VIETNAM NATIONAL UNIVERSITY – HOCHIMINH CITY
HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING

A Novel Adaptive Beacon-based Scheme for Warning Messages Dissemination in Vehicular Ad-Hoc Networks

Author:

Nguyen Dinh Trung Truc – 51304505

Huynh Quang Bao – 51300225

Supervised by:

Dr. Pham Hoang Anh

Referee:

Dr. Le Trong Nhan

A thesis submitted in partial fulfillment of the requirements for the degree of Bachelor of Engineering

in the

Faculty of Computer Science and Engineering
Ho Chi Minh City University of Technology

Ho Chi Minh City, December 18th, 2017

And whatsoever ye do, do it heartily,
as to the Lord, and not unto men;
— Colossians 3:23

In dedication to the Lord, my academic advisor, beloved brothers and sisters …

Acknowledgements

We would like to give the biggest respect to our supervisor, Dr. Pham Hoang Anh, for his trust, support and guidance. The enthusiasm and energy in him really inspire us in our work.

Thank our beloved friends and all the people who have encouraged us during this stage of our life. Thank Bach Khoa University and the Department of Computer Science and Engineering for providing us a chance to experience this study.

Achievements

Parts of this thesis have been successfully published in the following conference and journal:

1. T. D. T. Nguyen, T.-V. Le and H.-A. Pham, "Novel Store-carry-forward Scheme for Message Dissemination in Vehicular Ad-hoc Networks," in *Journal of Information and Communication Technology Express (ICT-Express)*, 2017. ISSN 2405-9595. DOI 10.1016/j.icte.2017.11.009

2. T. D. T. Nguyen, Q.-B. Huynh and H.-A. Pham, "An Adaptive Beacon-based Scheme for Warning Messages Dissemination in Vehicular Ad-Hoc Networks," in *2017 International Conference on Advanced Computing and Applications (ACOMP)*, Ho Chi Minh City, 2017, pp. 47-53. DOI 10.1109/A-COMP.2017.19

Abstract

Vehicular Ad-hoc Networks (VANETs) that allow vehicles to form a network among themselves are considered as one of the key technologies to improve traffic safety. In VANETs, vehicles conventionally disseminate warning messages to their nearby vehicles as soon as a dangerous situation occurs. As an significant solution for improving traffic safety, this mechanism faces two major challenges that are broadcast storm and network partition problems.

In this thesis, after elaborating on fundamental concepts of VANETs and state-of-the-art related works in this field, we propose a novel Adaptive Beacon-based Data Dissemination (ABDDis) scheme to tackle both of the broadcast storm and network partition problems using only one-hop neighbor information. The proposed ABDDis scheme features a novel adaptive beacon mechanism to adaptively adjust the beacon interval to reduce the channel load while maintaining the correct information about nearby vehicles. It aims to handle a drawback of periodically broadcasting beacons. Furthermore, the ABDDis scheme adopts a novel store-carry-forward mechanism to handle the network partition problem.

To evaluate the performance, three state-of-the-art schemes have been compared to the ABDDis scheme by means of realistic simulations. Those schemes are implemented in the Veins simulation framework that provides a bidirectional coupling between the network simulator OMNeT++ and the traffic simulator SUMO. The experimental results show that ABDDis scheme gains an outstanding performance by significantly mitigates the broadcast storm compared to other schemes and maintains a good coverage across various traffic densities. Additionally, the ABDDis scheme also proves its robustness by tolerating reasonable GPS drifts.

Key words: Vehicular network, VANET, Intelligent transportation system, Smart traffic, VEINS, SUMO, OMNet++, Data dissemination scheme.

Tóm tắt

Vehicular Ad-hoc Networks (VANETs) được xem là một công nghệ cho phép xe cộ tự kết nối với nhau tạo thành một mạng để trao đổi dữ liệu với mục tiêu cải thiện an toàn giao thông. Trong mạng VANETs, các xe sẽ phân tán các gói tin cảnh báo tới những xe xung quanh ngay sau khi một tình huống nguy hiểm xảy ra. Mặc dù đây là một giải pháp hiệu quả để cải thiện an toàn giao thông, cơ chế phân tán dữ liệu trong VANETs đối mặt với hai thách thức chính là vấn đề broadcast storm (dư thừa các gói tin phát tán và có thể dẫn đến nghẽn mạng) và network partition (phân chia mạng).

Trong đề tài luận văn này, sau khi nghiên cứu chi tiết các khái niệm cơ bản của mạng VANETs cũng như các đề tài nghiên cứu mới nhất trong lĩnh vực này, chúng tôi đề xuất một giải pháp có tên gọi là Adaptive Beacon-based Data Dissemination (ABDDis) để khắc phục cả hai vấn đề broadcast storm và network partition trong khi chỉ sử dụng thông tin cục bộ giữa các xe gần nhau. ABDDis đề xuất một cơ chế linh hoạt để điều chỉnh chu kỳ gửi beacon (adaptive beacon) để giảm tải kênh truyền trong khi vẫn duy trì tính chính xác của thông tin về các xe xung quanh. Mục đích của cơ chế adaptive beacon là xử lý vấn đề định thời phát tán các gói tin để giảm mức độ broadcast storm. Hơn nữa, ABBDis còn đề xuất cơ chế store-carry-forward để cải thiện vấn đề network partition.

Để đánh giá hiệu suất của giải pháp đề xuất, ba giải thuật đã được đề xuất trong các nghiên cứu gần đây sẽ được so sánh với ABDDis bằng mô phỏng thực tế. Các giải thuật sẽ được hiện thực trên framework mô phỏng Veins - một sự kết hợp giữa công cụ mô phỏng mạng OMNeT++ và công cụ mô phỏng giao thông SUMO. Kết quả thử nghiệm cho thấy ABDDis đạt được hiệu suất vượt trội bởi việc giảm đáng kể vấn đề broadcast storm khi được so sánh với ba giải thuật còn lại trong khi vẫn phân tán gói tin hiệu quả trên nhiều mật độ phương tiện khác nhau. Thêm vào đó, ABDDis còn có thể hoạt động tốt ngay cả khi bị nhiễu GPS.

Từ khóa: mạng giao thông, VANET, hệ thống giao thông thông minh, VEINS, SUMO, OMNet++, giải thuật phân tán dữ liệu.

Contents

List of Figures

List of Tables

Abbreviations

BSS ..... Basic Service Set

CCH ..... Control Channel

DV-CAST ..... Distributed Vehicular Broadcast

DSRC ..... Dedicated Short Range Communication

EDCA ..... Enhanced Distributed Channel Access

FDMA ..... Frequency-division multiple access

GPRS ..... General Packet Radio Service

GPS ..... Global Positioning System

GSM ..... Global System for Mobile Communications

GUI ..... Graphical User Interface

IEEE ..... Institute of Electrical and Electronics Engineers

IPv4 ..... Internet Protocol version 4

IPv6 ..... Internet Protocol version 6

ITS ..... Intelligent Transportation System

IVC ..... Inter-Vehicle Communication

JSF ..... Junction Store and Forward

LLC ..... Logical Link Control

LTE ..... Long Term Evolution

MANET ..... Mobile Ad-hoc Network

MIB ..... Management Information Base

MLME ..... MAC sublayer Management Entity

MMS ..... Multimedia Messaging Service

NED ..... Network description

NJL ..... Nearest Junction Located

NSF ..... Neighbor Store and Forward

OBE ..... On-Board Equipment

PLME ..... Physical Layer Management Entity

QoS ..... Quality-of-Service

RSE ..... Road-Side Equipment

SCH ..... Service Channel

SUMO ..... Simulation of Urban Mobility

TraCI ..... Traffic Control Interface

UID ..... Unique identifier

V2I ..... Vehicle-to-infrastructure

V2V ..... Vehicle-to-vehicle

V2X ..... Vehicle-to-everything

VANET ..... Vehicular Ad-hoc Network

VEINS ..... Vehicles in network simulation

WAVE ..... Wireless Access in Vehicular Environment

WBM ..... WaveBroadcastMessage

WSA ..... WAVE Service Advertisement

WSM ..... WAVE Service Message

WSMP ..... WAVE Short Message Protocol

Wi-PAN ..... Wireless Personal area network

1 Introduction

For many years, traffic safety is among the major issues in the society. According to an annual road accident statistic [12], nearly 1.3 million people die in road crashes each year, up to 50 million people are injured or disabled, which is more than the population of Canada [13]. Furthermore, road traffic crashes rank as the 9th leading cause of death, and also, they have cost $518 billion globally. Consequently, World Health Organization has suggested that by 2030, road crashes will become the 5th leading cause of death [14].

There have been many efforts aiming to improve road safety, along with reducing pollution and congestion. As a consequence, in recent years, Vehicular Ad-hoc Networks (VANETs) have become a trending research topic in both academic institutes and industries associations. Considered as a subclass of Mobile Ad-hoc Networks (MANETs), VANETs are formed by the association of moving nodes (i.e., vehicles) among themselves without any centralized servers [15]. However, due to the high mobility of vehicles, VANETs present some noticeable differences such as rapid changes in network topology, frequent fragmentation, and variable network density [16, 15]. Additionally, VANETs have the potential to provide a wide range of applications, from critical safety services to infotainment applications [17]. In this work, we aim to contribute a study on dissemination of warning messages to the development of emergency warning service under VANETs for alerting nearby vehicles about a dangerous situation.

Currently, broadcast storm and network partition problems are two major challenges in designing a broadcast protocol for VANETs. Broadcast storm problem occurs when a massive dissemination of messages overloads the transmission channel causing many packet collisions, thus reducing the efficiency of

the message delivery. Network partition problem occurs when the number of nodes in the area to participate in disseminating the broadcast messages is not sufficient [18]. Moreover, traffic density is also an important issue since a dense traffic is prone to broadcast storm problem while a sparse one usually encounters the network partition problem. For example, a road may experience high traffic in rush hours but will be sparse at late night. Therefore, it is suggested that an efficient data dissemination scheme should be adaptable to various traffic densities [19].

To achieve a cooperative awareness with nearby nodes (i.e., neighbors) in a typical vehicular networking environment, vehicles exchange their local information using periodic one-hop broadcast messages called beacons. However, on account of the rapid changes in network topology as above-mentioned, the information about neighbors' position is fast outdated. This issue can lead to the failure of some protocols that require a precise information about neighbors [20, 18, 21, 22]. To handle this issue, vehicles usually increase the frequency of sending beacons. This solution can maintain the accuracy of location information but it will consume a significant amount of network bandwidth and cause many collisions, especially in a high vehicle density environment.

In this thesis, we propose a novel adaptive beacon-based data dissemination (ABDDis) scheme to perform data dissemination in VANETs which aims to handle the broadcast storm and network partition problems while maintaining a good coverage and being able to work under both dense and sparse traffics. The ABDDis scheme adopts a novel mechanism to adaptively adjust the beacon interval based on current traffic data to reduce the channel load while being able to cope with changes in the network topology. By combining the broadcast suppression technique, store-carry-forward and adaptive beacon mechanism, ABDDis scheme is expected to distribute the warning message with low overhead and high delivery ratio. In this work, we assume that all vehicles are equipped with sensors and wireless communication devices.

After this chapter, the remainder of this thesis is organized in the following manners:

- Chapter 2 presents some fundamental knowledge and issues about vehicular ad-hoc network (VANET). After that, we give an extensive study on related works about state-of-the-art data dissemination schemes in VANET.
- Chapter 3 describes our proposed ABDDis scheme. First, we provide an overview of the scheme then we will elaborate on each component.

- An investigation of some simulators are illustrated in **??**.
- Chapter 4 presents experiments and evaluation of data dissemination schemes. We illustrate our experimental design and then we will discuss the result of each experiment.
- Finally, chapter 5 gives some concluding remarks and approaches for future works.

2 Literature review

2.1 Vehicular Ad-hoc Networks (VANETs)

VANETs is formed by the association among themselves or with roadside infrastructure to exchange data wireless medium, it is a new research topic in both scientific and industrial community, which is a key approach to intelligent transportation system (ITS), it opens an ability to develop many useful services on vehicles ranging from safety application such as intersection movement assist, emergency braking warning, blind spot detection, etc.. and comfort application such as Internet access for navigation, advertisement, music, etc.

This section aims to give a general knowledge about VANETs, which will provide the understanding for further investigation about VANETs' problems. For the detail, we first introduce the characteristics of VANETs, which is the key points make VANETs different to previous ad-hoc networks. Then the general architecture and its applicable possibility is presented. Finally, the standardized technology for wireless communication in VANETs is described, although there are many wireless technology is proposed as in [23], but in our work, we just concern on WAVE/DSRC (explained later) standard.

2.1.1 Characteristics

VANETs is a sub-domain of MANETs (Mobile Ad-hoc Networks) [6], which are formed by the association of vehicles among themselves without any centralized servers, makes it different in comparison with MANETs as follows:

- High mobility: mobile nodes in VANETs is vehicles, which move very fast

at high speed on the road as well as changes in moving direction, moving roads. It's difficult to predict the vehicles' position.

- Mobility modeling: vehicles in VANETs do not move in random directions like mobile nodes in MANET but in some patterns, on the road with constraints about speed, turning rule,...
- Network connectivity: due to high mobility and road scenario (urban or highway), the vehicle density is not ensured, so it can leads to the disconnected network when the density of vehicles is low, especially on highway scenario.
- Frequent exchange of data: the ad hoc architecture and the high mobility characteristic motivate the nodes in VANETs to exchange their information frequently in order to maintain its connectivity with their neighbors.
- Transmission environment: the communication in VANETs is based on wireless communication standard, so it has a specific communication range as well as can be affected by obstacles on the road such as building, trees,...
- Delay constraint in data dissemination: VANETs plays an important role in ITS safety application, so when an emergency case occurs, it needs to be transmitted to relevant nodes in an acceptable amount of time to warrant the other vehicles can have time for a reaction to that emergency case.
- Generally, vehicular network system components may consist of an On-Board Equipment (OBE) which is a network device located in the moving vehicle and connected to both wireless network and to in-vehicle network, Road-Side Equipment (RSE), which can be described as a device installed in the side-road infrastructure (e.g., Light pole and road signs) that connects the moving vehicle to the access network which in turn is connected to the core network [24].
- Network boundary: VANETs can be deployed in a city, multiple cities, a country, continents,...

2.1.2 Architecture

The exchange of data of vehicles-to-vehicles and vehicles-roadside infrastructures account for most parts of communication in VANETs as well as applications in ITS. Therefore, the communication in VANETs is categorized into two main types: non-infrastructure-based (ad hoc) communications as vehicle-to-vehicle

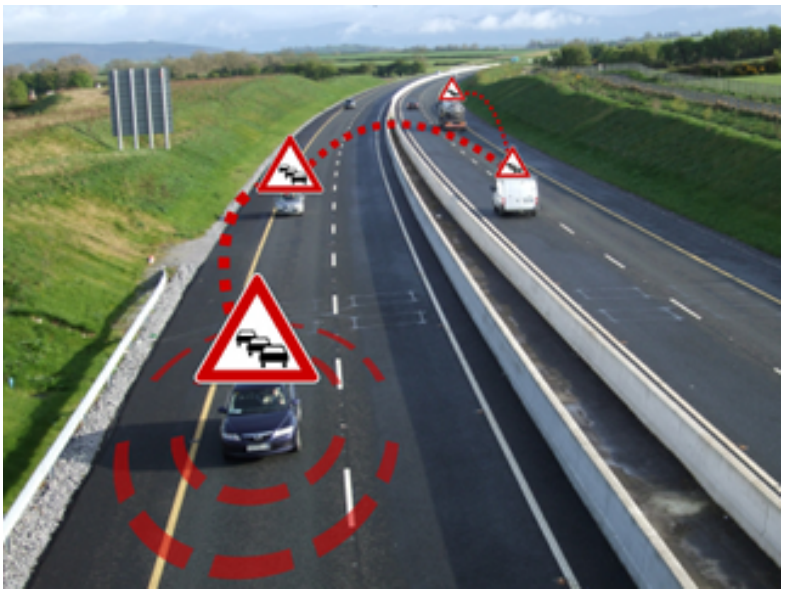


Figure 2.1: Communication in V2V [1]

communications (V2V) and infrastructure-based communications as vehicle-to-infrastructure communications (V2I), which is based on the presence of roadside infrastructures. Furthermore, there is also a term called vehicle-to-everything (V2X) that combines V2V and V2I.

V2V communications

V2V communication is very important in safety applications, it relies on the exchange of data between vehicles to detect if it is necessary to warn the driver about a potential risk. For example, V2V could assist intersection crossing by alerting if there is a car is approaching with high speed. Figure 2.1 shows the exchange of data between vehicle.

V2V communications systems are composed of devices, installed in vehicles, that use Dedicated Short-range Radio Communication (DSRC) to exchange messages containing vehicle information (e.g., vehicle's speed, heading, braking status). V2V devices use this information from other vehicles and determine if a warning to the vehicle's driver is needed, which could prevent a vehicle crash [25].

V2V communication data can be divided into two types: basic message and warning message:

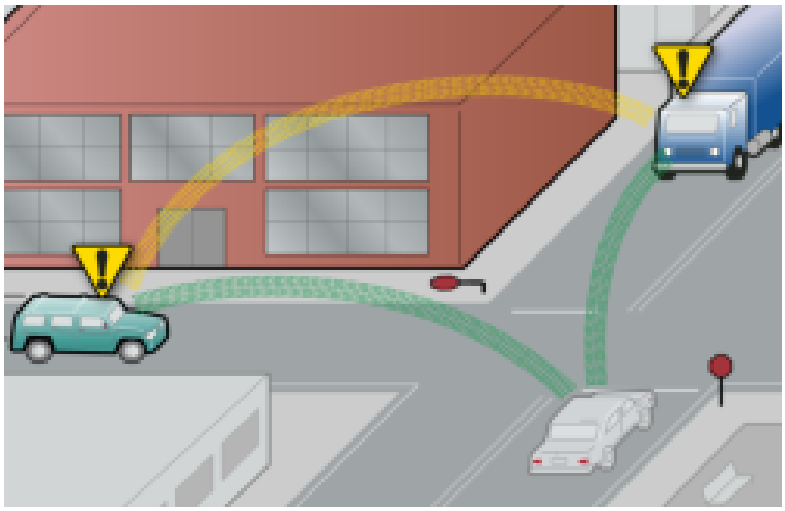


Figure 2.2: Example of the intersection movement assist application [2]

- Basic message: exchanged between vehicles periodically which can be used for maintaining a neighbor list, extracting for detecting abnormal behaviors of other vehicles to dictate collision threats. The exchanged data contains information about speed, heading direction, wheel steering angle, braking status of the vehicle.

- Warning message: this type of message will be broadcast to involved vehicles in case of emergency (accident, car crash, etc).

With V2V communication, some useful safety applications can be deployed to address the limited line-of-sight based on basic message information:

- Intersection crossing and left turn assist: the vehicle can detect it's not safe to pass the intersection (as shown in figure 2.2) or to turn left because of the potential threat for colliding with incoming vehicle(s).

- Urgent braking avoidance (figure 2.3): the vehicle can detect if ahead vehicle is braking, it will warn the driver to slow down the speed.

- Detect blind spot: the vehicle will notify its driver if there is a vehicle is present in its blind spot (figure 2.4), which will be useful for lane changing assist.

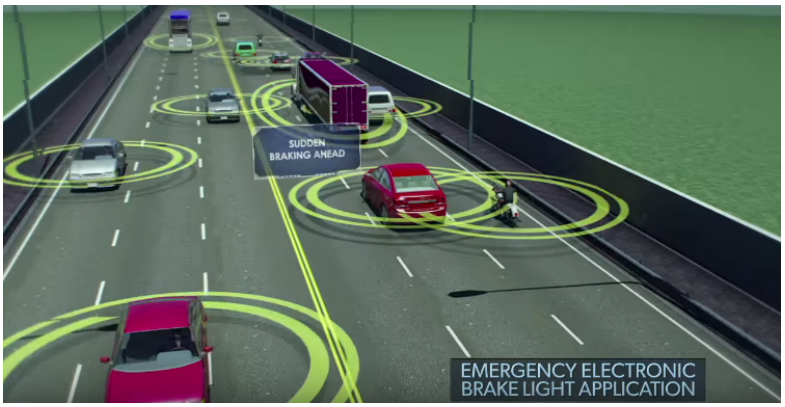


Figure 2.3: Example of the emergency braking avoidance application [3]

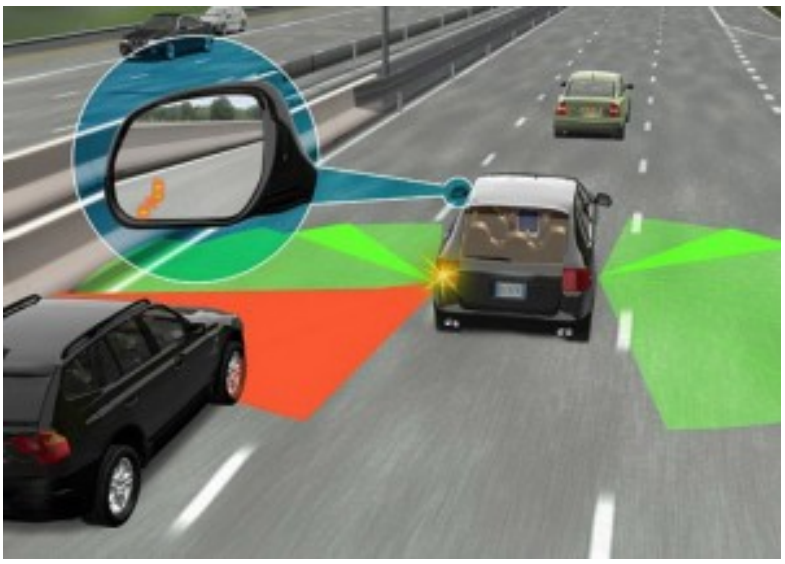


Figure 2.4: Example of the blind spot detection application [4]

2.1.3 V2I communications

VANETs can take advantages of roadside infrastructure. It's based on exchanged data from vehicles to roadside units (like traffic lights) and data between roadside units to support more ITS applications as well as to compensate the disconnected network and obstacles affection to data dissemination between vehicles. Figure 2.5 shows that vehicles receive messages from a roadside unit - traffic light.

To address the disconnected network and obstacles affection in V2V communication, some ideas are proposed:

- Roadside infrastructure can receive basic periodically data from vehicles to detect traffic condition in its area, so it will know if the network is disconnected, then it can hold data to forward later when there is a vehicle entering its area (in the case of emergency message like a car crash).

- Roadside infrastructure like traffic lights is placed at the intersection, so it can help to forward data that is blocked by building at the corner of the street.

The roadside infrastructure (consists of many roadside units) can gather information from many sources:

- Periodic data from vehicles.
- Statistic information from other roadside units in the city, for instance.
- Roadside infrastructure can be connected to the Internet.

Base on information gathered, roadside infrastructure can support a wide range of application:

- Sends warning to vehicles about traffic jams, accidents so they can change their route to avoid sticking in traffic flow.
- Acts as a router to allow vehicles to access the Internet for infotainment, navigation, advertisement applications.
- Another application is based on local position of the roadside units like lane curve speed warning, traffic light warning, slowdown/work-zone warning

Figure 2.5: V2I communications [5]

2.1.4 Technology

Recently, the promises of wireless communications to support vehicular safety applications have led to several research projects around world: the Vehicle Safety Communications Consortium developing the DSRC technology (USA), the Internet ITS Consortium (Japan), the PReVENT project (Europe) or the 'Network on Wheels' project (Germany) are some samples [26].

To cater to the emerging wireless communication needs with regard to vehicles, in July 2003 ASTM and IEEE adopted the Dedicated Short Range Communication (DSRC) standard (ASTM E 2213-03). The aim of this standard is to provide wireless communications capabilities for transportation applications within a 1000 m range at typical highway speeds. It provides seven 10 MHz channels at the 5.9 GHz licensed band for ITS applications, with different channels designated for different applications, including one specifically reserved for vehicle-to-vehicle communications [26].

With DSRC standard for transmission medium in VANETs, the Institute of Electrical and Electronics Engineers (IEEE) 1609 Family of Standards for WAVE (Wireless Access in Vehicular Environment) defines the architecture and standardizes set of services and interfaces that enable secure wireless communication and physical access for high speed (up to 27MB/s), short range (up to 1000m), and low latency wireless communication in the vehicular environment.

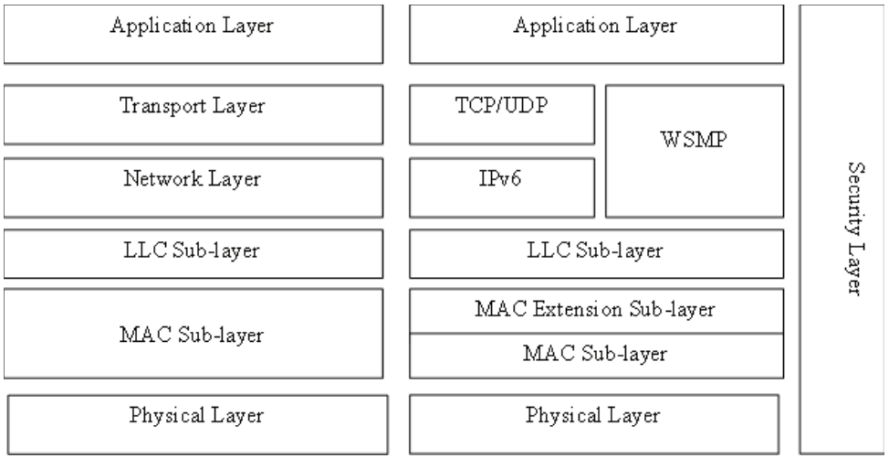


Figure 2.6: Comparison of WAVE protocol stack with TCP protocol stack [6]

Figure 2.6 shows the difference between WAVE protocol stack and TCP protocol stack, WAVE protocol differs from TCP protocol in the transport layer and network layer. The physical layer of WAVE is based on IEEE 802.11p standard and the MAC layer is based on IEEE 1609.4 standard. The IPv6, WSMP stacks is based on IEEE 1609.3, the security stack is based on IEEE 1609.2 standard. The specifications of WAVE protocol stacks are discussed further in next sections.

WAVE protocol physical layer - IEEE 802.11p

IEEE 802.11p is considered for dedicated short-range communications (DSRC). As the communications of vehicle-to-vehicle and vehicle-to-infrastructure only exist for a short time, the IEEE 802.11p is an amendment of IEEE 802.11 to operate in the vehicular network environment. IEEE 802.11p defines the way to exchange data without the need of establish a basic service set (BSS) and wait for association and authentication process before exchanging data. Therefore, IEEE 802.11p can reduce delay in data transmission.

WAVE physical layer has been described by IEEE 802.11p standard, protocol WAVE relies on IEEE 802.11a Orthogonal Frequency Division Multiplexing (OFDM) mechanism to support different data rates that are determined by coding rate and modulation type. The 75 MHz bandwidth in 5.9 GHz spectrum is divided into 7 smaller operation channels each of 10 MHz bandwidth [6].

IEEE 802.11p uses 75MHz bandwidth channel in the 5.9GHz band (5.85-5.92GHz), which is divided into 7 smaller operation channels (10MHz each channel). As shown in figure 2.7, 7 channels is numbered from 172 to 184, the

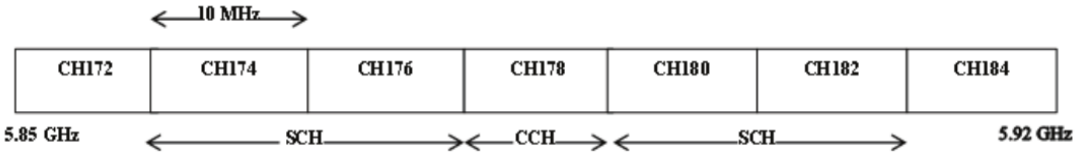


Figure 2.7: IEEE 802.11p operation channels [6]

first and the last channels is reserved for future use. Channels 174, 176, 180, 182 are used as Service Channel (SCH). Channel 178 is used as Control Channel (CCH). The SCH channels are used to carry the data of IPv6 stack, the CCH channel is used for service announcement and WAVE short messages that are used for safety applications.

WAVE protocol MAC layer - IEEE 1609.4

MAC layer is described by IEEE 1609.4 standard, which is based on IEEE 802.11e Enhanced Distributed Channel Access (EDCA) Paradigm and extended to cover two channels operations. As shown in figure 2.8, the WAVE MAC layer model consists of two planes: data plane and management plane. Data plan provides data services such as inbound and outbound high layers data, while the management plane performs the management command such as synchronization and channel access.

Physical Layer Management Entity (PLME) is the management entity that provides management function to the physical layer. IEEE 1609.4 specifies the extension to 1EEE 802.11 MAC sublayer Management Entity (MLME) to provide channel coordination, which is required when there are one or more switching devices with concurrent alternating operation on CCH and SCH, so that data packets are transmitted on the proper channel at the right time. MLME is providing services for both data plane and management plane, data plane services which are performed by MLME include channel coordination, channel routing and user priority while management services are: multi-channel synchronization, channel access, Vendor specific actions frames, Management Information Base (MIB) maintenance and readdressing. Three types of frames are allowed in WAVE: 1- Control frames which are used as per IEEE 802.11 standard. 2- Management frames which are either Time Advertisement frames (TA), used to advertise time synchronization information, or, Vendor specific Action (VSA) frames. Management frames can be transmitted on CCH or SCH. 3- Data frames which might be WSMP data or IP datagram, WSMP frames which

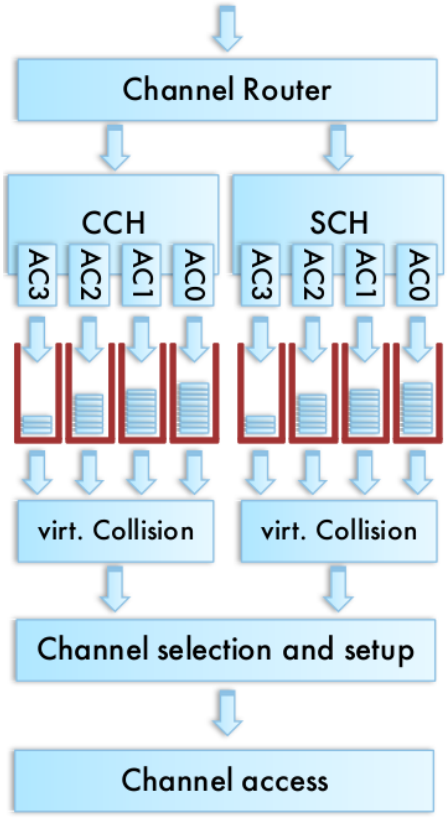


Figure 2.8: Description of WAVE MAC protocol layer [6]

containing WAVE Service Messages (WSM) may get transmitted on CCH or SCH, However, IP datagram can only be transmitted on SCH. Figure 6 describes the internal architecture of Multi-Channel operation MAC, Figure 6 contains both CCH and SCH. Queues are provided to perform access of the prioritized data [6].

When data is received from Logical Link Control (LLC), the channel router checks the ethertype field in the header of the data, if it is WSMP datagram, then it should be routed to the proper queue based on channel identifier and data priority. However, IP datagram transmission is slightly different, all IP data should be registered with the transmitter profile in the MIB which contains SCH number, power level and the adaptable status of power level and data rate. When LLC passes an IPv6 datagram to the channel router, it routes the datagram to the data buffer that corresponds to the specified SCH. Those procedures are allowing the higher layer to control the transmission parameters of the physical layers. Channel selector carries out many functions such as, drops the data when the channel is no longer valid [6].

WAVE protocol network services

The network services are defined by the IEEE 1609.3 Standard protocol and describe data plane and management plane services. Two types of WAVE devices are defined: Provider device which is the device transmitting WAVE Service Advertisement (WSA) to indicate its availability for data exchange on one or more SCHs, provider WAVE device is the sender of WSA and it is the initiator of the service. User device is the device that monitoring for received WSAs, with a possibility of participation in the SCH data exchange, the user device is the receiver of the WSA and joiner of the service. WAVE device may assume one, both or neither device [6].

Data plane services support two protocol stacks, WSMP and IPv6:

- WSMP: WAVE Short Message Protocol (WSMP) is a WAVE network layer unique protocol to support high priority and time sensitive communication. Upon reception of WAVE Short Message (WSM) data unit from upper layers, WSMP generates WSMP header to be included in the received unit, then make packet transmission request to LLC, LLC set the ethertype field value to encapsulate the packet and pass the data to the lower layers. When the LLC receives MAC data unit, it checks the ethertype field value, then delivers it to either IPv6 or WSMP stack [6].
- IPv6: supports for UDP/TCP/IP protocol suite. IP traffic is sent and received through LLC sublayer [6].

Management plane services provide many functions:

- WAVE Management Entity (WME): provides networking management services functions: providing channel assignments, processing service requests for higher layers.
- Management Information Base (MIB): includes channel related information, system configuration information (number of channels supported, registration port, advertiser identifier), system status information (Provider Service Request Table, User Service Request Table, CCH Service Request Table).

Due to the applicability in various type of applications (safety and comfort), we only concentrate on traffic safety applications such as shown in figure 2.4,

figure 2.3, figure 2.2. The key challenge for these safety applications is the reliability when exchanging the information between vehicles in which they spreads warning message in order to involve neighboring vehicles about danger circumstances, this process is called data dissemination and will be presented in next section.

2.2 Simulation Tools

2.2.1 Introduction

VANET composes of the movement of traffics on the road and their communication in wireless environment, with below features:

- The movement of traffics is under strict transportation rules about their behaviors.
- The communication environment is wireless, so the transmission signal can be blocked by obstacles (like buildings).
- Frequent changes of topology due to high mobility of traffics.

To simulate the VANET comprehensively, the simulator must satisfy two types of simulation: mobility and network.

2.2.2 Overview of simulation tools in VANET

This section will introduce about simulation tools that is used widely in research about VANET. Figure 2.9 list some simulation tools and categorizes them into types: VANET simulator (all in one software, software coupling network simulator and mobility simulator to simulate VANET), network simulator, mobility simulator.

To satisfy the requirements of VANET simulation of this topic, a mobility simulator must have some features:

- GUI should be available.
- User-defined simulation map, traffic routing or extract information from available online real map (e.g., OpenStreetMap [27]).
- Ability to simulate two main traffic map scenario: urban and highway.
- Ability to simulate full features of road: lanes, directions on lanes, traffic lights, speed restriction, intersection management,...

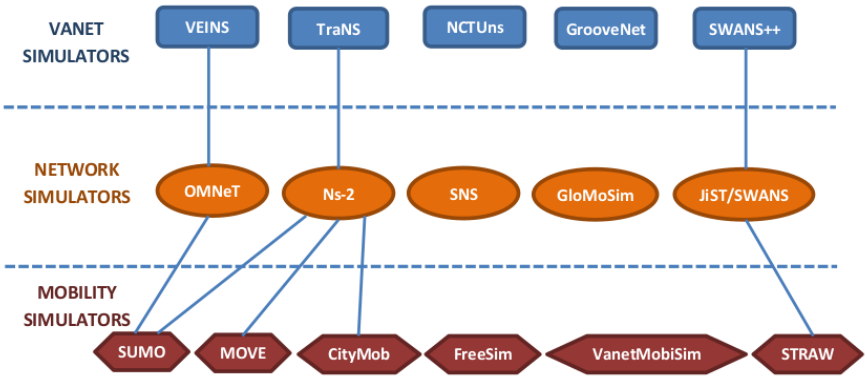


Figure 2.9: Classification of VANET's simulators [7]

- Microscopic simulation: capable of controlling each movement objects in the simulation.
- Multi-model vehicle simulation: personal vehicles (car), community vehicles (bus),...
- Simulation of additional objects on the road: buildings.

Table 2.1 presents a detailed comparison of different traffic simulators.

To satisfy the requirements of VANET simulation of this topic, a network simulator must have some features:

- Large network simulation capacity.
- GUI should be available.
- Simulation of 802.11p communication standard.
- Capable of simulating the influence of obstacles to transmission signal.

Table 2.2 presents a detailed comparison of different network simulators.

There are two types of VANET simulators:

- Integrates mobility simulator and network simulator into one software (all-in-one).
- Plays immediate role that integrates mobility simulator and network simulator by standardization the communication of the two simulator .

Table 2.3 presents a detailed comparison of different simulators that can be used in VANET.

After investigating on VANET simulators, the Veins simulator [28] is chosen

Table 2.1: A comparison of traffic simulators [11]

	VanetMobiSim	SUMO	MOVE	STRAW	FreeSim	CityMob
Software						
Portability	✓	✓	✓	✓	✓	✓
Freeware	✓	✓	✓	✓	✓	✓
Opensource	✓	✓	✓	✓	✓	✓
Console	✗	✓	✓	✓	✗	✓
GUI	✓	✓	✓	✓	✓	✓
Available examples	✓	✓	✓	✓	✓	✗
Continuous development	✗	✓	✗	✗	✓	✓
Ease of setup	Moderate	Moderate	Easy	Moderate	Easy	Easy
Ease of use	Moderate	Hard	Moderate	Moderate	Easy	Easy
Maps						
Real	✓	✓	✓	✓	✓	✗
User defined	✓	✓	✓	✓	✗	✗
Random	✓	✓	✓	✓	✓	✓
Mobility						
Manhattan	✗	✓	✓	✗	✗	✓
Urban	✗	✓	✗	✗	✗	✓
Highway	✓	✓	✓	✓	✓	✓
Traffic models	✓	✓	✓	✓	✓	✓
Macroscopic	✗	✗	✗	✗	✓	✗
Microscopic	✓	✓	✓	✓	✓	✓
Multi-lane road	✓	✓	✓	✓	✓	✓
Lane changing	✓	✓	✓	✓	✓	✓
Separate directional flows	✓	✓	✓	✓	✓	✓
Speed constraints	✓	✓	✓	✓	✓	✓
Traffic signs	✓	✓	✓	✓	✓	✓
Intersection management	✓	✓	✓	✓	✓	✗
Overtaking criteria	✓	✓	✓	✓	✓	✗
Large road networks	✓	✓	✓	✓	✓	✓
Collision free movement	✓	✓	✓	✓	✓	✓
Different vehicle types	✗	✓	✓	✓	✗	✓
Route calculation	✓	✓	✓	✓	✓	✗
Traces						
NS-2	✓	✗	✓	✗	✗	✓
GloMoSim	✓	✗	✓	✗	✗	✗
SWANS	✗	✗	✗	✓	✗	✗
Import difference formats	✓	✓	✓	✓	✓	✓
OMNet++	✓	✓	✓	✗	✗	✗

Table 2.2: A comparison of network simulators [11]

	NS-2	GloMoSim	JiST/SWANS	SNS	OMNET++
Software					
Portability	✓	✓	✓	✓	✓
Freeware	✓	✓	✓	✓	✓
Opensource	✓	✓	✓	✓	✓
Available examples	✓	✓	✓	✓	✓
Continuous development	✓	✗	✓	✗	✓
Large networks	✗	✓	✓	✓	✓
Console	✓	✓	✓	✓	✓
GUI	✓	✓	✓	✓	✓
Scalability	Poor	High	High	High	High
Ease of setup	Easy	Moderate	Hard	Easy	Moderate
Ease of use	Hard	Hard	Hard	Hard	Hard
VANET					
802.11p	✓	✗	✗	✗	✓
Obstacles	✗	✗	✗	✗	✓

since it satisfies all the requirements mentioned above together with some great features:

- Allows to change traffic's route and behaviors when simulation is running in reaction to data from network simulator.
- Support wireless communication standard for VANET: IEEE802.11p, IEEE1609.4 DSRC/WAVE.
- Allows to import real map to simulate road traffic (e.g., OpenStreetMap [27]).
- Large network simulation capacity (city).

2.2.3 VEINS - VANET simulator

Introduction

VEINS is the combination of two simulation tools, OMNET++ to simulate network, SUMO to simulate road traffic. Two simulator is running in parallel and communicate with each other via TraCI (Traffic Control Interface) protocol, allows two simulator to bi-directionally communicate with each other. The movement of traffics in SUMO also reflects into movement of nodes in OMNET++. Nodes can interact with traffics in SUMO to simulate the influence of IVC application on movement of traffic on the road.

Table 2.3: A comparison of VANET simulators [11]

	VEINS	TraNS	GrooveNet	NCTUns	SWANS++
Mobility generator	SUMO	SUMO	GrooveNet	NCTUns	STRAW
Network generator	OMNET++	NS-2	GrooveNet	NCTUns	SWANS
Simulation type	Microscopic, space-continuous and time-discrete				
Lane models	Multi-lane streets with lane changing				
Road topology	Any	Any	Any	User-defined	Any
Traffic flow model	Car following				
traffic lights	Manual	Manual	Manual	Auto-generated	Manual
Trip model	Djikstra, user-defined	Random, user-defined	Djikstra, sightseeing	User-defined	User-defined
Free-ware	✓	✓	✓	✓	✓
Open-source	✓	✓	✓	✓	✓
Ease of use	Hard	Moderate	Hard	Hard	Hard

OMNET++ - Network simulator

OMNET++ is an object-oriented modular discrete event network simulation framework. It is not a simulator but provides infrastructure and tools for writing simulations.

OMNET++ can be used in various problem domains:

- Modeling of wired and wireless communication networks.
- Protocol modeling.
- Modeling of queuing networks.
- Modeling of multiprocessor and other distributed hardware systems.
- Validating of hardware architectures.
- Evaluating performance aspects of complex software systems.
- In general, modeling and simulation of any system where the discrete event approach is suitable, and can be conveniently mapped into entities communicating by exchanging messages.

Modeling concepts An OMNET++ model consists of modules that communicate with message passing. The active modules are termed simple module, they are written in C++, using the simulation class library. Simple modules can be grouped into compound modules and so forth; the number of hierarchy levels is unlimited. The whole model, called network in OMNeT++, is itself a compound module. Messages can be sent either via connections that span modules or directly to other modules [8]. Figure 2.10 represents compound module, simple

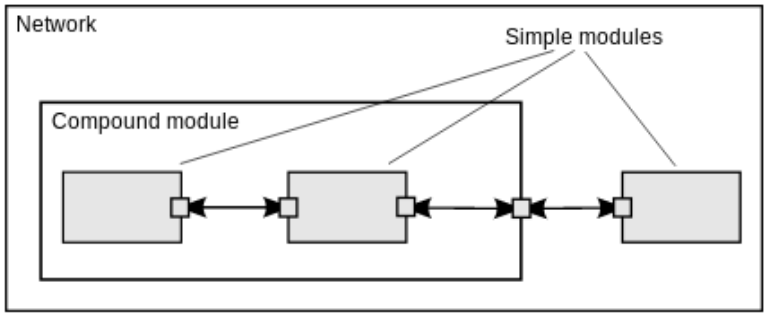


Figure 2.10: Simple and compound modules [8]

modules and their connections.

OMNeT++ provides efficient tools for the user to describe the structure of the actual system. Some of the main features are the following:

- Hierarchical modules:
  - An OMNET++ model consists of hierarchically nested modules that communicate by passing messages to each other.
  - OMNET++ model as top level module - system module, containing submodules, which can contain submodules themselves.
  - The depth of nesting modules unlimited, allowing the user to reflect the logical structure of the actual system in the model structure.
  - Model structure is described in OMNeT++'s NED language (Network description language).
  - Modules that contain submodules called compound modules.
  - Modules at the lowest level (contain no submodules) called simple modules, which contain algorithms of the model, implemented in C++, using OMNET++ simulation library.
- Modules are instances of module types.
- Modules communicate with messages through channels.
- Flexible module parameters.
- Topology description language: The user defines the structure of the model using NED.

SUMO - Traffic simulator

SUMO is a free and open traffic simulation suite. SUMO allows modelling of multi-modal traffic systems including road vehicles, public transport and pedes-

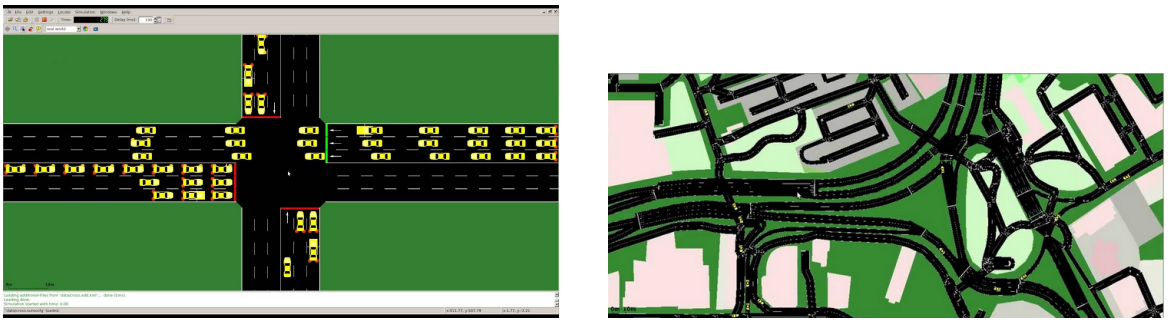


Figure 2.11: SUMO simulator

trians. Included with SUMO is a wealth of supporting tools which handle tasks such as route finding, visualization, network import and emission calculation. SUMO can be enhanced with custom models and provides various APIs to remotely control the simulation. So, SUMO is used by the V2X community for both, providing realistic vehicle traces, and for evaluating applications in an on-line loop with a network simulator.

More about some features of SUMO:

- Microscopic simulation - vehicles, pedestrians and public transport are modeled explicitly.
- Simulation of multimodal traffic, e.g., vehicles, public transport and pedestrians.
- Time schedules of traffic lights can be imported or generated automatically by SUMO.
- Fully simulate lane model: multi-lanes, bi-directional lane, lane changing model.
- No artificial limitations in network size and number of simulated vehicles.
- Supported import formats: OpenStreetMap, VISUM, VISSIM.
- Allow external program to communicate with the running simulation via TCP connection (standardized as TraCI protocol).

Basic concepts about simulating road traffic:

- In traffic simulation, there are four types of simulation: macroscopic, microscopic, sub-microscopic, mesoscopic.
- Macroscopic: models traffic flows as the basic object.
- Microscopic: models the movement of every single vehicle, the behavior of vehicle depends on vehicle's physical ability and driver's controlling.

- Sub-microscopic: similar to microscopic, but it divides each vehicle into sub-structures like engine's speed, engine's rotation,...
- Mesoscopic: also models every single vehicle, unlike microscopic, its data is not updated continuously, but at specific time.
- SUMO uses the microscopic model to simulate its traffic on the road.

Mobility model - Car following Krauss model Car following is how vehicles can avoid collision between each other during the simulation. The idea is controlling the velocity of vehicle below a safe velocity (v_{safe}), so the gap (g) between the following vehicle and the leading vehicle is larger than a desired gap (g_{des}).

The safe velocity is computed using the below formula [29]:

$$v_{safe} = v_l + \frac{g - v_l\tau}{\frac{\overline{v}}{b(\overline{v})} + \tau} \tag{2.1}$$

Where:
v_f is velocity of the following vehicle
v_l is velocity of the leading vehicle
g is the gap between the following vehicle and the leading vehicle.
$\overline{v}$ is mean value of the leading and the following's velocity.
$b(\overline{v})$ is deceleration function.

The desired velocity is the minimum value of three value: the vehicle's maximum velocity (physical feature), the vehicle speed plus maximum acceleration and the above computed safe velocity [29]:

$$v_{des} = min(v_{safe}, v + a, v_{max}) \tag{2.2}$$

Traffic Control Interface - TraCI

SUMO allows external program (network simulator - OMNET++ in this case) to communicate with the running simulation via TCP connection for retrieving or manipulating object's values: vehicles' values (speed, type, id), routes' value,

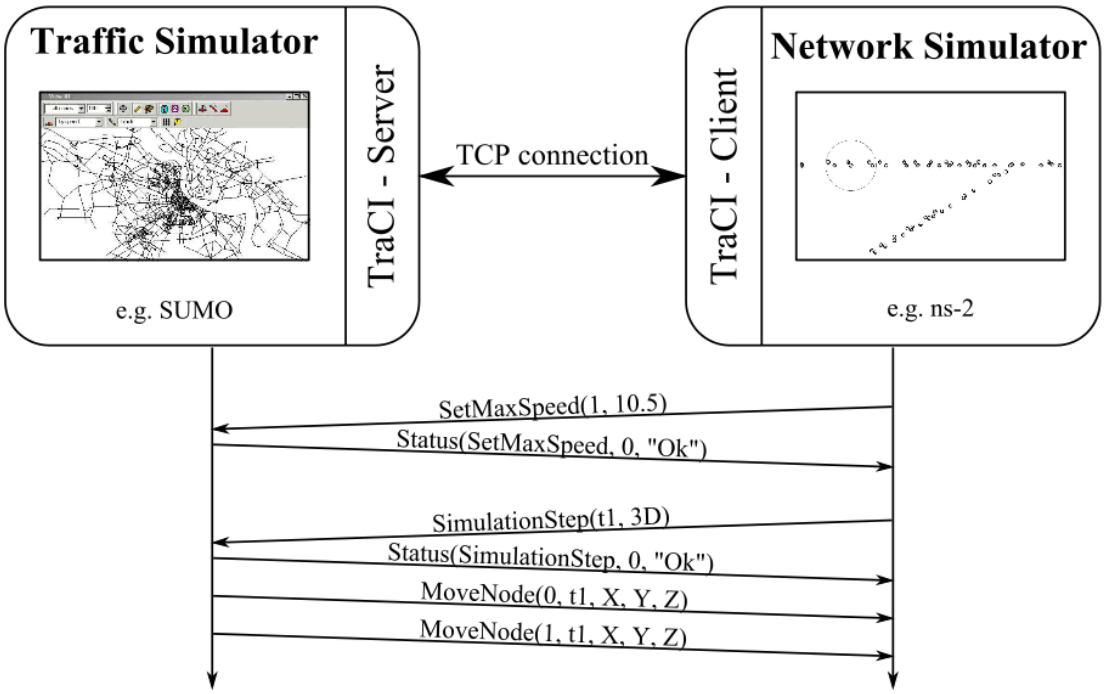


Figure 2.12: Example about communication between traffic simulator and network simulator via TraCI [9]

lanes' value,...

The communication between network simulator and traffic simulator is bidirectional, which is standardized as Traffic Control Interface (TraCI). TraCI is based on architecture of TCP client-server, in which OMNET++ acts as client, SUMO acts as server.

Using TraCI protocol, SUMO provides mobility trace for the network simulation and the network simulation can influence to the traffic simulation by sending command back. Figure 2.12 shows the example about how the traffic simulator and network simulator can interact with each other, the network simulator sends command *set speed value* to the traffic simulator and get the feedback from it.

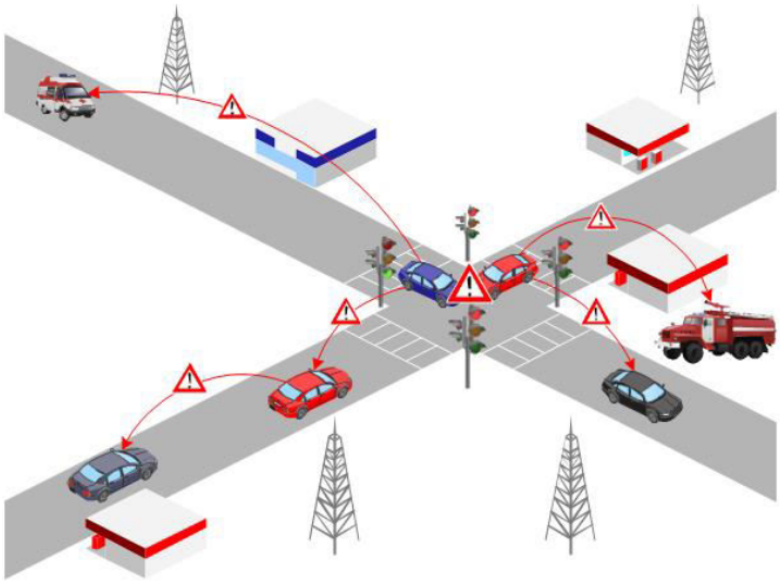


Figure 2.13: Data dissemination model

2.3 Data Dissemination Schemes

2.3.1 Introduction

In the category of safety application, data dissemination usually refers to the process of spreading a warning message over a distributed wireless network [7]. Whenever an accident occurs, the vehicles will send warning messages to their neighbors immediately, and these messages will also be forwarded by receiving vehicles in order to:

- Prevent additional risks.
- Optimize rescue process.

Thus, most of data dissemination schemes share these main goals:

- Reduce message delivery latency.
- Ensure the correct reception of warning messages.

Figure 2.13 illustrates a concept of data dissemination in VANET.

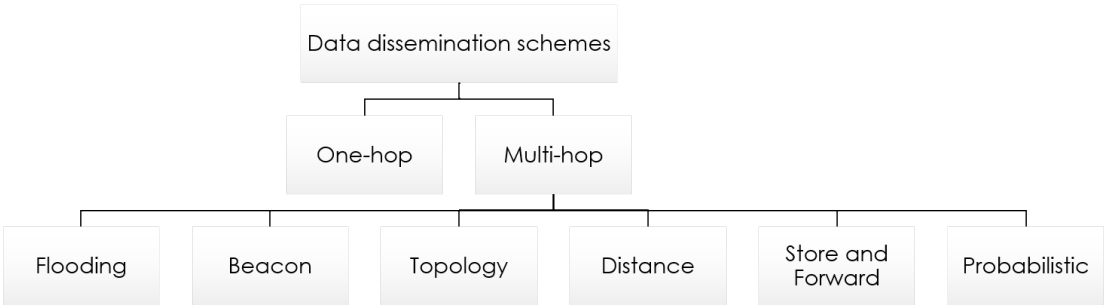


Figure 2.14: Taxonomy of data dissemination schemes

This technique requires the broadcast capability at the data link layer in which a frame can be transmitted to all the vehicles within the radio scope. It usually leads to the broadcast storm problem if the dissemination is not well-handled. Therefore, an efficient scheme should calculate to select most appropriate vehicle to forward the message, thus maximizing the number of vehicles informed, while reducing the latency and channel load [19].

In rest of this chapter, we provide a taxonomy of data dissemination schemes. After that, we will briefly discuss some well-known multi-hop data dissemination schemes.

2.3.2 Taxonomy of schemes

Existing schemes can be classified into one-hop or multi-hop schemes depending on whether or not forwarding is allowed [19]. Figure 2.14 presents an overview of the taxonomy.

One-hop One-hop dissemination messages are those periodically exchanged between vehicles and are not forwarded. As above-mentioned in section 2.1.4, the DSRC standard divides the band into seven channels of 10 MHz bandwidth. Consequently, one-hop messages are usually generated at the rate of 10 Hz to provide information about traffic condition [19]. An one-hop dissemination message is also referred as a *beacon*. Some noticeable works are listed below:

- Xu et al. [30] proposed a DSRC-based QoS (Quality-of-Service) safety messaging. This mechanism focuses on ensuring high reception ratio of messages within the transmission range. Still, in order to achieve that,

packets must be resent multiple times within their lifetime.

- Torrent-Moreno et al. [31] proposed a mechanism to manage the channel load in scenarios with high traffic density. However, this scheme was evaluated in a very simple straight road which resulted in high performance.

In general, these schemes are mostly used to provide local information, hence requiring more effort to adapt in safety applications covering a wide area. This may result in the increasing of overhead which will affect the transmission of messages.

Multi-hop In vehicular network, a multi-hop data dissemination scheme usually works in the following way:
When a vehicle gets into an accident or detects one, it instantly broadcasts warning messages to nearby vehicles. These messages will also be rebroadcasted by receiving vehicles to notify others about the situation, thereby preventing additional risks. Normally, the broadcasting process will be stopped once the message reaches a defined Region of Interest (ROI) [18].

The common issue with multi-hop broadcast is that the massive dissemination of messages easily leads to the broadcast storm problem if we do not have a mechanism to prevent it. According to [19], most schemes mitigate the broadcast storm by reducing the number of retransmitting nodes basing on different parameters, thus minimize the channel load as well as redundancy and contention. The authors also propose a classification of multi-hop data dissemination schemes with respect to their unique characteristics and techniques they use to select the appropriate node to forward a message. In particular, they introduce six categories:

- *Flooding-based schemes.* This is a very straight-forward approach to perform broadcasting, it works by making nodes rebroadcast every received message.

- *Beacon-based schemes. Beacons* are periodic messages which usually contain information about position, speed, condition, ... Beacons normally have a lower priority than alert messages. However, those information could be used so that vehicles can have knowledge about theirs surrounding area (i.e., neighbors), thereby taking decisions.

- *Topology-based schemes.* Topology plays a very important role in a vehicular network. It influences the mean distance between vehicles and the presence of obstacles. The information about the topology is very useful to optimize the propagation performance. For example, Sanguesa et al. [32] proposed the Nearest Junction Located (NJL) scheme in which the only vehicles allowed to forward warning messages are those located nearest to the junction.

- *Distance-based schemes.* This approach uses the relative distance between the sender and the receiver to determine whether a vehicle should rebroadcast. The sender's position can be simply obtained from the Global Positioning System (GPS) and it is then included in the message so that the receiver can calculate the distance. In some cases, the distance can also be determined using the received signal strength at the receiver [10]. Generally, the farther the received vehicle the more likely it forwards the message.

- *Store and Forward schemes.* The policy works as follows: when a vehicle receives a warning message, it stores the message and then waits for the appropriate condition to rebroadcast. Usually, a node forward the message once it recognizes a new neighbor has entered its area. Schemes in this category generally suit to sparse environments where a vehicle's neighbor list is infrequently changed.

- *Probabilistic-based schemes.* In this category, messages are broadcasted with a given probability. This method effectively reduces the redundant retransmissions, however, the message will reach a lower number of destinations because it depends on the node that randomly decide to forward the packet [33]. When the probability is 100 percent, schemes in this approach are obviously equal to flooding. The probability can be predefined and dynamically calculated at the receiver.

2.3.3 Multi-hop data dissemination schemes

In this section, four schemes are introduced. Three of them were originally proposed by Wisitpongphan et al. [10], however, they were evaluated in a very simple scenario.

Flooding

This is the simplest scheme, also called as *1-persistence*. The first time a vehicle receives a warning message, it retransmits the message immediately. The complexity of this scheme is $O(N^2)$ with N is the number of vehicles, therefore it will not work well in a large network. As a consequence, this issue clarifies the above-mentioned idea that an efficient scheme should calculate to select most appropriate vehicle to forward the message.

Weighted p-persistence

This is a scheme proposed by Wisitpongphan et al. [10], they describe the mechanism as follows:

> Upon receiving a packet from node i, node j checks the packet ID and rebroadcasts with probability p_{ij} if it receives the packet for the first time; otherwise, it discards the packet.

Denoting D_{ij} as the relative distance between node i and j, R as the average transmission range, the forwarding probability p_{ij} can be calculated using equation (2.3).

$$p_{ij} = \frac{D_{ij}}{R} \tag{2.3}$$

Note that the receiving node must wait for WAIT_TIME (e.g., 5 ms) before calculating the forwarding probability so that it can receive the same packets from multiple sources, then it should select the smallest p_{ij} among them as its final forwarding probability. That is, each vehicle should use the distance to the nearest sender, therefore, the nodes that are farther away rebroadcast with higher probability. This scheme is illustrated in figure 2.15.

If node j decides *not to* forward, it will keep the message for an additional WAIT_TIME + δ ms, where δ is considered as the propagation delay. If node j does not receive any retransmission packet after waiting (other nodes may decide not to forward as well), it will rebroadcast the packet to prevent message die out and ensure 100 percent reachability.

With those characteristics, this is a distance-and-probabilistic-based scheme since it relies on the relative distance between the sender and receiver, moreover, messages are rebroadcasted with a probability.

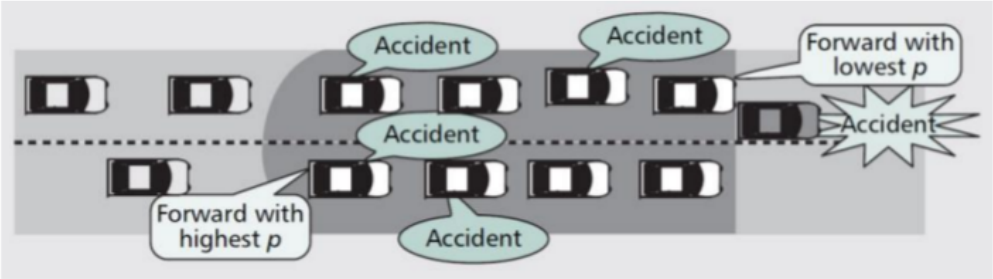


Figure 2.15: Weighted p-persistence [10]

Slotted 1-persistence

This scheme is also proposed in [10], it is described as follows:

> Upon receiving a packet, a node checks the packet ID and rebroadcasts with probability 1 at the assigned time slot $T_{S_{ij}}$ if it receives the packet for the first time and has not received any duplicates before its assigned time slot; otherwise, it discards the packet.

Denoting D_{ij} and R the same as the above Weighted p-persistence, N_s as the predetermined number of slots, the assigned slot number S_{ij} can be expressed as equation (2.4).

$$S_{ij} = \left\lfloor N_s \times \left(1 - \frac{min(D_{ij}, R)}{R}\right) \right\rfloor \tag{2.4}$$

With the obtained S_{ij}, the assigned time slot $T_{S_{ij}}$ can be easily calculated using equation (2.5) where τ is denoted as the one-hop delay, which includes the medium access delay and propagation delay.

$$T_{S_{ij}} = S_{ij} \times \tau \tag{2.5}$$

This approach shares the same logic as the previous Weighted p-persistence, but instead of calculating the forwarding probability, the vehicle uses the source's position information to determine the waiting time until retransmitting. Figure 2.16 best describes this scheme. As can be seen, the transmission range is divided into four slots, the farther the node, the shorter the waiting time. Furthermore, like Weighted p-persistence scheme, if a node receives duplicate packets in its WAIT_TIME, it will take on the smallest D_{ij} value.

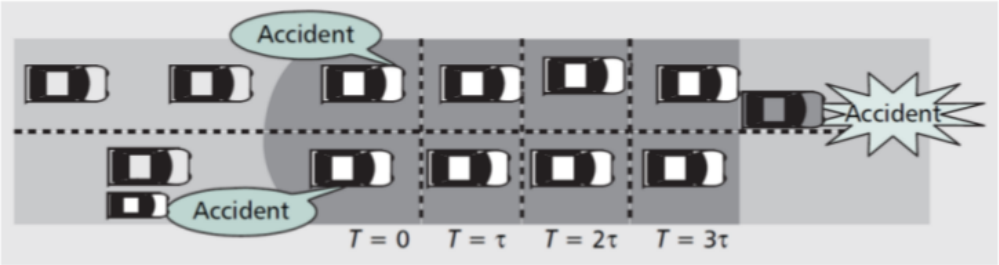


Figure 2.16: Slotted 1-persistence [10]

The value of N_s must be carefully chosen. In [10], they suggested that this value should theoretically be a function of the traffic density, that is, the denser the traffic, the larger the number of slots. However, it is not easy for a vehicle to detect the current traffic density, hence, N_s should be fixed or adaptively change over time. For example, there should be five slots during the rush hour and three slots for non-rush.

With the above characteristic, this scheme is considered as a distance-based scheme since the decision is made based on the distance between the sender and the receiver. Additionally, if topology information is used to get knowledge of current traffic density in order to dynamically change the value of N_s, it is also a topology-based scheme.

Slotted p-persistence

Wisitpongphan et al. [10] described this scheme using the following rule:

> Upon receiving a packet, a node checks the packet ID and rebroadcasts with the predetermined probability p at the assigned time slot $T_{S_{ij}}$, as expressed by equation (2.5), if it receives the packet for the first time and has not received any duplicates before its assigned time slot; otherwise, it discards the packet.

This scheme works like a combination of Weighted p-persistence and Slotted 1-persistence. Note that the probability here is predefined so there may be Slotted 0.5-persistence, 0.7-persistence, 0.9-persistence, ... Just like Weighted p-persistence, there is a chance that all vehicles decide *not to* rebroadcast. Therefore, each node in this scheme buffers the message for a period of time (e.g., $(N_s - 1) \times WAIT\_TIME + \delta\ ms$) and retransmit after waiting if it does not receive any retransmission packets. Figure 2.17 presents the concept of this scheme with

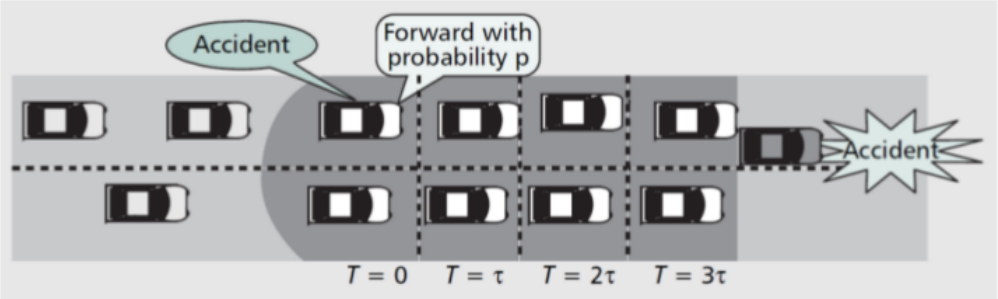


Figure 2.17: Slotted p-persistence [10]

four slots.

With those characteristics, this is considered as a distance-and-probabilistic-based scheme because the decision is given based on the distance and the predefined probability.

2.3.4 Other schemes

Apart from the "primitive" schemes above, there are some other noticeable works below:

- *Distributed Vehicular Broadcast (DV-CAST)*. The schemes mentioned above does not perform well in a disconnected network, Tonguz et al. [18] proposed a way to overcome this problem while alleviating the broadcast storm and without increasing the network overhead much. Specifically, DV-CAST uses information about neighbors to detect the current status of the network and select the best dissemination process. This scheme bases its mechanism on beacon, distance and store-and-forward.

- *Junction Store and Forward (JSF)*. Sanguesa et al. [34] proposed this scheme to take the advantage of topology information. Specifically, JSF suggests that a vehicle should wait to be near a crossroad to rebroadcast messages. Unlike other schemes that retransmit as soon as possible, JSF keeps the messages until it finds a better communication environment. This is a topology-based scheme.

- *Neighbor Store and Forward (NSF)*. In [20], the authors proposed a scheme that is similar to JSF. However, NSF does not need any information about the topology, instead it relies on the neighbor's list. A vehicle keep track of its neighbors, whenever a new neighbor enters the area, the neighbor's list

will be updated. At that time, the vehicle will rebroadcast the message. It can be simply stated that vehicles using NSF will keep the message until they recognize a new neighbor. This is a beacon-based scheme.

3 Adaptive Beacon-based Data Dissemination (ABDDis)

In the ABDDis scheme, every vehicle uses beacons to exchange its local information that is used to perceive the surrounding area. To effectively solve the broadcast storm and network partition problems, the proposed ABDDis scheme provides some features:

- Rely only on local one-hop neighbor information.
- Be able to operate with various traffic densities even if network partition problems occur.
- Adaptively adjust the beacon interval to reduce unnecessary and redundant beacons while keeping accurate neighbor information.

A novel mechanism for message dissemination in the proposed scheme is described by the state machine shown in Figure 3.1. Vehicles receiving a new warning message will adopt the slotted 1-persistence [10] scheme to decide whether they should rebroadcast the message or not. The slotted 1-persistence scheme serves as a core broadcast suppression technique which chooses the forwarding nodes based on the distance between the sender and receiver. If vehicles decide to rebroadcast, they will broadcast the message to their appropriate neighbors. Then, they examine if they should resort to the Store-carry-forward (SCF) mechanism or not. In the following sections, we will describe the two major components in the ABDDis scheme that are (1) the novel adaptive beacon mechanism and (2) the novel store-carry-forward (SCF) mechanism.

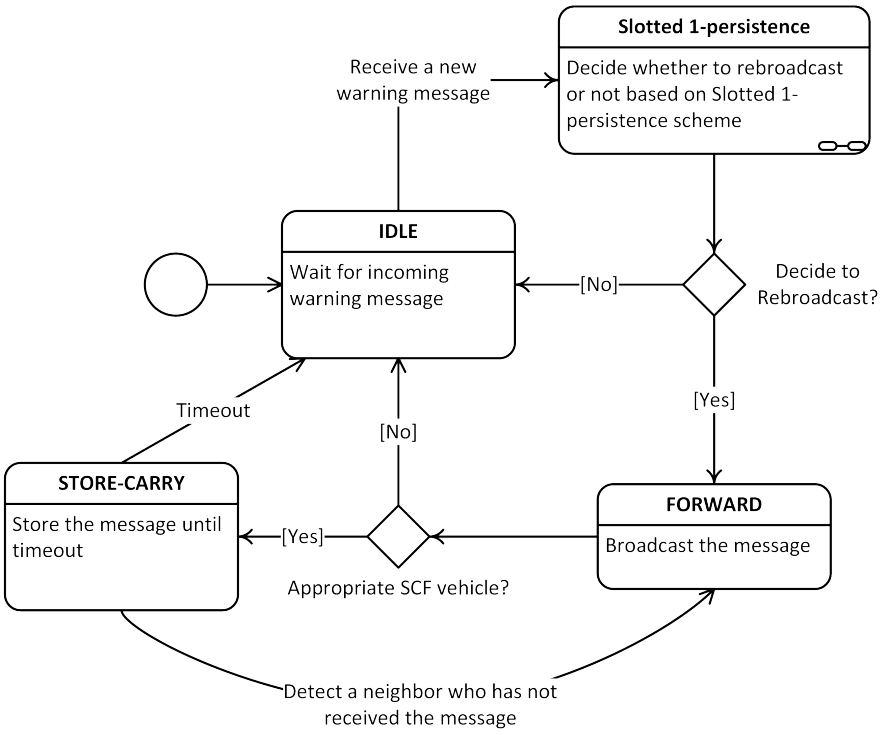


Figure 3.1: Novel mechanism for data dissemination in the ABDDis scheme

3.1 Adaptive beacon mechanism

In the ABDDis scheme, the information exchanged between vehicles includes position, speed, heading direction, and hash of the most recent received warning message. The position is obtained from Global Positioning System (GPS) [35]. The heading direction is an angle whose value is in $[-\pi, \pi)$ and 0 represents heading toward East. Figure 3.2 gives a clearer view in which the heading direction value of the left car is $\frac{\pi}{4}$ and the other is $-\frac{3\pi}{4}$

It can be seen that ABDDis relies heavily on neighbor information to operate, thus, that information must be highly accurate. Simply increasing the frequency of broadcasting beacons helps maintain the accuracy of neighbor information, however, it also exacerbates broadcast storm problems. In this work, we propose a strategy for adaptively adjusting the beacon interval based on the speed of neighboring vehicles.

For example, in dense traffic, vehicles often move in a stable cluster, with few differences among vehicle speeds. In this case, the topology does not change rapidly. Therefore, exchanging beacons at a high frequency is not efficient. By observing this circumstance, a vehicle can obtain the variance of neighbors'

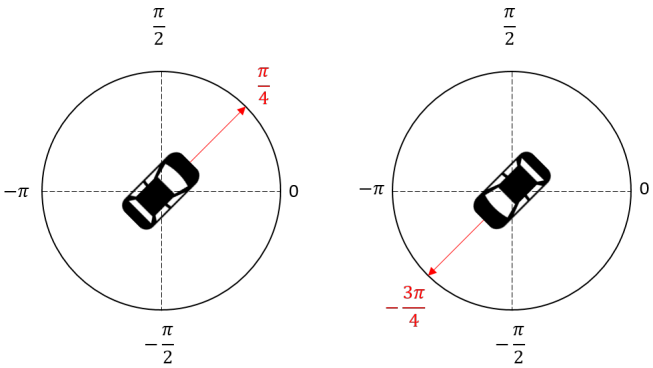


Figure 3.2: Heading direction of a vehicle

speed to determine an appropriate beacon interval. In the following sections, we present two approaches toward this mechanism.

3.1.1 A simple approach

A vehicle can use the ratio between its speed and the average speed of its neighbors to determine the appropriate beacon interval. That ratio is calculated using the following equation:

$$ratio = \begin{cases} \frac{v}{\overline{v_{nb}}} & \text{if } v > \overline{v_{nb}} \\ \frac{\overline{v_{nb}}}{v} & \text{otherwise} \end{cases} \tag{3.1}$$

where:

v is the current speed of the vehicle.

$\overline{v_{nb}}$ is the average speed of its neighbors.

As shown in Equation (3.1), the *ratio* is in $[1, \infty)$ (it equals to ∞ when the denominator is zero). A high *ratio* implies that the topology is more likely to change, thus, the beacon interval should be small so that vehicles can cope with the change.

As a result, the beacon interval can be obtained as follows:

$$interval = 1 + \frac{max.Interval - 1}{ratio} \quad \text{(second)} \tag{3.2}$$

where $max.Interval$ is the preset maximum beacon interval (e.g., 3 seconds). As can be seen from Equation (3.2), the interval value is in $[1, max.Interval]$ and it increases as the $ratio$ decreases.

3.1.2 A statistical approach

Abuelenin et al. [36] proved that the vehicle's speed follows the Gaussian distribution with the below probability density function:

$$f(v) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(v-\mu)^2}{2\sigma^2}} \tag{3.3}$$

where v is the vehicle's speed. By collecting the speeds of neighbors, a node can calculate the coefficient of variation (CV) among the collected speeds using the following equation:

$$CV = \frac{\sigma}{\mu} \tag{3.4}$$

A high value of CV denotes that the difference between vehicle speeds in a cluster is high. That is, the topology changes more frequently. To manage these changes, the beacon interval should be small. Likewise, a low value of CV implies a stable cluster, where the beacon interval should be high to reduce channel load.

As a result, the beacon interval T can be obtained as follows:

$$T = \begin{cases} T_{max} - (T_{max} - 1) \times CV & \text{if } CV < 1 \\ 1 & \text{otherwise} \end{cases} \tag{3.5}$$

where T_{max} is the preset maximum beacon interval (e.g., 3 seconds). As can be seen from Equation (3.5), the interval value is in $[1, T_{max}]$ and it increases as the CV decreases.

Furthermore, we adopt a prediction model for improving the neighbor information accuracy. Boukerche et al. [37] suggested that a vehicle can use the information in the last received beacon from a neighbor to predict its current position. In this work, we propose a prediction model that accounts for the position, speed and heading direction of a vehicle. First, the traveled distance D

since the last beacon is estimated by Equation (3.6).

$$D = v \times (T_1 - T_0) \tag{3.6}$$

where T_0 is denoted as the last beacon's timestamp and T_1 is the current time. Then, the current position $P(x, y)$ can be predicted by Equation (3.7).

$$\begin{aligned} x_P &= x_A + D \times \cos(\alpha) \\ y_P &= y_A + D \times \sin(\alpha) \end{aligned} \tag{3.7}$$

where $A(x, y)$ and α are the position and heading direction retrieved from the last beacon.

In the beaconing mechanism, a node only obtains the position from GPS only when it broadcasts a new beacon. Whenever a node acquires its current position, it will calculate its own $P(x, y)$ based on the last sent beacon and use it instead of retrieving from GPS again. This will help every node in a group operate on the same neighbor information.

3.2 Store-carry-forward (SCF)

In the ABDDis scheme, a novel SCF mechanism will be adopted to handle the network partition problem. An SCF vehicle will store and forward a message whenever it detects a new neighbor that has not previously received that message. To determine whether a neighbor has previously received a message, it compares the hash retrieved from the received beacon packet with the hash of the storing message. Each SCF vehicle typically stores the message in a predefined interval (e.g., 2 minutes), after that, it discards the message.

In order to overcome the network partition problem, vehicles that reside at the edge of a cluster should resort to the SCF mechanism. In other words, a vehicle is chosen as an SCF vehicle if it does not stand between the source and any neighbors relative to its heading direction. Figure 3.3 illustrates the strategy for assigning SCF vehicles based on the SCF mechanism. The purple node is the *source* vehicle (S) that initiates the warning message, the red nodes are assigned as SCF vehicles.

Vehicles deciding to rebroadcast will then use the local neighbor information to determine which vehicles locate at the edge. To compare their relative positions, in the SCF scheme, we use coordinate rotation to normalize coordinates

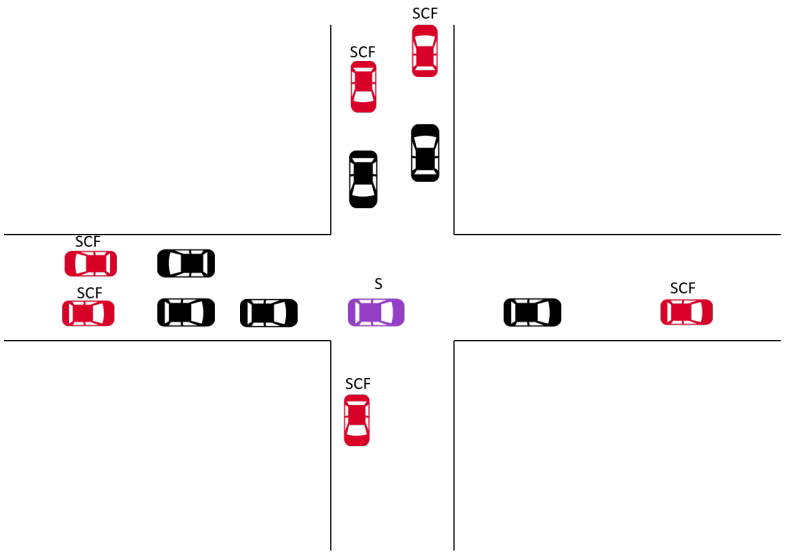


Figure 3.3: SCF vehicles

with respect to the heading direction. Denoting S as the source vehicle, A as the node assigned to rebroadcast, α as the heading direction of A, and B as the neighbor of A (heading in the same direction as A), we calculate their new x-coordinate using Equation (3.8).

$$\begin{aligned} A'_x &= A_x \times cos(-\alpha) - A_y \times sin(-\alpha) \\ B'_x &= B_x \times cos(-\alpha) - B_y \times sin(-\alpha) \\ S'_x &= S_x \times cos(-\alpha) - S_y \times sin(-\alpha) \end{aligned} \tag{3.8}$$

Vehicle A is selected as an SCF vehicle if and only if:

$$\nexists B \in NB, (B'_x < A'_x < S'_x) \vee (B'_x > A'_x > S'_x) \tag{3.9}$$

where NB is a list of A's neighbors, who have the same direction as A. Note that in the SCF scheme, two vehicles are decided to have the same direction if the difference between their heading directions is less than $\frac{\pi}{6}$.

4 Performance Analysis

The performance of the proposed ABDDis scheme has been evaluated via simulation experiments. We use the Veins 4.4 framework [28] that provides a bidirectional coupling between a discrete event-based network simulator called Objective Modular Network Testbed in C++ (OMNeT++) 5.0 [38] and a traffic simulator called Simulation of Urban MObility (SUMO) 0.25 [39]. Veins implements PHY and MAC layers based on the DSRC/WAVE (Dedicated Short Range Communication/Wireless Access in Vehicular Environments)[6] stack which is defined by IEEE 802.11p. Additionally, ABDDis scheme is compared to Flooding and other beacon-based schemes such as DV-CAST [18] and NSF [20].

4.1 Methodology

A fragment of A712 highway located in Dumfries and Galloway, Scotland is selected as the simulation area as shown in Figure 4.1. The geographic data was retrieved from OpenStreetMap [27]. This two-way highway is 2.5 km long and has two lanes in each direction with the speed limit of 27.78 m/s. The experiments are performed in five simulation scenarios with different traffic flows that represent the rate of inserting new vehicles to the network. These flows range from very sparse to dense traffic that include 500, 800, 1000, 3000 and 5000 vehicles/hour. Furthermore, to achieve a realistic car following behavior, each car obtains its own expected speed per lane from an assigned speed factor. This speed factor is the multiplier of the lane's speed limit and follows a normal distribution with $\mu = 0.8$ and $\sigma = 0.2$. This configuration results in a speed distribution where 95% of vehicles drive between 60% and 100% of the speed limit. Moreover, cars follow the mobility model that was proposed by Krauss et al. [40]. The simulation duration is 200 seconds. The first 100 seconds is for

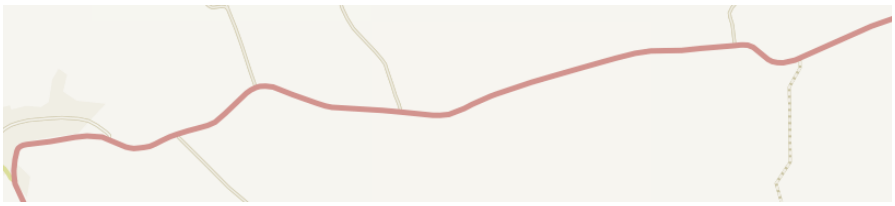


Figure 4.1: A fragment of A712 highway

vehicles to be generated and move around the area. After that, an arbitrary vehicle will initiate and broadcast a warning message. The goal is to disseminate this message to as many vehicles as possible in a timely manner.

For configuring the MAC/PHY layer of the transceiver device, the data rate is set to 6 Mbps, the transmission power is set to 300 mW and the receiver sensitivity is set to -100 dBm. The Free-Space Path Loss (FSPL) propagation model is used with the loss exponent is assigned to 3. This configuration achieves the transmission range of approximately 366m.

The beacon interval of DV-CAST and NSF are set to one second according to their recommendation [18, 20]. For DV-CAST, we use the Weighted p-persistence as its broadcast suppression technique. In the ABDDis scheme, the Slotted p-persistence is configured with four time slots and the slot time is 6 ms. Each data point is the mean of 32 replications with a confidence interval of 95%. All the simulation parameters are summarized in Table 4.1.

We evaluate the performance of the proposed scheme based on seven metrics as followings.

- *Coverage:* this is the percentage of vehicles that received the warning message. It implies the reliability of a scheme.
- *Delay:* this metrics measures the average latency between the sending of the warning message and its reception by other vehicles. It illustrates the average end-to-end delay. Additionally, it also relates to the coverage since a high coverage means farther vehicles can receive the message, thus the average delay is high.
- *Warning notification time:* it illustrates the number of informed vehicles as a function of time. Therefore, it determines how fast the warning message travels across the network. In this metric, we only consider two traffic flows which are 500 and 5000 vehicles per hour.

Table 4.1: Simulation Parameters

Parameter	Value
Map	A712 highway, Scotland
Traffic flows	500, 800, 1000, 3000 and 5000 vehicles/hour
Road length	2.5 km
Number of lanes	4 (2 per direction)
Speed limit	27.78 m/s
Simulation time	200 s
Mobility model	Krauss [40]
MAC/PHY	802.11p
Frequency	5.890 GHz
Propagation model	FSPL
Loss exponent	3
Bit rate	6 Mbps
Transmit power	300 mW
Receiver sensitivity	-100 dBm
Transmission range	$\sim$ 366m
Propagation delay	$\sim$ 5.120 ms
T_{max}	3 s
Beacon interval (DV-CAST and NSF)	1 s
Packet timeout	120 s
Number of runs	32

- *Collision ratio:* this is the percentage of collisions out of total packets received at the MAC layer by all vehicles. For DV-CAST, NSF and ABDDis, the packet refers to both warning messages and beacons, however, for Flooding, it only refers to the warning messages. The collision mainly occurs when too many packets overload the channel which is the indication of the broadcast storm problem. A high value of collision ratio implies a serious of the broadcast storm.

- *Number of messages received per vehicle:* it measures the amount of warning messages received per vehicle after the dissemination process. It implies the overhead of a scheme.

- *Number of beacons received per vehicle:* it is the total number of beacons received per vehicles. In a dense traffic scenario, too many beacons may lead to the broadcast storm. Flooding will not be accounted for this metric because it does not use beacon.

- *Efficiency:* we evaluate the efficiency of each scheme based on the achieved

coverage (CV), collision ratio (CR), number of messages received per vehicle (OV) and delay (D). In particular, an efficient scheme should have a high coverage, few collisions, small overhead and short delay across various traffic flows. The efficiency is calculated by Equation (4.1).

$$Efficiency = \frac{CV^2}{CR \times D \times OV} \tag{4.1}$$

4.2 Experimental results and Discussion

All the experimental results are presented in belows where Figure 4.2 shows the coverage of each scheme. As can be seen, the ABDDis and NSF schemes achieve the same and highest performance with about 100% coverage across various traffic scenarios. Moreover, in a very sparse environment (500 vehicles/hour), flooding scheme is the worst protocol with only more than 60% vehicles are notified, the DV-CAST scheme is better with about 90% of nodes informed. The reason is the Flooding scheme lacks of SCF mechanism to overcome the network partition problem that usually occurs in sparse environments. At 800 and 1000 vehicles/hour, the Flooding and DV-CAST schemes reach more than 90% of nodes. For 3000 and 5000 vehicles/hour, all schemes achieve 100% that implies the network is well-connected at these traffic flows. Based on this result, it can be concluded that the ABDDis scheme is very reliable since it has a good SCF mechanism that is able to handle the disconnected network problem to achieve a high delivery ratio under different traffic densities.

For the average end-to-end delay, the Flooding scheme has the smallest delay among other schemes at 500, 800 and 1000 vehicles/hour as shown in Figure 4.3. However, at those traffic flows, the Flooding scheme also has the smallest coverage ratio, that means, the higher delay of other schemes mainly come from the SCF mechanism. Furthermore, it can be observed that the ABDDis scheme has a higher latency than the NSF and DV-CAST schemes. Comparing to the DV-CAST scheme, the higher delay comes from the higher coverage. On the other hand, the higher latency than the NSF scheme is due to the fact that the ABDDis scheme chooses less SCF vehicles than the NSF scheme (every vehicle is a SCF vehicle with NSF scheme [20]). For well-connected scenarios, all schemes achieve nearly the same performance.

Figures 4.4a and 4.4b shows more experimental results in terms of the delay at 500 and 5000 vehicles/hour, respectively. At 500 vehicles/hour, a very sparse network, we can compare the effectiveness of the SCF mechanism. It can be

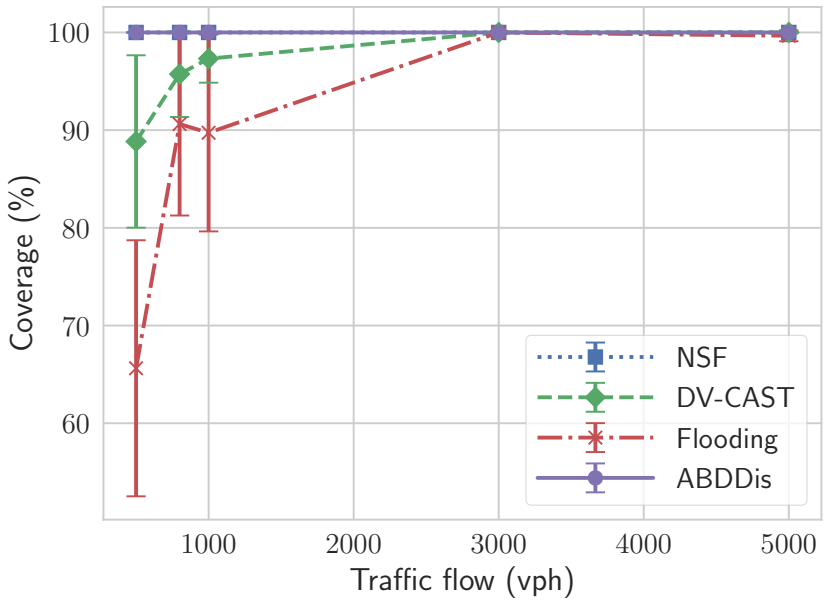


Figure 4.2: Coverage

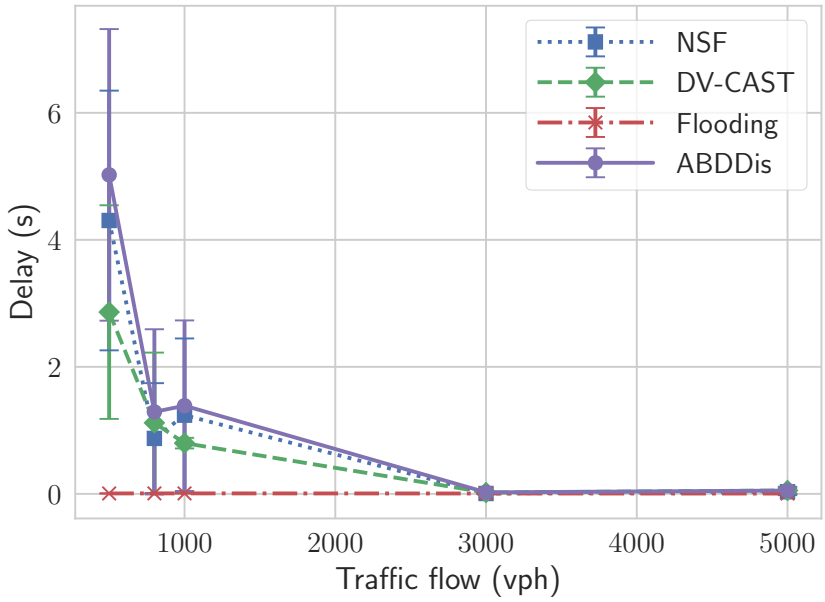
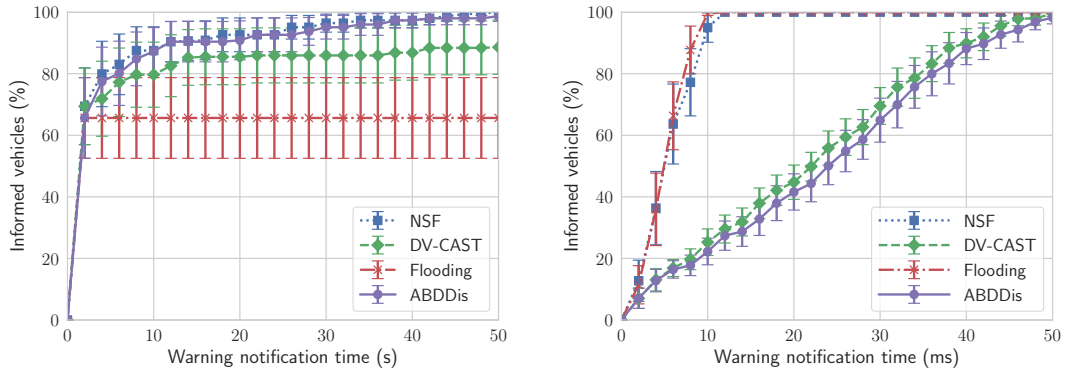


Figure 4.3: Delay



(a) 500 vehicles/hour

(b) 5000 vehicles/hour

Figure 4.4: Warning notification time

seen that the SCF mechanism of the ABDDis scheme is faster and better than the DV-CAST scheme, however, it is slightly slower than the NSF scheme. At 5000 vehicles/hour, a well-connected network, the Flooding and NSF schemes are very fast and have nearly the same speed. Moreover, the ABDDis and DV-CAST schemes are slower because of the delay introduced by the broadcast suppression techniques. Therefore, the latency is considered as a downside of the proposed scheme. However, this latency is acceptable since it takes only about 50ms to reach all vehicles in the network.

Figure 4.5 shows the experimental results in terms of the collision ratio. As expected, the Flooding scheme produced much more collisions compared to other schemes since it lacks of a broadcast suppression technique. As the traffic becomes denser, the broadcast storm is more serious in the Flooding scheme. Specifically, at 5000 vehicles/hour, up to 70% of packets are failed to be received due to collision for Flooding. Other schemes have a ratio of less than 10%, the ABDDis and DV-CAST schemes achieve the best performance with less than 1% across various traffic densities. This means that packet delivery using ABDDis scheme is more efficient than NSF and Flooding schemes.

Additionally, Figure 4.6 illustrates the number of messages received per vehicle or the overhead of each scheme. Note that it only accounts for the successfully received messages, that is why the Flooding scheme has the lowest overhead since only 50% and 30% of packets are received successfully at 3000 and 5000 vehicles/hour, respectively, according to the collision ratio. In this graph, it is also observed that the NSF scheme has the highest overhead, thus, the NSF scheme is prone to broadcast storm problem due to the lack of broadcast

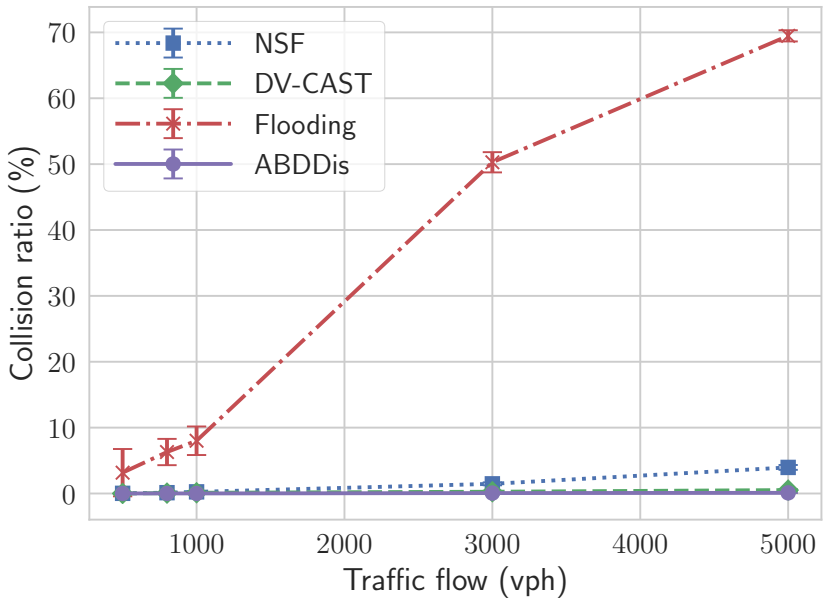


Figure 4.5: Collision ratio

suppression technique. The ABDDis scheme achieves the best performance regardless of the traffic flows in this metric.

The number of beacons received per vehicle is shown in Figure 4.7. Since the NSF and DV-CAST schemes use a fixed beacon interval (i.e., 1 second), they have the same performance. With an adaptive beacon mechanism that adjusts the beacon interval according to the current traffic, the ABDDis scheme significantly reduces about 60% of the beacon load under all traffic flows. It can be concluded that this beaconing mechanism is effective because not only it reduces the number of generated beacons but also maintains a good neighbor information accuracy so that the ABDDis scheme can properly operate (as proved by having a high coverage). From Figures 4.5 to 4.7, it can be seen that the ABDDis scheme achieves the best performance in those metrics with low collisions, low number of beacons and low number of received messages. Therefore, the ABDDis scheme solves the broadcast storm problem very well.

Figure 4.8 illustrates the efficiency of each scheme according to Equation (4.1). Since this metric considers previous results, the ABDDis scheme is expected to perform better than other schemes. The NSF scheme is the worst scheme due to the fact that it encounters a serious broadcast storm. From 500 to 3000 vehicles/hour, the efficiency of the ABDDis scheme is increased because the

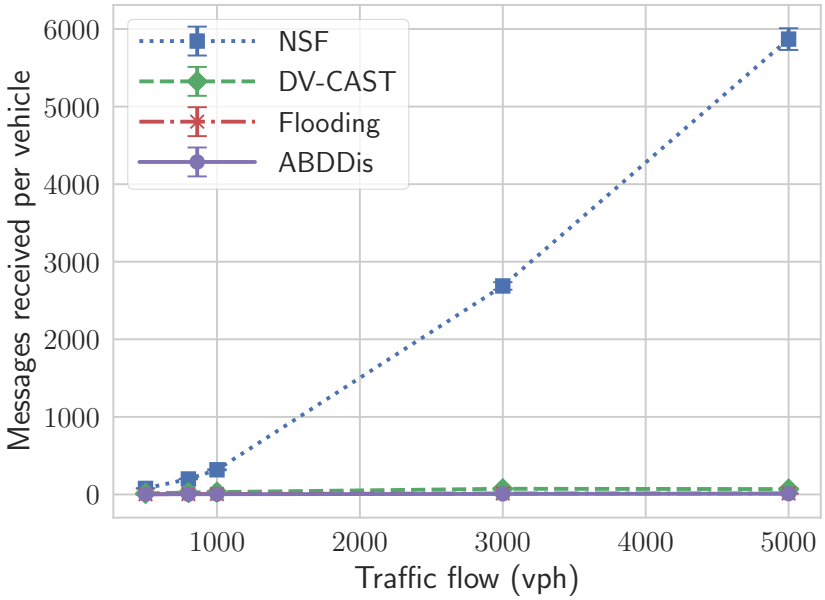


Figure 4.6: Number of messages received per vehicle

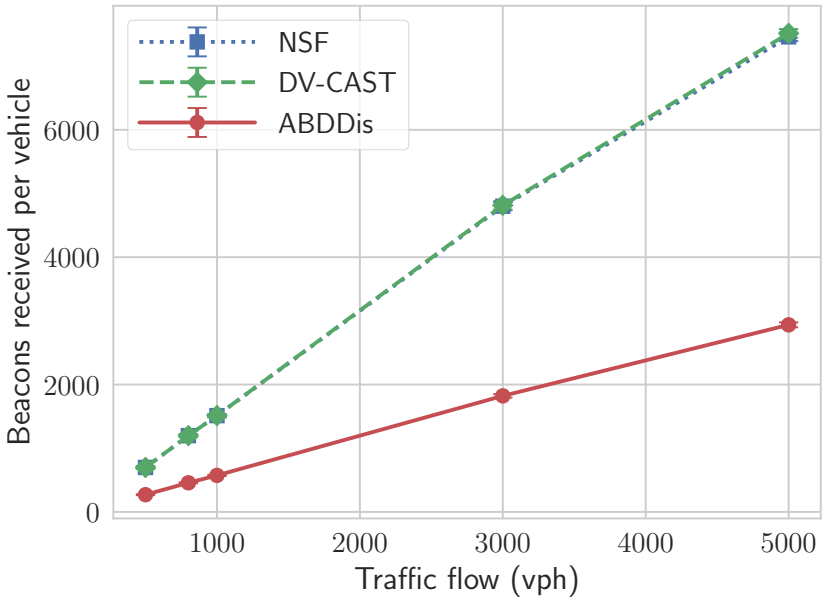


Figure 4.7: Number of beacons received per vehicle

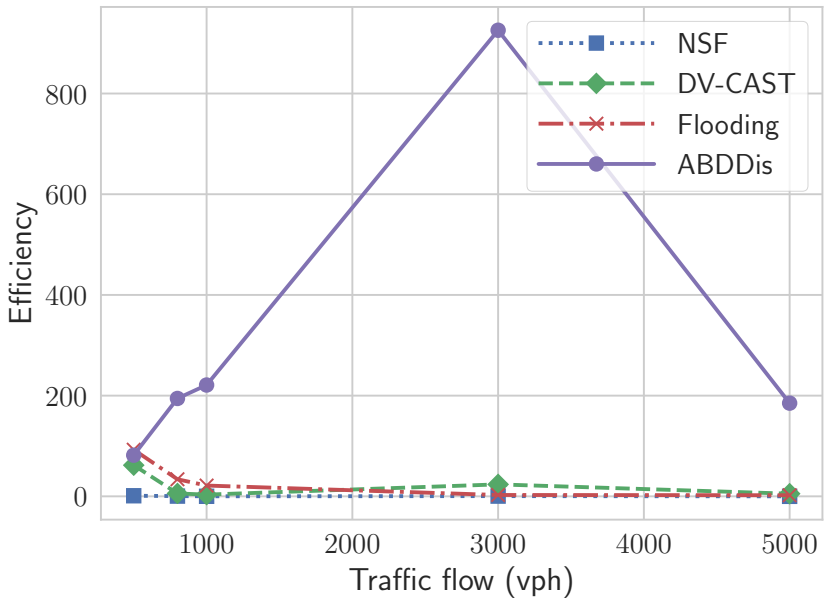


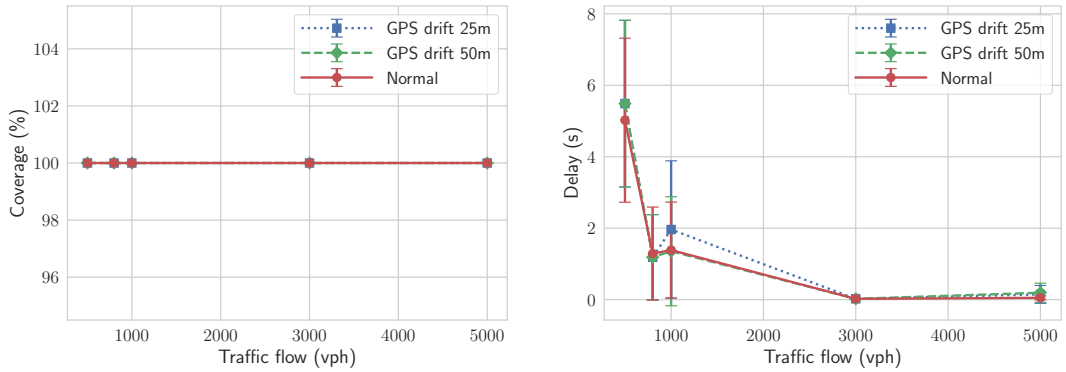
Figure 4.8: Efficiency

delay becomes shorter as the network becomes denser and it is well-connected at 3000 vehicles/hour. However, the efficiency drops at 5000 vehicles/hour because more messages are generated with higher delay due to more vehicles but the coverage is still the same.

In summary, the above experimental results show that the proposed ABDDis scheme outperforms other schemes in mitigating the broadcast storm as it significantly reduces the amount of beacons, messages and collisions in the network. Furthermore, by adopting the SCF mechanism, it is capable of handling the network partition problem by maintaining a high coverage in sparse traffic scenarios.

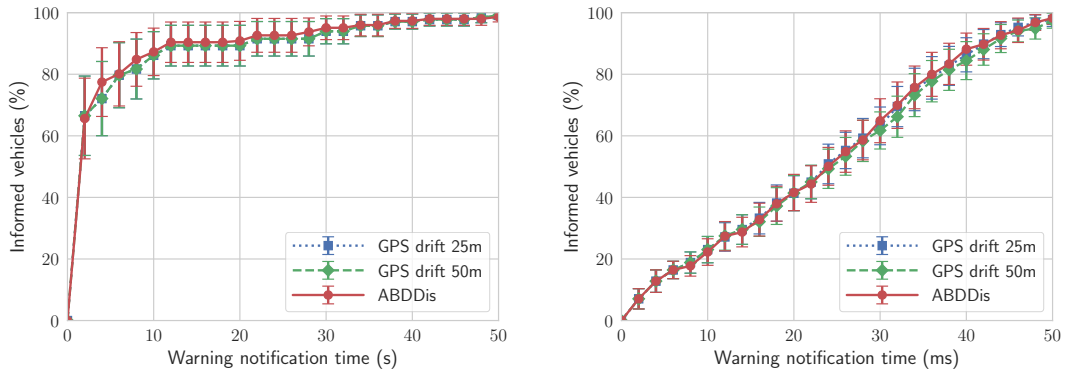
4.2.1 GPS drift

The accuracy of GPS is another essential issue since the proposed ABDDis scheme relies much on the GPS information. There are many causes of GPS drift such as obstruction, signal attenuation, and differences between GPS receivers that make it an inevitable problem. Therefore, we also investigate this circumstance by adding a random skew in [-25m, 25m] and [-50m, 50m] to the obtained GPS position.



(a) Coverage (b) Delay

Figure 4.9: Impact of GPS drift on Coverage and Delay

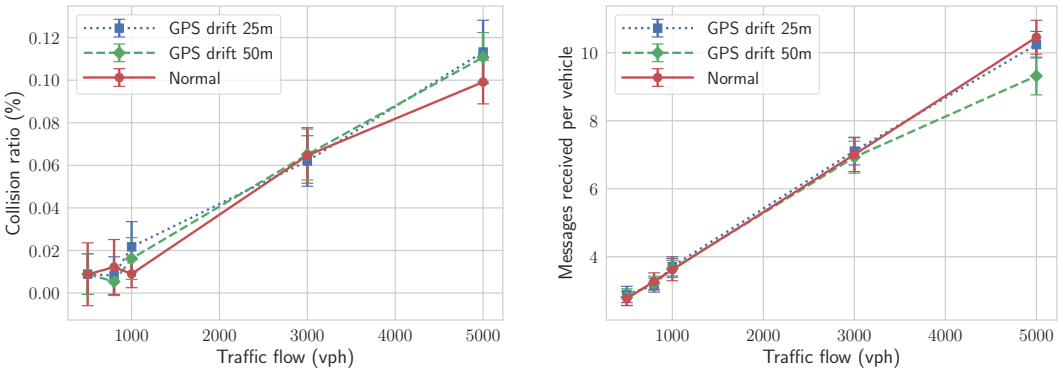


(a) 500 vehicles/hour (b) 5000 vehicles/hour

Figure 4.10: Impact of GPS drift on warning notification time

Figure 4.9 illustrates the impact of GPS on the coverage and delay of the ABDDis scheme where Figure 4.9a shows that the ABDDis scheme still have a good coverage regardless of the GPS accuracy. Nevertheless, the delay of this scheme is affected by the drift at sparse scenarios as illustrated in Figure 4.9b. This is because the SCF mechanism relies on the position of vehicles to operate. However, the impact is relatively small, less than 1 second.

As can be seen in Figures 4.10a and 4.10b, there is not much difference when we add errors to the GPS position. Figure 4.11a shows that the impact of GPS on the collision ratio. GPS drift tends to cause more collisions, however, they are still very low compared to other schemes. In Figure 4.11b, at 5000 vehicles/hour, there is a small difference that is mainly caused by the Slotted 1-persistence.



(a) Collision ratio

(b) Number of messages received per vehicle

Figure 4.11: Impact of GPS drift on collision ratio and number of received messages

As described in section 3.1, the beaconing mechanism does not account for the GPS position, therefore, there is no impact of GPS drift as shown in Figure 4.12.

Finally, the efficiency becomes unstable, especially in sparse scenarios as shown in Figure 4.13 because the delay and collision ratio are affected by GPS drift. However, if we account for the ability to solve the broadcast storm and network partition problem by keeping a high coverage and small amount of packets, the impact of GPS drift on the proposed scheme is negligible.

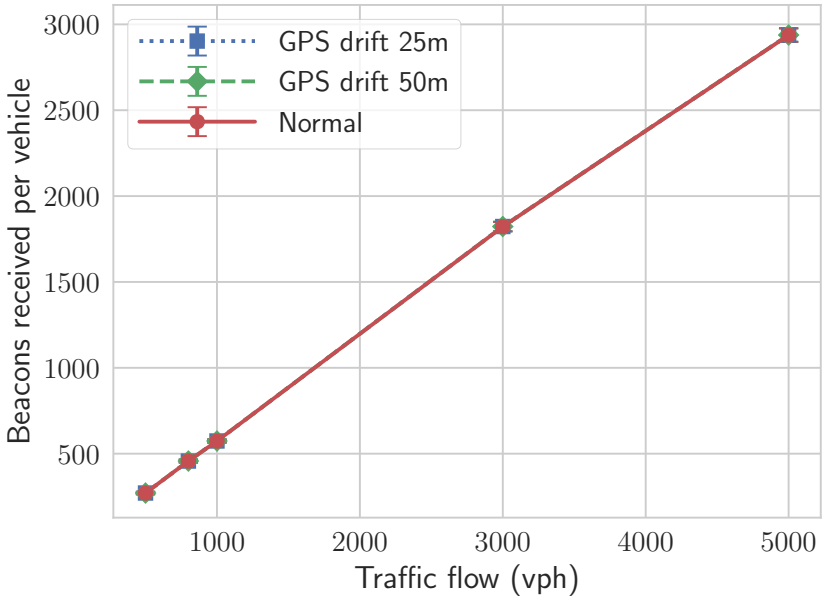


Figure 4.12: Impact of GPS drift on number of beacons

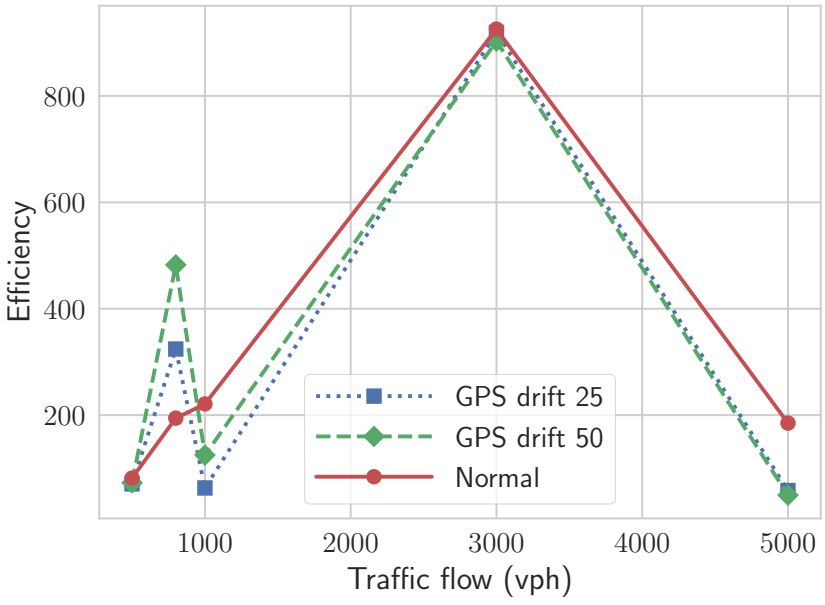


Figure 4.13: Impact of GPS drift on the Efficiency

5 Conclusion and future works

5.1 Conclusion

In this thesis, we have proposed a novel scheme combining the broadcast suppression technique, adaptive beacon and store-carry-forward mechanism which aims to efficiently handle the broadcast storm and network partition problems in VANETs while maintaining a good coverage regardless of traffic density. The ABDDis scheme is designed to be capable of operating properly under both dense and sparse traffics.

The most valuable contributions of the ABDDis scheme are (1) the novel mechanism to adaptively adjust the beacon interval based on neighbors' speed by Equation (3.5) to reduce the channel load while being able to cope with changes in the network topology, and (2) the novel store-carry-forward (SCF) mechanism to handle the network partition problem. Additionally, a novel prediction model has been adopted to improve neighbor information accuracy by Equation (3.7).

The simulation result shows that the ABDDis scheme outperforms other schemes in mitigating the broadcast storm and has a high delivery ratio across various traffic densities. Furthermore, the ABDDis scheme is robust enough to tolerate reasonable GPS drift.

On the basis, our work has effectively contributed to the study of warning messages dissemination in VANET. With this research, traffic safety can be further improved and by diminish the broadcast storm, the transmission bandwidth that has been freed up can be utilized for other applications.

5.2 Future works

Some approaches are pointed out below:

- Reducing the latency introduced by the adoption of a broadcast suppression technique and the SCF mechanism.
- Carrying out detailed experiments to evaluate the effectiveness of the adaptive beacon mechanism.
- Investigating on embracing some security mechanism to the proposed scheme.

A Guide to SUMO, OMNeT++ and VEINS Installation

This is an installation guide of SUMO, OMNeT++ and Veins on Linux OS (Ubuntu version).

Since SUMO and OMNeT++ are absolutely separating software, each of them has many released version. To run Veins simulator, SUMO and OMNeT need suitable version of each other to be coupled to run Veins simulator. On this project, we use the latest version of Veins-4.4, which requires version 0.25.0 of SUMO and 5.0 of OMNeT++.

The required versions of SUMO and OMNeT++ for each Veins version can be found at [41].

A.0.1 Preparation and dependencies

First, we need to get source code for installation, refer to [42], [43] and [41] for OMNeT++ (5.0), SUMO (0.25.0) and VEINS (4.4), respectively.

After that, open terminal and create *omnetpp* in *home/user-name* directory (it can be somewhere else). Then, extract SUMO, OMNeT++ source code and put them inside *omnetpp* directory. The directory structure is as follow:

```
/home/user-name/omnetpp
├── omnetpp-5.0
└── sumo-0.25.0
```

Then, open Linux Terminal, copy and paste the following script to install

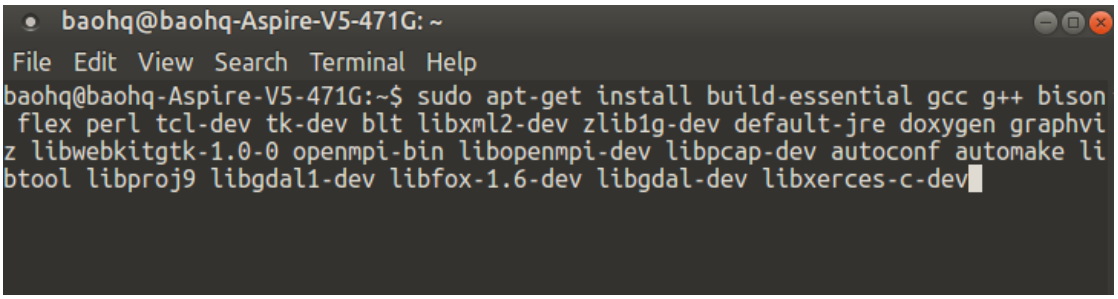


Figure A.1: Install dependencies for the simulators.

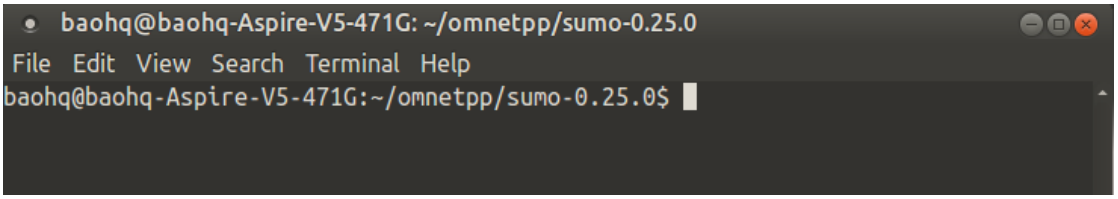


Figure A.2: First, go to extracted SUMO source code directory.

dependencies for the simulators (as in figure A.1):

```
$ sudo apt-get install build-essential gcc g++ bison
  flex perl tcl-dev tk-dev blt libxml2-dev zlib1g-dev
  default-jre doxygen graphviz libwebkitgtk-1.0-0 openmpi
  -bin libopenmpi-dev libpcap-dev autoconf automake
  libtool libproj9 libgdal1-dev libfox-1.6-dev libgdal-
  dev libxerces-c-dev
```

A.0.2 SUMO Installation

We go to *sumo-0.25.0* directory in *omnetp* as shown in figure A.2.

Next, run configure script to setup source code build process as following:

```
~/omnetpp/sumo-0.25.0$ ./configure --prefix=/home/user-
  name/omnetpp/sumo-0.25.0 --enable-debug
```

After configuring step, we run **make -j4** command to build the source code as shown in figure A.4.

After building source code successfully, we run **sudo make install** command to install SUMO binaries as shown in figure A.5.

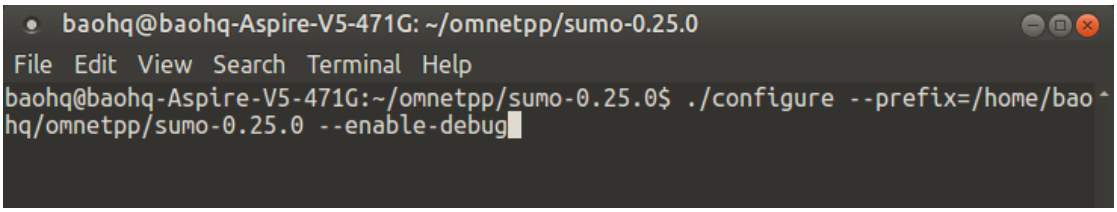


Figure A.3: Run configure script in extracted SUMO source code directory.

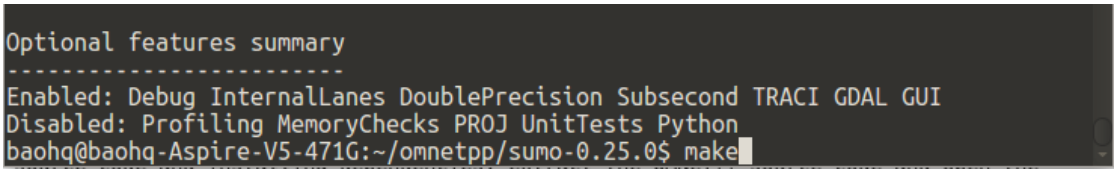


Figure A.4: Run **make -j4** to build SUMO source code.

Next, we need to set the environment variables for SUMO by editing **.bashrc**:

```
~/omnetpp/sumo-0.25.0$ nano ~/.bashrc
```

Add the following line at the end of the file as shown in :

```
export PATH=$PATH:/home/user-name/omnetpp/sumo-0.25.0/
  bin/
```

Then, validate the change in *.bashrc* by running:

```
~/omnetpp/sumo-0.25.0$ source ~/.bashrc
```

Finally, we verify the installation by running SUMO using following command:

```
~/omnetpp/sumo-0.25.0$ sumo-gui
```

Figure A.7 is the SUMO when running.

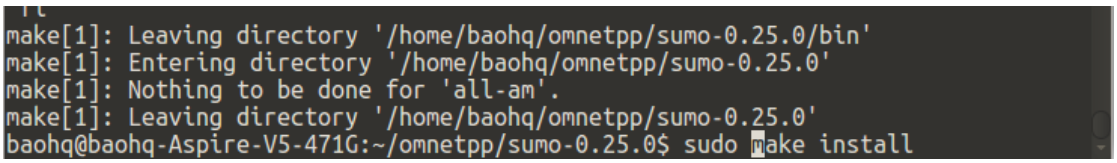


Figure A.5: Run **sudo make install** to install SUMO binaries.

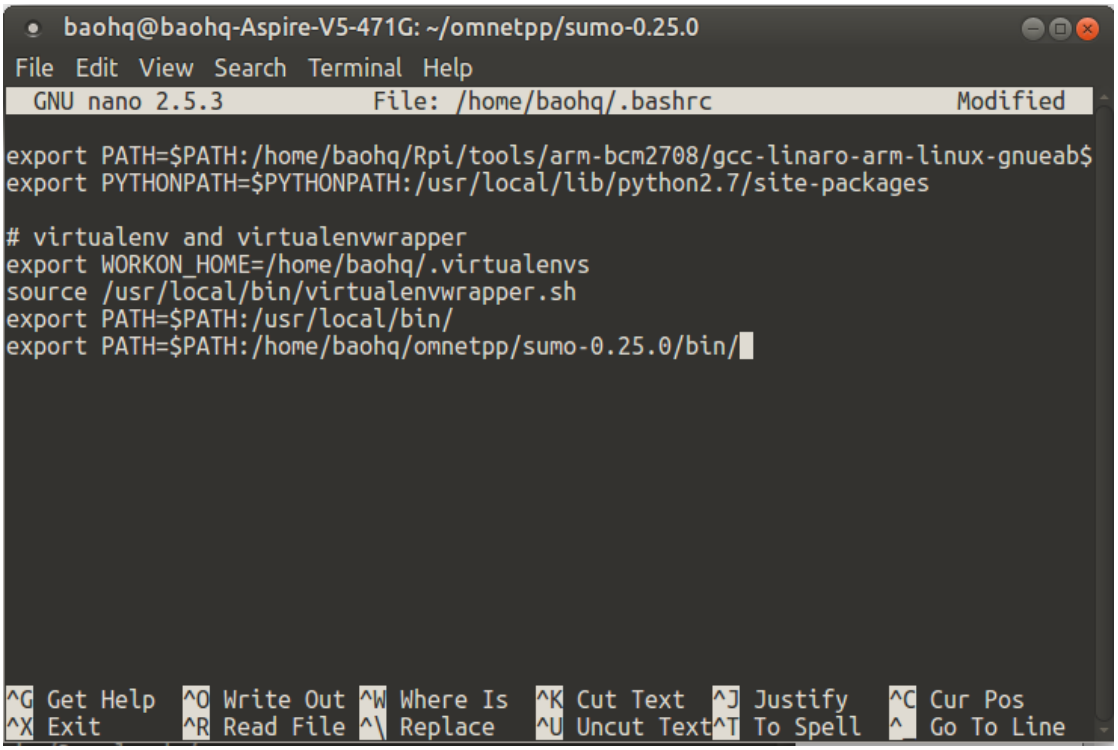


Figure A.6: Set the environment variables for SUMO by editing *.bashrc*.

A.0.3 OMNeT++ Installation

We go to *omnetpp-5.0* directory in *omnetpp* as shown in figure A.8.

Next, we need to set the environment variables for OMNeT++ by editing **.bashrc**:

```
~/omnetpp/omnetpp-5.0$ nano ~/.bashrc
```

Add the following line at the end of the file as shown in :

```
export PATH=$PATH:/home/user-name/omnetpp/omnetpp-5.0/
  bin/
```

Then, validate the change in *.bashrc* by running:

```
~/omnetpp/sumo-0.25.0$ source ~/.bashrc
```

Next, run configure script to setup source code build process as following:

```
~/omnetpp/omnetpp-5.0$ ./configure
```

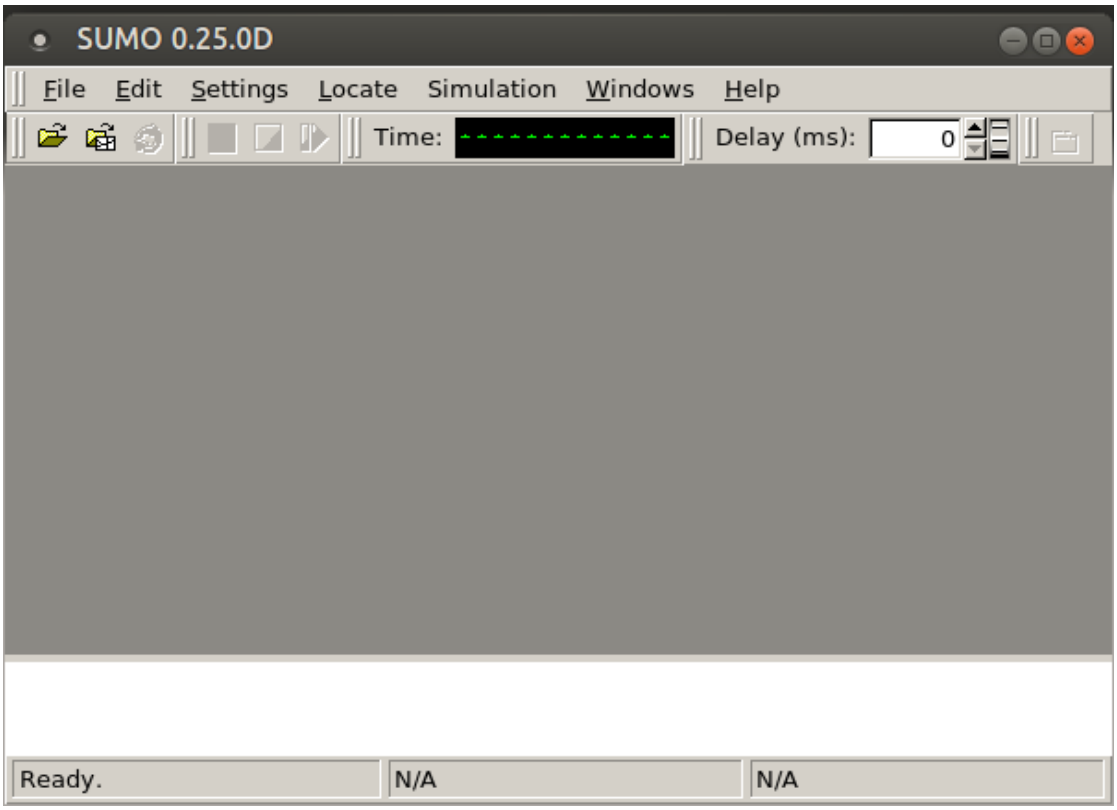


Figure A.7: SUMO was installed successfully.

Figure A.8: Go to extracted OMNeT++ source code directory.

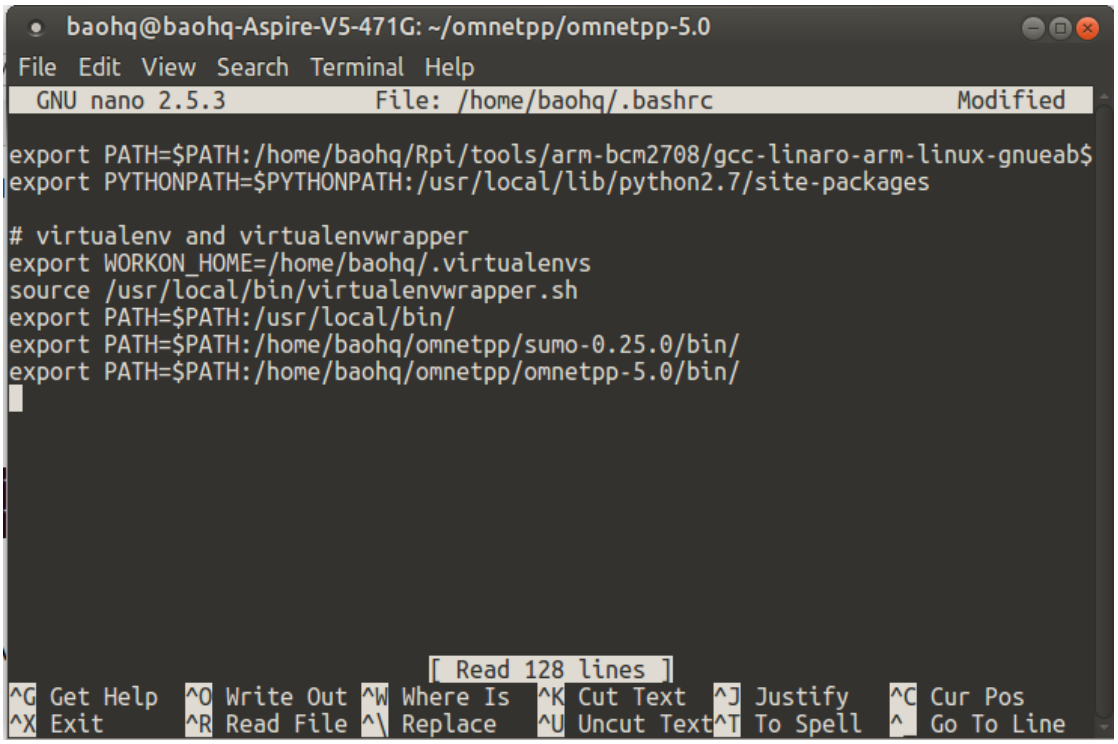


Figure A.9: Set the environment variables for OMNeT++ by editing *.bashrc*.

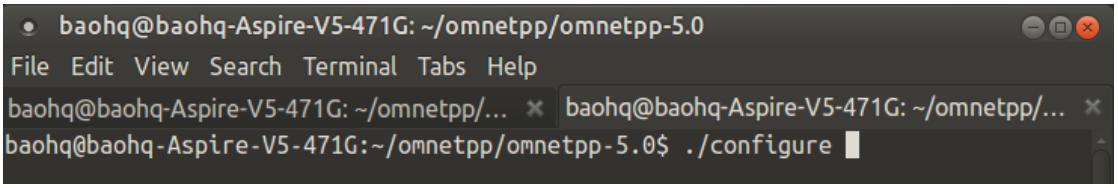


Figure A.10: Run configure script in extracted OMNeT++ source code directory.

```
Scroll up to see the warning messages (use shift+PgUp), and search config.log
for more details. While you can use OMNeT++ in the current configuration,
be aware that some functionality may be unavailable or incomplete.

Your PATH contains /home/baohq/omnetpp/omnetpp-5.0/bin. Good!
baohq@baohq-Aspire-V5-471G:~/omnetpp/omnetpp-5.0$ make -j4
```

Figure A.11: Run **make -j4** to build OMNeT++ source code.

```
Creating executable: out/gcc-debug//sockets
make[2]: Leaving directory '/home/baohq/omnetpp/omnetpp-5.0/samples/sockets'
make[1]: Leaving directory '/home/baohq/omnetpp/omnetpp-5.0'

Now you can type "omnetpp" to start the IDE
baohq@baohq-Aspire-V5-471G:~/omnetpp/omnetpp-5.0$ omnetpp
```

Figure A.12: Run OMNeT++.

After configuring step, we run **make -j4** command to build the source code as shown in figure A.11.

Finally, verify the installation by running OMNeT++ as shown in figure A.12:

```
1 ~/omnetpp/omnetpp-5.0$ omnetpp
```

As the OMNeT++ IDE starts, a window appears to prompt us install **INET framework** and OMNeT++ example (see figure A.13), click **OK** to start installation because VEINS need the framework to operate.

To create desktop icon or menu item, execute these commands:

```
1 ~/omnetpp/omnetpp-5.0$ make install-desktop-icon
2 ~/omnetpp/omnetpp-5.0$ make install-menu-item
```

Then, OMNeT++ was installed successfully.

A.0.4 VEINS Installation

Extract VEINS source code and put the extracted directory in *omnetpp*, the same place with **sumo-0.25.0** and **omnetpp-5.0**:

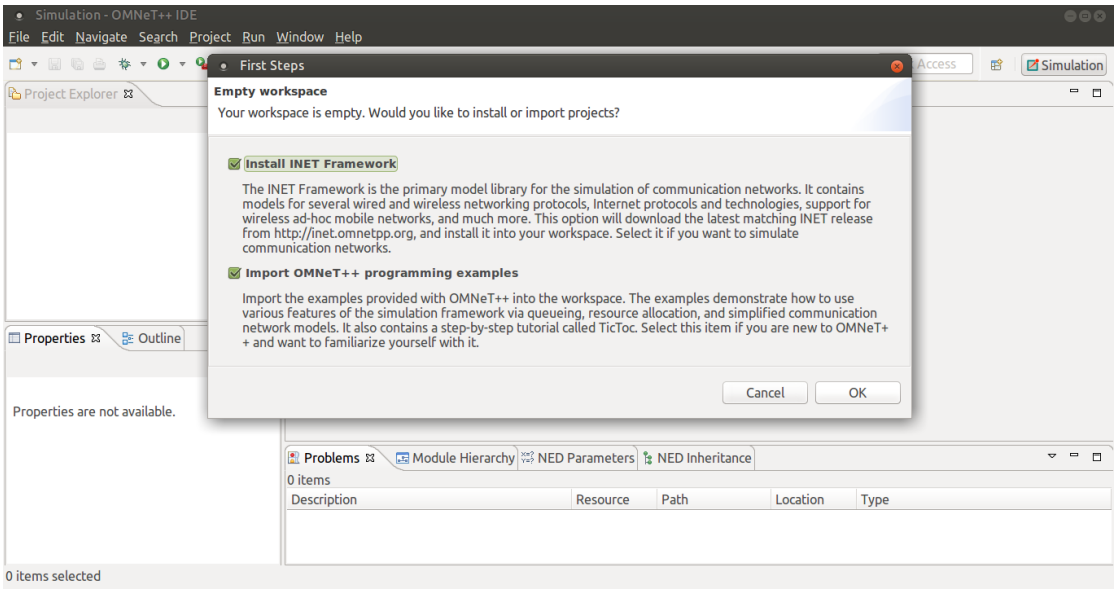


Figure A.13: INET framework installation.

```
/home/user-name/omnetpp
├── omnetpp-5.0
├── sumo-0.25.0
└── veins-veins-4.4
```

Then, start OMNeT++, import **VEINS** as a project to OMNeT++:

- On OMNeT++ IDE window, select **File -> Import...** as shown in figure A.14
- The **Import** window opens, select **General -> Existing Projects into Workspace**, then click **Next** as shown in figure A.15.
- A window shows up to choose the project, select **Browse**, then browse to **omnetpp** which contains **sumo-0.25.0**, **omnetpp-5.0** and **veins-veins-4.4** directory, select **veins-veins-4.4** and click on **OK**, as shown in figure A.16.
- Click **Finish** as shown in figure A.17.
- Project list is as shown in figure A.18, **veins** project at the bottom is our VEINS simulator. If the imported project's is not **veins**, it should be manually renamed to **veins**.
- Then, select **Project -> Build all**, as shown in figure A.19.

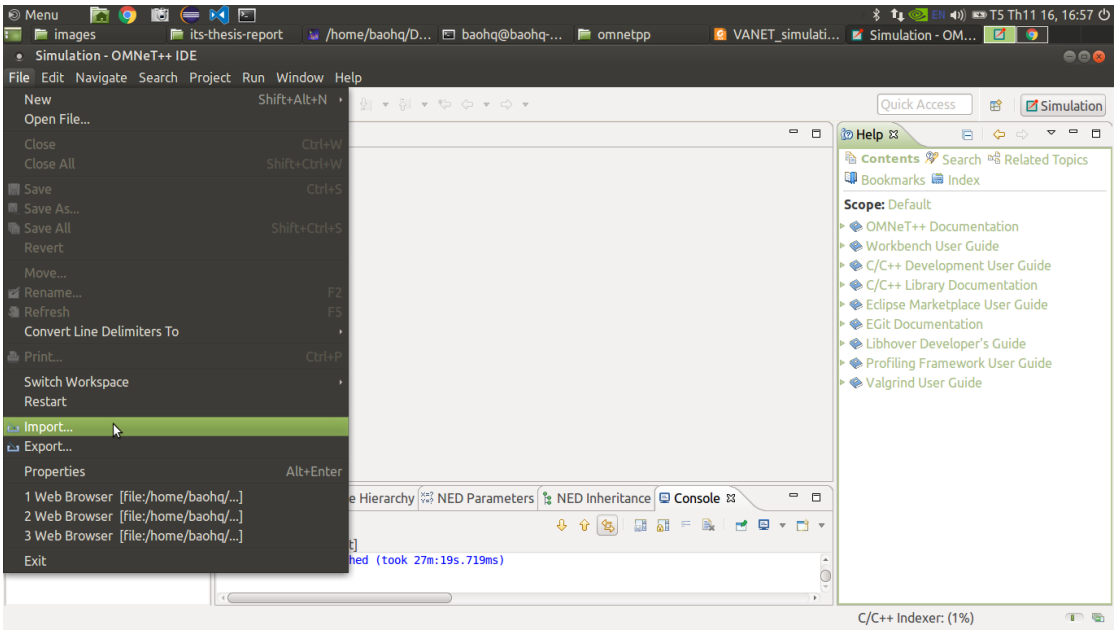


Figure A.14: Open OMNeT++, select select **File -> Import...**.

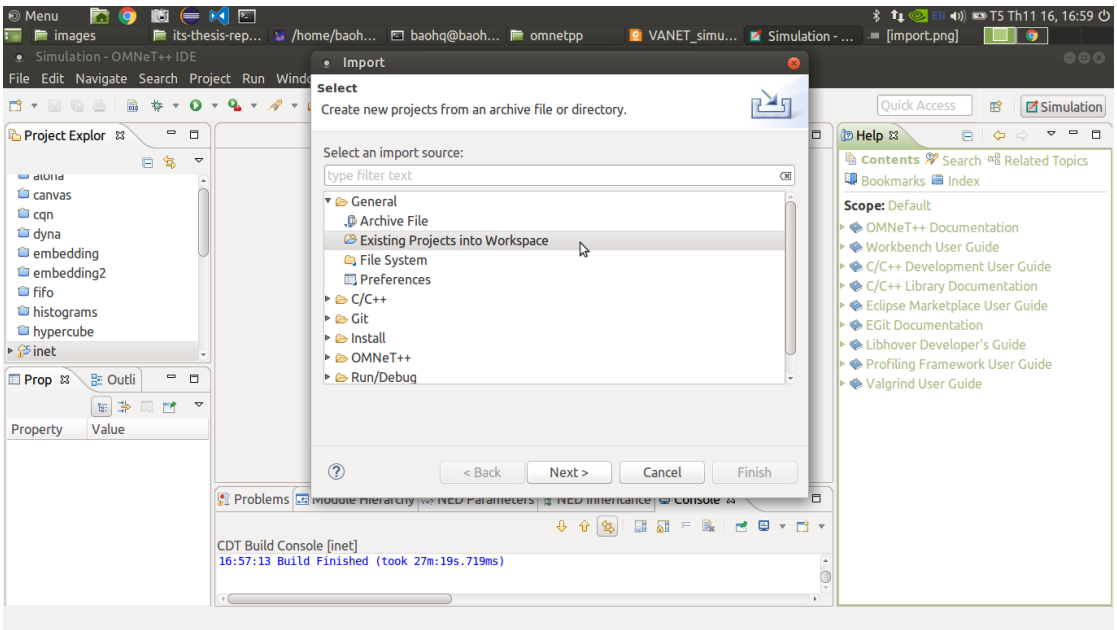


Figure A.15: Select **General -> Existing Projects into Workspace**, then click **Next**.

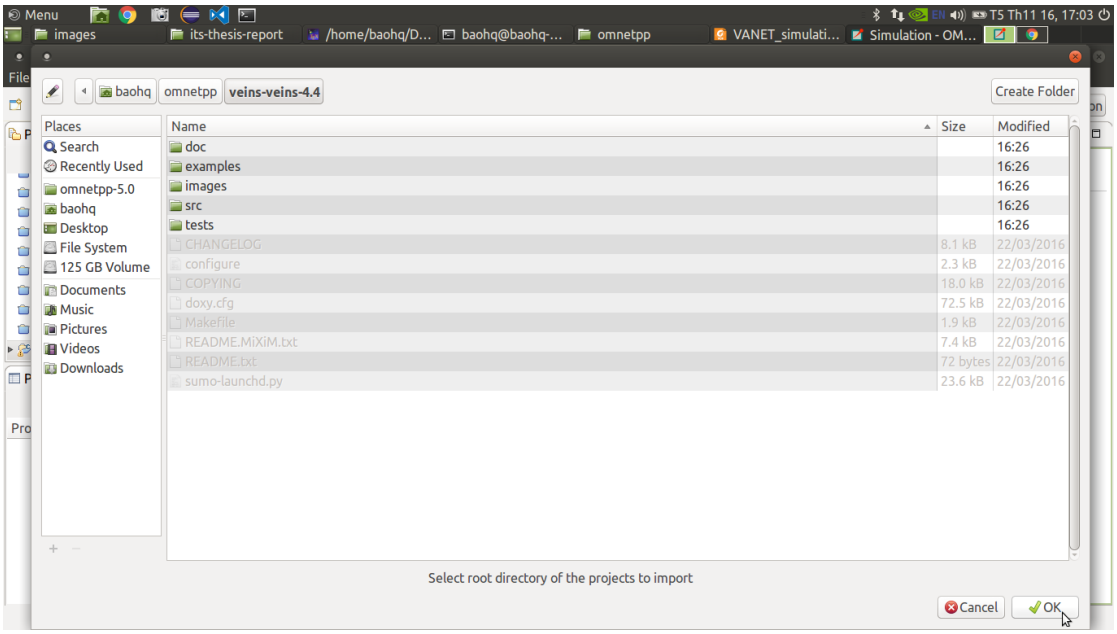


Figure A.16: Browse to omnetpp directory and select **veins-veins-4.4**.

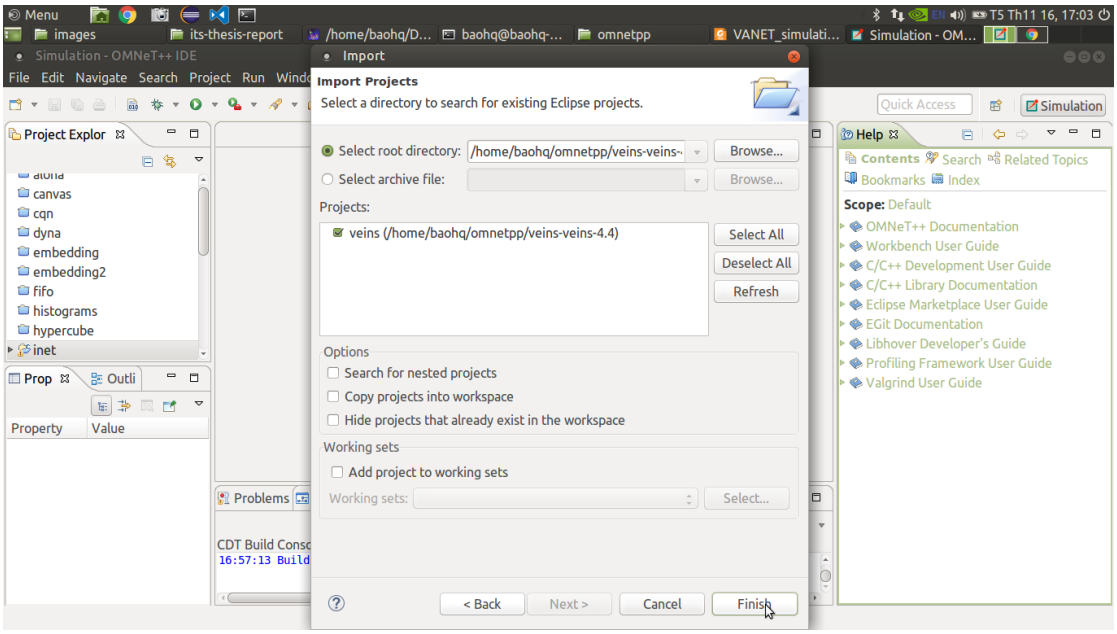


Figure A.17: Click **Finish**.

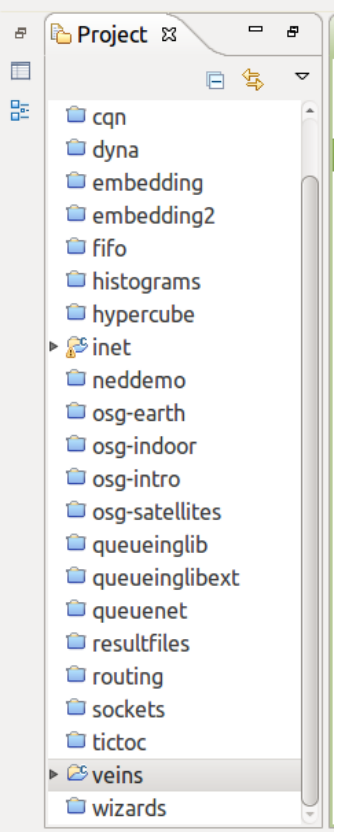


Figure A.18: Project explorer should have **veins** project in it.

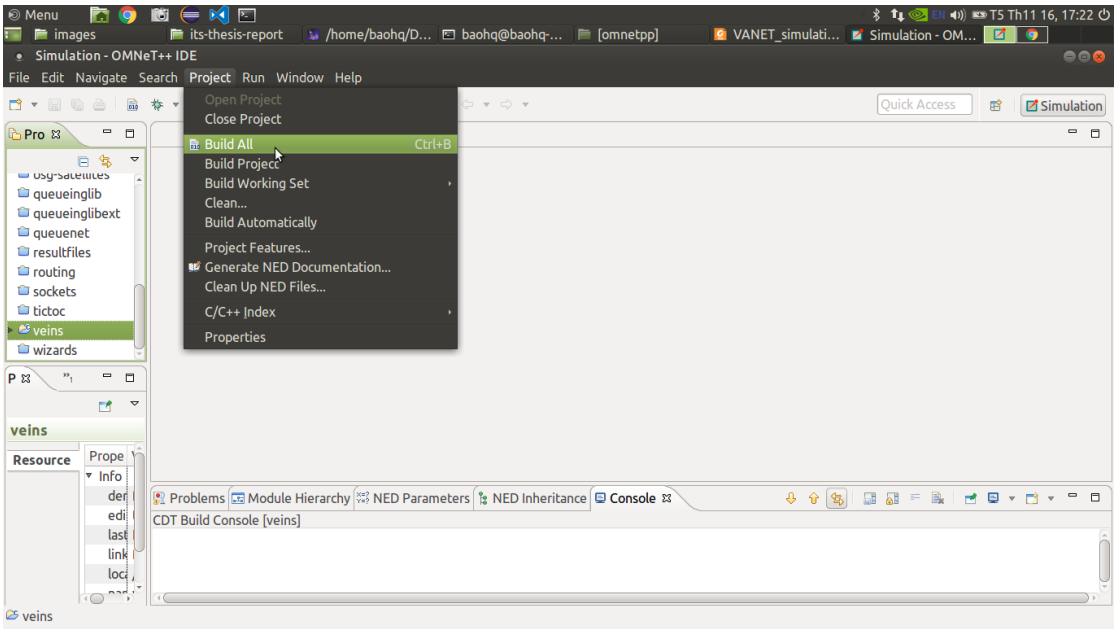


Figure A.19: .

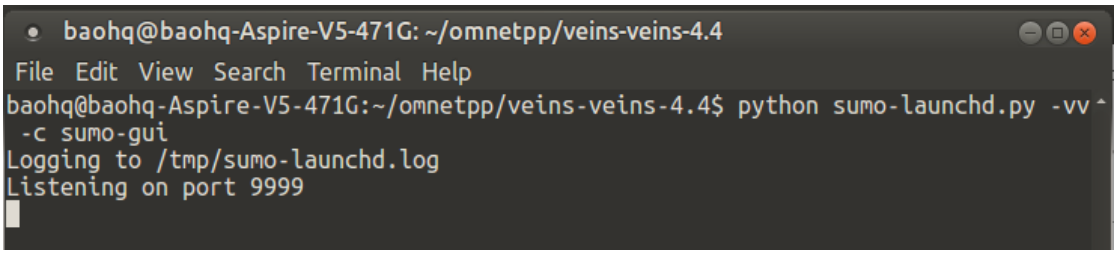


Figure A.20: Run sumo-launchd.py.

A.0.5 Run Veins example

To verify the Veins simulator installation, we run an available example in **VEINS** project. Since Veins simulator need the coupling between SUMO simulator and OMNeT++ simulator, we need to run python script **sumo-launchd.py** first.

- **sumo-launchd.py**: is designed to run in the background, listening for incoming requests. On each incoming connection, it receives the simulation setup in XML format, then launches a separate instance of SUMO and proxies requests between OMNeT++ and SUMO [44].

The **sumo-launchd.py** script reside in **veins-veins-4.4** directory, to run it, execute the following command:

```
1 ~/omnetpp/veins-veins-4.4$ python sumo-launchd.py -vv -c
  sumo-gui
```

After veins project is built, find the **omnetpp.ini** file in veins -> examples -> veins -> omnetpp.ini, then run it as shown in figure A.21.

After running **omnetpp.ini** file, an OMNeT++ simulation window will show up as in figure A.22, then click **OK**. The window will progress as in figure A.23, then press **F5** key or click on **run** button on the window to start the simulation.

When the simulation running, a SUMO simulation window is opened, which simulates the traffic as shown in figure A.25 and OMNeT++ simulation window simulates network as shown in figure A.24.

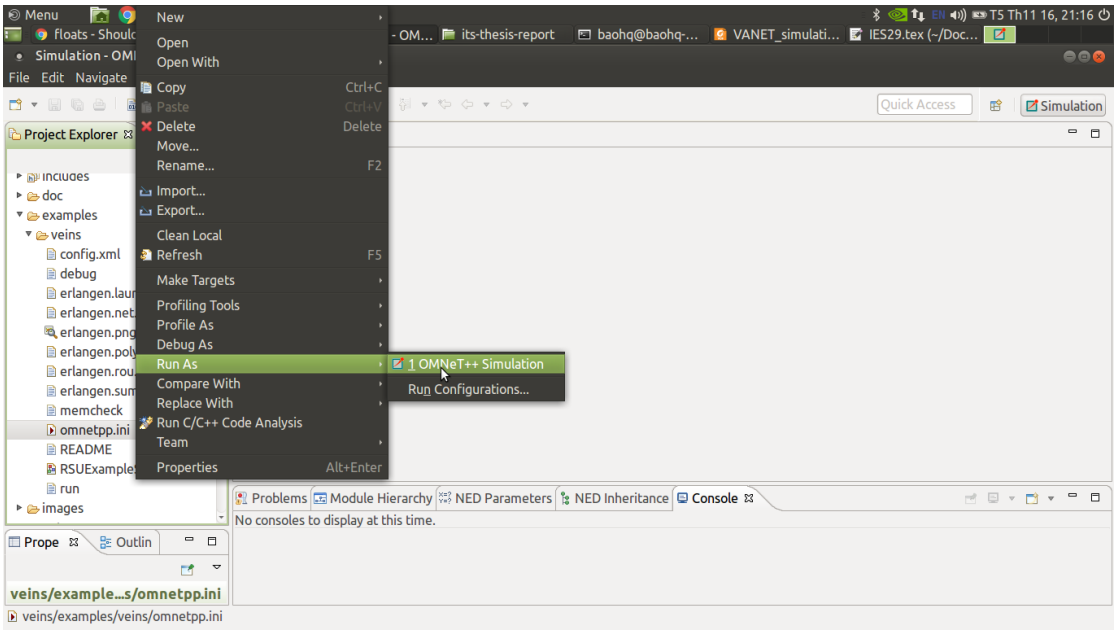


Figure A.21: Right click on omnetpp.ini, select Run as -> OMNeT++ Simulation.

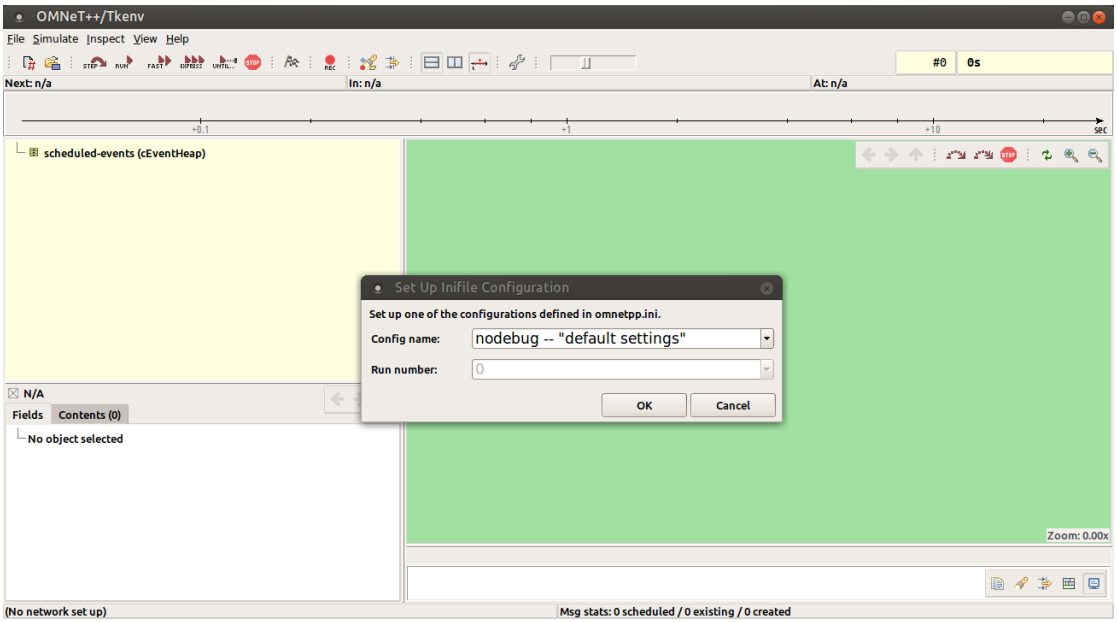


Figure A.22: OMNeT++ simulation window.

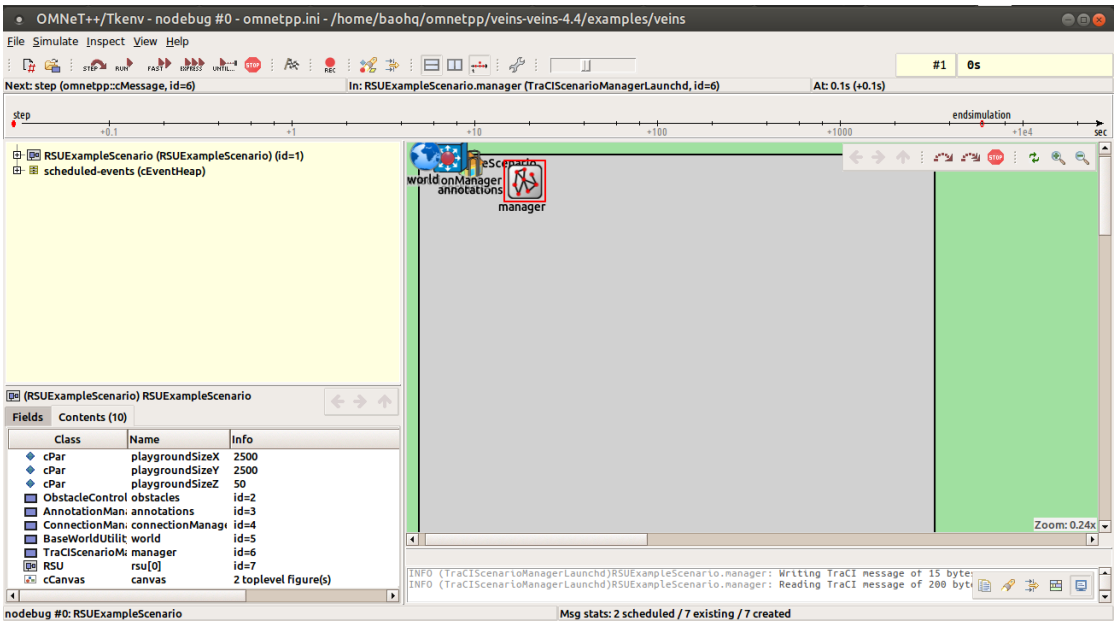


Figure A.23: OMNeT++ simulation window.

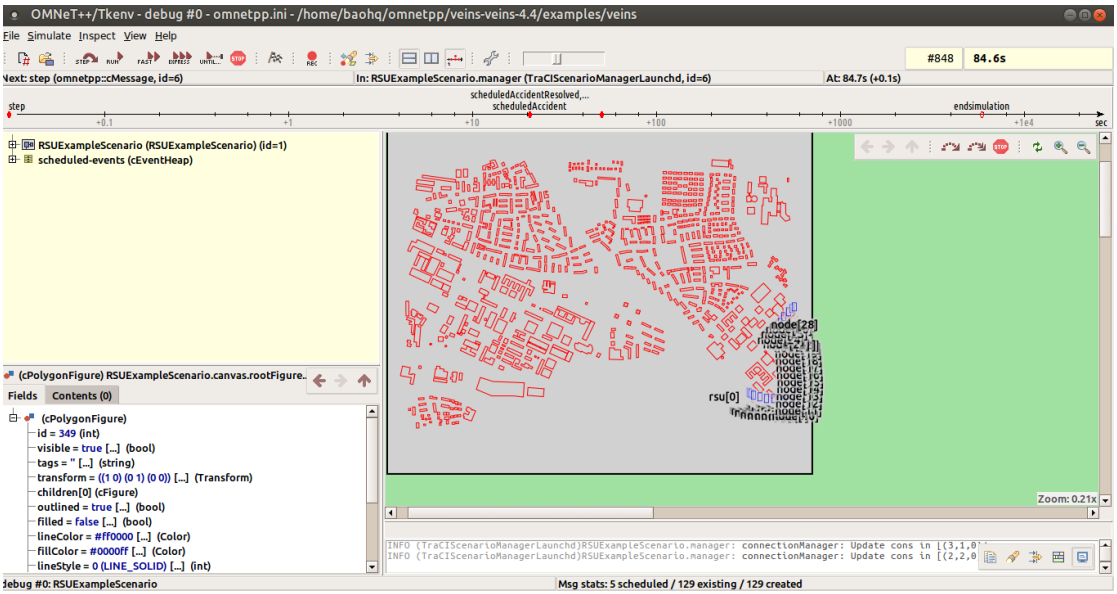


Figure A.24: OMNeT++ simulation window when running.

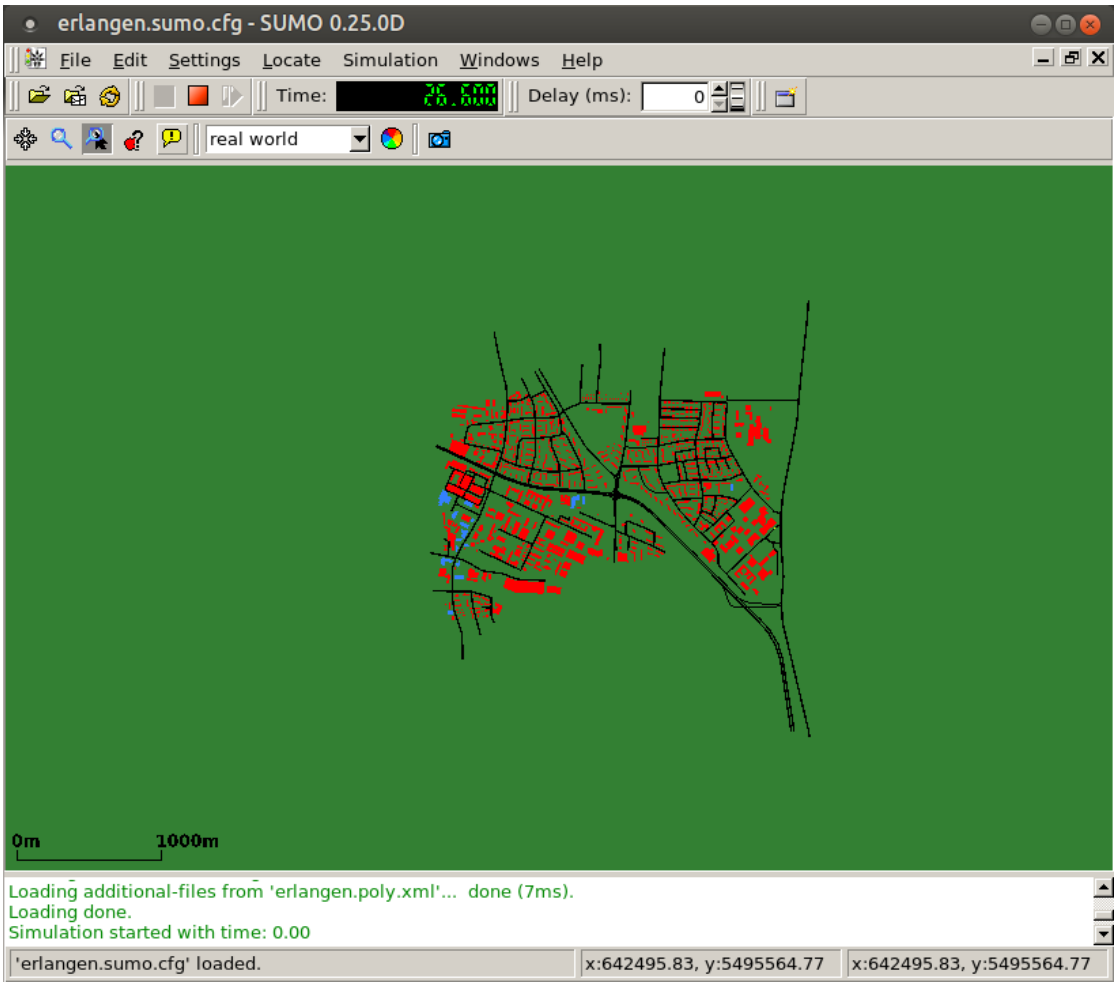


Figure A.25: SUMO simulation window when running.

References

[1] "Gi/itg kuvs fachgespräch inter-vehicle communication." [Online]. Available: http://fg-ivc.car2x.org/

[2] "Example of v2v intersection movement assist warning scenario." [Online]. Available: https://icsw.nhtsa.gov/safercar/v2v/

[3] "Vehicle to vehicle communications." [Online]. Available: https://www.nhtsa.gov/technology-innovation/vehicle-vehicle-communications

[4] "Knowledge centre: Blind spot detection." [Online]. Available: http://www.autoserviceworld.com/features/watching-cant-see/

[5] "Vehicle-to-infrastructure (v2i) resources." [Online]. Available: https://www.its.dot.gov/v2i/index.htm

[6] S. A. Ahmed, S. H. Ariffin, and N. Fisal, "Overview of wireless access in vehicular environment (WAVE) protocols and standards," *Indian Journal of Science and Technology*, vol. 6, no. 7, pp. 4994–5001, 2013.

[7] D. L. R. Herrera, "Simulation of basic multi-hop broadcast techniques in vehicular ad-hoc networks using veins simulator," Master's thesis, Universitat Politècnica de Catalunya, 2016.

[8] A. Varga, "OMNeT - simulation manual." [Online]. Available: https://omnetpp.org/doc/omnetpp/manual/

[9] A. Wegener, M. Piórkowski, M. Raya, H. Hellbrück, S. Fischer, and J.-P. Hubaux, "TraCI: an interface for coupling road traffic and network simulators," in *Proceedings of the 11th communications and networking simulation symposium*. ACM, 2008, pp. 155–163.

[10] N. Wisitpongphan, O. K. Tonguz, J. Parikh, P. Mudalige, F. Bai, and V. Sadekar, "Broadcast storm mitigation techniques in vehicular ad hoc networks," *IEEE Wireless Communications*, vol. 14, no. 6, pp. 84–94, dec 2007.

[11] 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, no. 7, pp. 813–828, oct 2009.

[12] "Road crash statistics." [Online]. Available: http://asirt.org/initiatives/informing-road-users/road-safety-facts/road-crash-statistics

[13] "Canada population (live)." [Online]. Available: http://www.worldometers.info/world-population/canada-population/

[14] "Global status report on road safety 2013: supporting a decade of action." [Online]. Available: http://www.who.int/violence_injury_prevention/road_traffic/en/

[15] P. Ranjan and K. K. Ahirwar, "Comparative study of vanet and manet routing protocols," in *Proc. of the International Conference on Advanced Computing and Communication Technologies (ACCT 2011)*, 2011, pp. 517–523.

[16] S. Yousefi, M. S. Mousavi, and M. Fathy, "Vehicular ad hoc networks (vanets): Challenges and perspectives," in *2006 6th International Conference on ITS Telecommunications*, June 2006, pp. 761–766.

[17] F. J. Ros, P. M. Ruiz, and I. Stojmenovic, "Reliable and efficient broadcasting in vehicular ad hoc networks," in *VTC Spring 2009 - IEEE 69th Vehicular Technology Conference*, April 2009, pp. 1–5.

[18] O. Tonguz, N. Wisitpongphan, and F. Bai, "DV-CAST: A distributed vehicular broadcast protocol for vehicular ad hoc networks," *IEEE Wireless Communications*, vol. 17, no. 2, pp. 47–57, apr 2010.

[19] J. A. Sanguesa, M. Fogue, P. Garrido, F. J. Martinez, J.-C. Cano, and C. T. Calafate, "A survey and comparative study of broadcast warning message dissemination schemes for VANETs," *Mobile Information Systems*, vol. 2016, pp. 1–18, 2016.

[20] ——, "Using topology and neighbor information to overcome adverse vehicle density conditions," *Transportation research part C: emerging technologies*, vol. 42, pp. 1–13, 2014.

[21] B. Tian, K. M. Hou, and J. Li, "Trad: Traffic adaptive data dissemination protocol for both urban and highway vanets," in *Advanced Information Networking and Applications (AINA), 2016 IEEE 30th International Conference on*. IEEE, 2016, pp. 724–731.

[22] L. A. Villas, A. Boukerche, G. Maia, R. W. Pazzi, and A. A. Loureiro, "Drive: An efficient and robust data dissemination protocol for highway and urban vehicular ad hoc networks," *Computer Networks*, vol. 75, pp. 381–394, 2014.

[23] M. S. Anwer and P. C. Guy, "A survey of vanet technologies," *Journal of Emerging Trends in Computing and Information Sciences*, 2014.

[24] G. Karagiannis, O. Altintas, E. Ekici, G. Heijenk, B. Jarupan, K. Lin, and T. Weil, "Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions," *IEEE communications surveys & tutorials*, vol. 13, no. 4, pp. 584–616, 2011.

[25] U. N. H. T. S. Administration, "NHTSA ISSUES ADVANCE NOTICE OF PROPOSED RULEMAKING AND RESEARCH REPORT ON GROUND-BREAKING CRASH AVOIDANCE TECHNOLOGY: "VEHICLE-TO-VEHICLE COMMUNICATIONS: READINESS OF V2V TECHNOLOGY FOR APPLICATION"," United States Department of Transportation, Tech. Rep., 2014.

[26] S. Yousefi, M. S. Mousavi, and M. Fathy, "Vehicular ad hoc networks (vanets): Challenges and perspectives," in *2006 6th International Conference on ITS Telecommunications*, June 2006, pp. 761–766.

[27] "OpenStreetMap." [Online]. Available: http://www.openstreetmap.org/

[28] C. Sommer, R. German, and F. Dressler, "Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis," *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, pp. 3–15, January 2011.

[29] B. Tian, "Data dissemination protocols and mobility model for VANETs," phdthesis, Blaise Pascal - Clermont-Ferrand II, 2017. [Online]. Available: https://tel.archives-ouvertes.fr/tel-01469693

[30] 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*. ACM, 2004, pp. 19–28.

[31] M. Torrent-Moreno, P. Santi, and H. Hartenstein, "Fair sharing of bandwidth in vanets," in *Proceedings of the 2nd ACM international workshop on Vehicular ad hoc networks*. ACM, 2005, pp. 49–58.

[32] J. Sanguesa, M. Fogue, P. Garrido, F. J. Martinez, J.-C. Cano, C. T. Calafate, and P. Manzoni, "On the selection of optimal broadcast schemes in VANETs,"

in *Proceedings of the 16th ACM international conference on Modeling, analysis & simulation of wireless and mobile systems - MSWiM '13*. ACM Press, 2013.

[33] M. Chitra and S. S. Sathya, "Efficient broadcasting mechanisms for data dissemination in vehicular ad hoc networks," *International Journal of Mobile Network Communications & Telematics (IJMNCT)*, vol. 3, no. 3, pp. 47–63, 2013.

[34] J. A. Sanguesa, M. Fogue, P. Garrido, F. J. Martinez, J.-C. Cano, and C. T. Calafate, "Topology-based broadcast schemes for urban scenarios targeting adverse density conditions," in *Wireless Communications and Networking Conference (WCNC), 2014 IEEE*. IEEE, 2014, pp. 2528–2533.

[35] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins, *Global positioning system: theory and practice*. Springer Science & Business Media, 2012.

[36] S. M. Abuelenin and A. Y. Abul-Magd, "Empirical study of traffic velocity distribution and its effect on vanets connectivity," in *2014 International Conference on Connected Vehicles and Expo (ICCVE)*, Nov 2014, pp. 391–395.

[37] A. Boukerche, C. Rezende, and R. W. Pazzi, "Improving neighbor localization in vehicular ad hoc networks to avoid overhead from periodic messages," in *GLOBECOM 2009 - 2009 IEEE Global Telecommunications Conference*, Nov 2009, pp. 1–6.

[38] 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*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008, p. 60.

[39] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "SUMO–simulation of urban mobility: an overview," in *Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation*. ThinkMind, 2011.

[40] S. Krauß, P. Wagner, and C. Gawron, "Metastable states in a microscopic model of traffic flow," *Physical Review E*, vol. 55, no. 5, p. 5597, 1997.

[41] "Veins - download." [Online]. Available: http://veins.car2x.org/download/

[42] "Omnet++ discrete event simulator - download." [Online]. Available: https://omnetpp.org/omnetpp

[43] “Download - sumo.” [Online]. Available: http://sumo.dlr.de/wiki/Downloads

[44] “sumo-launchd.” [Online]. Available: http://veins.car2x.org/documentation/sumo-launchd/