence, they allow researchers to test scenarios that might be particularly difficult or expensive to do using real hardware, especially in VANETs. Network simulators are particularly useful to test new networking protocols or to propose modifications to existing ones in a controlled and repeatable manner.

Computer network design and implementation involves the communication of different networking devices including servers, NICs, switches, routers, and firewalls. Mostly, it is ineffective with respect to time, money, and effort to assess the performance of a live network. Computer network simulators are often used to evaluate the system performance without building a real network. However, the operation of a network simulator relies on various stochastic processes, including random number generators. Therefore the accuracy of simulation results and model validation is an important issue. A main concern in wireless network

simulations or any simulation efforts is to ensure a model is valuable and represents reality. If this can't be guaranteed, the model has no real credit and can't be used to answer desired questions [72]. For selecting an appropriate network simulator for a particular application, it is important to have good knowledge about the available simulator tools, along with their strengths and weaknesses.

According to [73] the physical layer is usually the least detailed modelled layer in network simulations. Since the wireless physical layer is more complex than that of wired physical layer and may affect the simulation results drastically. Physical layer parameters such as received signal strength, path loss, fading, interference and noise computation, and preamble length greatly influence WLAN performance. Therefore, simulation results may differ from real testbed evaluation both qualitatively and quantitatively [74].

Although drawing a conclusion from simulation run has to be done very carefully, network developers are rarely independent from using simulators. Network simulation packages offer a rich environment for the rapid development, performance evaluation and deployment of networks. Perhaps simulation results between two competing architectures or protocols might be more important than quantitative results [75].

Depending on the networking area and the layer for which new algorithms or protocols are developed, the selection of the appropriate simulator is feasible. Therefore network researchers and developers should use an appropriate simulation package for their simulation tasks. However, if a major design decision has to be made and different simulators suggest different conclusions, implementing a real testbed would be appropriate. Although authors in [76] did not find the differences in their evaluation between OPNET and ns-2, most of the other authors could observe that changing the network simulator can result in completely different conclusions. Some recommendations are being proposed to run a hybrid simulation based network evaluation which simulate the physical layer and execute more abstract layers of the protocol stack. However, this would add a huge amount of complexity and could lead to scalability problems when dealing with large number of wireless nodes.

Several network simulators can be used to simulate the communication between vehicles themselves or vehicles and infrastructures. Next in Table 2.3 the main characteristics of some of the most promising network tools to simulate CV scenarios are presented.

**Table 2.3 A few Different types of wireless simulation models - their main features and drawbacks**

Discrete event simulation environments			
Name	Main features	Main drawbacks	References
Ns-2 (Network Simulator)	It includes node mobility, a realistic physical layer with a radio propagation model, radio network interfaces, and the IEEE 802.11 MAC protocol using the Distributed Coordination Function (DCF). The revised PHY is a full featured generic module capable of supporting any single channel frame based communications. The key features include cumulative SINR computation, preamble and PLCP header processing and capture, and frame body capture. The revised MAC accurately models the basic IEEE 802.11 carrier senses multiple accesses with collision avoidance (CSMA/CA) mechanism, as required for credible simulation studies.	The ns-2 distribution code had some significant shortcomings both in the overall architecture and the modeling details of the IEEE 802.11 MAC and PHY modules. It is not so well structured software architecture and the mixture of compilation and interpretation made it difficult to analyze and understand the code. Also, simulation running will be very slow especially when the network simulated contains many nodes.	[77], [78]
SNS (Staged Network Simulator)	It is a staged simulator based on ns-2. SNS executes approximately 50 times faster than regular ns-2 and 30% of this improvement is due to staging, and the rest to engineering. This level of performance enables SNS to simulate large networks which NS-2 is incapable of.	This implementation is not specifically designed to simulate VANET scenarios.	[79]
NCTUns (National Chiao Tung)	The most unique feature of NCTUns is its capability of modeling the CV communication standards (i.e., the IEEE 802.11p and IEEE 1609 family). In addition, unlike NS-2, which	It is an open-source network simulator that is designed to only run on a Linux platform. Also, the manipulation at every node has to be done node by node, or all the nodes	[80]

University network simulator)	employs a TCL, NCTUns provides an easy-to-use GUI to create or edit communication network modules.	to the same time. The programming is not supported by the NCTUns, therefore, simulation parameters set only by graphical user interface.	
OMNet++ (Object Modular network Testbed in C++)	It provides a component-based, hierarchical, modular and extensible architecture. Components, or modules, are programmed in C++ and new ones are developed using the C++ class library which consists of the simulation kernel and utility classes for random number generation, statistics collection, topology discovery etc.	The original implementation does not offer a great variety of protocols, and very few have been implemented, leaving users with significant background work if they want to test their own protocol in different environments. For instance, the mobility extension for OMNeT++ is intended to support wireless and mobile simulations within OMNeT++ but it is fairly incomplete. Therefore, OMNeT++ has remained relatively obscure.	[81]
J-Sim (Java based simulation)	It is built according to the component-based software paradigm and written in Java. The lower layer Core Service Layer (CSL) comprises every OSI layer from network to physical, the higher layer comprises the remaining OSI layers. Initially designed for wired network simulation, its wireless extension proposes an implementation of the IEEE 802.11 MAC. This extension turns J-Sim to a viable MANET simulator. J-Sim also features a set of components which facilitates basic studies of wireless/mobile networks, including three distinct radio propagation models and two stochastic mobility models.	The execution time is much longer than that of NS-2. Since J-Sim was not originally designed to simulate wireless networks, the inherently design of J-Sim makes users hardly add new protocols or node components. Its wireless extension is only MAC supported so far.	[82]
	It provides a comprehensive set of tools with many components for custom network modeling and simulation. Models in source code	The worst drawback of the QualNet is its extreme high CPU utilization and its implementation in Java which	

QualNet (Scalable Network technologies)	form provide developers with a solid foundation from which to build new functionality or to modify existing functionalities. QualNet does have a range of wired as well as wireless models but its main strength is in the wireless area. QualNet is written in Parsec (Qualnet, web link).	makes it run very slowly on most machines.	[83]
OPNET (Optimized Network Engineering Tools)	It is the most widely used network simulator. The simulator provides an environment for designing protocols and technologies as well as testing and demonstrating designs in realistic scenarios. It defines a network as a collection of sub-models representing sub-networks or nodes, therefore it employs hierarchical modeling. The topology used in a simulation can be manually created, changed using C++, imported or selected from the pool of predefined topologies.	The disadvantage of using OPNET is that the simulation requires a lot of processing power and can be very time consuming particularly for network with a large number of transmitter and receivers. This is mainly due to the detailed radio pipeline stage that OPNET uses since every packet transmitted is required to go through these stages. The simulation time can be reduced by using parallel processors.	[84]

There are several issues that need to be considered when selecting a network simulation package for simulation studies. For example, use of reliable random number generators, an appropriate method for analysis of simulation output data, and statistical accuracy of the simulation results (i.e., desired relative precision of errors and confidence interval). These aspects of credible simulation studies are recommended by leading network researchers [85-88]. In [89] a case study was presented in which four popular wireless network simulators, i.e. J-Sim, OMNeT++, ns-2 and ShoX [90], were used to evaluate a topology control protocol. The authors mentioned the missing features of the simulators. They also compare the amount of effort needed for installation, familiarization, implementation and visualization from feature and usability point of view (instead of correlation of the individual simulation results). Authors in [91] provided a map of the main characteristics that MANETs simulation tools should feature and the current support of these. It gives a description of a list of simulators (DIANEmu [92], NAB [92], GloMoSim, GTNets [93], J-Sim, Jane [94], , ns-2, OMNeT++, OPNET Modeller, QualNet, and

SWANS [95]), provides an estimation of their popularity, and gives some hints on which simulator to use for what needs. Authors in [76], Simply presented a comparative study of two network simulators, OPNET Modeller and Ns-2, and provides a guide to researchers undertaking packet-level network simulations. The simulator outputs were compared to the output from a live network testbed. In [96] a project called Integrated Risk Reduction of Information-based Infrastructure Systems (IRRIIS), was delivered. The purpose was to identify, list and compare tools and components suitable for simulation of critical infrastructures. The simulators used are OPNET Modeller, NS-2, QualNet, OMNeT++, J-Sim, SSF and Backplane [92].

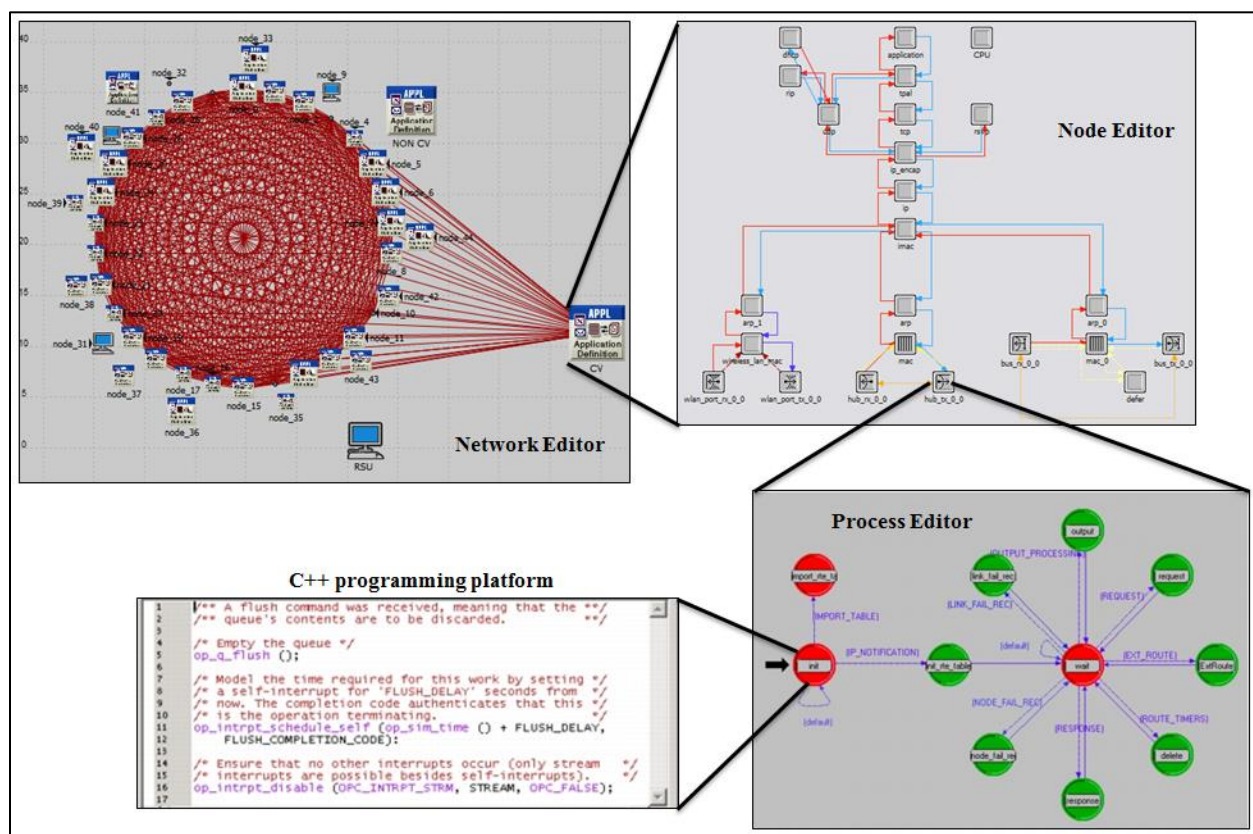
Some of the studied simulators provide open source code and are available freely on the internet (e.g. NS-2). Users can modify and enhance the code further, creating new versions. The major shortcoming is the lack of considerations for VANETs. For example, vehicular traffic flow models are not considered and 802.11p MAC is not included into the simulators (except for OPNET). Also, physical layer issues, obstacles, and road topologies present in a vehicular environment are often neglected. But, OPNET provides a comprehensive development environment for the specification, simulation and performance analysis of communication networks. From all the studied simulators, we select OPNET simulator to implement V2V and V2I communications based on DSRC. Because, OPNET offers full protocol stack modeling capability with the ability to model all aspects of wireless transmissions, including RF propagation, interference, transmitter/receiver characteristics, node mobility including handover, and the interconnection with wired transport networks. Several wireless networking technologies such as MANET, IEEE 802.11, 3G, Ultra Wide Band, IEEE 802.16, Bluetooth, and Satellite are supported in OPNET. Therefore in this research, OPNET is selected as the discrete event simulator as the means of analyzing system performance and their behavior during the simulation of the proposed system. Here, in the next section, we introduce OPNET with more details.

### **2.6.1 OPNET**

OPNET is one of the most popular network simulators as the package is available to academic institutions at no cost under OPNET academic program. It contains numerous models of commercially available network elements, and has various real life network configuration

capabilities. This makes the simulation of a real life network environment close to reality. However, network researchers are often reluctant to use this package because they may not aware of the potential strengths of this package and also because of the lack of good tutorial on wireless network simulation using OPNET.

OPNET Modeler provides different levels of modeling (network, Node, and Process) depending on the necessities and requirements of the simulation. These modeling environments are sometimes referred to as the modeling domains of OPNET, since they essentially measure all the hierarchical levels of a model. The remaining specification editors correspond to no particular modeling domain since they mainly support the three principal editors. The capabilities offered by the three modeling domains mirror the types of structures found in an actual network system and models developed at one layer can be used by another model at a higher layer [84]. Each model is briefly described in following and shown in Figure 2.1:



**Figure 2.1 The three-tiered OPNET hierarchy**



**Network model** - Network Editor is used to specify the physical topology of a communications network, which defines the position and interconnection of communicating entities. Network topology described in terms of subnetworks, nodes, links, and geographical context. Its models consist of nodes and links which can be deployed within a geographical context. Most nodes require the ability to communicate with some or all other nodes to perform their function in a network model. Several different types of communication link architectures are provided to interconnect nodes that communicate with each other. A node can either be fixed, mobile or satellite. OPNET provides simplex (unidirectional) and duplex (bidirectional) point-to-point links to connect nodes in pairs. A bus link provides a broadcast medium for an arbitrary number of attached devices. The Radio version adds the capability for fixed, satellite, and mobile nodes to communicate with each other via radio links. Each type of link can be customized by editing parameters or supplying new logic for the underlying link models. To break down complexity and to simplify network protocols and addressing, many large networks make use of an abstraction known as a subnetwork. A subnetwork is a subset of a larger network's devices that forms a network in its own right. OPNET provides fixed, mobile, and satellite subnetworks to enhance network models. These subnetworks can then be connected by different types of communication links, depending on the type of subnetwork.

**Node model** - The Node Domain provides for the modeling of communication devices that can be deployed and interconnected at the network level. In OPNET terms, these devices are called nodes, and in the real world they may correspond to various types of computing and communicating equipment such as routers, bridges, workstations, terminals, mainframe computers, file servers, fast packet switches, satellites, and so on.

Node models consist of modules and connections. Modules are information sources, sinks, and processors which are expressed as interconnected modules. These modules can be grouped into two distinct categories. The first set is modules that have predefined characteristics and a set of built-in parameters such as packet generators, point-to-point transmitters and radio receivers. The second group contains highly programmable modules. These modules referred to as processors and queues, rely on process model specifications. Each node is described by a block structured data flow diagram. Each programmable block in a Node Model has its functionality defined by a Process Model. Modules are interconnected by either packet streams or statistic

wires. Packets are transferred between modules using packet streams. Statistic wires could be used to convey numeric signals.

**Process model** - Processor modules are user-programmable elements that are used to describe the logic flow and behavior of processor and queue modules. The tasks that these modules execute are called processes. Processes may be created and destroyed based on dynamic conditions that are analyzed by the logic of the executing processes. Communication between processes is supported by interrupts.

In OPNET, it is possible to add some abilities to the nodes, modify the overall performance of the simulation and control the behavior of processes (protocols, algorithms, applications) using process model. Process models are expressed in a language called Proto-C, which is specifically designed to support development of protocols and algorithms. Proto-C is based on a combination of STDs, an extensive library of over 300 KPs, and the general facilities of the C or C++ programming language. Proto-C models allow actions to be specified at various points in the finite state machine. STD defines a set of primary states that the process can enter or move to another state. The condition needed for a particular change in state to occur is called a transition.

Nodes contain a set of transmission and reception modules, representing a protocol layer or physical resource, to ensure their connection to communication links. Interactions between modules are handled by exchanging messages. Users are able to configure applications installed on a node, and set nodes and links to fail or recover during simulation at specified times. Before simulation execution, one should make a selection of desirable output statistics. It is possible to specify a set of network simulations and pass a range of input parameters or traffic scenarios (which can be characterized by models for various applications like FTP, HTTP, etc.) to them. OPNET Modeller can execute several simulation scenarios in a concurrent manner.

OPNET Modeler can model protocols, devices and behaviors with a number of special modelling functions and has easy to use GUI and therefore it is easier to develop a network model for simulations [97]. A skilled user can develop a complex model containing many different hierarchical layers within a short period of time especially interoperability of WANs and LANs can be modelled efficiently. Other features of OPNET include comprehensive library of network protocols and models, source code for all models, and graphical presentation of simulation results.

More importantly, OPNET has gained considerable popularity in academia as it is being offered free of charge to academic institutions. That has given OPNET a great positive point in comparison with NS-2 in both marketplace and academia. OPNET supports huge amount of low level details (CPU speed of servers, number of cores, CPU utilisation, etc.).

While modeling the effect of low level protocol modification is generally difficult to achieve, OPNET appears to be more suitable for high level network simulations (e.g. evaluating the scalability of a network architecture) and less suitable for low level protocol performance evaluation [21]. Although OPNET has limited functionality, it supports various servers, routers and other networking devices with different manufacturer specifications which are adequate for developing models (small scale) of real-world network scenarios.

As mentioned earlier, in order to fulfil the second goal, we need to integrate the PARAMICS with OPNET. But first, in the next section, we review the previous works done on the similar integration, their shortcomings, and how our implementation will complete preceding studies.

## **2.7 Overview of Previous Research on Integration of traffic and wireless communication simulators**

With recent interest in CV applications, a few works have focused on creating connection between traffic and communication simulators, because communication effectiveness plays a key role in determining the overall performance of the CV system. Detailed and realistic simulation of both traffic and communication interaction can assist researchers in testing various functional architecture designs, implementation algorithms, and parameter configurations.

Earlier work on integrated traffic and communications simulations focused on creating simplified models of communication characteristics [28, 98]. This approach included the validation of different traffic concepts without too much concern about the details of communication efficiency and reliability. On the other hand, several studies have adopted a simplified vehicular movement model (e.g., the random way point model) to feed geographic and kinetic data of nodes for detailed communication network modeling [99, 100]. Although randomized node movement and message generation models are commonly used by the MANET research community to validate networking protocols for generic applications, they are insufficient for real time validation of specific vehicular traffic operations.

As envisioned in CV systems, equipping vehicles and roadside infrastructures with wireless communication interfaces will make it possible to improve the availability and quality of traffic information transmission by using real wireless protocols. These improvements would enhance the performance and capability of V2V and V2I communication and consequently safety and mobility patterns.

In [29], authors have proposed their own platforms that combine the capabilities of traffic networks and wireless communication simulators, in an attempt to provide a vehicular communication network platform. These simulation platforms often lack good modeling of either the traffic network, or the wireless network, or even both aspects. This is because they are designed by first creating a mathematical model for the traffic network, then another model for the wireless network. Also, most of these simulators were designed only for a specific task or to test a certain capability.

In [101], the CORSIM vehicular traffic simulator was integrated with the QualNet wireless network simulator [83] using a third-party distributed simulation software package to allow both models to synchronize their operations. A common message format was also defined to facilitate the exchange the vehicle status and position information. They focused on optimizing communication between simulators to examine different traffic conditions such as congestion due to an incident, while they simulated only unidirectional feedback.

In [102], authors developed an integrated simulation model by enabling bidirectional couplings with SUMO [103], a microscopic traffic simulation model, and OMNeT++ [81], an open-source network simulation model. Since OMNeT++ did not support the WAVE/DSRC communication standard for IntelliDriveSM, the wireless communication that was adopted was Wi-Fi utilizing existing IEEE 802.11a/b/g standard. This is the main gap of this study as IEEE 802.11p is particularly defined for vehicular internet access.

Two works, [104] and [105], have proposed a hybrid simulation approach using an on-line communications simulation with offline vehicular trajectories. In [104], the VISSIM [14], microscopic simulation program, was used to create vehicular trajectories that are fed into the network simulator NS-2 [78]. However, the communication standards used were far from those of IntelliDriveSM because the NS-2 version used in this study did not support the communication protocols of WAVE/DSRC.

In [106], the authors designed TraNS (Traffic and Network Simulation environment) which combines a traffic network simulator (SUMO) and a wireless traffic simulator (NS-2). The network-centric mode simply extracts mobility traces from SUMO, parses these traces into a form readable by NS-2 and then inputs this readable form into the NS-2 simulation. A drawback of this approach was the use of an oversimplified driver behavior model where the actions of the driver were constrained to increase-decrease speed and change lane.

Attempts have been made to link PARAMICS with QUALNET [107], NS-2 [108], and [109]. While vehicles are programmed to record and broadcast link travel times, there is, in both cases, no explicit modeling of V2I communications. A similar limitation is found in [110], which attempts to link VISSIM with NS-2.

In CV systems it is assumed that vehicles establish connection with the closest RSU and that both connections and data transfers occur instantaneously. In reality, a short delay may occur before a connection is fully established and data transfer may be spread over a certain interval. In [27] authors tried to complete the development of an interface between PARAMICS and NS-2 simulator. While an interface has already been tested, they could not complete the modeling of DSRC communication protocols and data routing mechanisms within NS-2.

In [111] authors highlighted the effects on data latency of rules prohibiting vehicles to communicate more than once with a RSU. They also looked at data aggregation issues associated with the potential need to process large amounts of data. From a communication standpoint, latency often refers exclusively to transmission delays between the moment a data packet is put in line for broadcast and the moment it reaches its destination. Also, the time a vehicle will need to get to a RSU communication range, will be an important contributor to the overall data latency.

Real systems will eventually have to deal with lots of vehicles running multiple applications and seeking to communicate simultaneously with the same RSU or data server. Significant communication delays, and eventually data losses, could happen more often if the underlying infrastructure is not designed to handle the data traffic. While it is interesting to explore time lag effects that may result from large data loads, this is not possible as long as the CV system does not fully model wireless communication systems.

In [112], a CV simulation environment integrating both microscopic traffic simulation and a wireless communications network simulator based on the WAVE/DSRC standards was developed. The results of their case study demonstrates the feasibility of integrating traffic and communications simulation models, and also illustrates the need to consider both components in IntelliDriveSM evaluation. In two papers [18, 113] authors developed IntelliDriveSM simulation testbed that can be used to quantify benefits of a traffic monitoring application in the Tyson's Corner network in Virginia using AIMSUN, and a ramp metering application in Irvine, California, using PARAMICS. But they assumed a perfect communications performance such as having no delays if vehicles are within the communication zone. In reality, transmission delays as well as packet drops during radio communications are inevitable. Thus, their impacts need also be modelled.

By reviewing the previous works done on the integration, we design our integrated simulators and evaluate the DSRC communications.

### **2.7.1 Integration of PARAMICS with OPNET**

This research, as previously mentioned, implements an integration of the network simulator OPNET, which is the most widely used network simulator providing an environment for designing protocols and technologies as well as testing and demonstrating designs in realistic scenarios, and the traffic simulator PARAMICS, which consists of several tools that aid in designing, modeling, analyzing and testing a wide range of transportation networks.

PARAMICS is time-step behavior-based microscopic traffic simulation software which represents the traffic network as a link-node system, and thus describes a vehicle's location by tracking the link it is on, and the distance to the link's stop line. The vehicles' locations and other essential characteristic of vehicles, such as type, are stored in trajectory file as the output. This output is used as input into the OPNET to implement the same testbed network and assess the connectivity between vehicles and infrastructures while the output of the OPNET such as the location of the closest RSUs can be captured by PARAMICS.

The main convincing factor of selecting these two simulators was their flexible ability of adding extended modules to model the safety and mobility impacts of CV. Through the use of

API, CV system was developed and V2V/ V2I communications efficiency, using DSRC protocol, were evaluated.

## **2.8 Summary**

In this chapter, we study the CV systems and the use of DSRC protocol on CV applications, then the motivations behind using simulations instead of real world testbed is described and the importance of having traffic network simulator and a wireless network simulator in modeling vehicular communication systems is explained.

The importance of using traffic network simulator to represent the details of the transportation network consisting of arterial streets, freeways and different types of vehicles stopping at traffic lights and abiding by traffic rules is highlighted. Then different types of traffic simulators are explored, concluding with the need for a microscopic traffic simulator in our work. Subsequently, we also highlight the importance of having a wireless network simulator to model communication specifications such as network protocols.

Then different types of communication network simulators are reviewed. In vehicular networking and ITS applications, it is necessary to simulate the wireless system performance under realistic traffic conditions. As a result, there have been many research attempts to combine the benefits of both these simulators. We also reviewed various researches done in this field to consider the need for representing vehicular movement in wireless network simulators.

In the chapter 3, we used the lesson learned from the chapter 2 and describe the methodology of designing and implementing of CV system using two simulators in details.

## Chapter 3: Methodology

### 3.1 Introduction

In this chapter we explain the design and implementation of CV system to determine the DSRC based V2V and V2I communication in two perspectives: traffic simulator itself with the use of added APIs and the integration of traffic simulator and wireless communication simulator. The reason is that PARAMICS and OPNET solely do not have the ability to model the proposed CV system and satisfy our goals from its implementation.

One of our set of goals is to apply ITS applications, which use ICT, to improve transportation system safety, mobility and traffic efficiency. Therefore, PARAMICS, which is designed to replicate car-following, lane-changing, gap-acceptance and other driver behaviors and is capable of simulating and testing advanced traffic systems, is used to model CV system. However, as mentioned before, this microsimulation package does not have the ability to simulate CV system or generate incidents. So we designed our proposed system using MaSE methodology and then implement it using APIs. This extension is then is used to evaluate impacts of CV on safety and mobility indices by calculating travel time and crash likelihood measurements.

We utilized the MaSE methodology, step by step, to model CV as a multi-agent system (MAS). Since our proposed system is consists of several modules, such as CV and infrastructures, which send messages to each other in order to connect and transmit warning messages, we selected MAS as the approach to the system. MAS composed of multiple interacting intelligent agents used to solve issues which are hard for an individual agent or module to solve. The primary focus of MaSE is to help a designer take an initial set of requirements and analyze, design, and implement a working MAS. The proposed design extends the abilities of PARAMICS to simulate CV systems.

PARAMICS has a C++ language interface that provides a library of functions allowing interactions with a range of model elements such as links and vehicles. Given that, we implemented the MaSE artifacts as extensions for the PARAMICS using two APIs to add the ability to simulate CV systems. API#1 is programmed to create incidents in the PARAMICS network to study the system under hazard situations. API #2 then attempts to increase driver



awareness of the upstream and downstream vehicles through V2V and V2I communications and thus reduces the probability of a secondary collision.

Our second set of goal is to optimize the different DSRC ranges, represent critical communication network elements, such as physical layer noise, delay, fading, data packet collision or loss, throughput, and loads to test and detect failure or latency of V2V and V2I communication. These elements were found to be critical factors on the evaluation results of CV. However, PARAMICS itself does not have the ability to assess them in a realistic representation of CV applications leading to the need of creating a connection between traffic simulators and wireless communication simulators to examine essential issues concerning both traffic and wireless networks.

Therefore, we coupled PARAMICS traffic simulator to OPNET Modeller Wireless Suite as the wireless network simulator to provide modeling, simulation, and analysis of wireless networks in CV systems. We selected OPNET since literature review did not find any study investigating on creating connection between OPNET and PARAMICS. Perhaps it is because OPNET is only free for academic programs. In OPNET multiple scenarios can be modeled for a network which leads to a useful comparison between various network behaviors. Consequently, we did not find a study, evaluating the IEEE802.11p using WLAN protocol as the DSRC range in V2V and V2I communications in integrated simulators.

The results of our two goals are extracted from 3 evaluating scenarios which will be presented in the chapter 4.

## **3.2 MaSE Methodology**

The MaSE methodology is a detailed approach which will cover information needed in models. Also it is a top-down methodology for analyzing, designing, and developing heterogeneous MAS. MaSE views MAS as a supplementary abstraction of the object-oriented model which applies its techniques to the specification and design of MAS.

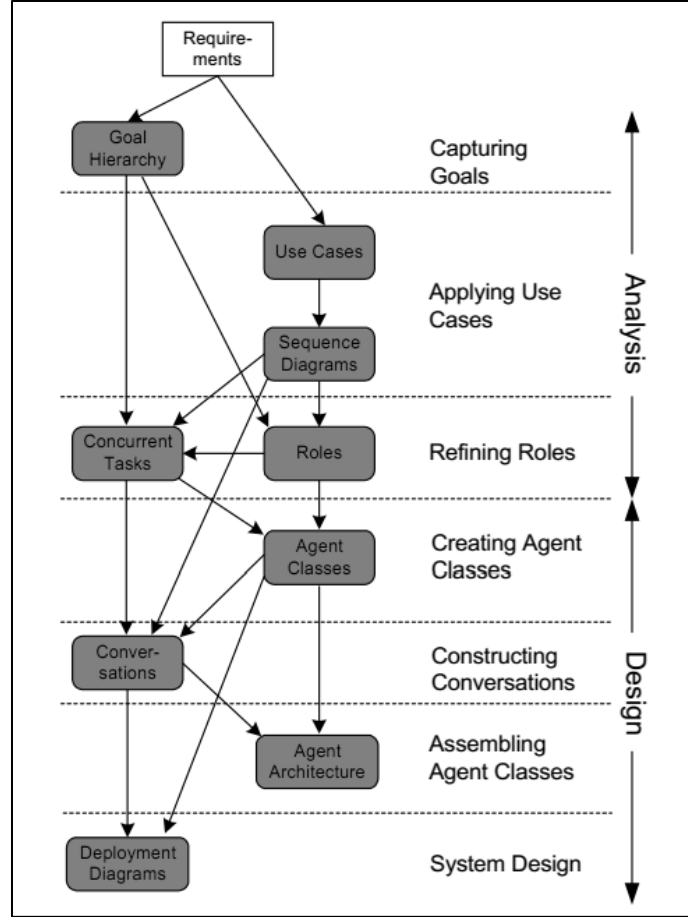
In the MAS, agents are specialization objects. Agents coordinate their actions via conversations to attain individual and community goals instead of objects whose methods are invoked by other objects [114].

The main advantage of MaSE over other methodologies for analysis and design of MAS, such as Gaia [115], is its scope and completeness. The major strength of MaSE is the ability to track changes throughout the process [114]. Every object which created during the analysis or design phases can be traced forward or backward to other related objects. For instance, a goal extracted from the “Capturing Goals” step can be tracked down to a specific role, task, and agent class. Similarly, an agent class can be traced back throughout tasks and roles to the fulfilled system level goal.

MaSE uses the MAS concept to develop distributed software systems to achieve the goal and uses graphical models to describe the types of agents in a system, their interfaces to other agents, and detailed definitions of the internal agent design.

MaSE takes many ideas and combines them into a complete methodology [116, 117]. Its methodology is the foundation for the agentTool development system [114, 118], which also serves as a validation platform and a proof of concept. The agentTool system is a graphical, interactive software engineering tool for the MaSE methodology. AgentTool supports the analysis and design in each of the seven MaSE steps, (as illustrated in Figure 3.1), as well as confirmation of communications in agents and code generation for multiple MAS structures.

The MaSE methodology and agentTool are independent of any specific agent architecture, programming language, or communication frame. MAS designed in MaSE can be implemented in several different ways from the same design. MaSE follows the phases and steps shown in Figure 3.1 [119]. The MaSE Analysis phase includes three steps: Capturing Goals, Applying Use Cases, and Refining Roles. The Design phase has four steps: Creating Agent Classes, Constructing Conversations, Assembling Agent Classes, and System Design. The rounded rectangles indicate the MaSE models used to get the output of each step while the arrows between them signify how the models affect each other. The intent is that the analyst or designer be allowed to move between steps and phases in a way that additional detail can be added and finally, a complete and consistent system design can be produced with each consecutive pass. The drawn diagram for every step is based on MaSE particular notations.



**Figure 3.1 MaSE phases [119]**

### 3.3 Analysis and Design of Connected Vehicles in PARAMICS

In order to implement CV in PARAMICS, first the MaSE methodology was employed, step by step, to model CV as MAS. Then by using two APIs the output design from MaSE was implemented and extended the PARAMICS capabilities to simulate CV systems. This section provides the detailed explanation of classes and their functions.

Our proposed CV multi-system includes 4 independent agents, which can offer different functions and simulations. In this modeling process, there will be V2V, Non V2V, Database and PARAMICS agents.

### **3.3.1 ANALYSIS**

The MaSE Analysis phase consists of three steps: Capturing Goals, Applying Use Cases, and Refining Roles. The overall approach in the Analysis phase is defining the system goals from a set of functional requirements and then defining the required roles to get those goals. While it is possible to map directly from goals to roles, MaSE suggests the use of Use Cases to help validate the system goals and acquire an initial set of roles.

#### **3.3.1.1 Capturing goals**

The proposed system consists of four components: CV, non-CV, RSU and VMS. CV is equipped with a wireless system to create V2V and V2I connection. RSU, the infrastructure placed within the network, is capable of collecting information about traffic conditions and communicate with the CV located within a DSRC range, and sending this information to the VMS. VMS located in network to show warning messages and traffic information collected by RSU to communicate with non-CV. Non-CV do not have the equipment to connect with other vehicles or RSUs. Messages are issued by CV to disseminate important information in a specific point of a road, such as occurrence of collision, presence of lane blockage, etc.

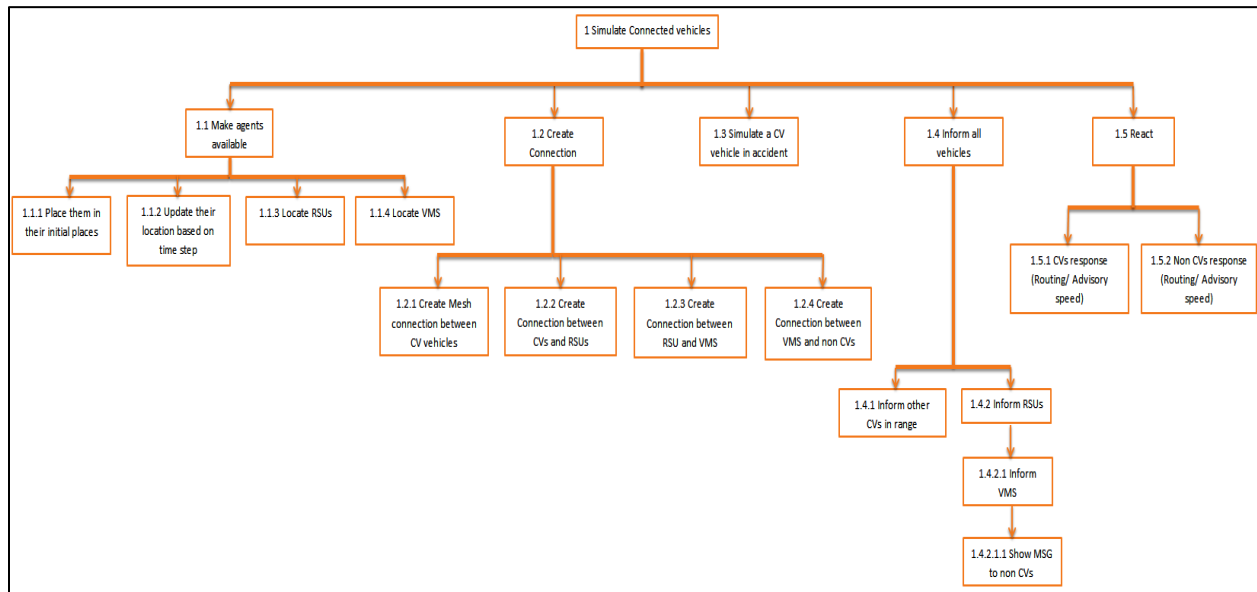
In the first step, a set of functional requirements was created:

1. Two types of vehicles are generated in the network (CV/Non-CV).
2. CVs can communicate with each other (V2V) and with RSUs (V2I).
3. Non-CVs have a virtual connection with VMSs to simulate that they can read the VMS message and then response to it.
4. RSUs and VMSs are linked.
5. If a vehicle is in accident, its flag is set to true.
6. All the relevant CVs and RSUs, within DRSC range, are informed about the accident.
7. Non-CVs are notified about the accident via VMS.
8. In light of the information disseminated, some drivers might change their route or speed since their drivers' awareness and aggression are improved.

From the requirements these goals are extracted:

1. Generate vehicles (CV/Non-CV) and infrastructures (RSU/VMS).
2. Create connections.
3. Transmit messages.
4. Change route/speed.

Next, the goals are analyzed and put into a hierarchical form. A Goal Hierarchy Diagram is a controlled graph where the nodes shows goals and the arcs indicate a sub-goal relationship. The overall system goal is placed at the top of it, which is simulating CV as an independent agent added to PARAMICS module. Once a basic goal hierarchy is in place, goals may be disintegrated into new sub-goals and each sub-goal must support its parent goal. System goals are represented in a goal hierarchy diagram in Figure 3.2.

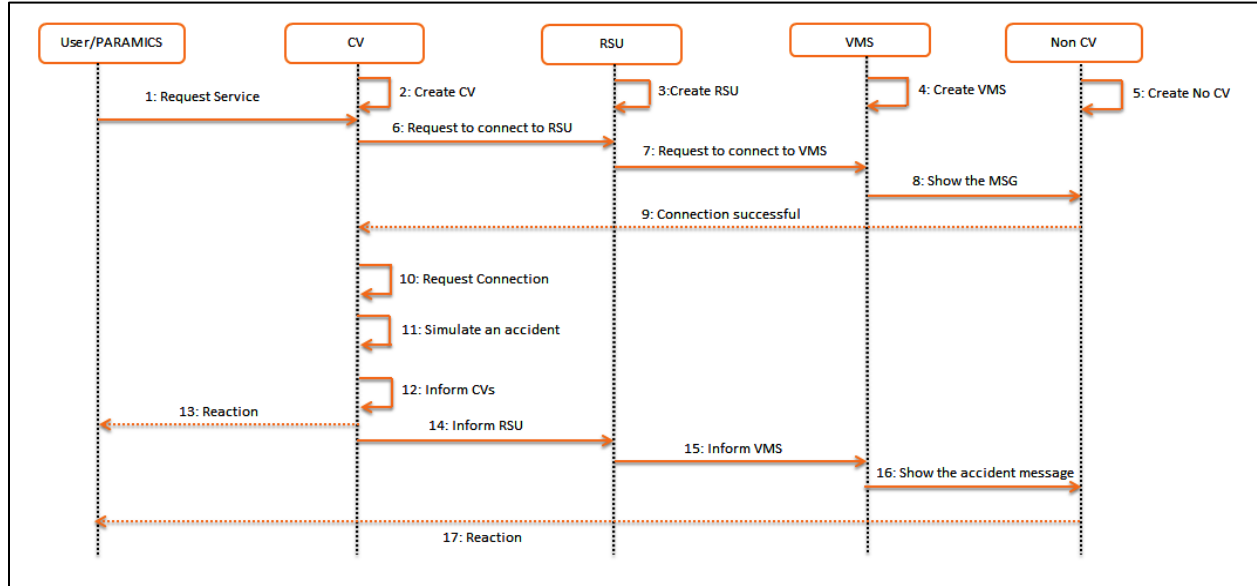


**Figure 3.2 Goal Hierarchy Diagram**

### 3.3.1.2 Applying Use Cases

The objective of the Applying Use Cases step is to seize a group of use cases from the initial system context and create a set of Sequence Diagrams to determine an initial set of roles and communications links within the system. Use cases of our system are extracted from the system requirements, describe sequences of events which show system behavior and are examples of

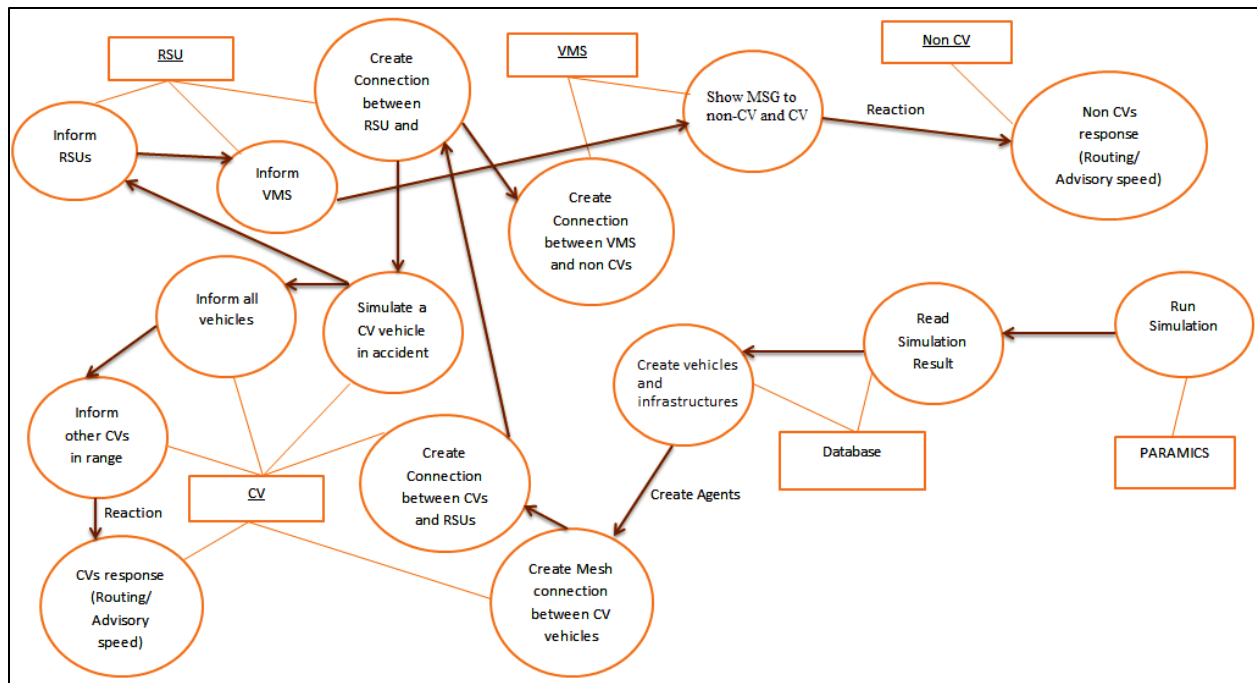
how system should behave. By creating use cases paths of communication can be established. Figure 3.3 shows the sequence diagram and how each goal can be accomplished.



**Figure 3.3 Sequence Diagram [36]**

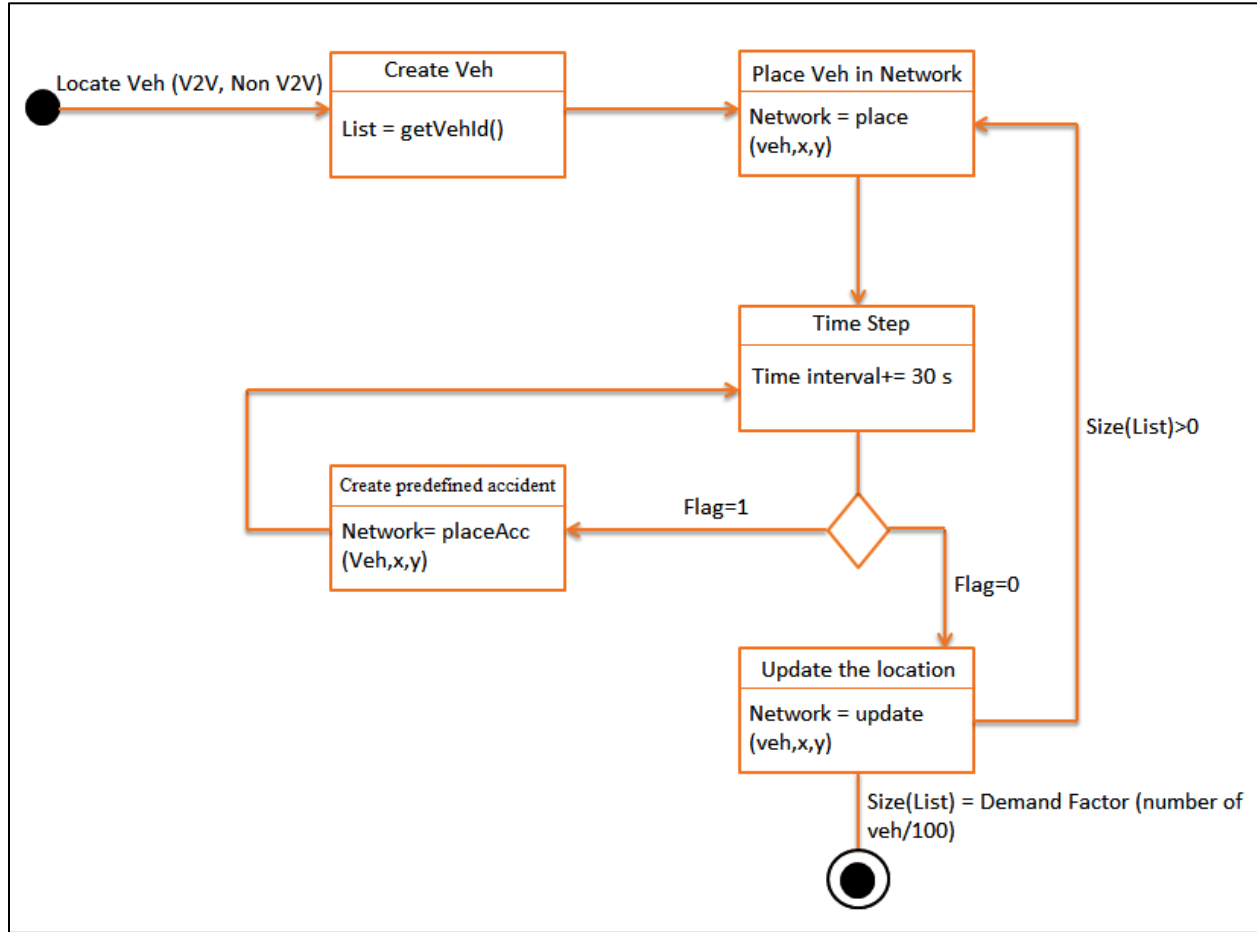
### 3.3.1.3 Refining Roles

The purpose of the Refining Roles step is to convert the Goal Hierarchy Diagram and Sequence Diagrams into roles and their relevant tasks, which are more suitable for designing MAS. MaSE Role Model includes information on interactions between role tasks and is more complex than traditional role models [120]. From Role Model, the definitions of roles can be determined. Figure 3.4 illustrates system role model diagram that shows CV systems' roles and behaviours in PARAMICS.



### Figure 3.4 System Role Model

**The concurrent task model** provides a built in timer activity. An initial Concurrent Task Model can be built from the scenarios of creating CVs and creating incidents by taking the sequence of messages sent or received by the roles of CV/non-CV and use them to create a sequence of corresponding states and messages. After creating the initial concurrent task diagram, the internal processing, which should be performed by role to be able to satisfy the use case, is determined. This internal processing is represented as activities within the existing states. The information about the data passed in the messages and additional messages required for information exchange was also occupied. Concurrent tasks in our system are defined in Figure 3.5 and are specified as finite state automata, which consists of states and transitions.



**Figure 3.5 Concurrent Task Diagram**

### 3.3.2 DESIGN

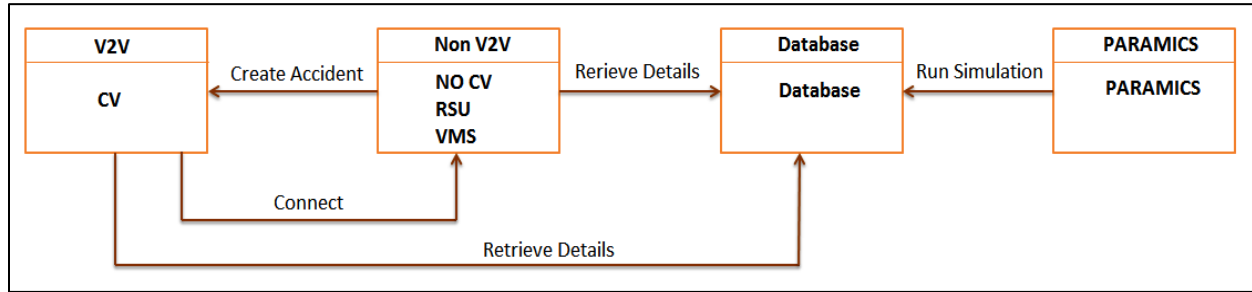
There are four steps to design a system with MaSE methodology. In the first step we assigned roles to the 4 agent types in the system to create Agent Classes. In the second step, Constructing Conversations, the actual conversations between agent classes are defined. In the third step, assembling Agents Classes, the internal architecture and reasoning processes of the agent classes are designed. Finally the number and location of agents in the deployed system are defined. Each of these steps is discussed below.

#### 3.3.2.1 Creating Agent Classes

In the Creating Agent Classes step of the Design phase, agent classes are created from the roles defined in the Analysis phase. The result of this phase is an Agent Class Diagram, which



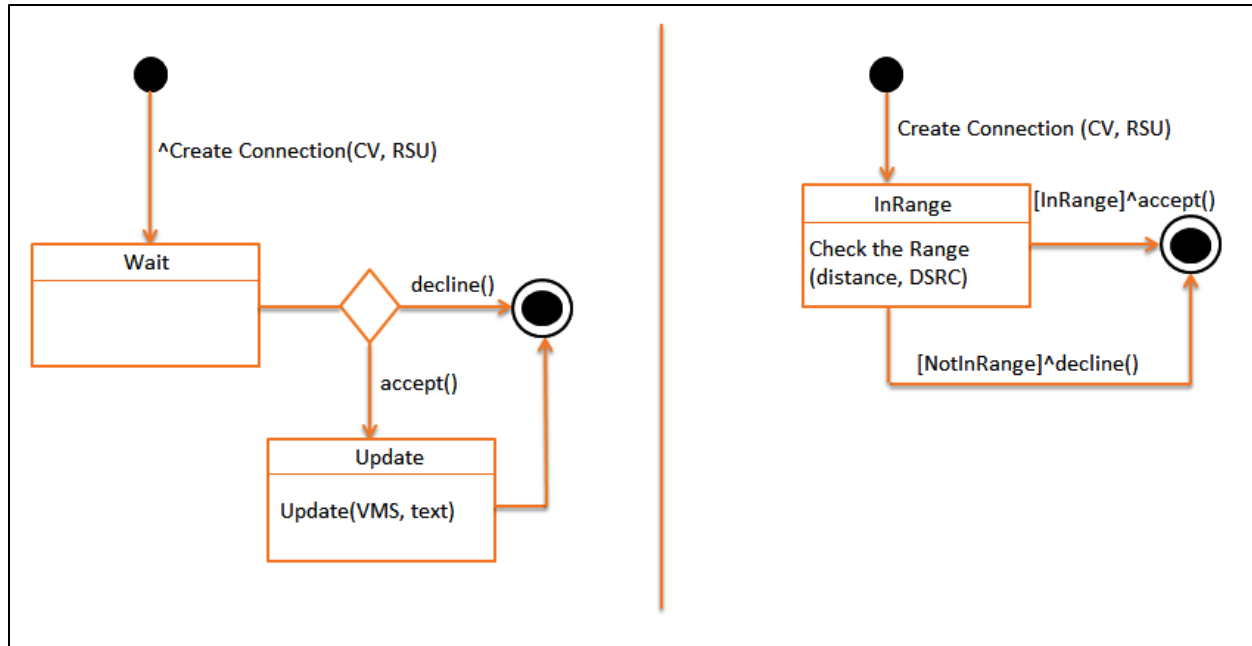
illustrates the overall agent system organization consisting of agent classes and the conversations between them. At this point, the roles and tasks, which an agent class must play, are determined. The conversations, that an agent class must participate in, are derived from the external communications of the agent roles. The Agent Class Diagram is the first design object in MaSE that shows the entire MAS in a way it can be implemented. The CV MAS mainly defines 4 agent classes as shown in Figure 3.6. An agent is an actual instance of an agent class.



**Figure 3.6 Agent Class Diagram**

### 3.3.2.2 Constructing Conversations

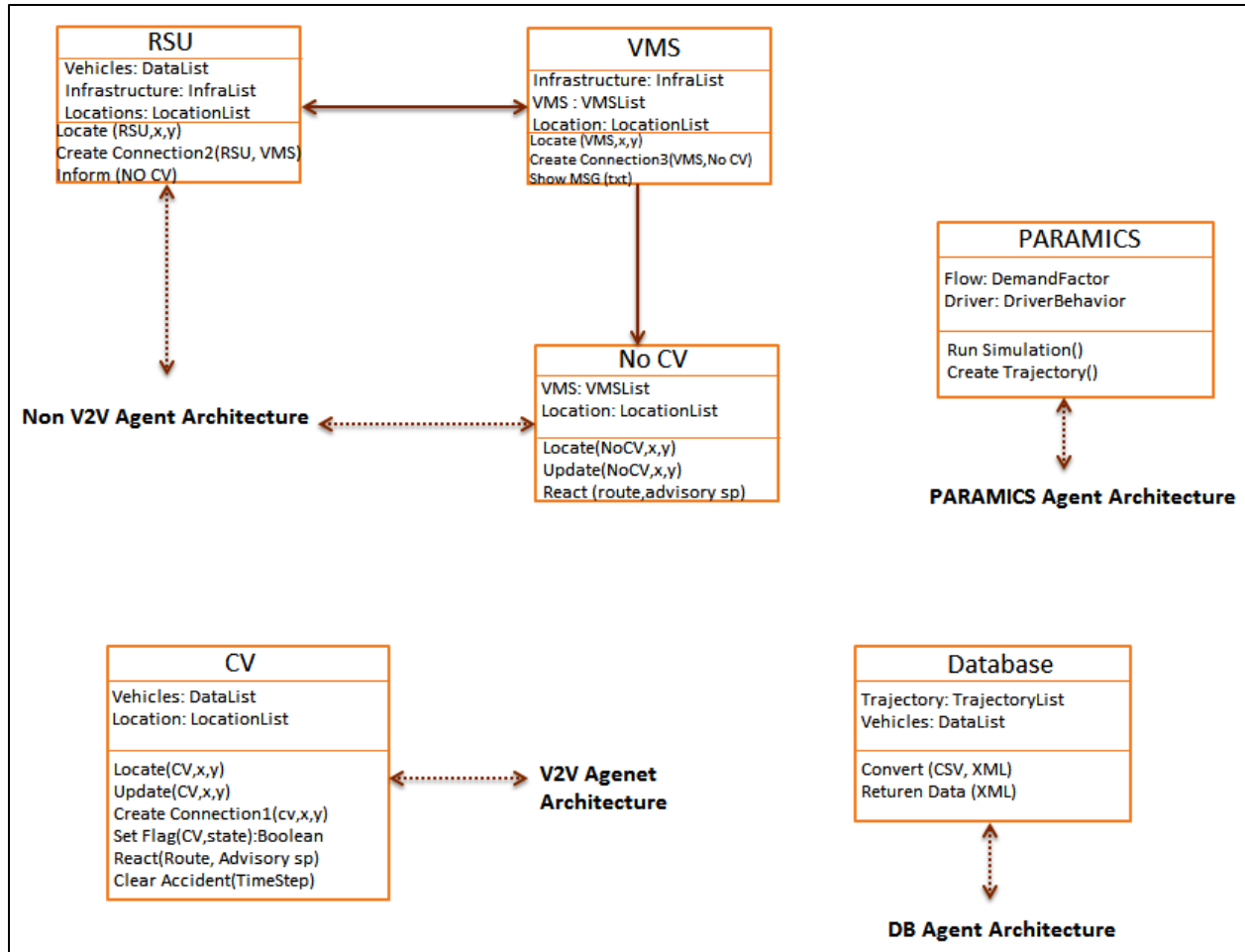
A MaSE conversation defines an organized procedure between two agents. The Communication Class Diagram, as shown in Figure 3.7, is similar to a Concurrent Task Model and defines the conversation states of V2V and non-V2V as the two participant agent classes. This conversation is about constructing connection between CVs and RSUs. CV, the initiator, begins the conversation by sending “create connection” as the first message. When the agent receives the message, it checks if CV is in range of corresponding RSUs. Only the vehicles which their distance from RSUs fall into DSRC range can establish connection. Otherwise, the request will be declined and RSU cannot get the message from the CV and show it on VMS.



**Figure 3.7 Create Connection conversation initiator and responder**

### 3.3.2.3 Assembling Agents

During the Assembling Agents step of the Design phase, the internals of agent classes are created. This is accomplished via two sub-steps: defining the agent architecture and defining the components that make the architecture. Components consist of attributes, methods, and may have sub-architecture. Internal component behaviour may be depicted by formal operation definitions and state-diagrams which state happened events happened between components. Basically, each task from each agent role defines a component in the agent class. Figure 3.8 shows agents' architecture of CV MAS.

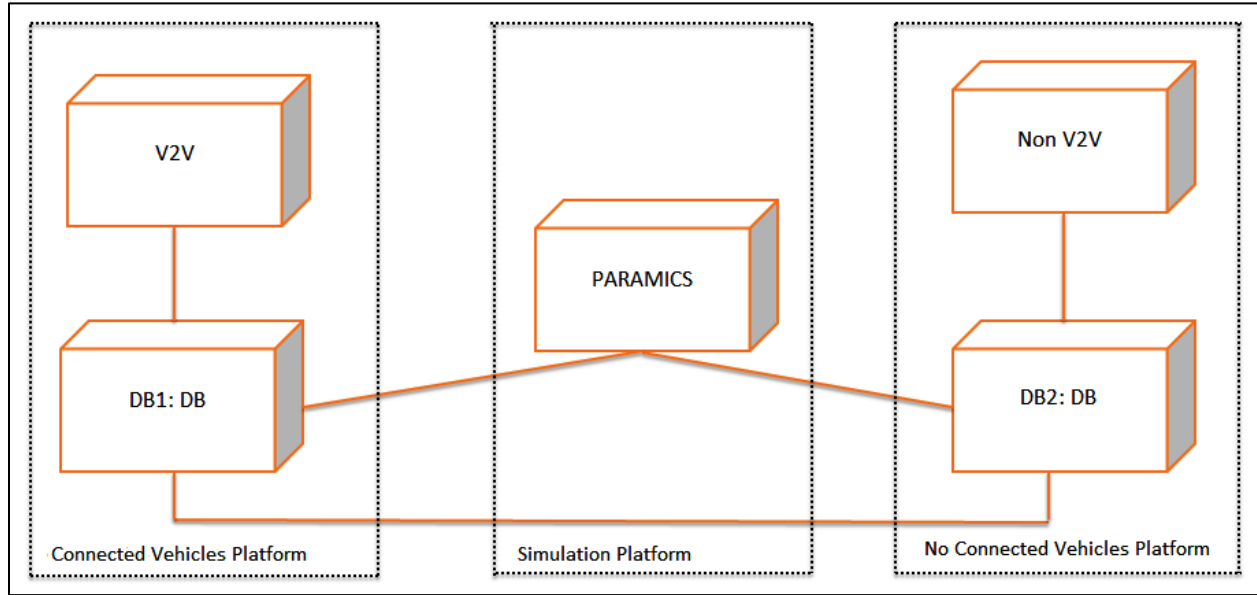


**Figure 3.8 System Architecture for agent class**

### 3.3.2.4 System Design

The last step of the MaSE methodology takes the agent classes defined before and expresses actual agents using Deployment Diagram to show the numbers, types, and locations of agents within a system. This is similar to instantiating objects from object classes in object-oriented programming. Deployment Diagrams describe a system based on predefined agent classes. Strength of MaSE is that a designer can make these modifications after designing the system organization, thus generating a variety of system configurations. System Deployment Diagram includes 5 agents, there into, V2V Agent and DB1 Agent run in the same physical node, Non V2V Agent and DB2 Agent in one node, and PARAMICS Agent run in the other physical node, as shown in Figure 3.9. PARAMICS agent acts as the PARAMICS software itself where the other agent classes should be implemented and connected to it. The Connected Vehicles and No

Connected Vehicles platforms should be added to the simulation platform to apply CV system as the extension to the software. This is done by programming APIs.



**Figure 3.9 System Deployment Diagram**

### 3.4 Implementation using Application Programming Interface

By using MaSE methodology, the essential agents, relationships and conversations between them, and the particular parameters of proposed CV system were identified. Moreover, MaSE specified how every agent should act in possible scenarios (Figure 3.3) to eventually reach the requirements, goals and sub-goals defined in Analysis phase (Figure 3.2). This specified design is implemented by adding two APIs to the PARAMICS programmer module. In API #1, the connection between PARAMICS agent to V2V and Non-V2V agents was created through the use of Database. However, the main duty of API #1 is to generate predefined or randomly [121] incidents for running test case scenarios, explained in detail in next chapter.

Most of the proposed design by MaSE methodology is used in API #2, where both V2V and No-V2V agent classes and their relevant agents (i.e. CV, non-CV, RSU, VMS) were created (Figure 3.6). Additionally, the roles and tasks for each agent class and the conversations between these agent classes were programmed, such as the example conversation showed in Figure 3.7. It specifies the procedure of creating connection and message transmission between CVs and RSUs

when they are in predefined DSRC range, and then showing the message context on the VMS. API #2 consists of 2 packages: V2V package in which V2V agent class was implemented and V2I package where the Non-V2V agent class were created.

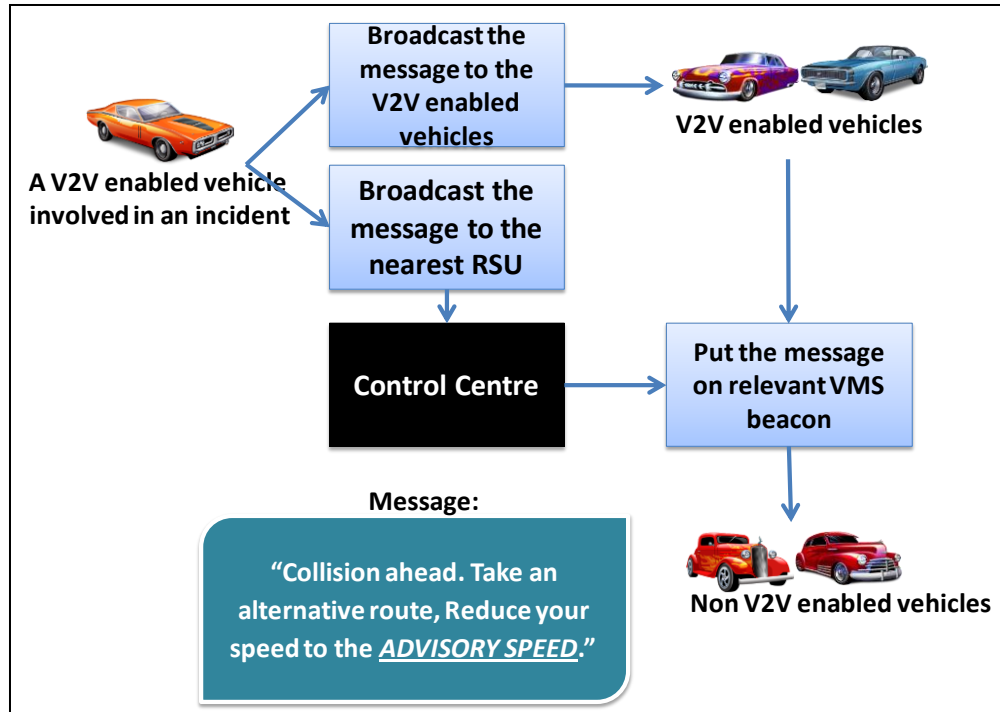
Once incidents were created (randomly or specifically) applying API #1, API #2, accordingly simulates the broadcasting of V2V and V2I communication in the network and models its effects on drivers' behavior. This API was developed to allow vehicles communicate with each other and infrastructures to apply the connection and conversations between V2V and Non-V2V agent classes.

When a CV is involved in an incident, API #2 turns the vehicle's color to red to indicate this condition. This allows users to visually pick out which vehicle on the studied network was involved in the incident. API #2 then calculates the distance of surrounding CVs, sends them messages to notify them of the incident ahead and provides an advisory speed. At this point, all notified CV will have improved awareness and decreased aggressiveness (vehicle parameters that programmed intentionally). These two modules will be set randomly for different V2V enabled vehicles. In other words, their awareness ranges between 6 and 9, whilst their aggression rate will be between 1 and 4 (the threshold of awareness module is 1-9).

API#2 has also been developed enabling CV to communicate to RSUs to simulate the connection between V2V and No V2V agent classes in V2I package. When CVs encounter an incident, they send a message to the nearest RSU in DSRC range, along the road, to transmit the location and the type of the incident to the unit. The corresponding RSU will forward aforementioned information to the control center. The control center defined as a decision maker chooses which VMS should be enabled to reflect the message to the upstream and downstream vehicles and what must be the context of message.

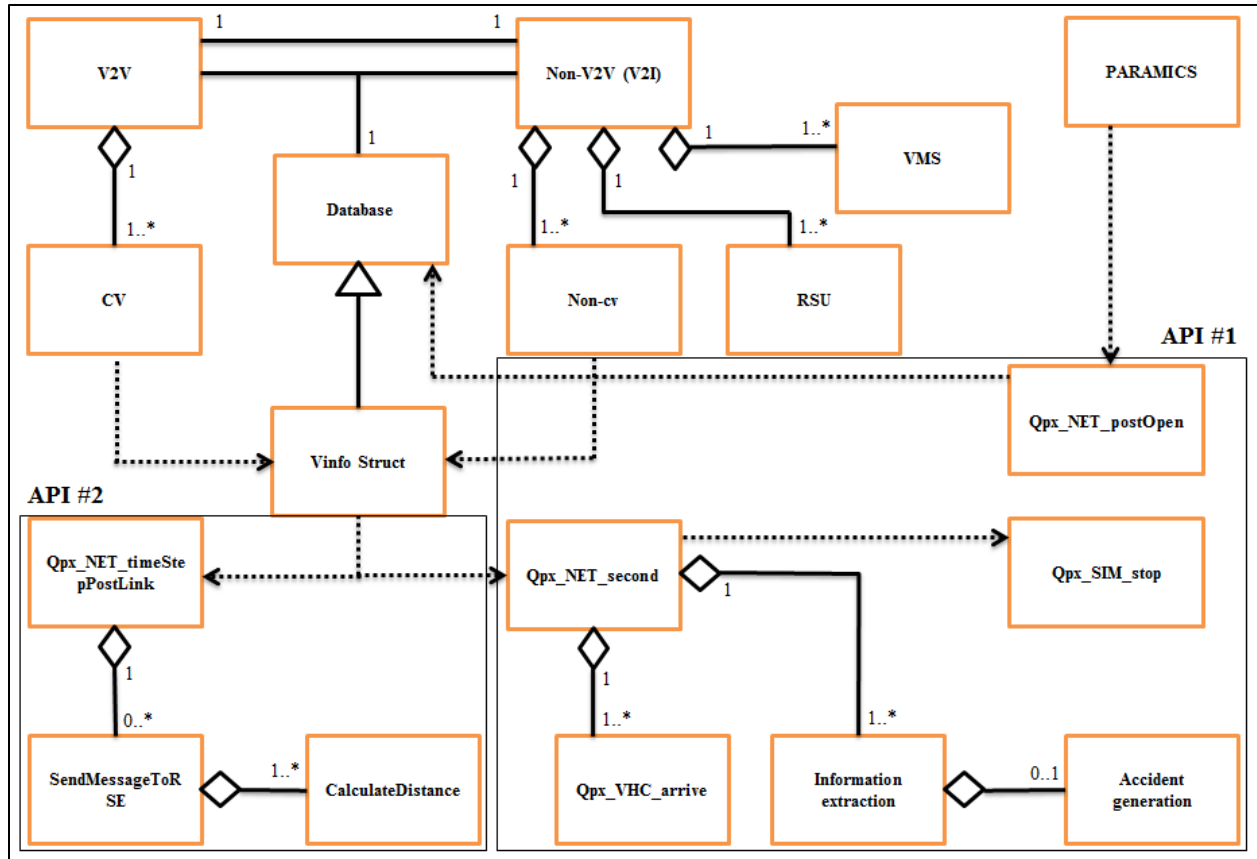
Usually the message contains warning information for drivers to let them know that there is a collision ahead alongside an advisory speed message to prevent the occurrence of a secondary collision. The advisory speed is lowered in increments as the distance from collision decreases and it is increased for downstream of the location of the incident to let the traffic around the incident be cleared easily.

When the traffic is restored to its normal condition and the queue created by the incident is cleared, the incidents messages will be removed from the affected VMSs (Figure 3.10).



**Figure 3.10 Illustration of the CV simulation developed by PARAMICS APIs [4]**

The class of functions involved in creating incidents, information storage, information retrieval and sending V2V and V2I messages will be described in the following sections. The developed APIs along CV agents is shown in Figure 3.11.



**Figure 3.11 Implemented APIs using CV agents**

### 3.4.1 API #1: Incident creation

API #1 creates incidents and requires a set of functions to extract vehicle information from the network. It retrieves and stores the simulated vehicles' information from PARAMICS and is completed through the vehicle map. Each vehicle's information is stored in the Vinfo struct class. Vinfo struct consists of information related to vehicles including: Float x, Float y, Float z, Int ID, Float bearing, Float gradient, Float length, Float width, Int age, Bool incident, Int usertag, Bool stopped, Bool collide\_stop, Float stop\_time, Float distance, VEHICLE \*vpoint, LINK \*link, Int lane.

API #1 also creates a readable database of vehicles, checks vehicle location for any incident that might have occurred, and finally identifies and registers the incidents. A majority of its classes of functions are described below.

**Qpx\_NET\_postOpen** is a PARAMICS extension (prefix qpx) built-in function. When a model is first opened in PARAMICS, the postOpen function is called. It pulls the information for each vehicle and stores it in the Vinfo structure belonging to that vehicle each time step. If a vehicle does not already exist in the map, the Vinfo structure is added to the map. The total number of links and lanes in the network is recorded by its functions and vehicles and infrastructures are generated in the traffic network.

**Qpx\_NET\_second** is executed every second when the simulation is running. It is essentially the core of the program. The subroutine includes all of the functions in API#1.

**Qpx\_VHC\_arrive** is called every time a vehicle arrives at its destination zone. It removes the vehicle from the map structure.

**Void informationExtraction** sets up the vehicle x, y coordinates. The vehicle location is where vehicle information is stored. API #2 retrieves vehicle information from the vehicle map.

**Void accidentGeneration** checks the entire vehicle map for any incidents that have occurred (based on incident flags) and determines if an incident should occur. It can be several numbers of incidents based on the probability that user provided when the model is first loaded [121] or a predefined incident.

**Bool accidentFlag** is called when the incident happens. It sets the incident flag of the corresponding vehicle to true.

### 3.4.2 API #2

API #2 retrieves CV information from the Vinfo struct. The retrieved information includes coordinate information for calculation of the DSRC range. It then searches for occurred incidents in the network and sends warning messages to surrounding vehicles and the nearest RSUs fall in DSRC protocol. These messages increase driver awareness and enable them to avoid possible consequent collisions. When the incident has been totally cleared, the situation resumes to the initial state.

To facilitate flexibility to PARAMICS networks, RSUs are further modeled using PARAMICS' Beacons objects. These are predefined objects representing points within a



network where information can be delivered to drivers. While they are normally used to model VMS, they have by default no explicit functionality and can be programmed to execute any suitable task. For each RSU, the communication range is currently determined using a simple distance radius.

#### **3.4.2.1 V2V package**

V2V package first adds a number of vehicles to the V2V map as CV. This number is based on the penetration rate defined at the start time of the simulation. Next it must determine whether an incident occurred or not. It decides based on distance from the accident, which vehicles should be notified. The incident range will be calculated and vehicles within the DSRC range will be informed. Vehicles notified of the incident ahead will change colour to show the user that V2V is occurring.

**Qpx\_NET\_timeStepPostLink** checks every link and adds CV to the V2V map and infrastructures to the network. The V2V map only consists of CV. It updates vehicle positions in the V2V map every time step and checks the incident flag for each vehicle. If there has been an accident, it calls the SendMessage class of functions. If accident has been cleared, it gets back to the initial state.

**Void SendMessage**, when a vehicle is involved in an accident, its Vinfo struct is sent to the SendMessage. It iterates through the V2V map and calls calculateDistance. If any CV is within the DSRC range (e.g. 1000m) of the vehicle involved in an accident, that vehicle is signaled of the danger ahead. With the increased driver awareness, the vehicle does not partake in the same incident.

**Float CalculateDistance** receives vehicles' Vinfo struct classes, calculates and returns the distance between the two V2V enabled vehicles.

**Bool inRange** is called by SendMessage and checks every vehicle's Vinfo in the map to find out which ones are in the same road with the vehicle in accident. It uses bearing in the Vinfo and outputs from CalculateDistance function.

**Qps\_VHC\_awareness** is used to set the awareness value stored with a vehicle. It sets the awareness value of all notified CV to a number in [6 – 9] threshold.

**Qps\_VHC\_aggression** is used to set the aggression value stored with a vehicle. It sets the aggression value of all notified CV to a number in  $[1 - 4]$  threshold.

**Qps\_DRW\_vehicleColour** is used to set the colour of a specific vehicle. It changes the color of the vehicles involved in accident to red and all the notified CV to blue to make them noticeable in visual simulation.

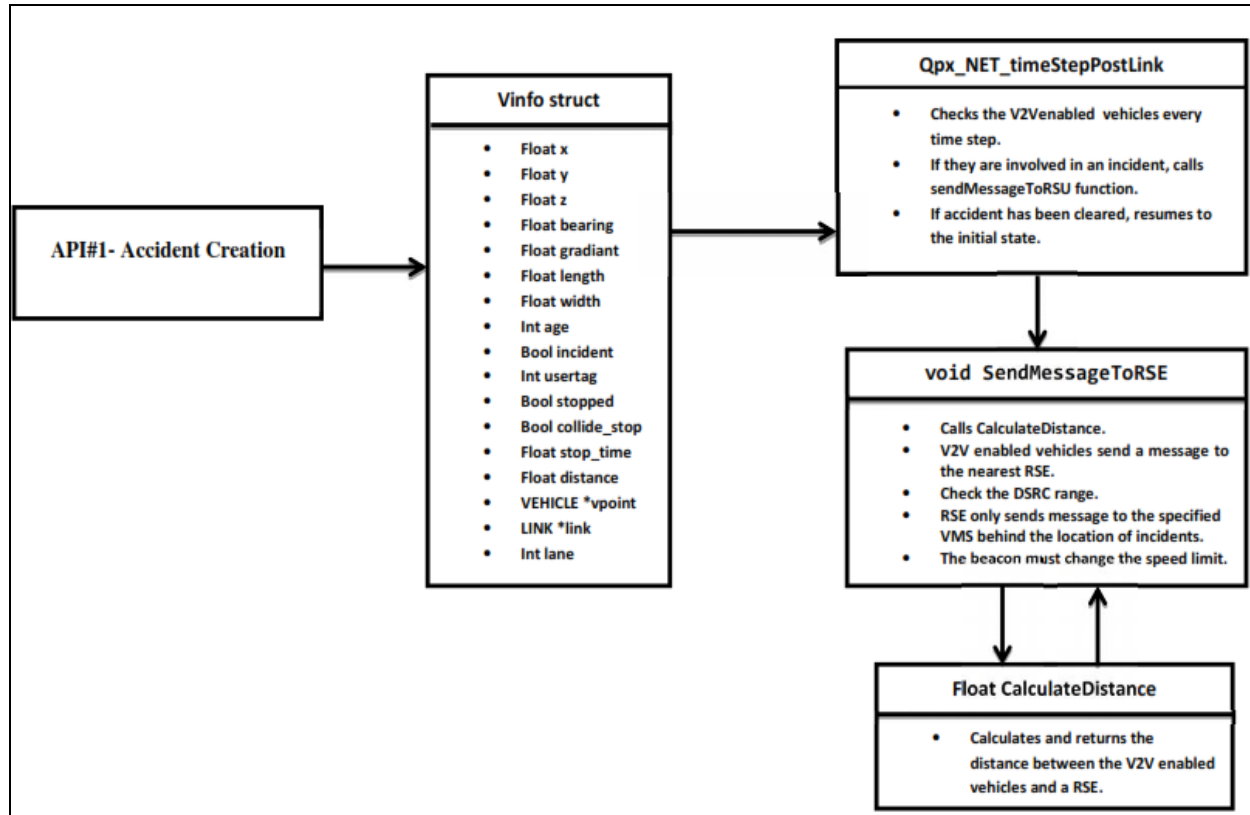
### 3.4.2.2 V2I package

Based on the DSRC, CV involved in an incident also sends message to RSU, RSU sends it to control center and control center decides about the advisory speed and which VMS should show that. It should be noted that the range of DSRC standard can be changed accordingly for experimenting its effects on CV system (Figure 3.12).

**Void SendMessageToRSU** is basically same as SendMessage in V2V package; however it creates communication between CV and RSU.

**Bool inRange** is called by SendMessageToRSU to check which RSUs are in the same road with the CV in accident and determine if beacon should receive the message. Beacon should be within DSRC range of the CV.

**Float CalculateDistance** receives vehicles' Vinfo struct classes, calculates and returns the distance between CV and RSUs or VMSs.



**Figure 3.12 Extended API for V2I implementation**

In order to implement advisory speed, API was used to allow for placement of VMS in the network to display the posted advisory speed reduction so that drivers can change their speeds accordingly as they pass the VMS beacon. The advisory speeds can differ depending on the location of VMS:

1. **Upstream of the incident location:** In [15], authors found that gradually changing speed is better than abruptly; thus the advisory speed, for traffic upstream of the incident, ranges from 60 km/h to 100 km/h. Vehicles nearby incident are advised to reduce their speed to 60 km/h, vehicles 500 meters behind should reduce their speed to 70 km/h, 1,000 metre behind to 80 km/h, 1,500 meter behind to 90 km/h and 2,000 meter behind to 100 km/h.
2. **Downstream of the incident location:** The downstream traffic is advised to increase their speed up to 110 Km/h to accelerate traffic flow. The vehicles should clear up the

space near the incident location to let the vehicles upstream of the accident easily pass the situation. Therefore they should be even advised to increase their speed [122].

The control center is the main decision maker. It sends an advisory speed to each VMS beacon falling in the DSRC range along the freeway. The message “Incident ahead” is common for all VMS beacons falling in the DSRC range; whilst, the advisory speed varies according to the VMS beacon’s distance from the incident. Consequently, non CV vehicles will be aware of the traffic situation along the freeway very shortly.

**Qpg\_NET\_beacons** returns the number of beacons in the network.

**Qps\_BCN\_message** is used to display a message on the specified VMS beacon. Such as "Speed Limit: 37.3mph".

### 3.5 Simulation Configuration in PARAMICS

For the simulation in PARAMICS, two types of vehicles are provided, CV and non-CV. Although physical attributes are the same for them (i.e. length, width, height, mean age, weight), their demand and assignment and kinematics are different. Here are the features whose default values are overridden to better simulate CV scenarios.

**Proportion (%)**: the relative number of particular type of vehicle (expressed as a percentage). By changing the proportion, different percentage of vehicles can be simulated in the network.

**Familiarity (%)**: the percentage of vehicles with familiar routing capabilities for every vehicle type. Each vehicle in the simulation is designated as having either a familiar or unfamiliar driver. This will affect their route choice decisions for a given OD trip as the simulation runs.

CV system is designed in a way that drivers of CVs will have more frequency in updating their travel time information than non-CV since they are more aware of road conditions with V2V communication. This is modelled in PARAMICS by assigning higher frequency of feedback and familiarity % to CV-drivers.

**Mean driver reaction factor:** the mean reaction time of each driver, in seconds. The value is associated with the lag in time between a change in speed of the preceding vehicle and the following vehicles reaction to the change.

**Driver perception reaction time:** reaction time of the vehicles which can be different for vehicle types.

**Base parameters:**

- Start time: the time to start the simulation, relative to 00:00:00 on Monday of the simulation, in HH/MM/SS format.
- Duration: the length of the simulation, relative to the start of the simulation, in HH/MM/SS format.
- Seed: specifies the value used by the random number generator. Identical networks simulated using the same processor will return the same results using the same seed value.
- Time steps: specifies the number of discrete simulation intervals that are simulated per second. The simulation time steps determine when calculations are carried out during every second of simulation. The default time step is 2 which means that calculation are done every 0.5 seconds of simulation.
- Demand factor (%): it specifies the dynamic demand for the current simulation ranging from 0 to 200% of the current global demand. The same traffic demand inputs may result in different travel time patterns for different seed values due to the random nature of the microscopic traffic simulation model.

**Measurements:** it allows the user to specify which statistics will be gathered during the simulation. These statistics are used by Analyser module to examine the data gathered during a simulation. Here is the simulation attributes defined in measurements:

- Statistics collection warm up period: it determines the time after the simulation start time that the periodic statistics will start to be collected. If no warm up period is specified then statistics will be collected from the start of the simulation.

- **Statistics collection duration:** it allows the user to specify a different statistic collection end-point. If no duration is specified then statistics will be collected from the end of the warm up period till the end of the simulation.
- **Gather interval:** the frequency of Analyser data which is written out to the Analyser binary files. If no interval is specified Analyser statistics will not be recorded.

**Loop Detectors:** PARAMICS detector station is located in the through lanes for the presence detection of through vehicles and it covers all lanes of a link. There will be one file per gather-period per detector; this file will contain data for each lane that the detector occupies.

- **Gather interval:** the frequency of detector data which is generated. If no interval is specified statistics will not be recorded.
- **Selected loop detectors:** to include the detectors required for data collection the checkbox for each of the detectors should be selected.
- **Requested Statistics:** Each output file contains time, lane, vehicle type and vehicle ID. However, acceleration (mps), flow (v/hour), gap (s), headway (s), occupancy (s), and speed (kph) can be requested to be included in detector data.

**Trajectory:** it allows the user to export information on each vehicle as it moves through the simulation. It has several export types: Comma Separated Values (CSV), SSAM Trajectory (TRJ), 3DS Max Design Civil View (SIM), and PHEM (CSV). The output file includes data about: Type, Origin, Destination, Lane, Gradient, Bearing, Length, Speed, Acceleration, Link Name, Link Length, Link Speed, Link Gradient, Curved, Radius, Lanes.

- **Gather interval:** the frequency of trajectory data which is generated.
- **Sample period:** it includes start time and end time of the simulation which trajectory statistics are recorded.

**Speed:** the speed limit on the link can be defined in core attributes. If the maximum speed of the vehicle is higher than the speed limit, the vehicle will try to attain a speed that depends on the aggression value assigned to the particular vehicle. In free-flow conditions, some vehicles will exceed the speed limit, depending on the behaviour parameters assigned to the drivers.

## **3.6 Safety and Mobility applications**

When an incident happens for a vehicle in a roadway such as overheat or running out of gas, secondary events, including major car crashes, can occur. Such events have effects on the average peak-period delay, which is experienced by vehicles traversing on the roads, the average increase in travel time in freeways, and the number of injury crashes occurring during the incident in roadways.

Safety and mobility indices help quantifying the impact of CV system on reducing the number of crashes and the amount of travel time. The value of these measures are either directly extracted from PARAMICS Analyser module (i.e. mobility indicators) or calculated by models and formulas (i.e. safety indicators). To apply different safety models, its elements are extracted from Analyser module.

In this study, various experiments are investigated on the V2V and V2I functionalities to analyze their ability to reduce crash risk and improve travel time. Their effectiveness is evaluated by comparison crash risk measurement with and without CV for safety index. Further, the mobility impact is measured by the resulting changes in the average travel time of the vehicles traveling on the network.

### **3.6.1 Safety measurements**

Crashes on freeways are normally more hazardous than crashes on urban streets [123]. The historical records of crashes in North America [23] have reported that the proportion of fatal crashes on freeways and rural roads is significantly higher than the proportion on urban roads and streets. These statistics indicate that more efforts are required to reduce the number of freeway crashes.

Crash risk varies under different traffic and environmental conditions and identifies significant factors which are the reason of their occurrences. In general, as traffic flow becomes more congested, drivers are required to adjust speeds more frequently resulting in increased crash risk.

### 3.6.1.1 Crash likelihood measure

Authors in [16, 124, 125], developed statistical models to calculate a measurement for getting real-time crash potential. The model developed by [16], which is used to assess crash likelihood for one of the case study scenarios, is described below with details.

This model was separately designed for the moderate to high-speed and low-speed traffic speed pattern. The threshold for separating two patterns was set at 60 kph based on visual examination of traffic speed distributions. Above this speed, the moderate to high-speed model is used. It gets average occupancy and flow as the input. Below this speed, the low-speed model, including average volume, occupancy, and coefficient of variation in speed variation as the inputs, is used. The authors stated that these models can be used to assess the crash potential at any location using loop detector data and they are specific for every location which shows the risk of crash in that place. However, it can only be used to compare crash risk at the same station before and after implementing certain strategies.

**Crash prediction low speed model** - The low speed model is based on average speeds being below 60 kph. The crash probability is calculated by the following equation:

$$\text{Crash Likelihood} = 2.64827\text{LogCVSF2} + 0.88842\text{LogCVSF3} + 1.33966\text{LogAOE2} + 0.97766\text{LogAOH3} - 0.43603\text{SVF2} \quad (3.1)$$

Where:

*LOGCVSF2* is the log of the standard deviation of speed divided by the average speed at the station of interest 5–10 min before the time of interest.

*LOGCVSF3* is the log of the standard deviation of speed divided by the average speed at the station of interest 10–15 min before the time of interest.

*LogAOE2* is the log of average occupancy 0.5 mile upstream of the station of interest 5–10 min before the time of interest.

*LogAOH3* is the log of average occupancy 1 mile downstream of the station of interest 10–15 min before the time of interest.

*SVH2* is the standard deviation of volume (Veh /hr) 1 mile downstream of the station of interest 5–10 min before the time of interest.



**Crash prediction high speed model** - The high speed model is based on average speeds being above 60kph. Similarly, the crash probability is calculated by the following equation:

$$\text{Crash Likelihood} = -0.93423\text{LogAOF2} + 1.14584\text{LogAOH3} - 0.22878\text{SVH2} - 0.10055\text{AVG2} + 0.5932\text{AVE3} \quad (3.2)$$

Where:

*LogAOF2* is the log of average occupancy at the station of interest 5–10 min before the time of interest.

*LogAOH3* is the log of average occupancy 1 mile downstream of the station of interest 10–15 min before the time of interest.

*SVH2* is the standard deviation of volume 1 mile downstream of the station of interest 5–10 min before the time of interest.

*AVG2* is the average volume 0.5 mile downstream of the station of interest 5–10 min before the time of interest.

*AVE3* is the average volume 0.5 mile upstream of the station of interest 10–15 min before the time of interest.

This calculation model was used as the safety measure for assessing one of the case studies. The drawback of this work is that the model was designed specifically for a section of interstate 4 in Orlando, Florida, 36 miles from Tampa to Daytona. Therefore the ORCI is utilized for the second scenario as the primary measures of safety for rear-end and lane-change crash risks. This statistical model is capable of determining the crash risk on the freeway in real time and was initially proposed in [15].

### **3.6.1.2 Overall Risk Change Index (ORCI) measure**

The index denotes the change in rear-end and lane-change crash risk between any particular test case and the base case. The ORCI is calculated in the following manner: First, the crash risk is calculated for each 5 minute period at every location. Then, the crash risk at each location is averaged over the entire simulation length. Next, a plot of the average crash risk value vs.

location is created for the base case and the test case. The area between the two crash risks curve represents the ORCI. This measure is shown below.

$$ORCI = \sum_I [ \sum_{t=1}^T ((Risk\_profile)_{tI} / T)]_{Base} - \sum_I [ \sum_{t=1}^T ((Risk\_profile)_{tI} / T)]_{Test} \quad (3.3)$$

Where:

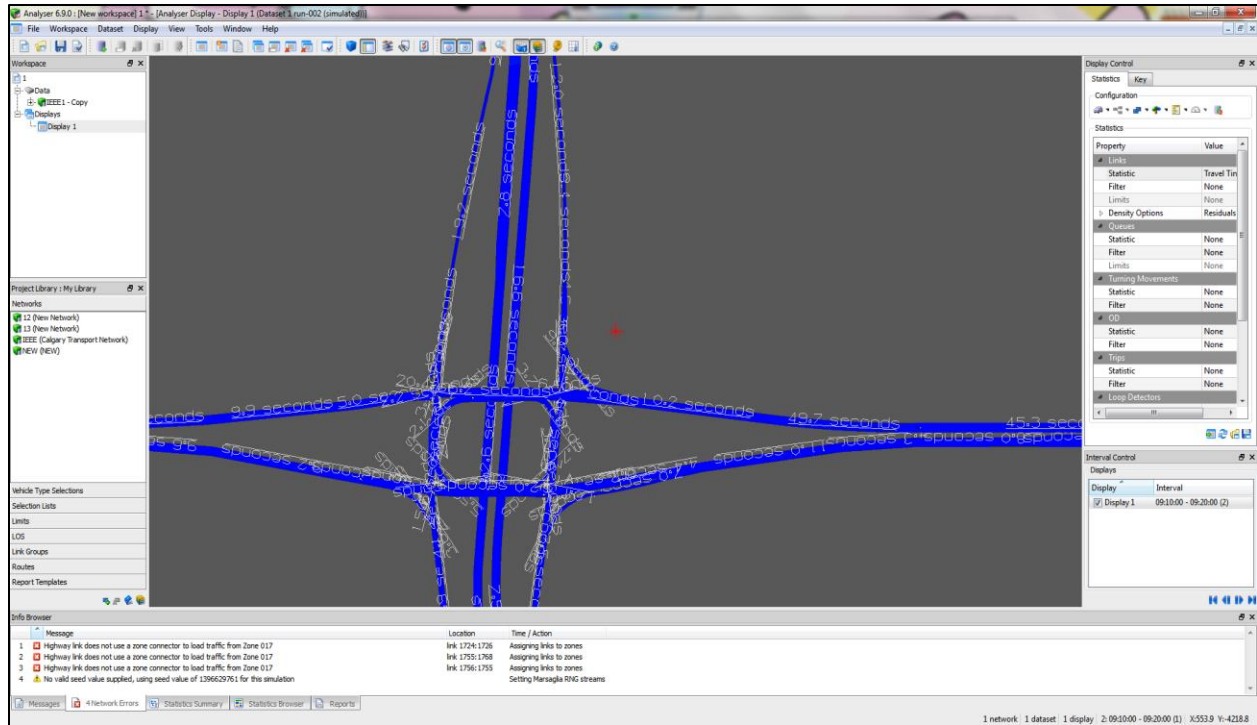
$(Risk\_profile)_{tI}$  is the average rear-end crash risk at time  $t$  and station  $I$ ;  $T$  is the number of time periods in the simulation run.

### 3.6.2 Mobility measurements

The other benchmark that is considered in this study is the point-to-point travel time along freeways. This value is calculated by the PARAMICS Analyzer module and is equal to the summation of the individual vehicle travel times along the studied corridor for the length of the simulation period [58]. The travel time index is considered as a relevant proxy for mobility improvement of a freeway.

Analyzer module reports the average time vehicles spend traversing the link. By defining a particular route, which may include numbers of links, journey time can be extracted as a data file. Each Analyzer data file contains information that has been gathered over a measurement interval. For example, if measurement interval is set to 10 minutes, the start time to 09:00:00 and the end time to 09:20:00, two data file with timestamps of 09-10-00 and 09-20-00 will be created by Analyzer. They contain journey time information collected during the two Analyzer intervals.

Figure 3.13 shows the screenshot of one of the examined scenarios in Analyzer module. It indicates travel times of the first time interval for one intersection and its relevant links.



**Figure 3.13 Travel time indicator in Analyzer module**

### 3.6.3 ANOVA test

ANOVA is used to determine whether there are any significant differences between the means of three or more independent groups. ANOVA generalizes t-test to three or more groups and is useful when testing three or more variables for statistical significance [126]. In this study, we have used ANOVA test to determine the level of confidence.

## 3.7 Integration of PARAMICS and OPNET

PARAMICS can model the dynamics of large-scale traffic networks, but it lacks the ability to model the performance of communication systems; this excludes its use for evaluating different communication protocols and network parameters during the implementation of CV. Additionally, with the development of cellular and various short range wireless technologies such as WiFi and Bluetooth, there has been increasing interest in providing internet access to moving vehicles relying on V2V and V2I wireless communications. The development of

equipping vehicles with this facility requires the use of WLAN networks such as IEEE 802.11 to support.

Since PARAMICS cannot simulate WLAN networks, the integration of PARAMICS and OPNET was developed for the purpose of implementation and evaluation of vehicular internet access using DSRC based communication protocols and their applications in the context of ITS.

The three important elements for the integration of traffic network and communication network are the road, the vehicles, and the wireless communication devices. The first two elements can sufficiently be modeled with high details in traffic simulator, while communication network simulators provide the great environment for modeling wireless devices. The traffic simulator is thus used to model the road and traffic control devices, generate traffic and CV vehicles based on penetration rates, and simulate driving behavior. The traffic simulator is also used to model ITS applications, vehicle controls and generating the messages to be carried by the communication network. All communication functions, including the determination of which vehicles receive the propagated messages, are left to the communication network simulator.

PARAMICS microsimulator was used to model the traffic flow and CV system to create location-specific incidents and collect traffic information. This information was directed to the OPNET discrete event simulator. Adaptive V2V and V2I communications, including addressing, routing, and propagating messages, were modeled in the OPNET simulation environment.

In CV system, the communication network is expected to become congested due to the increased data traffic transmission in many different V2V and V2I connections. Therefore, detailed analysis of the communication system, with the consideration of various communication technologies which support the information exchange in V2V and V2I applications, is required to meet the requirements of a real world implementation.

OPNET Modeler 17.5 was used to simulate a network with features similar to the network simulated by PARAMICS. OPNET has the ability to model several aspects of wireless transmissions including RF propagation, interference, transmitter-receiver characteristics, node mobility including handover, and the interconnection with wired transport networks. Several wireless network protocols such as MANET, IEEE 802.11, 3G, Ultra Wide Band, IEEE 802.16, Bluetooth, and Satellite can be simulated in OPNET.

Network protocols define how different nodes can communicate with each other over a network. These include a set of standards that define the parameters and methods followed by different layers of the network. The IEEE 802.11, 802.15 standards define protocols of the first two layers of OSI model, used in WLAN and WPAN. IEEE 802.11 is the base of products marketed as WiFi and 802.15.1 known as Bluetooth technology. Communication protocols describe modulation techniques, maximum power level allowed, the radio frequency spectrum utilized and other details depending on the requirements of the system. IEEE 802.11p improves 802.11 to support ITS applications. IEEE 802.11 is a set of MAC and PHY specifications for implementing WLAN computer communication. PHY, the first layer of the seven-layer OSI model, is a direct point to point data connection which is not necessarily reliable. MAC is the sub-layer of layer 2, the data link layer, which is a reliable direct point to point data connection. In order to simulate DSRC based CV system, WiFi network topology has been used in OPNET.

In this integration, either PARAMICS or OPNET is running at a time. We start by running a traffic network (Deerfoot trail) in the PARAMICS simulation. In PARAMICS, the vehicles are generated according to the OD matrix. The OD matrix stores the number of trip interchanges between each origin and destination. However, OPNET simulator does not have the same OD matrix logic and there is no meaning for zones, thus the mobility trace file, extracted from PARAMICS, is needed to implement vehicles as the nodes in similar environment in wireless network simulation. Other predefined elements such as the types of vehicles, link or lane restrictions and speed limits should be simulated in OPNET as they were in PARAMICS.

### **3.7.1 Mobility Traces**

The system consists of five components: CV, non-CV, RSU, VMS and Control Center. CV is equipped with a wireless system to create V2V and V2I connection. RSU, the infrastructure placed within the network, is capable of collecting information about traffic conditions and communicate with the CV located within a DSRC range, and sending this information to the VMS. VMS located in network to show warning messages and traffic information collected by RSU to communicate with non-CV. Non-CV do not have the equipment to connect with other vehicles or RSUs. Control Center is the decision making module which defines warning messages and selects the RSUs to transmit those messages.

In this system, there are many objectives which need to be fulfilled. First, the RSU node should be able to sense WiFi signals from vehicles and send information to the other RSUs, to CV and possibly to the VMS. Second, VMS should be able to show the given information to non-CV. Also, CV should be able to directly send the information to other CVs and RSUs that fall into DSRC range.

However first in OPNET, vehicles should be initialized, their location should be updated during time and get cleared from the network as they reach their destination. In order to do that, an API plugin which extracts the mobility traces from all vehicles in the network, throughout the simulation, is activated.

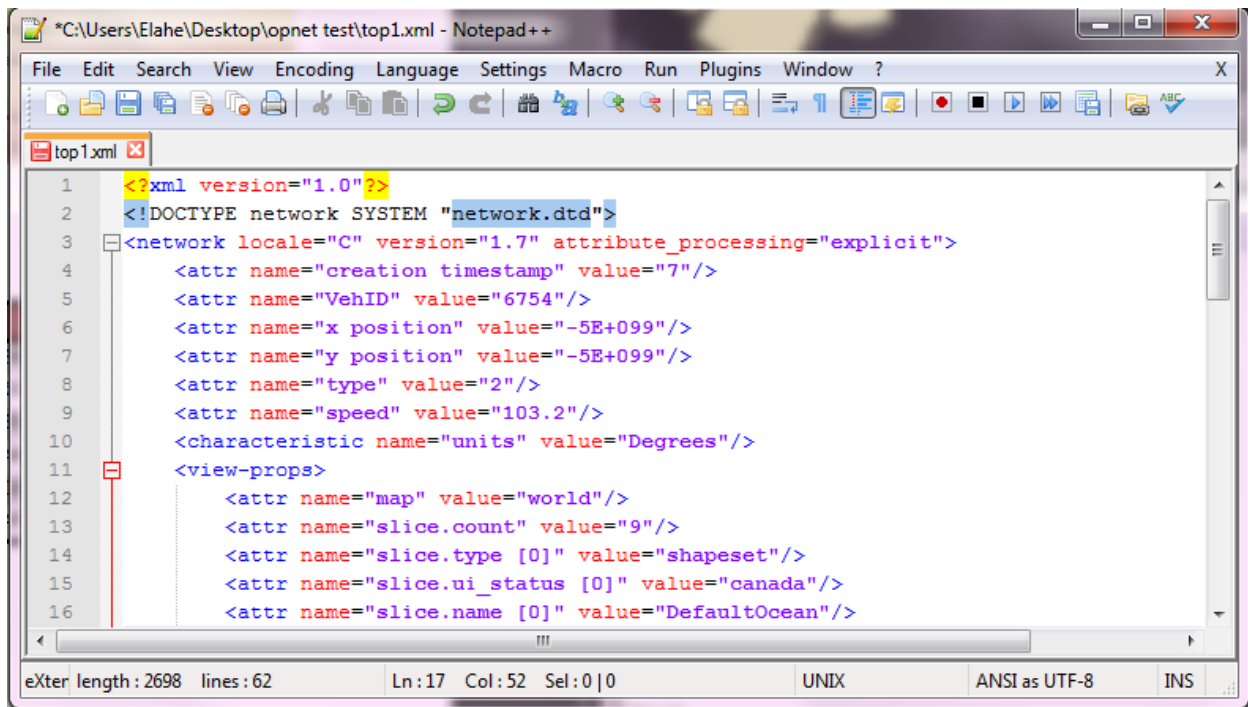
After the simulation ends in PARAMICS, the trajectory of vehicles can be extracted as the output in the form of a CSV file which includes <Vehicle type, Origin, Destination, Lane, Gradient, Bearing Length, Speed, Acceleration, Link name, Link ID, Link length, Link speed, Link gradient, Curved, Radius, Lanes> columns for every time interval that can be hours, minutes or seconds, depending on the accuracy required. In this study we only extracted information on <car-id, type, x-position, y-position, speed> columns of mobility traces. This means that the ID, position and speed of each vehicle in the network are stored for each time interval. Since there are two types of vehicles simulated in the network the “type” column is CV (type-1) or non-CV (type-2).

OPNET reads the trajectory file as the input and populates a node for every vehicle with the same coordinates in file. But first this CSV file, which is output of PARAMICS with the details of the mobility traces for all the vehicles, should be converted to XML file format to be accepted by OPNET as the input. By using a C++ program this change is made. Table 3.1 and Figure 3.14 represent the trajectory file in CSV and converted XML file format.

**Table 3.1 Trajectory file includes mobility traces in CSV file**

time	ID	type	x	y	speed kph
7:00:00	6754	1	2308.72	-6285.94	17.34
7:00:00	6431	1	3065.09	-4329.61	91.5
7:00:00	6435	2	2005.09	-9947.91	17.34
7:00:25	6461	2	2221.75	-9265.44	102.99
7:00:25	6567	1	2166.3	-9431.16	101.25
7:00:25	6545	1	2181.68	-9381.45	104.54

As an example, after the warm up period, the vehicle with id = 6754 has first appeared at time 7:00:00 at position (2308.72,-6285.94) with speed 17.34 as shown on the first line of the traces file in Table 3.1. This vehicle will thus appear in the XML file with its own <VehID, x position, y position, type and speed> which has <creation timestamp> attribute with the value of 7. All the coming positions of this vehicle will be stored in the mobility XML file shown in Figure 3.14. On the other hand, when a vehicle reaches its destination, it does not appear any more in the traces file. Thus a corresponding line will be written in the file to make sure the vehicle is not appear in the next time step in the simulated network.



**Figure 3.14 Converted mobility traces in XML file**

### 3.7.2 Simulation configuration in OPNET

Each vehicle in PARAMICS has a unique ID and is represented as a mobile node in OPNET which can be CV or non-CV. Mobile node, such as cellular telephone or router, is a device whose location and point of attachment to the internet may regularly be changed. Particular actions should be made to keep the mobile node connected to the internet while it moves from one network or subnet to another and change the device's IP address every time they are connected to internet via another network or subnet.

With ID, the movement of each vehicle can be tracked throughout in the network. In particular, every node has several abilities such as sending, receiving, analysing, tracking, storing and multiplexing messages. If the type of vehicle was indicated as type-2 in the file, the simulated corresponding mobile node would not have the ability of sending messages. But if it is type-1, the node would have all the abilities of wireless node provided by OPNET. It is modeled as a node that is able to communicate over a wireless interface using the IEEE 802.11p (Wi-Fi) protocols. Note that it can be represented by any other wireless module if desired. This node



itself consists of several other modules such as packet streams, static wires, processor modules, queue modules, transmitter modules, receiver modules and antenna modules.

Moreover, each fixed node in OPNET corresponds to an infrastructure such as RSU or VMS and performs different functions such as receiving warning message, processing, exchanging with control center, and dissemination within IEEE 802.11p. Fixed node is a network terminal or device whose geographical positions cannot be changed over time but can be assigned arbitrary at the start of simulation. Normally it has the ability to communicate to other nodes.

In OPNET, there is another type of node called satellite node. It is similar to a fixed node, but it changes its location automatically on an assigned orbit route only in Radio products. We did not use this type of node in our simulations.

Once the nodes have been defined in the wireless simulator, the data elements and properties of network should be set. Some of them were chosen directly in OPNET modeler and some of them were programmed in process model using Proto-C language. Figure 3.15 shows an explicit overview of the modeled data elements in both simulators. The selection of each parameter, such as the location and distance between RSUs, depends on the required system performance. In [4], we found the proper amounts for these parameters in PARAMICS simulator. But in order to satisfy smooth communication and implement V2V and V2I connection in wireless communication simulator, the distance between the RSUs needs to be adjusted such that vehicles can always sense a strong signal from the nearest infrastructure. If they are too far apart, there might be transmission delays, message drops or even areas without DSRC coverage; on the other hand, if the RSUs are too close together, excessive handover might take place. That means vehicle would need to handover its communication from more than one RSU which might lead to a high number of packets being dropped than allowed. Therefore, these settings need to be adjusted adequately in wireless network simulator.

Network parameters include configuration data, such as number and location RSUs, the percentage of each type of vehicles, and parameters regarding the updates of their location. These parameters are kept for both the current and other predefined intervals. For instance, for each RSU, static data are used to specify various communication ranges, size of the buffer, and a list of VMSs which are linked to it. Dynamic data for RSU are used to model the number of

vehicles which are currently within its communication range, the number of message transmissions, and statistics such as load and latency (Figure 3.15).

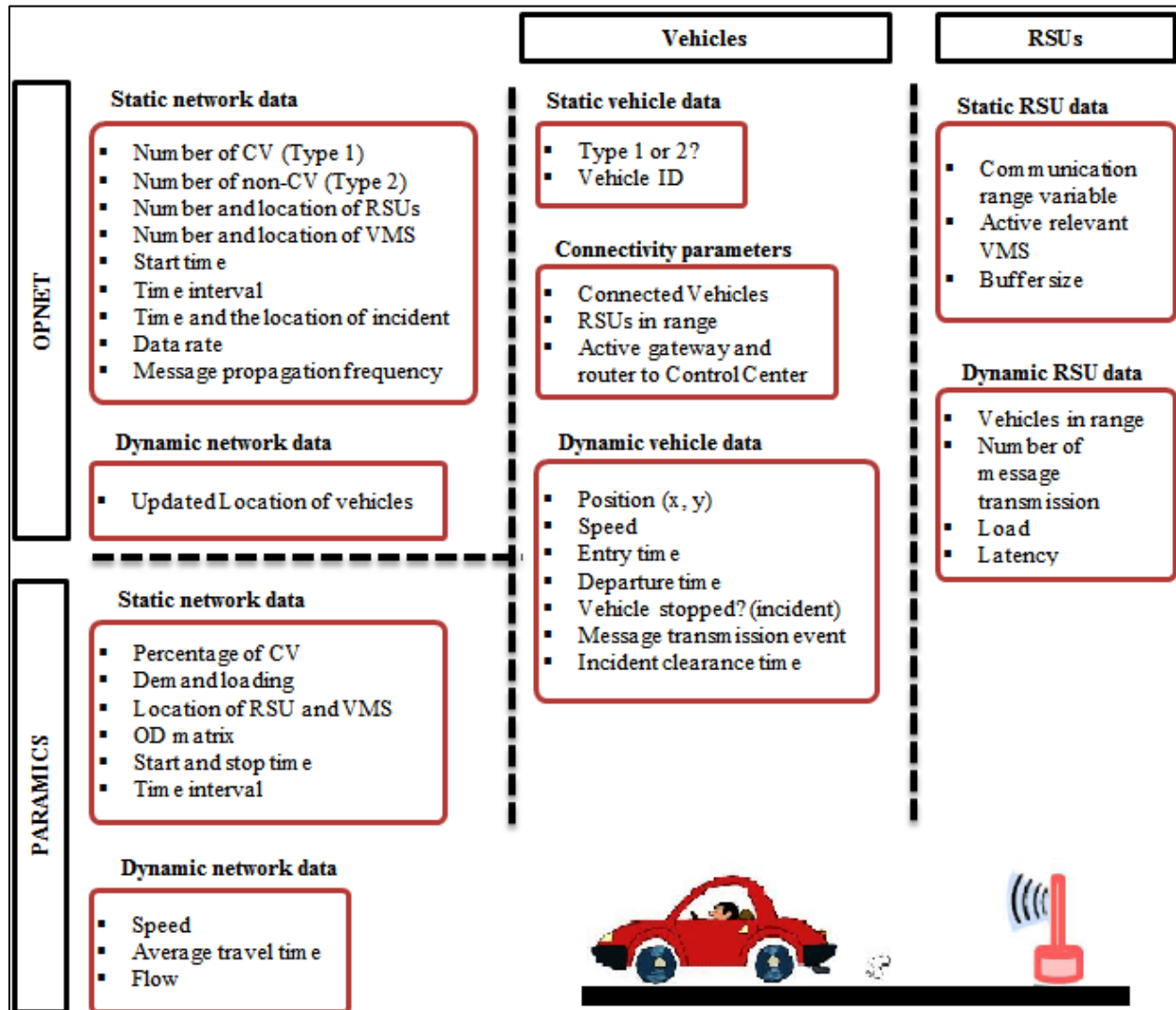


Figure 3.15 Data modeling in two layers of simulation

In OPNET simulation in order to simulate CV in the same way as in PARAMICS, we need to ensure that the vehicles only appear in the simulation when they first appear in PARAMICS and then they disappear once they reach their destination. Also, we need to ensure that the vehicles follow the same path as they did in the PARAMICS simulation. During the simulation, the scheduled tasks were programmed in a way that OPNET updates the position of the nodes based

on the mobility traces in every time interval. With the passage of time, if there is no change in the location of the node (i.e. the vehicle reached its destination), OPNET would destroy it. When the number of nodes that were created is equal to the number of nodes destroyed, the simulation run will be ended.

### **3.7.2.1 Data Collection**

After creating the network and placing vehicles and infrastructures, dynamic simulations can be run in order to study their system performance and behavior. Based on the results from the simulation, changes can be made to the model's specification and attributes to duplicate different scenarios and execute additional simulations.

There are many metrics that can be measured in OPNET to evaluate system performance. Global statistics, node statistics, attribute statistics and animation statistics are some statistics types which can be collected by OPNET. It is needed to decide which information should be extracted from the simulation, such as application-specific statistics, behavioral characterizations, and application-specific visualization. All the statistics for specific simulation run such as throughput or delay, whether between vehicles or between vehicles and RSUs, can be measured at the global or object level. In our test cases we used statistics at both levels. In global level, these parameters will be measured for the whole simulation run with its different modules. In object level, these measures can be extracted for only one specified object in the network, such as one RSU.

## **3.8 Summery**

In this chapter, we explained the detailed approach of simulating CV system as our ultimate goal. The design and implementation of this goal is provided in two ways: simulation using PARAMICS traffic simulator, which is accompanied with our added APIs, and simulation using the integration of PARAMICS and OPNET wireless communication simulator.

In the first approach, CV system was designed as MAS using MaSE methodology to specifically determine its different types of agents. Then, these agents, their connections and conversations were programmed using two APIs. Moreover, in order to run test case simulations, particular parameters and their assumed values in PARAMICS were explained. At the end, the safety and mobility indices, used to evaluate system, were introduced.

In the second approach, by using mobility traces file, extracted from PARAMICS, we simulated CV system and V2V/V2I communication in OPNET. This includes creating different CV devices in the network, their interaction and responses. Some extensions were programmed in process model to exactly simulate the same scenarios from first approach. The specific parameters in OPNET were explained where the type of evaluation metrics from OPNET were introduced.

In the next chapter, we describe, in detail, 3 scenarios carried on the proposed methodologies, output results and their interpretations.

## Chapter 4: Case Studies

### 4.1 Introduction

This chapter presents 3 scenarios (Table 4.1) as the case studies to assess the developed methodology. First two scenarios are to fulfil our first goal, described in chapter 3, which means evaluating the CV system for estimating and improving traffic safety and mobility parameters in the network.

In this research we utilize PARAMICS traffic micro-simulation tool to evaluate the impact of deploying CV on a section of Deerfoot trail, Calgary, Alberta. We have implemented a V2V assisted V2I system for PARAMICS which uses DSRC protocol to acquire traffic data, calculate and compare important traffic safety and mobility parameters. The study demonstrated that the CV technology can enhance traffic safety and mobility in freeways. In other words, equipping freeways with relevant infrastructures such as V2V and V2I as the CV technology significantly improves CV efficiency and leads to higher safety and mobility enhancement in freeways.

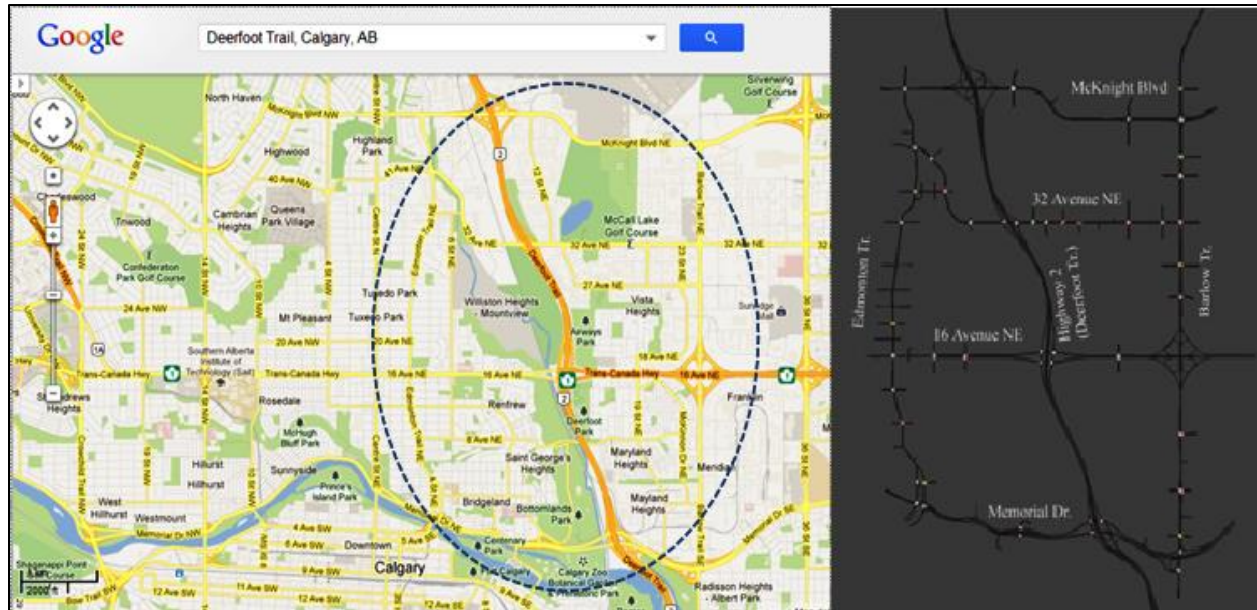
Moreover, the third scenario uses the integration of simulators to conducts analysis in order to satisfy our second goal where it examines the impact of factors, such as the percentage of CV, congestion levels of a road network, and the defined DSRC range for RSUs in the network. The results of its first test case on the integrated simulations show the sensitivity of the results to data transmission rate, data loss and delay to DSRC communication range of roadside units (RSUs). Also its second test case shows that how extra-equipping network with CVs would have negative impacts on overall throughput and data packet drops of the system and eventually on safety and mobility applications.

**Table 4.1 Three proposed scenarios accepted in referred conferences**

Analysis, design and implementation of an agent based system for simulating CV [127]		
Scenario 1	Scenario 2	Scenario 3
Modeling and simulation of advisory speed and re-routing strategies in CV systems for crash risk and travel time reduction [4]	A simulation-based benefit analysis of deploying CV using DSRC [6, 36]	Integrated traffic and communication performances of CV systems for evaluation of DSRC

#### **4.1.1 Study area and design of experiment**

The PARAMICS micro-simulation suite has been used to model Deerfoot trail in Calgary, Alberta. As mentioned before, the analysis was conducted on an 8-km southbound section of Deerfoot Trail in Calgary, Alberta, Canada and the study area extended from McKnight Boulevard to Memorial Drive. The simulated network also included Barlow Trail which is an urban arterial that provides a north-south connection throughout the northeastern part of the City (Figure 4.1). The inclusion of this arterial is important to examine the impact of rerouting in case of incidents. The network was calibrated based on count data provided by Alberta Transportation and travel time data and OD data provided by the City of Calgary [6]. The mean headway factor and the mean reaction time were calibrated to simulate observed traffic patterns. These two parameters control three components of the individual vehicle movement: car following, gap acceptance and lane change behaviours.



**Figure 4.1 The real vs. PARAMICS simulated network study area (source of real map: Google Maps)[6]**

In the network 176 loop detectors have been placed. These detectors yield measures of speed, lane occupancy, and volume on each of the 6 mainline lanes at 10 minutes intervals. The loop detectors provide reference points for locations along the freeway. This loop detector data from station 7, 8 and 9 was used in statistical models to create the crash likelihood, ORCI and travel time measures that are used in this study. The 70 minutes morning peak period was modeled to evaluate CV effectiveness in the network; however the first 10 minutes were used as the warm-up period and no statistics were collected during this time to ensure that simulations start with a realistically loaded network.

In the next sections all the 3 scenarios are explained in details.

## 4.2 Scenario 1

As mentioned before, conventional traffic simulator systems do not support the simulation of CV systems. The focus of this case study is to evaluate V2V and V2I communications by using the extended functionalities of PARAMICS which were developed by two APIs for the simulation of

CV. Moreover, this extended simulation system was also designed to examine the implementation of advisory speed recommendation and re-routing guidance for urban freeways under various load conditions to recommend the optimum treatments and reduce rear-end and lane change crash risks. The advisory speed recommendation was only for upstream traffic of the location of accident where speed differences between upstream and downstream vehicles were high. We use these strategies as a tool for safety and mobility improvements on a section of Deerfoot trail, Calgary, Alberta. Results of the experiments demonstrate the overall effectiveness of the approach.

There are 2 types of vehicles in the network: CVs and non-CVs. The probability of incident occurrence is ascertained at the start of simulation run. The incident will happen only for CVs. It sends messages to other CVs [121]. Random incidents are produced in the network by taking user inputs on the probabilities of collisions and weather-related incidents [121].

In order to develop V2I in API #2, as mentioned in chapter 3, RSUs and VMSs are designed using PARAMICS beacons and have been placed throughout the network. Moreover, control center, which has been created as the decision maker, gets the data from RSUs to determine which VMS should show the warning messages and what would be the context of the message. Generally the message contains information about advisory speed and re-routing guidance. The advisory speeds are designed to be incrementally decreased in upstream to prevent possible congestion. For example, when an incident occurs, nearby vehicles are advised to reduce their speed to 60 km/h, vehicles 500 metre behind would reduce their speed to 70 km/h, 1,000 metre behind 80 km/h, 1,500 meter behind 90 km/h and 2,000 meter behind 100 km/h. Re-routing guidance was also provided in message to disseminate information on the incident and help non-CV drivers decide if they may want to change their route to bypass the accident location.

By using API#2, CV can also communicate with the RSU. In the other words, in V2I application, CV encountered an incident can send message to the nearest RSU in the road to inform it about the position and type of incident. RSU will send this information to the control center. Control center decides which VMS beacon should be enabled to show the message and what the context of message is for every VMS. This message includes warning information to let the drivers know that there is an incident ahead and an advisory speed to prevent any further



accidents. Also when the accident is cleared from the network, control center has to update the messages on the VMS beacons.

In this work, two test scenarios have been examined with the base scenario (no-CV): 30% and 50% CVs, respectively. The reason for considering only these three percentages of CVs (0%, 30% and 50%) is to study the effect of CV penetration rate on the safety and mobility improvements. As the results show their effectiveness, we complete our study by investigating more different values for CV percentages in scenario 2.

For the simulation, some inputs are provided based on preferences, such as percentage of CVs, driver behaviour, driver familiarity, advisory speed and probability of various simulated incidents such as weather related incidents, travel lane blockage, vehicle break downs and collisions.

#### **4.2.1 Scenario evaluation**

All examined scenarios are comparing the CV cases with the base scenario, i.e. the simulation without V2V and V2I communication - 0% CV and no RSUs and VMSs in the whole network, which reflects existing traffic condition and traffic control strategies. The demand loading for this scenario is 100% which mimic the weekday's AM peak traffic flow in the network. The effectiveness of implementing V2V and V2I is measured by comparing two indices of effectiveness of the test case scenarios with the base case to show the improvement in safety and mobility applications:

1. Reduction in crash likelihood; and
2. Savings in travel time (sec) in simulation intervals.

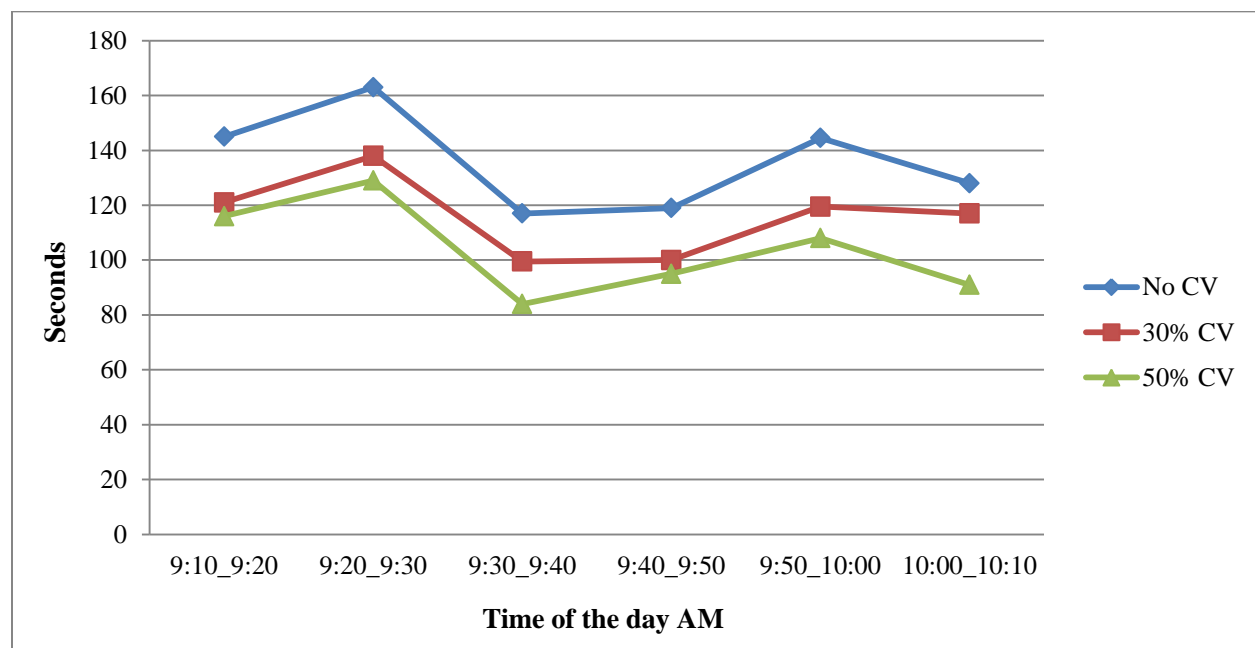
In this case study, the simulation has run for 30 times where the percentage of CVs changes but the other elements including seed and demand factor remain constant. In other words, 10 runs were designed with 10 different seeds (S1 S2... S10) and these runs were applied for 3 configurations, i.e. 0%, 30% and 50% CVs.

The average travel time for a route in particular link (route near 32 Ave in the link from memorial to McKnight) in Deerfoot trail, in the tested network, has been calculated for every run and the average of them is used for the test and base cases. Flow, occupancy and speed are

extracted from loop detectors' data of the 30 simulation runs (10 for each configuration). For instance, since there are 6 intervals and each of them is for 10 minutes (70 minutes simulation, 10 minutes warm-up), every run delivers 6 values for speed which are the average speed of lanes of a link. Then the average of these 6 values is used for calculating the crash likelihood. The same approach is used for calculating occupancy and flow.

## 4.2.2 Results

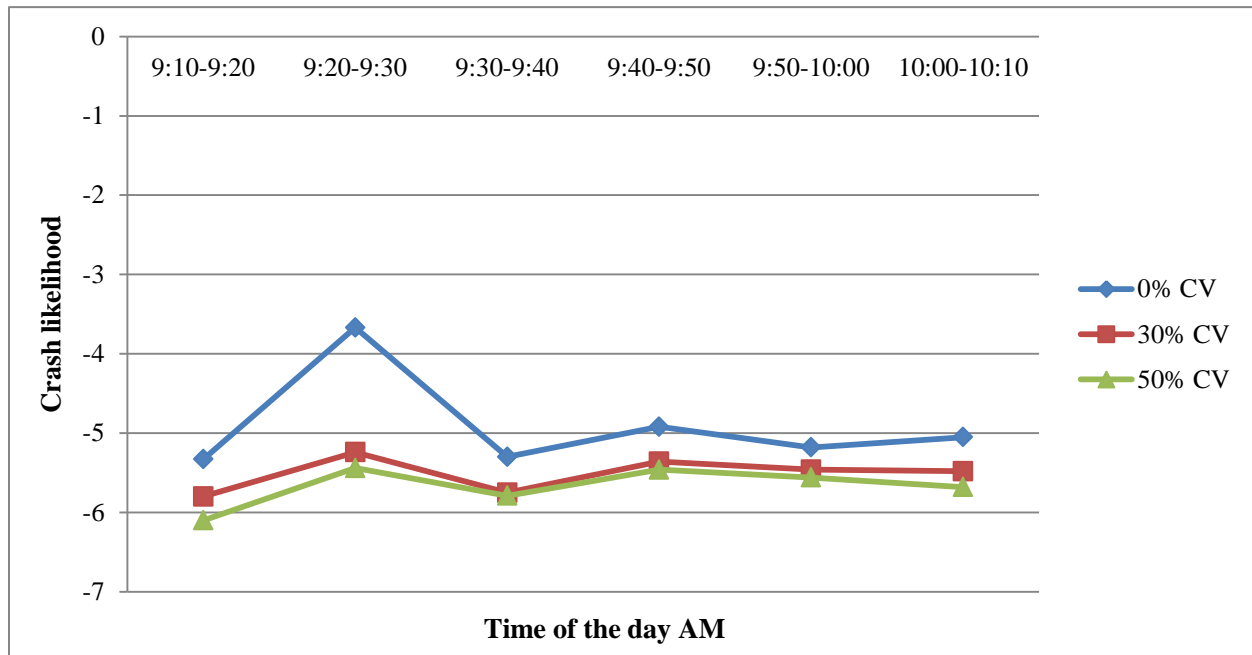
Figure 4.2 shows that there is a consistent reduction in travel time, because the test cases reveal a significant lower value of travel times in the system, due to the implementation of advisory speed and re-routing guidance in CV system. This indicates that mobility improvement can be achieved in Deerfoot trail as the tested traffic network.



**Figure 4.2 Average travel time comparison between base case (no connected vehicle), 30% and 50% V2V (seconds)**

As mentioned before, the travel time is extracted from the results that PARAMICS Analyser module has been provided.

Figure 4.3 shows the crash potential which is calculated based on the model introduced in [16]. It is used in this scenario to evaluate the effectiveness of CV system in safety index. Thus, average volume, occupancy, and factor of speed variations from loop detector real-time data were used as the input to assess crash likelihood measure for the simulated network (Deerfoot trail).



**Figure 4.3 Crash potential for the base case (No CV) and test cases (30%, 50% CV)**

As the Figure 4.3 shows, there is a reduction in crash potential when advisory speed, re-routing application and V2I have been implemented in the network in which there are 30% and 50% CVs. This reduction improves the safety index in the Deerfoot trail. Figure 4.3 indicates that when the crash potential is low from 9:30 to 10:10, its probability is even lower in test case rather than base case. Moreover, as shown in Figure 4.3, by having 50% CV, lower amounts of crash risks and higher safety benefits can be achieved.. In the next scenario, we even found out that increasing the percentage of CV too many will have negative effects on the outputs.

The crash likelihood and travel time results of the examined scenario were found to be sensitive to the percentage of the CV in the network. Therefore, next scenario includes experiments that penetration rate of CVs varies with higher traffic congestion levels.

## 4.3 Scenario 2

The main objective of this case study is to assess the potential of DSRC enabled V2V and V2I capabilities in traffic safety and mobility enhancement using PARAMICS microsimulator environment. It uses DSRC protocol to acquire traffic data, calculate and compare important traffic safety and mobility parameters and their impacts on CV by testing five test cases differentiated by the percentage of CVs (0% to 40% market penetration of CV). Our experiments showed that if there are more than 40% CV in the network it will have adverse effects on the outputs. The reason is that higher percentage of CV leads to higher percentage of drivers being aware about the advisory speed. As more vehicles, approaching the incident, lower their speed, the route they are taking become congested, which negatively impacts the travel time at the network [13]. Regarding this, scenarios with more than 40% of CV have been eliminated for further investigation.

Although the previous case study found the improvement in travel time and crash likelihood in Deerfoot trail, we did not develop the DSRC range as a factor for distributing messages in V2I module and we only focused on upstream traffic. In this scenario we demonstrate effect of considering DSRC, re-routing guidance and advisory speed for upstream and downstream traffic. This case study shows that CV technology can enhance traffic safety and mobility in freeways, if the percentage of CVs is significant (e.g. 30-40%) and the CV technology is accompanied by advisory speed reflected on VMS on both upstream and downstream of the incident location using DSRC range.

### 4.3.1 Study area and design of experiment

The study area for this scenario is the same as scenario 1 and was done on the southbound Deerfoot Tr. between McKnight Blvd. and Memorial Dr. In the study network in PARAMICS, the loop detector data from stations 7, 8 and 9 from 176 loop detectors were used in statistical models to create the ORCI and travel time measures that are used in this study. The 70 minutes morning peak period was modeled, however the first 10 minutes were used as a warm-up period and no statistics were collected during this time.

This scenario is designed to have only one predefined incident. This way the simulation specifications would be pretty the same and leads to more precise results. The incident location is

about 600 meter upstream of the Memorial Drive exit. It should be noted that the link and the lane on which the incident happened remain unchanged for all simulation runs. Furthermore, it was set the vehicle involved in the incident to be a CV vehicle. The warning messages would be sent frequently until the accident is completely cleared in the network.

There are 2 types of vehicles in the network: CVs and non-CVs. Once the incident happens, the involved cars will promptly send message to other CVs to inform them about the incident and provide the advisory speed and re-routing recommendations. Concurrently, a message is sent to the nearest RSU in DSRC range. The RSU sends the aforementioned message to the control center. The control center sends an advisory speed to each VMS beacon falling in the DSRC range along the freeway to inform non-CVs about the incident and show an advisory speed. The message “Incident ahead” is common for all VMS beacons falling in the DSRC range; whilst, the advisory speed varies according to the VMS beacon’s distance from the incident. Note that not all the VMSs will show the warning messages, only the ones that fall into DSRC range. Checking the DSRC range and creating advisory speed for downstream of traffic were programmed as an extension in the second API. Consequently, non-CV will be aware of the traffic situation along the freeway very shortly.

Depending on the location, the advisory speed limit might vary. For upstream traffic the advisory speed is lowered in increments as the distance from accident decreases, the same as defined in scenario1. The downstream of incident location are also advised to increase the speed up to 110 Km/h to accelerate traffic flow and clear the area near the incident location to let upcoming vehicles bypass the situation. When incident happens, the potential of crashes will be increased since queues started forming and there is a group of vehicles arriving at the existing queues. Therefore it is reasonable to slow down the upstream vehicles to prevent them from hitting the queue and increase up the downstream speed to help the queue [5].

In this work, fifteen test scenarios have been examined based on the following configurations:

- CV distribution percentage: Non-CV, 10% CV, 20% CV, 30% CV, 40% CV;
- Demand loading (demand factor): 60% demand loading; 80% demand loading; 100% demand loading.

60% and 80% demand loading replicates a typical Holiday/Sunday and typical Saturday traffic counts, respectively; where 100% repeats a typical AM peak weekday when traffic flow is high [128].

The demand loading percentages are correlated to the DSRC range [129]. A higher demand percentage leads to opt for a shorter DSRC range in order to avoid data interference and latency. DSRC range selection is based on choosing one in boundary (1000), one in above, and one in below the boundary. For this case study the following DSRC ranges have been adopted based on the finding in [129]:

- 60% demand loading → 1200 m DSRC;
- 80% demand loading → 1000 m DSRC;
- 100% demand loading → 800 m DSRC.

It is to be noted that the reported run for each of the 15 scenarios correspond to the average of 10 PARAMICS runs with 10 different random seeds. These random numbers are utilized by PARAMICS to calculate different traffic assignment parameters, such as car following, lane changing, route choice and release of demand. Thus, PARAMICS creates a dynamic traffic model for each seed number and varying traffic demand on the freeway section. The same set of random seeds was used for the simulation of the different examined different scenarios. The messages include an “Incident ahead” message along an advisory speed. Since the CV has more information about the road conditions, they have the higher amount of driver familiarity input which leads to having more awareness of updated cost to their destination each interval. In this scenario, the awareness and aggression indices of CV drivers are programmed to be a number between the threshold of [6-9] and [1-4], respectively. They will react in lower amount of time to the accident ahead compared to non-CV and may change their route to reduce their travel time and reduce the number of further accidents. The aggressiveness and awareness amounts for non-CVs are the default values provided by PARAMICS for the typical vehicles.

### **4.3.2 Scenario evaluation**

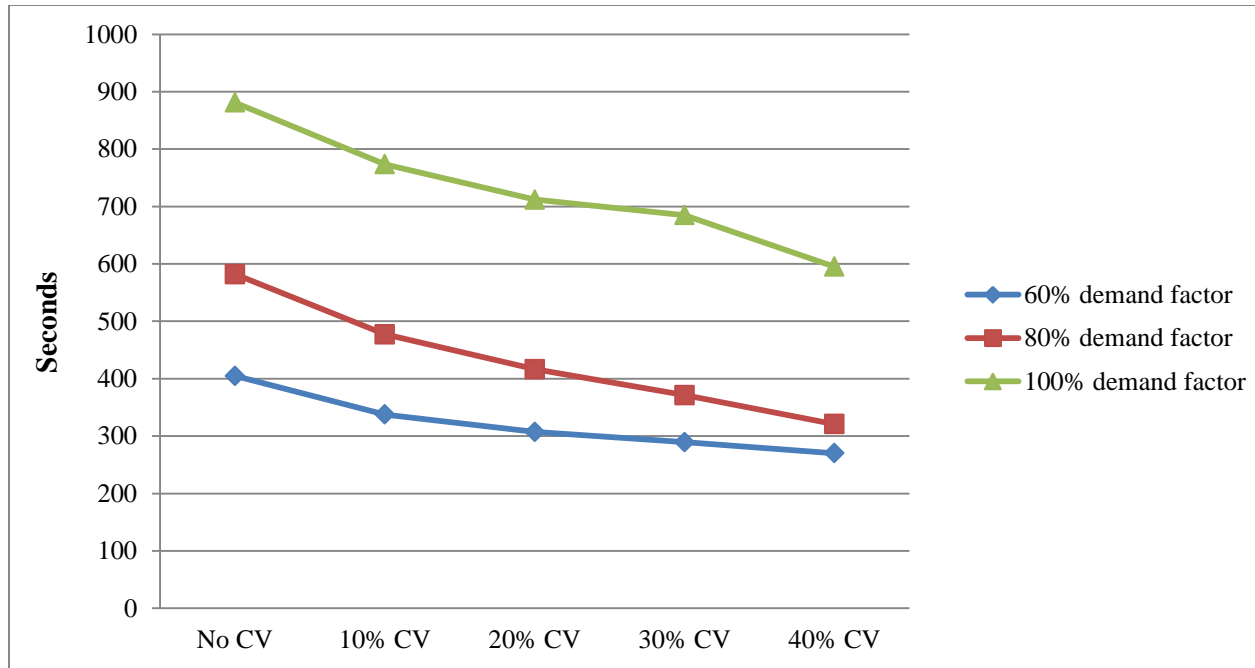
All fifteen scenarios were assessed based on two factors: mobility, and safety. As pointed out earlier, the mobility benchmark is measured based on the average estimated P2P travel time on Deerfoot Trail between McKnight Blvd. and Memorial Drive. The safety index is measured by

crash likelihood and ORCI models for the rear-end and lane-change crash risks along the freeway. Moreover, since 150 simulation runs were executed, the ANOVA test was conducted to acquire the level of confidence for the results. ANOVA tests the null hypothesis that the means between and among of all CV percentage's group of results are equal; where, the hypothesis is that the means of the results for every CV percentage has significant difference with means of the results for other tested percentage, at 95% level of confidence. In other words, the hypothesis tested by ANOVA is that the population means for all conditions have noticeable differences with each other. We selected 95% for level of confidence since it is a standard level of confidence widely used in Statistics [126]. For some cases, when the accuracy is inevitably essential, scientists might use 99% as well. The selection depends on the required level of certainty from the analysis of variation calculation [130].

### **4.3.3 Results**

Given the fifteen tested scenarios differentiated by the CV percentage penetration (0%, 10%, 20%, 30%, and 40%), and demand loading (60%, 80%, and 100%) implicitly representing peak and off-peak traffic; the proposed 3 test cases demonstrated that the CV technology can enhance traffic safety in freeways, if the percentage of CVs is significant (e.g. 30-40%) and the CV technology is accompanied by advisory speed reflected on VMSs on not only upstream but also downstream of the incident location despite of scenario 1. In other words, applying V2V, V2I and advisory speed application significantly improve CVs efficiency and leads to higher safety and mobility enhancement in freeways.

Figure 4.4 shows the average travel time for all 3 test cases. It illustrates how applying CV system leads to decreasing travel time and improving mobility index in different congestion levels.



**Figure 4.4 The average P2P travel time for all 3 test cases (sec) [6, 36]**

Table 4.2 shows the overall ORCI for every demand loading and CV percentage. As shown, applying 40% CV in our proposed scenario in which V2I includes transmission of warning messages with advisory speed and re-routing guidance, the highest increase on safety benefit can be achieved. Advisory speed helps drivers adjust their speed and change their lane at the proper time in upstream and downstream of incident. The findings conform to the other research's findings in the context of the network capacity development [131].



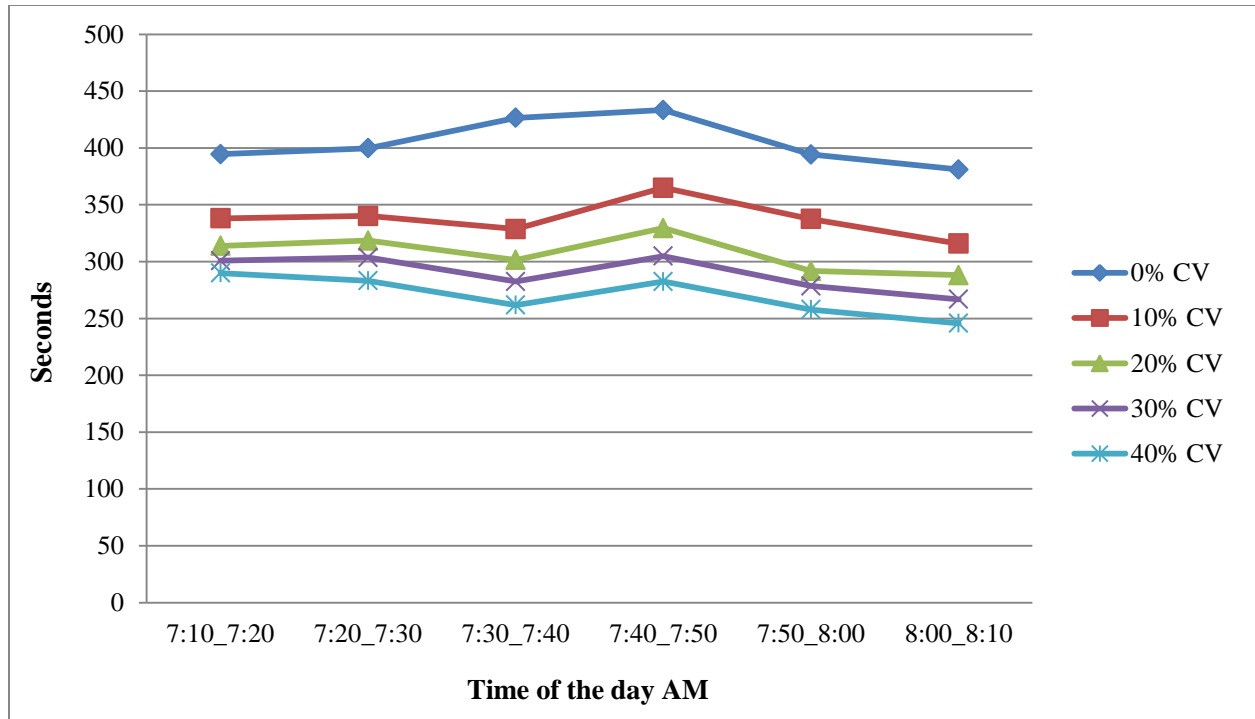
**Table 4.2 The overall ORCI for all 3 test cases [6, 36]**

	ORCI			
	No CV - 10% CV	No CV - 20% CV	No CV - 30% CV	No CV - 40% CV
60% demand factor	1.32	1.71	3.41	4.47
80% demand factor	1.51	1.79	3.63	4.43
100% demand factor	1.52	2.19	3.32	4.11

In the rest, we study every 3 test cases individually in terms of travel time and crash likelihood.

#### **4.3.3.1 Test case 1: 60% demand loading**

Figure 4.5 shows the average P2P journey time along the specified route in the network for six 10 minutes intervals from 7 AM to 8 AM when demand factor is 60%. It also indicates that developing CV system decrease P2P travel time for all intervals leading to mobility improvements.



**Figure 4.5 P2P journey time (sec) in tested network (60% demand load) [6]**

Although implementing 40% CVs cause to have the highest mobility improvements, its travel time difference with 20% and 30% CV seems not to be significant. Thus, an ANOVA test has been performed.

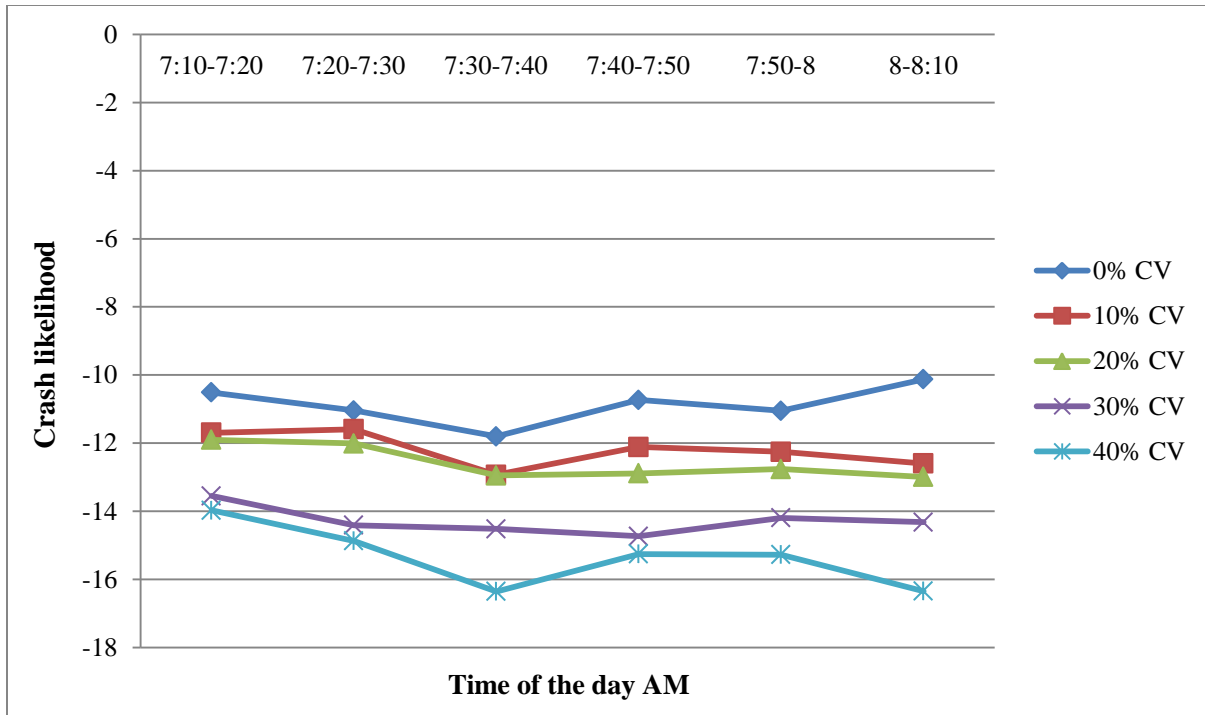
Results of ANOVA test (Table 4.3) shows that on the subject of travel time there is a significant difference between and among groups.

**Table 4.3 ANOVA Test for 60% demand loading [6]**

Anova: Single Factor						
SUMMARY						
Groups	Count	Sum	Average	Variance		
20% V2V	6	1843.457	307.2428	260.8487		
30% V2V	6	1737.139	289.5232	248.835		
40% V2V	6	1621.007	270.1678	304.8276		
ANOVA						
Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	4126.342	2	2063.171	7.599051	0.005258	3.68232
Within Groups	4072.557	15	271.5038			
Total	8198.899	17				

The first column shows the sources of variation, the second column shows the sums of squares (SS), the third shows the degrees of freedom (df), the fourth shows the mean squares (MS), the fifth shows the F ratio, the sixth shows the probability value, and the last one (F\_crit) shows the critical value of F which is a function of the degrees of freedom and the significance level (95%). If  $F \geq F\_Crit$ , the null hypothesis is rejected. The mean squares are always the sums of squares divided by degrees of freedom.

The crash likelihood measure which is calculated based on the formula presented in [16], is showed in Figure 4.6. It shows that developing CV reduce the crash risks and consequently improving safety applications. Higher percentage of CV leads to higher safety benefits achieved during the network.



**Figure 4.6 Crash likelihood prediction (60% demand load) [6]**

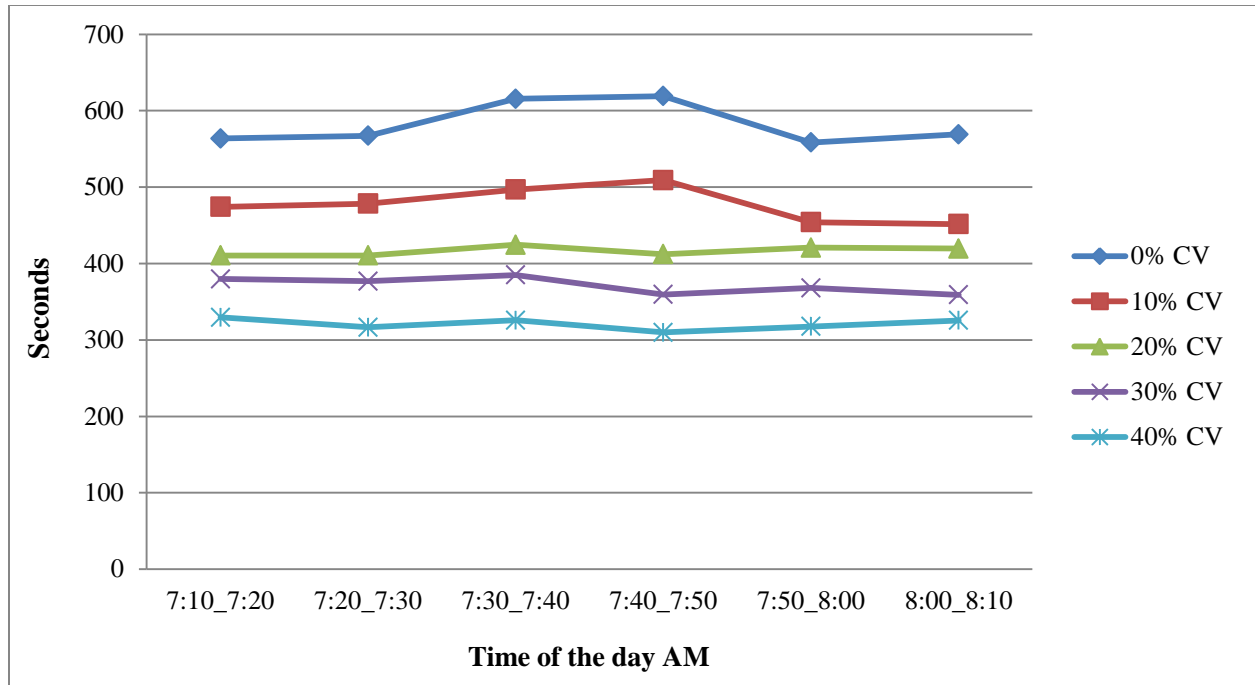
The ANOVA test (Table 4.4) shows that the difference between the first three CV groups (0% CV, 10% CV, and 20 % CV) is significant at the 95% level of confidence, however it cannot be concluded that the difference between 10% CV and 20 % CV is also significant.

**Table 4.4 ANOVA Test for 60% demand loading [6]**

Anova: Single Factor						
SUMMARY						
<i>Groups</i>	<i>Count</i>	<i>Sum</i>	<i>Average</i>	<i>Variance</i>		
0% V2V	6	-65.26	-10.8767	0.32717		
10% V2V	6	-73.1726	-12.1954	0.265781		
20% V2V	6	-75.4926	-12.5821	0.244166		
ANOVA						
<i>Source of Variation</i>	<i>SS</i>	<i>df</i>	<i>MS</i>	<i>F</i>	<i>P-value</i>	<i>F crit</i>
Between Groups	9.594338	2	4.797169	17.19177	0.000131	3.68232
Within Groups	4.185581	15	0.279039			
Total	13.77992	17				

#### **4.3.3.2 Test case 2: 80% demand loading**

Figure 4.7 shows the average P2P travel time from 7 AM to 8 AM in the tested route in the network. Similar to test case 1, CV system improves mobility index by reducing journey time in the network, especially in the case of 40% CV where 44% mobility enhancement is achieved (Figure 4.4).



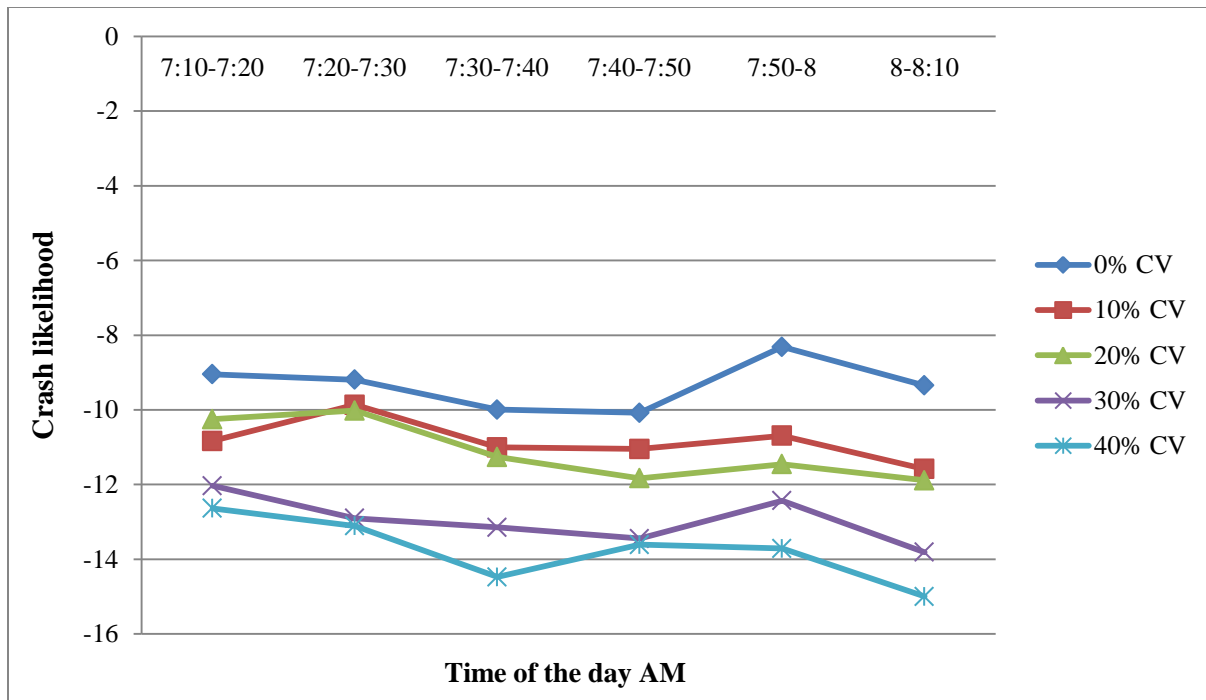
**Figure 4.7 P2P journey time (sec) (80% loading) [6]**

Moreover, the ANOVA test proves that there is a significance difference amongst all CV categories at the 95% level of confidence (Table 4.5).

**Table 4.5 ANOVA Test for 80% demand loading [6]**

Anova: Single Factor						
SUMMARY						
Groups	Count	Sum	Average	Variance		
Non V2V	6	3493.142	582.1903	757.4929		
10% V2V	6	2864.371	477.3952	520.8418		
20% V2V	6	2498.108	416.3513	36.64266		
30% V2V	6	2228.229	371.3715	118.1943		
40% V2V	6	1924.6	320.7667	54.65047		
ANOVA						
Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	245390.2	4	61347.55	206.1656	9.48E-19	2.75871
Within Groups	7439.111	25	297.5644			
Total	252829.3	29				

Figure 4.8, similar to test case 1, shows that CV reduces the crash risk and as the result improves the safety index for 80% demand loading. This improvement is also showed in Table 4.2 in terms of ORCI index where the impact of CV is noticeable for 40% CV by receiving the value of 4.43.



**Figure 4.8 Crash likelihood prediction (80% loading) [6]**

Figure 4.8 also illustrates that the effect of 10% and 20% CV is approximately similar. Thus, the ANOVA test is calculated (Figure 4.6). It shows that there is a significant difference at 95% level of confidence.



**Table 4.6 ANOVA Test for 80% demand loading [6]**

Anova: Single Factor

SUMMARY

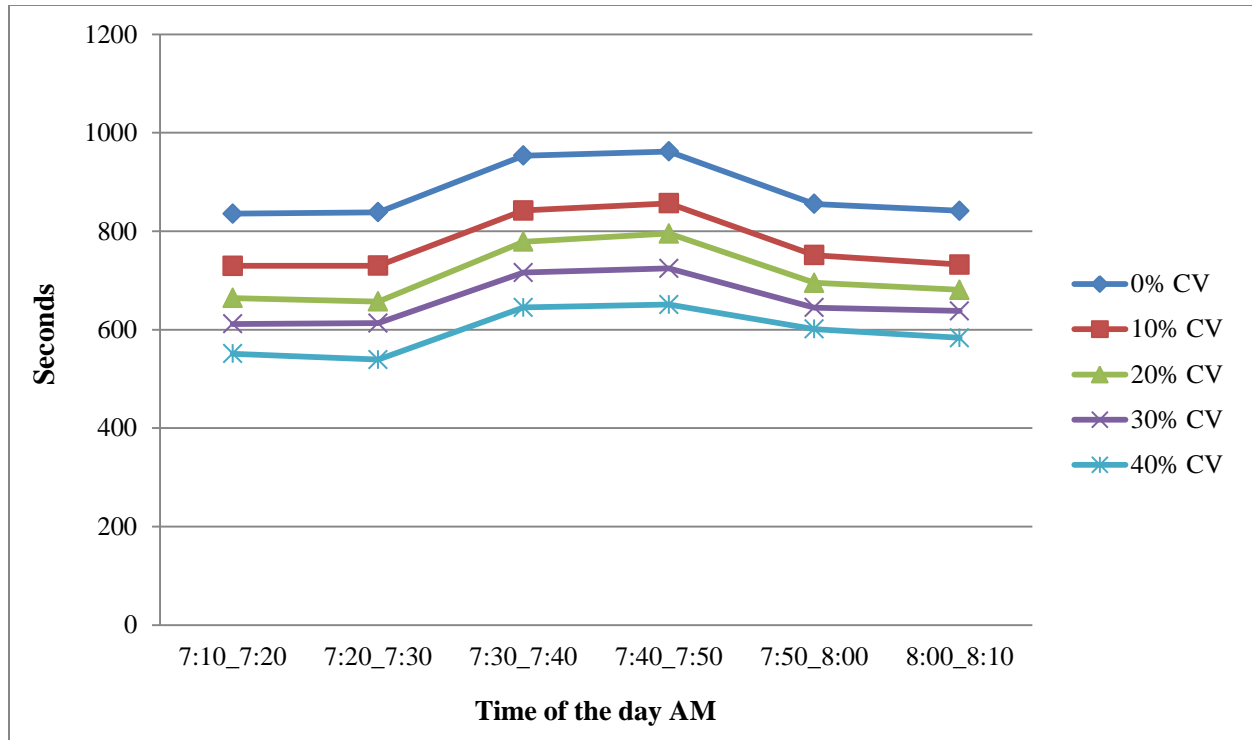
Groups	Count	Sum	Average	Variance
0% V2V	6	-55.9758	-9.3293	0.426793
10% V2V	6	-65.022	-10.837	0.31915
20% V2V	6	-66.7052	-11.1175	0.637872

ANOVA

Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	11.09926	2	5.549628	12.03115	0.000763	3.68232
Within Groups	6.919074	15	0.461272			
Total	18.01833	17				

#### 4.3.3.3 Test case 3: 100% demand loading

Similar to test cases 1 and 2, the experiments have been designed to replicate 1 hour AM peak (7-8 AM) for 6 time intervals. As shown in Figure 4.9, for 100% demand factor, the average P2P journey time is decreased where CV is implemented in the network. In this test case, implementing 40% CV leads to 31% mobility improvements as Figure 4.4 already showed.



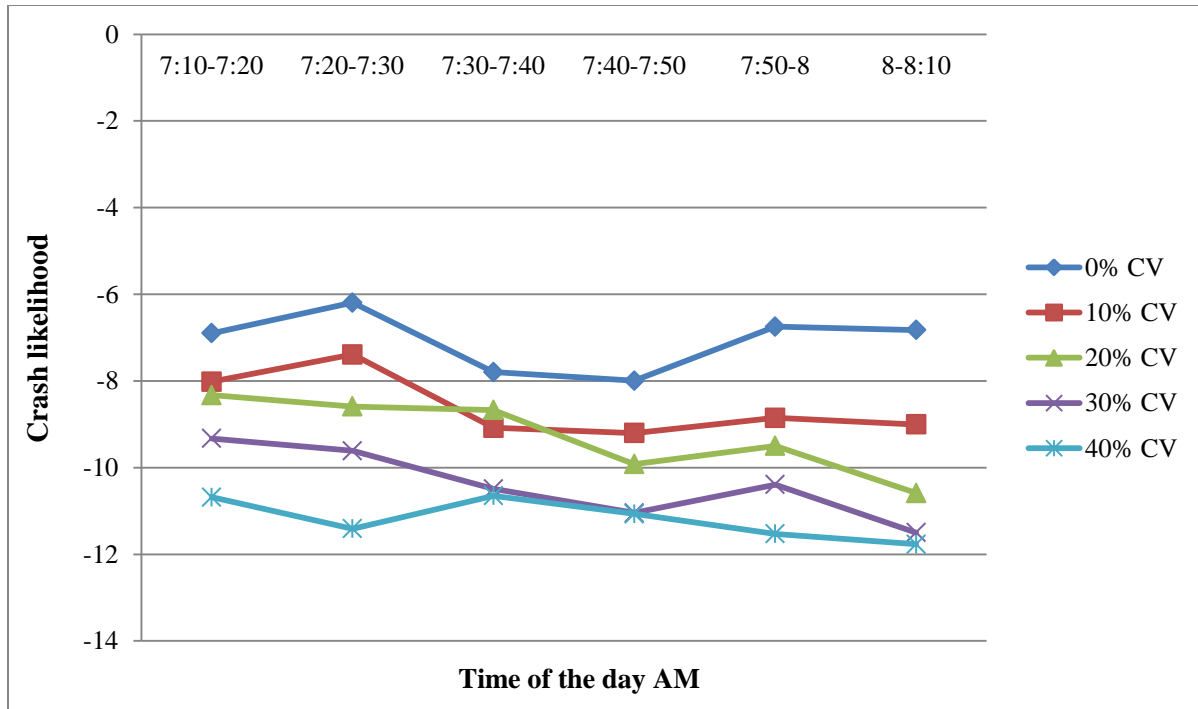
**Figure 4.9 P2P journey time (sec) in tested network (100% demand load) [6]**

The ANOVA test demonstrates a significance difference amongst all CV categories for test case 3 at the 95% level of confidence (Table 4.7).

**Table 4.7 ANOVA Test for 100% demand loading [6]**

Anova: Single Factor						
SUMMARY						
<i>Groups</i>	<i>Count</i>	<i>Sum</i>	<i>Average</i>	<i>Variance</i>		
Non V2V	6	5286.71	881.1183	3581.319		
10% V2V	6	4641.8	773.6333	3525.815		
20% V2V	6	4271.688	711.9479	3585.488		
30% V2V	6	3948.02	658.0033	2505.048		
40% V2V	6	3570.94	595.1567	2175.278		
ANOVA						
<i>Source of Variation</i>	<i>SS</i>	<i>df</i>	<i>MS</i>	<i>F</i>	<i>P-value</i>	<i>F crit</i>
Between Groups	289506.3	4	72376.58	23.54024	3.62E-08	2.75871
Within Groups	76864.74	25	3074.59			
Total	366371.1	29				

Crash likelihood measure for this test case shows that CV decreases the incident risk and consequently improves the safety benefit (Figure 4.10). Moreover, Table 4.2 showed that 40% CV has the highest effect in 100% demand loading where the ORCI value is more than 4.



**Figure 4.10 Crash likelihood prediction in tested network (100% demand load) [6]**

Unlike test cases 1 and 2, Figure 4.10 shows that the effect of 30% and 40% CV on improving incident probability is somehow similar as in some intervals their values are almost the same (violet line tangents the light blue line).

#### 4.3.4 Interpretations of results

Comparing all these three cases indicates that the value of mobility index gets lower as the demand factor increases. The average P2P journey time in the tested route for 80% demand loading is 44% higher than the case of 60%. Also, this amount for 100% demand factor is 117% and 51% higher than corresponding value for 60% and 80% respectively.

The impact of the CV on traffic safety index in the all test cases leads to a further finding. As the ORCI values in Table 4.2 indicate, higher percentage of CV in the network (40%) has the more efficiency in non-overloaded freeways (4.47 vs. 4.11).

## 4.4 Scenario 3

Previous case studies have mainly focused on the potential of using CV system to improve mobility and safety under several congestion levels in the examined network. In this case study, we have developed an integration of a wireless network simulator and a vehicle traffic simulator for the purpose of implementation and evaluation of DSRC based vehicular communication protocols and their applications in the context of ITS. PARAMICS simulator was used to model the traffic flow and required CV extensions to create location-specific incidents and collect traffic information. However, it does not have the ability to simulate different communication protocols and network parameters; therefore, OPNET discrete event simulator has been used to implement and evaluate the real-time V2V and V2I communications, including addressing, routing, and propagating messages. Integration of these two simulators helped us to evaluate various DSRC ranges to detect failure and latency of communications.

The results of designated test case 1 on the integrated simulations show the sensitivity of data transmission rate, data loss and delay to DSRC communication range of RSUs. Moreover, results of test case 2 indicate how the penetration rate of CV has impacts on overall throughput and data packet drops of the whole network.

### 4.4.1 Study area and design of experiment

Evaluation scenarios using the traffic network (Figure 4.1) were simulated for the morning AM peak traffic from 7:00 AM to 8:10 AM. The first 10 minutes was considered as a warm-up period and was disregarded from the evaluation. Since evaluations were meant to demonstrate wireless simulation capabilities, the traffic demand used in the experiments was calibrated to provide a reasonable reflection of peak-hour traffic conditions. For each scenario, statistics were recorded in CSV file every 25 seconds and used as the input for OPNET simulation.

Specification of the simulation is mainly the same as scenario 2 (Figure 3.3), an incident that blocks one lane was generated at predefined location and start time between 7:20-7:30 A.M. After 15 minutes, the incident and its relevant congestion would be cleared from the network.

Since PARAMICS does not allow location collision to be specified, additional program modules have been added to ensure that a specified vehicle would be implicated in a collision and the time and location of the incident are exactly defined. The CV that detects the incident or is involved in the incident sends warning messages to other CVs. Next all the CVs which were informed about the incident send messages to the nearest RSU in DSRC range. RSU then notifies the control center which decides about the next action and informs all the relevant RSUs and they, in turn, advise all the vehicles which fall into the DSRC range defined for them. Also, RSUs should notify the VMS to inform non-CV about the warning messages.

After simulation ends in PARAMICS, the CSV trajectory file was converted to the XML input file for the OPNET. As mentioned in chapter 3, the file includes information about id, type, coordinates and speed of vehicles. In the wireless simulation environment the fixed nodes were created and the location of mobile nodes would be initiated. In particular, each fixed node in OPNET corresponds to a CV device (i.e., RSU or VMS) and each vehicle in PARAMICS is represented as a mobile node in OPNET. Each node performs different functions such as data collection, exchange, process, and dissemination.

The facility of providing vehicular internet access has to be supported by fixed WLAN such as IEEE 802.11 (WiFi) [21] or by WPAN such as IEEE 802.15 (Bluetooth) [132]. In this scenario, the simulation environment is defined to have WiFi communication protocol to implement IEEE 802.11p as the vehicular internet access system for V2V and V2I communications.

After every time intervals (25 seconds) the placement of mobile nodes was updated [27]. When the location of one vehicle did not change after 4 times interval in OPNET, which means in programmed task that the incident happened, the same scenario of the V2V and V2I connections, message propagation and accident clearance, simulated in PARAMICS, were modeled in OPNET and the results were recorded.

When a vehicle comes in the predefined DSRC range of a RSU, a communication link is established. This link is created as the vehicle comes within range and will be disconnected after it passes the coverage threshold of DSRC range. If the communication is possible with more than one RSU from a given location, the closest RSU is always selected which communication should

be established. This will eventually help to have strong signals or other communication parameters.

The essential details of the nodes such as buffer size are the default amounts defined in OPNET for WiFi simulation environment. However, bandwidth of every RSU is defined 54 Mbps [133].

## **4.4.2 Results**

All the results of the test cases are extracted from the OPNET Modeler at global and object level.

### **4.4.2.1 Test case 1**

The first proposed case study was duplicated in OPNET with various DSRC range (100-1000 meters) defined for RSUs to determine its impacts on data traffic transmission rate, data dropped and delay. This data was recorded for 10 minutes interval in OPNET; therefore, the average values were provided for further interpretation. In conducting the experiment, to have better results, while the communication range was varying, other implicit factors such as the percentage of penetration rate of CVs or demand loading was assumed constant. The selection of these parameters was based on the best results obtained from scenario 2. Given that, the percentage of CV is selected to be 40%, while the demand loading is 100% to reflect a typical AM Peak weekday traffic level.

Table 4.8 presents the results from 10 simulation runs with 10 various communication ranges executed in OPNET. They are at the object level and extracted from RSU node statistics. The output results show that longer communication ranges lead to a higher frequency of data transmissions as vehicles are able to come within range of more RSUs and more vehicles are able to communicate with them. It is due to the ability of infrastructures to capture new vehicles on additional links in the network. Therefore CV may then upload the messages to more than one RSU.

A certain delay usually exists between the moment a message is sent and the moment it is received at a RSU. In the simulation model, the essential factor affecting latency is the time needed for a vehicle to come within range of an RSU. In reality, data communication factors could also add more delays. The importance of considering latency is linked to its potential

impacts on safety and mobility applications. The age of received information may also play a critical role, as received data will gradually lose usability. An increase in communication range leads to lower delay at every RSU.

This reduction in delay, as range of RSU increases, is explained by a greater number of opportunities for vehicles to come within range of an RSU. However, as it can be seen in Table 4.8, after increasing the communication range to 800 meters and more, there is a substantial increase in delay and also data drops. It can be explained by having the buffer full. With an increase in the number of vehicles communicating with RSUs, the buffer may become full. When this happens, older messages are discarded to make room for newer ones. This delay links with the number of messages dropped due to a full buffer.

**Table 4.8 The effects of DSRC range for every RSU**

RSU in Range (m)	Average data traffic transmission rate per RSU (Packet/sec)	Average Data Dropped (Buffer overflow) Packet/Sec	Delay for every RSU (Sec)
100	58.1	0	55.6
200	66.3	0.1	53.6
300	73.4	0.45	42.4
400	85.1	7.4	33.1
500	98.7	11.3	28.9
600	122.5	29	16.4
700	147.5	32.5	11.6
800	169.4	69.8	27.4
900	194.2	91.2	46.5
1000	218.4	124.5	56.2



In order to validate our results, we used clustering method to compare distance amount between each group. Groups of DSRC range with their assigned values are used in Minkowski formula where  $r=1$  (Manhattan measure):

$$\text{Distance} = \sum_{k=1}^n (|p_k - q_k|^r)^{\frac{1}{r}} \quad (4.1)$$

The highest distance amount is between groups of 700 and 800 meters of DSRC ranges. This amount is also high in 800-900 and 900-1000 meters groups. Interpretation of these amounts shows that when we change the RSU ranges from 700 to 800 meters and then higher amounts, the output which are data packet drops and delay are getting increased and worsen.

Therefore, the optimum range for DSRC, as the effective communication range, is around 600 to 700 m, at most, whereas the demand loading is 100% in the network. This conforms to our assumption in scenario 2 when we considered these amounts for simulating DSRC based V2V and V2I in week day's traffic density.

As the consequence, if our goal is to place RSUs in the freeway to fully cover the network area and have the best transmission rate, the results recommend that the radius coverage of 600 - 700 meters is preferable and effective.

#### 4.4.2.2 Test case 2

In the second case study, to study the impact of penetration rate of CV on IEEE 802.11p MAC overall load, throughput and data drops of the whole network, we simulated scenarios with varying number of CV nodes ranging from 10% - 100%. The summary of results is shown in Table 4.9.

Table 4.9 indicates the average number of CVs in RSU defined DSRC range at every time interval. This is acquired from the number of transmission attempts to the RSU node from CV nodes in OPNET. Table 4.9 also shows the average data traffic transmission and data packet dropped for overall network in global level for the whole simulation run with its different nodes and modules (i.e. RSU, CV, etc.). This data was extracted from OPNET for every 10 minutes time interval and the average amounts of all intervals used for the interpretation.

In order to have precise results, the demand loading and DSRC range are constant, 100% and 800 meters respectively; while the percentage of CV is variable. Demand loading is to present typical week days and DSRC range is its associated number as in [27].

**Table 4.9 The effects of percentage of CV on the network parameters**

CV penetration rate (%)	Average number of CV in RSU range (Veh/RSU)	Average data traffic transmission – Throughput (Packet/Sec)	Average data packet dropped – Buffer overflow (Packet/Sec)
10	11.9	41.0	16.3
20	24.3	47.2	27.5
30	36.7	56.4	37.1
40	45.2	89.0	70.8
50	59.7	110.7	166.9
60	73.1	124.4	221.8
70	88.2	143.9	293.7
80	91.7	172.8	387.7
90	105.0	193.7	479.2
100	120.4	224.3	591.0

Table 4.9 shows the results summary of 10 test case simulations varying the percentage of CV in the network. As the percentage of CV increases, throughput and data dropped increase since more CV leads to more V2V and V2I connections. Although, as shown in Table 4.9, when there is 40% of CV and more, the throughput and number of packet dropped (due to buffer overflow) increase drastically. This can be explained by the fact that having more percentage of

vehicles lowering their speed, as they get aware about the incident ahead, will lead to more congestion on the routes the traffic is being advised to decrease speed and proves the adverse results on improving travel time while having more than 40% of CV in scenario 2 [13]. Moreover, when a large number of fixed and mobile nodes are contending to access connections and data transmissions, because of full buffer, the number of overall data packet dropped in the network grows. This may lead to the lost or delay of message transmission in RSUs and as the consequence, a percentage of vehicles will not receive the warning messages about the possible incidents in their route. In other words, if we have loss or delay for message transmission, RSUs will not transmit this message on time, so VMSs will have delay to show the message and some of the vehicles would miss the warnings. This will cause a congested area around the incidents which will have negative effects on safety and mobility indices.

These results are expected to vary based on the distances between RSUs. The above results are related to 800 meter DSRC range, but it can be different when we, for example, consider 600 meters as the communication range. However, reducing the distance between RSUs excessively might not be interesting for companies which are using these results to select the number of infrastructures for the real world implementation, since the expenses may get high and not affordable. This impact of different RSU distances, on the selection of optimum percentage of CV to have the least load and throughput, is left to the future works.

## **4.5 Summery**

This chapter we presented 3 scenarios in order to evaluate the proposed CV system. The first 2 scenarios were presented to assess the impact of CV system on improving mobility and safety indices. In scenario 1, we showed the percentage CVs in the network is effective on improving 2 indices where applied V2V and V2I is accompanied with advisory speed and re-routing guidance. However, this scenario had some limitations. For instance, DSRC was not implemented as the communication range for V2V and V2I. Also, the advisory speed was only designed and implemented for upstream of the incidents.

In scenario 2, we solved the mentioned limitations. Also, more percentages of CV have been examined. We developed predefined incident to precisely study the effect of CV system in collision event. In this way the conditions of all simulation runs were somehow the same. We

assessed 3 test cases which were different in terms of demand loading and again, we showed improvements on safety and mobility indices where CV system is applied in the network.

In scenario 3, in order to evaluate critical factors concerning V2V and V2I, we studied the throughput and load of the system. By investigating 2 test cases we presented the optimum selection of DSRC range and CV percentage. These selections decrease the data packet loss and delay of the communications to effectively ensure that warning messages will be received by RSUs and vehicles, and then safety and mobility improvements can be made.

## **Chapter 5: Conclusion and Future works**

### **5.1 Thesis Summery and Contributions**

In this research study, we aimed to implement, simulate and assess the performance of CV system, which are an emerging type ITS, to fulfill our proposed goals. We developed V2V and V2I communications based on DSRC to: (1) evaluate the effect of CV system in the case of collision occurrence on improving safety and mobility benchmarks, (2) study the total throughput and load of the system to provide insights about the effective values and sensitivity of DSRC range and CV penetration rate on the data loss and latency related to DSRC based V2V and V2I communications.

In order to reach our goals, we used the simulation environment instead of real world as the field test of CV system cannot be done before actually deploying the system [5], particularly when testing the system needs applying costly and temporary equipment. Given that, by doing the literature reviews and comparing different simulators and studies, we chose PARAMICS traffic microsimulator and OPNET wireless network simulator, as they seemed to be the best selections to meet our demands through the research objectives. However, during the implementing CV system, we encountered problems which were needed to be solved to satisfy our goals. The problems were limitations of PARAMICS to (1) model V2V and V2I communications as the essential part of demonstrating CV system, (2) create predefined incidents to evaluate CV system in the case of incident occurrence. Moreover, (3) limitation of OPNET and PARAMICS to solely simulate the DSRC based V2V and V2I connections, as OPNET does not have the ability to simulate traffic networks, junctions, vehicles, etc. and PARAMICS cannot simulate various network protocols and CV connections realistically.

We addressed the above challenges and fulfilled our goals by performing the following steps which are also the key contributions of the research described in this thesis. They can be listed as follows:

1. Applying the analysis and design of MaSE methodology to model CV system as the MAS. With this method, we analysed, designed and developed a DSRC based V2V assisted V2I traffic information system. It views the CV modules as the MAS and

- particularly determines its various types of agents, their connections and conversations, as described in section 3.3.
2. Implementation of CV system as the extension to the PARAMICS traffic microsimulator. As mentioned before, PARAMICS does not have the ability to simulate V2V and V2I connections; thus, we implemented the agent-based CV design of MaSE methodology and predefined incidents in PARAMICS using two APIs. The objective was to explore a strategy for improving safety and mobility indices under hazardous situations on freeways, specifically by using advisory speed and re-routing guidance in V2V and V2I systems, as described in section 3.4.
  3. Integration of PARAMICS and OPNET network simulators to evaluate DSRC protocol in CV system. Since PARAMICS itself does not have the ability to study critical communication factors concerning V2V and V2I, the combination of these two simulators have been created. We implemented CVs, CV infrastructures and IEEE 802.11p, using WLAN protocol to simulate DSRC standard, to create V2V and V2I connections, as described in section 3.7.
  4. Importing vehicular trajectory file extracted from PARAMICS into OPNET to specifically mimic the vehicles, as the mobile nodes, and move their locations based on the route they took in their trip in traffic network. This includes converting the CSV, output file from PARAMICS, to XML to be compatible with the types of input OPNET accepts. The mobility trace file consists of data related to the type of the vehicles (CV or non-CV) and their id, coordinates and speeds for every defined time interval. Some extensions, such as removing the vehicle nodes as they reach their destinations, was programmed in process model using proto-C language, as described in section 3.7.1.
  5. Examining various test cases on a section of Deerfoot trail, Calgary, Alberta, as the tested network, to evaluate the impacts of developed V2V and V2I communications by estimating and disseminating traffic safety and mobility parameters. The results of experimentation showed overall improvement in both mobility and safety indices. The former benchmark was measured by the P2P travel time along the freeway, while

the latter was represented by the crash likelihood and the ORCI measurements for the rear-end and lane-change crash risks, as described in sections 4.2 and 4.3.

6. Presenting a scenario on the integration of PARAMICS and OPNET. From the results, we presented the optimal selection of DSRC communication range of RSUs and ultimate percentage of CVs, for AM peak hour of weekday's congestion level in the network, to have the least data packet drops and time latency in V2V and V2I communications, as described in section 4.4.

These contributions allowed us to reach our two goals. In fulfilling the first goal, our approach was the first that performed the following 5 steps all together in the system, to reduce the travel time and crash risk which leads to the improvements of mobility and safety applications: (1) adding two APIs to PARAMICS to simulate CV system, (2) implementing and assessing both V2V and V2I in the tested network, (3) simulating DSRC standard as the vehicular internet access, (4) considering the implementation of advisory speed and re-routing applications in CV system, and (5) evaluating the whole system in the case of incident occurrence.

Moreover, our approach of reaching the second goal was novel because, as to our knowledge, no previous research integrated PARAMICS with OPNET to implement CV system, V2V/V2I and DSRC protocol. Moreover, these works all together were accompanied with implementing incident. By examining total throughput and load of the system, we proposed the best selection of DSRC range and percentage of CVs in particular network conditions to improve the effectiveness of the system by reducing time lag and data packet drops. This will eventually help to enhance the safety and mobility indices in the network.

## **5.2 Future Works**

While the proposed approaches have already shown promising results, there are several issues that can be addressed in future research:

1. In the first approach, the assumptions that all vehicles are always able to communicate with RSUs in range can be relaxes. The reason is that, as the buffer overflow happens for

the RSU, some messages may get lost or a considerable delay would happen for the transmission.

2. The integrated simulators can be used to assess data collection performance under various RSU placements and to determine the ideal number and placement of RSUs to achieve the desired performance. Performance could also be evaluated under various traffic demand conditions and percentage of CVs.
3. The combined simulators can also be used to develop a prioritization strategy for handling message jamming or conflicting information during V2V and V2I communications.
4. Different wireless protocols can be tested in CV system. For example, a Bluetooth interface (IEEE 802.15) can be simulated using WPAN protocol to compare its impacts with WiFi in different conditions to present the best selection for creating vehicular internet access. Moreover, simulation of HTTP or other protocols can be used to let vehicles connect to cloud and help them to reach their point of interests.
5. Pedestrians can be added to simulations by using iPhone or android device nodes in WiFi simulation. This can be helpful to design several scenarios such as sending messages to people to warn them about approaching vehicles travelling a high speed (i.e. reducing pedestrian collision and their fatality).
6. In order to study the behavior and the reaction of drivers after receiving warning messages, the possibility of sending OPNET feedbacks to PARAMICS should be examined. For example, different drivers' behavior implications may be made based on warning messages such as illegally changing the routes to avoid the congestion. These behavioral issues can be modeled and captured to propose the strategies or parameters in order to solve them.
7. Cost Benefit Analysis (CBA) can be used to assess the economic viability of CV scenarios in order to determine the optimized scenario.
8. Last but not least, the simulation results can be implemented in real world to realistically evaluate the proposed system and scenarios.



## References

- [1] Q. Xu, R. Segupta, D. Jiang, and D. Chrysler, "Design and analysis of highway safety communication protocol in 5.9 GHz dedicated short range communication spectrum," in *Vehicular Technology Conference, 2003. VTC 2003-Spring. The 57th IEEE Semiannual*, 2003, pp. 2451-2455.
- [2] C. M. V. T. *Government of Canada, Transport Canada, Statistics and Data*, available at: <http://www.tc.gc.ca/eng/motorvehiclesafety/resources-researchstats-menu-847.htm>.
- [3] B. Maitipe, U. Ibrahim, and M. I. Hayee, "Development and Field Demonstration of DSRC Based V2V-Assisted V2I Traffic Information System for the Work Zone," 2012.
- [4] E. Paikari, L. Kattan, S. Tahmasseby, and B. H. Far, "Modeling and simulation of advisory speed and re-routing strategies in connected vehicles systems for crash risk and travel time reduction," in *Electrical and Computer Engineering (CCECE), 2013 26th Annual IEEE Canadian Conference on*, 2013, pp. 1-4.
- [5] M. Abdel-Aty, J. Dillmore, and A. Dhindsa, "Evaluation of variable speed limits for real-time freeway safety improvement," presented at the Accident analysis & prevention, 2006.
- [6] S. Tahmasseby, E. Paikari, L. Kattan, B. Far, A. Radmanesh, and M. Moussavi, "Modelling and assessing the benefits of Connected Vehicle initiative in a micro-simulation environment," Unpublished report for Encom Wireless Data Solutions Inc. , NSERC Engage Grant, 2013.
- [7] F. Choa, R. T. Milam, and D. Stanek, "Corsim, paramics, and vissim: What the manuals never told you," in *Ninth TRB Conference on the Application of Transportation Planning Methods*, 2004.
- [8] S. L. Jones, A. J. Sullivan, N. Cheekoti, M. D. Anderson, and D. Malave, *Traffic simulation software comparison study* vol. 2217: University Transportation Center for Alabama, 2004.
- [9] S. Panwai and H. Dia, "Comparative evaluation of microscopic car-following behavior," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 6, pp. 314-325, 2005.
- [10] P. Hidas, "A functional evaluation of the AIMSUN, PARAMICS and VISSIM microsimulation models," *Road and Transport Research*, vol. 14, 2005.
- [11] J. W. Shaw and D. H. Nam, "Micro-Simulation: Freeway System Operational Assessment and Project Selection in Southeastern Wisconsin: Expanding the Vision," in *Transportation Research Board 81 st Annual Meeting, Washington, DC*, 2002.
- [12] N. AIMSUN, "Manual 6.0," *Transport Simulation Systems*, 2009.
- [13] L. Kattan, M. Moussavi, B. Far, C. Harschnitz, A. Radmanesh, and S. Saidi, "Evaluating the Potential Benefits of Vehicle to Vehicle Communication (V2V) under Incident Conditions in the PARAMICS Model," in *13th International IEEE Conference on Intelligent Transportation Systems Conference (ITSC 2010), Proceedings on the Workshop on " Emergent Cooperative Technologies In ITS", 19-22 September 2010, Madeira Island, Portugal*, 2010.
- [14] P. P. T. V. AG, *VISSIM 5.40 User Manual*: epubli GmbH, 2012.
- [15] M. Abdel-Aty, R. J. Cunningham, V. V. Gayah, and L. Hsia, "Dynamic variable speed limit strategies for real-time crash risk reduction on freeways," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2078, pp. 108-116, 2008.
- [16] M. Abdel-Aty, N. Uddin, and A. Pande, "Split models for predicting multivehicle crashes during high-speed and low-speed operating conditions on freeways," *Transportation research record: journal of the transportation research board*, vol. 1908, pp. 51-58, 2005.
- [17] J. Lee and B. Park, "Evaluation of route guidance strategies based on vehicle-infrastructure integration under incident conditions," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2086, pp. 107-114, 2008.

- [18] H. Park and B. L. Smith, "Investigating benefits of IntelliDrive in freeway operations: Lane changing advisory case study," *Journal of Transportation Engineering*, vol. 138, pp. 1113-1122, 2012.
- [19] H. Tanikella, B. L. Smith, G. Zhang, B. Park, and W. T. Scherer, "Simulating operations applications of vehicle infrastructure integration: Traffic monitoring case study," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2000, pp. 35-43, 2007.
- [20] B. Dalla Chiara, F. Deflorio, and S. Diwan, "Assessing the effects of inter-vehicle communication systems on road safety," *Intelligent Transport Systems, IET*, vol. 3, pp. 225-235, 2009.
- [21] N. I. Sarkar, "Performance Modeling of IEEE 802.11 WLAN using OPNET: A Tutorial," *Handbook of Research on Discrete Event Simulation Environments: Technologies and Applications*, p. 398, 2010.
- [22] Y. Yang, M. Marina, and R. Bagrodia, *Evaluation of multihop relaying for robust vehicular Internet access*: IEEE, 2007.
- [23] U. S. Department of transportation. *Research and Innovative Technology Administration - available at: [http://www.its.dot.gov/factsheets/dsrc\\_factsheet.htm](http://www.its.dot.gov/factsheets/dsrc_factsheet.htm)*.
- [24] V. Bychkovsky, B. Hull, A. Miu, H. Balakrishnan, and S. Madden, "A measurement study of vehicular internet access using in situ Wi-Fi networks," in *Proceedings of the 12th annual international conference on Mobile computing and networking*, 2006, pp. 50-61.
- [25] R. Gass, J. Scott, and C. Diot, "Measurements of in-motion 802.11 networking," in *Mobile Computing Systems and Applications, 2006. WMCSA'06. Proceedings. 7th IEEE Workshop on*, 2005, pp. 69-74.
- [26] J. Ott and D. Kutscher, "Drive-thru Internet: IEEE 802.11 b for" automobile" users," in *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, 2004.
- [27] F. Dion, J.-S. Oh, and R. Robinson, "Virtual testbed for assessing probe vehicle data in intellidrive systems," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 12, pp. 635-644, 2011.
- [28] A. Avila, G. Korkmaz, Y. Liu, H. Teh, E. Ekici, F. Ozguner, *et al.*, "A complete simulator architecture for inter-vehicle communication intersection warning systems," in *Intelligent Transportation Systems, 2005. Proceedings. 2005 IEEE*, 2005, pp. 461-466.
- [29] L. Bononi, M. Di Felice, M. Bertini, and E. Croci, "Parallel and distributed simulation of wireless vehicular ad hoc networks," in *Proceedings of the 9th ACM international symposium on Modeling analysis and simulation of wireless and mobile systems*, 2006, pp. 28-35.
- [30] P. Farradyne, "VII architecture and functional requirements," *Report prepared for the ITS Joint Programme Office, Federal Highway Administration, US Department of Transportation*, 2005.
- [31] B. A. Hamilton, "Vehicle infrastructure integration (VII) concept of operations," ed. Federal Highway Administration, U.S. Department of Transportation, Washington, DC., 2006.
- [32] M. Li, Z.-j. Zou, F. Bu, and W.-B. Zhang, "Application of vehicle infrastructure integration data on real-time arterial performance measurements," in *Transportation Research Board 87th Annual Meeting*, 2008.
- [33] B. L. Smith, B. B. Park, H. Tanikella, and G. Zhang, "Preparing to use vehicle infrastructure integration in transportation operations: phase i," 2007.
- [34] G. Abu-Lebdeh, "Exploring the potential benefits of intellidrive-enabled dynamic speed control in signalized networks," in *Transportation Research Board 89th Annual Meeting*, 2010.
- [35] Y. Pilli-Sivola, "State of the Art in Finland Concerning RWIS and Variable Message Signs," *Finnish National Road Administration*, vol. 2, 2000.
- [36] E. Paikari, S. Tahmasseby, and B. H. Far, "A Simulation-based Benefit Analysis of Deploying Connected Vehicles Using Dedicated Short Range Communication," in *Intelligent Vehicles Symposium, 2014 IEEE Intelligent Transportation Systems Society*, 2014, pp. 1-6.

- [37] U. S. Department of transportation. *Research and Innovative Technology Administration* - available at: <http://www.its.dot.gov>.
- [38] A. Standard, "E2213-03," *Telecommunications and information exchange between roadside and vehicle systems*, vol. 5, 2010.
- [39] R. Kandarpa, M. Chenzaie, M. Dorfman, J. Anderson, J. Marousek, I. Schworer, et al., "Final report: Vehicle infrastructure integration (VII) proof of concept (POC) test–Executive summary," ed: Feb, 2009.
- [40] F. Bai and H. Krishnan, "Reliability analysis of DSRC wireless communication for vehicle safety applications," in *Intelligent Transportation Systems Conference, 2006. ITSC'06. IEEE*, 2006, pp. 355-362.
- [41] A. F. Molisch, F. Tufvesson, J. Karedal, and C. Mecklenbrauker, "Propagation aspects of vehicle-to-vehicle communications-an overview," in *Radio and Wireless Symposium, 2009. RWS'09. IEEE*, 2009, pp. 179-182.
- [42] Q. Xu, T. Mak, J. Ko, and R. Sengupta, "Vehicle-to-vehicle safety messaging in DSRC," in *Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks*, 2004, pp. 19-28.
- [43] Y. Qian, K. Lu, and N. Moayeri, "A secure VANET MAC protocol for DSRC applications," in *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*, 2008, pp. 1-5.
- [44] J. Zhu and S. Roy, "MAC for dedicated short range communications in intelligent transport system," *Communications Magazine, IEEE*, vol. 41, pp. 60-67, 2003.
- [45] 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*, 2005, pp. 69-78.
- [46] H. Haj Salem, J. Chrisoulakis, M. Papageorgiou, N. Elloumi, and P. Papadakos, "The use of METACOR tool for integrated urban and interurban traffic control. evaluation in corridor peripherique, Paris," in *Vehicle Navigation and Information Systems Conference, 1994. Proceedings., 1994*, 1994, pp. 645-650.
- [47] T. Manual, "Travel Demand Modeling with TransCAD," ed: Version.
- [48] S. A. Boxill and L. Yu, "An Evaluation of Traffic Simulation Models for Supporting ITS," 2000.
- [49] S. Krauß, "Microscopic modeling of traffic flow: Investigation of collision free vehicle dynamics," Universität zu Köln., 1998.
- [50] N. B. Taylor, "The CONTRAM dynamic traffic assignment model," *Networks and Spatial Economics*, vol. 3, pp. 297-322, 2003.
- [51] M. Mahut and M. Florian, "Traffic simulation with dynameq," in *Fundamentals of Traffic Simulation*, ed: Springer, 2010, pp. 323-361.
- [52] W. Burghout, "Hybrid microscopic-mesoscopic traffic simulation," KTH, 2004.
- [53] M. Pursula, "Simulation of traffic systems-an overview," *Journal of Geographic Information and Decision Analysis*, vol. 3, pp. 1-8, 1999.
- [54] P. D. Prevedouros and Y. Wang, "Simulation of large freeway and arterial network with CORSIM, INTEGRATION, and WATSim," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1678, pp. 197-207, 1999.
- [55] R. Wiedemann, "Modelling of RTI-Elements on multi-lane roads," in *DRIVE CONFERENCE (1991: BRUSSELS, BELGIUM)*, 1991.
- [56] R. Wiedemann, "Simulation des Straßenverkehrsflusses. Schriftenreihe Heft 8," *Institute for Transportation Science, University of Karlsruhe, Germany*, 1994.
- [57] M. Fellendorf, "VISSIM: A microscopic simulation tool to evaluate actuated signal control including bus priority," in *64th Institute of Transportation Engineers Annual Meeting*, 1994, pp. 1-9.

- [58] Q. Paramics, "The paramics manuals, version 6.6. 1," *Quastone Paramics LTD, Edinburgh, Scotland, UK*, 2009.
- [59] N. T. Ratroun and S. M. Rahman, "A comparative analysis of currently used microscopic and macroscopic traffic simulation software," *The Arabian Journal for Science and Engineering*, vol. 34, pp. 121-133, 2009.
- [60] M. Ben-Akiva, H. Koutsopoulos, and Q. Yang, "User's guide for MITSIMLab and road network editor (RNE)," *ITS Program at the Department of Civil and Environmental Engineering, Massachusetts Institute of Technology, Boston, USA, Technical Report Draft*, 2002.
- [61] P. D. Prevedouros and H. Li, "Comparison of Freeway Simulation with INTEGRATION, KRONOS, and KWaves," in *Fourth International Symposium on Highway Capacity, Maui, Hawaii*, 2000, pp. 96-107.
- [62] M. A. Abdel-Aty and A. Dhindsa, "Coordinated use of variable speed limits and ramp metering for improving safety on congested freeways," in *Transportation Research Board 86th Annual Meeting*, 2007.
- [63] C. Lee, B. Hellenga, and K. Ozbay, "Quantifying effects of ramp metering on freeway safety," *Accident Analysis & Prevention*, vol. 38, pp. 279-288, 2006.
- [64] M. Smith, S. Druitt, G. Cameron, and D. MacArthur, "Paramics final report," *Technical Rep. EPCC-PARAMICS-FINAL, Univ. of Edinburgh, Scotland, UK*, 1994.
- [65] Professional ITS Design and development, PARAMICS, Modularity Introduction - available at: <http://www.realtraffic.cn/enweb/paramicsmod.asp?sortsid=7> [Online].
- [66] Q. Meng and H. L. Khoo, "Optimizing contraflow scheduling problem: model and algorithm," *Journal of Intelligent Transportation Systems*, vol. 12, pp. 126-138, 2008.
- [67] R. Kandarpa, M. Chenzaie, J. Anderson, J. Marousek, T. Weil, F. Perry, et al., "Final Report: Vehicle Infrastructure Integration Proof-of-Concept Results and Findings-Infrastructure," *FHWA JPO*, vol. 9, p. 057, 2009.
- [68] J. Anda, J. LeBrun, D. Ghosal, C.-N. Chuah, and M. Zhang, "VGrid: vehicular adhoc networking and computing grid for intelligent traffic control," in *Vehicular Technology Conference, 2005. VTC 2005-Spring. 2005 IEEE 61st*, 2005, pp. 2905-2909.
- [69] M. Bechler, W. J. Franz, and L. Wolf, "Mobile internet access in FleetNet," *13. Fachtagung Kommunikation in verteilten Systemen, Leipzig, Germany*, 2003.
- [70] 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, pp. 36-43, 2006.
- [71] S. Biswas, R. Tatchikou, and F. Dion, "Vehicle-to-vehicle wireless communication protocols for enhancing highway traffic safety," *Communications Magazine, IEEE*, vol. 44, pp. 74-82, 2006.
- [72] R. McHaney, *Computer simulation: a practical perspective*: Access Online via Elsevier, 1991.
- [73] M. Takai, J. Martin, and R. Bagrodia, "Effects of wireless physical layer modeling in mobile ad hoc networks," in *Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*, 2001, pp. 87-94.
- [74] S. Kurkowski, T. Camp, and M. Colagrosso, "MANET simulation studies: the incredibles," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 9, pp. 50-61, 2005.
- [75] F. Haq and T. Kunz, "Simulation vs. emulation: Evaluating mobile ad hoc network routing protocols," in *Proceedings of the International Workshop on Wireless Ad-hoc Networks (IWWAN 2005)*, 2005.
- [76] G. F. Lucio, M. Paredes-Farrera, E. Jammeh, M. Fleury, and M. J. Reed, "Opnet modeler and ns-2: Comparing the accuracy of network simulators for packet-level analysis using a network testbed," *WSEAS Transactions on Computers*, vol. 2, pp. 700-707, 2003.
- [77] K. Fall and K. Varadhan, "The ns Manual (formerly ns Notes and Documentation)," *The VINT project*, vol. 47, 2005.

- [78] Q. Chen, F. Schmidt-Eisenlohr, D. Jiang, M. Torrent-Moreno, L. Delgrossi, and H. Hartenstein, "Overhaul of iee 802.11 modeling and simulation in ns-2," in *Proceedings of the 10th ACM Symposium on Modeling, analysis, and simulation of wireless and mobile systems*, 2007, pp. 159-168.
- [79] K. Walsh and E. Sirer, "A staged network simulator (SNS)," *Computer Science Department, Cornell University*, 2003.
- [80] S.-Y. Wang and C.-C. Lin, "NCTUns 6.0: a simulator for advanced wireless vehicular network research," in *Vehicular Technology Conference (VTC 2010-Spring), 2010 IEEE 71st*, 2010, pp. 1-2.
- [81] A. Varga and R. Hornig, "An overview of the OMNeT++ simulation environment," in *Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops*, 2008, p. 60.
- [82] A. Sobeih, J. C. Hou, L.-C. Kung, N. Li, H. Zhang, W.-P. Chen, *et al.*, "J-Sim: a simulation and emulation environment for wireless sensor networks," *Wireless Communications, IEEE*, vol. 13, pp. 104-119, 2006.
- [83] Q. U. Manual, "v3. 8. Scalable Network Technologies, Culver City, CA," ed.
- [84] O. Users' Manual, "OPNET Architecture, OV. 415," ed, 2010.
- [85] W. D. Kelton and A. M. Law, *Simulation modeling and analysis*: McGraw Hill Boston, MA, 2000.
- [86] B. Schmeiser, "Simulation output analysis: A tutorial based on one research thread," in *Simulation Conference, 2004. Proceedings of the 2004 Winter*, 2004.
- [87] K. Pawlikowski, H.-D. Jeong, and J.-S. Lee, "On credibility of simulation studies of telecommunication networks," *Communications Magazine, IEEE*, vol. 40, pp. 132-139, 2002.
- [88] N. I. Sarkar and K. W. Sowerby, "The combined effect of signal strength and traffic type on WLAN performance," in *Wireless Communications and Networking Conference, 2009. WCNC 2009. IEEE*, 2009, pp. 1-6.
- [89] J. Lessmann, P. Janacik, L. Lachev, and D. Orfanus, "Comparative study of wireless network simulators," in *Networking, 2008. ICN 2008. Seventh International Conference on*, 2008, pp. 517-523.
- [90] P. Janacik, J. Lessmann, and M. Karch, "Distributed simulation environment for the ShoX network simulator," in *Networking and Services (ICNS), 2010 Sixth International Conference on*, 2010, pp. 206-211.
- [91] L. Hogue, P. Bouvry, and F. Guinand, "An overview of manets simulation," *Electronic notes in theoretical computer science*, vol. 150, pp. 81-101, 2006.
- [92] S. V. Mallapur and S. R. Patil, "Survey on Simulation Tools for Mobile Ad-Hoc Networks," *International Journal of Computer Networks and Wireless Communications (IJCNC)*, 2012.
- [93] F. J. Martinez, C. K. Toh, J. C. Cano, C. T. Calafate, and P. Manzoni, "A survey and comparative study of simulators for vehicular ad hoc networks (VANETs)," *Wireless Communications and Mobile Computing*, vol. 11, pp. 813-828, 2011.
- [94] V. D. Khairnar and S. Pradhan, "Comparative study of simulation for vehicular ad-hoc network," *arXiv preprint arXiv:1304.5181*, 2013.
- [95] K. Ibrahim and M. C. Weigle, "ASH: Application-aware SWANS with highway mobility," in *INFOCOM Workshops 2008, IEEE*, 2008, pp. 1-6.
- [96] S. Duflos, G. Grand, A. Diallo, C. Chaudet, A. Hecker, C. Balducelli, *et al.*, "Deliverable d 1.3. 2: List of available and suitable simulation components," Technical report, Ecole Nationale Supérieure des Communications (ENST)2006.
- [97] C. Zhu, O. W. Yang, J. Aweya, M. Ouellette, and D. Y. Montuno, "A comparison of active queue management algorithms using the OPNET modeler," *Communications Magazine, IEEE*, vol. 40, pp. 158-167, 2002.

- [98] R. Ghaman, L. Zhang, G. Mchale, and C. Stallard, "The role of traffic simulation in traffic signal control system development," in *Intelligent Transportation Systems, 2003. Proceedings. 2003 IEEE*, 2003, pp. 872-877.
- [99] M. Torrent-Moreno, "Inter-vehicle communications: assessing information dissemination under safety constraints," in *Wireless on Demand Network Systems and Services, 2007. WONS'07. Fourth Annual Conference on*, 2007, pp. 59-64.
- [100] Y. Zang, L. Stibor, G. Orfanos, S. Guo, and H.-J. Reuerman, "An error model for inter-vehicle communications in highway scenarios at 5.9 GHz," in *Proceedings of the 2nd ACM international workshop on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks*, 2005, pp. 49-56.
- [101] H. Wu, J. Lee, M. Hunter, R. Fujimoto, R. L. Guensler, and J. Ko, "Efficiency of simulated vehicle-to-vehicle message propagation in Atlanta, Georgia, I-75 corridor," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1910, pp. 82-89, 2005.
- [102] C. Sommer, Z. Yao, R. German, and F. Dressler, "Simulating the influence of IVC on road traffic using bidirectionally coupled simulators," in *INFOCOM Workshops 2008, IEEE*, 2008, pp. 1-6.
- [103] D. Krajzewicz, M. Bonert, and P. Wagner, "The open source traffic simulation package SUMO," *RoboCup 2006 Infrastructure Simulation Competition*, vol. 1, pp. 1-5, 2006.
- [104] S. Assenmacher, A. Leonhardt, F. Schimandl, and F. Busch, "Simulative Identification of Possibilities and Impacts of V2X-Communication," in *Proceedings of Workshop on traffic modeling: traffic behavior and simulation*, 2008.
- [105] J. a. P. Lee, B. , "Identifying the Characteristics of the Communication Latency of IntelliDrive Environment for a Freeway Merge," presented at the In Proceedings of the 89th Annual Meeting of the Transportation Research Board, Transportation Research Board of the National Academy, Washington, D.C., 2010.
- [106] 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, pp. 31-33, 2008.
- [107] H. Kim, M. Shin, B. Nam, and D. J. Lovell, "An integrated transportation and communication simulation framework for vehicular ad hoc network applications," in *Transportation Research Board 87th Annual Meeting*, 2008.
- [108] H. Xu and M. Barth, "Travel time estimation techniques for traffic information systems based on intervehicle communications," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1944, pp. 72-81, 2006.
- [109] B. Abdulhai, A. Shalaby, J. Lee, and A. Georgi, "Microsimulation modelling and impact assessment of streetcar transit priority options: The Toronto experience," in *National Research Council (US). Transportation Research Board. Meeting (81st: 2002: Washington, DC). Preprint CD-ROM*, 2002.
- [110] 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*, 2005, pp. 87-88.
- [111] S. E. Shladover and T. M. Kuhn, "Traffic probe data processing for full-scale deployment of vehicle-infrastructure integration," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2086, pp. 115-123, 2008.
- [112] A. M. Hyungjun Park, Joyoung Lee, Malathi Veeraraghavan, Byungkyu Park, Brian Lee Smith, "Integrated Traffic/Communications Simulation Evaluation Environment for IntelliDriveSM Applications Using SAE J2735 Dedicated Short Range Communications Message Sets," presented at the 2011 Annual Meeting of the Transportation Research Board and in Review for Publication in the Transportation Research Record, 2011.

- [113] H. Tanikella, G. Zhang, J. Guo, B. Park, and W. Scherer, "Development of a 6 Simulation Architecture to Evaluate Vehicle Infrastructure Integration," *ASCE Journal of Computing in Civil Engineering*, vol. 8, 2007.
- [114] A. Scott and M. Wood, "Developing multiagent systems with agenttool," in *Intelligent Agents VII Agent Theories Architectures and Languages*, ed: Springer, 2001, pp. 46-60.
- [115] M. Wooldridge, N. R. Jennings, and D. Kinny, "The Gaia methodology for agent-oriented analysis and design," *Autonomous Agents and Multi-Agent Systems*, vol. 3, pp. 285-312, 2000.
- [116] E. A. Kendall, S. Kalikivayi, U. Palanivelan, and N. Panvalkar, "Capturing And Structuring Goals: Analysis Patterns," in *EuroPloP*, 1998, pp. 113-122.
- [117] D. Kinny, M. Georgeff, and A. Rao, "A methodology and modelling technique for systems of BDI agents," in *Agents breaking away*, ed: Springer, 1996, pp. 56-71.
- [118] S. A. DeLoach, M. F. Wood, and C. H. Sparkman, "Multiagent systems engineering," *International Journal of Software Engineering and Knowledge Engineering*, vol. 11, pp. 231-258, 2001.
- [119] M. F. Wood and S. A. DeLoach, "An overview of the multiagent systems engineering methodology," in *Agent-Oriented Software Engineering*, 2001, pp. 207-221.
- [120] E. A. Kendall, "Agent roles and role models: New abstractions for intelligent agent system analysis and design," in *International Workshop on Intelligent Agents in Information and Process Management*, 1998.
- [121] A. Tang, T. Veltman, M. Chow, C. Harschnitz, M. Moussavi, L. Kattan, *et al.*, "ElectroSoft report," 2010.
- [122] X. Yan, M. Abdel-Aty, E. Radwan, X. Wang, and P. Chilakapati, "Validating a driving simulator using surrogate safety measures," *Accident Analysis & Prevention*, vol. 40, pp. 274-288, 2008.
- [123] Y. Gardes, A. D. May, J. Dahlgren, and A. Skabardonis, "Freeway calibration and application of the PARAMICS model," in *81st Annual Meeting of the Transportation Research Board, Washington, DC*, 2002.
- [124] C. Lee, B. Hellinga, and F. Saccomanno, "Assessing safety benefits of variable speed limits," *Transportation research record: journal of the transportation research board*, vol. 1897, pp. 183-190, 2004.
- [125] A. Pande, M. Abdel-Aty, and L. Hsia, "Spatiotemporal variation of risk preceding crashes on freeways," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1908, pp. 26-36, 2005.
- [126] D.M. Lane, D. Scott, M. Hebl, R. Guerra, D. Osherson, and H. Zimmer, in *Introduction to Statics*, ed University of Houston.
- [127] E. Paikari and B. H. Far, "Analysis, Design and Implementation of an Agent Based System for Simulating Connected Vehicles," in *Software Engineering and Knowledge Engineering (SEKE), The 26th International Conference on*, 2014, pp. 1-6.
- [128] The City of Calgary , Traffic volume flow maps - available at: <http://www.calgary.ca/Transportation/TP/Pages/Planning/Transportation-Data/Traffic-volume-flow-maps.aspx> [Online].
- [129] J. Vales-Alonso, F. Vicente-Carrasco, and J. J. Alcaraz, "Optimal configuration of roadside beacons in V2I communications," *Computer Networks*, vol. 55, pp. 3142-3153, 2011.
- [130] Adyz. ANOVA, *Business Excellence* - available at: <http://www.bexcellence.org/Anova.html>.
- [131] D. Ni, J. Li, S. Andrews, and H. Wang, "A methodology to estimate capacity impact due to connected vehicle technology," *International Journal of Vehicular Technology*, vol. 2012, 2011.
- [132] K. S. Kwak, S. Ullah, and N. Ullah, "An overview of IEEE 802.15. 6 standard," in *Applied Sciences in Biomedical and Communication Technologies (ISABEL), 2010 3rd International Symposium on*, 2010, pp. 1-6.

- [133] ENCOM Wireless Data Solutions, Broadband Wireless Networks - available at: <http://www.encomwireless.com/products/broadband-wireless-networks> [Online].