Sirindhorn International Institute of Technology

Thammasat University

Thesis EC-MS-2012-02

IDTSG: TIME-STABLE GEOCAST FOR POST CRASH NOTIFICATION IN VEHICULAR HIGHWAY NETWORKS

Phuchong Kheawchaoom

IDTSG: TIME-STABLE GEOCAST FOR POST CRASH NOTIFICATION IN VEHICULAR HIGHWAY NETWORKS

A Thesis Presented

by

Phuchong Kheawchaoom

Master of Science
Electronics and Communications Engineering Program
Sirindhorn International Institute of Technology

Thammasat University
April 11, 2013

IDTSG: TIME-STABLE GEOCAST FOR POST CRASH NOTIFICATION IN VEHICULAR HIGHWAY NETWORKS

A Thesis Presented

By

Phuchong Kheawchaoom

Submitted to
Sirindhorn International Institute of Technology
Thammasat University
In partial fulfillment of the requirement for the degree of

MASTER OF SCIENCE IN ENGINEERING

Approved as to style and content by

Advisor and Chairperson of Thesis Committee	Asst. Prof. Somsak Kittipiyakul, Ph.D.
Committee Member and Chairperson of Examination Committee	Assoc. Prof. Steven Gordon, Ph.D.
Committee Member	Dr. Kannikar Siriwong Na Ayutaya

External Examiner : Asst. Prof. Kultida Rojviboonchai, Ph.D.

April 11, 2013

Acknowledgement

This thesis is partially supported by Thailand Graduate Institute of Science and Technology (TGIST) under National Science and Technology Development Agency (NSTDA), contract no. TG-44-13-53-064M.

I express my gratitude to Dr. Somsak for supervising me. When I started the graduate program at SIIT, research in an area of computer network, especially in VANETs, was completely new to me. Dr. Somsak helped me to overcome the initial hurdles and guided me patiently. He also checked and corrected the fault of all publications including this thesis.

In addition, I am grateful for Asst. Prof. Dr. Steven Gordon and Dr. Kannikar suggestions and all their help.

Finally, my graduation would not be achieved without best wishes and supports from my farther and my mother who help me everything and always give me greatest love, willpower, and financial support.

Abstract

iDTSG: Time-Stable Geocast for Post Crash Notification in Vehicular Highway Networks

March 2013

by

Phuchong Kheawchaoom

B.Eng, Sirindhorn International Institute of Technology
Thammasat University, 2010

We study broadcasting of emergency messages in highways. A warning message must be disseminated to notify an accident to all incoming vehicles into a specific area. All the cars in the area must be aware of the emergency message for a certain time until the incident is taken care of. Such emergency messages help to reduce further accidents by informing the incident to other drivers in time. This is especially useful when the cars are moving at high speed and in low visibility environment such as at night or during rain. With fast-moving vehicles, the network topology changes rapidly and sometimes cars are too far apart from each other and result in partitioned networks. However, with slow-moving vehicles, the network may become too dense with many cars closed to each other. We propose a time-stable geocast protocol, called iDTSG, which uses vehicular ad-hoc networks and works in both sparse and dense car traffic. The protocol assumes each car is equipped with a GPS but does not require any knowledge of the neighbors. In sparse traffic scenario, the protocol uses opposite-direction vehicles to help relaying the message and hence connects the otherwise partitioned network. In dense scenarios, to avoid a broadcast storm problem resulted from a simple flooding scheme, the protocol uses a distributed relay selection mechanism that requires no coordination among the neighboring cars. The mechanism is based on a distance-based defer time. We develop our protocol from an existing protocol but we introduce several improvements. In addition, to better evaluate the performance of the protocol, we perform more realistic simulations such as fading channels and car-movement model. We study the importance of several protocol parameters to the performance of the protocol. The simulations show that iDTSG performs better than the existing protocol in term of a better message delivery while at a smaller number of message rebroadcasts. Furthermore, for the protocol to work reliably in any car traffic density, it must be able to estimate the traffic density and adapts its parameters accordingly. We propose a simple estimation algorithm and show by simulation that the protocol performs well with such algorithm.

Contents

Chapter **Title** **Page**

Signature Page i

Acknowledgement ii

Abstract iii

List of Figures viii

List of Tables xi

1 Introduction 1

1.1 Motivation and Objective 2

1.2 Contribution 2

1.3 Thesis Organization 3

2 Background and Related Work 5

2.1 Vehicular Ad-Hoc Networks (VANETs) 5

2.2 Wireless Access in Vehicular Environment (WAVE) 6

2.3 Broadcast and Dissemination Techniques 9

2.4 Time-Stable Geocast Protocol 11

3 VANET Simulation Tools 14

3.1 Network Simulator (NS-3) 14

3.2 Highway Mobility 17

4 iDTSG: Time-Stable Geocast for Post Crash Notification on Vehicular Ad Hoc Highway Networks 20

4.1 System Assumptions 20
4.2 Protocol Description 21
4.2.1 Direction of Propagation of the Warning Message 23
4.2.2 Broadcast Storm Suppression Part of iDTSG 23
4.2.3 Deterministic Distance-Based Defer Time 24
4.2.4 Store-Carry-Forward 25
4.2.5 Time-Stable Geocast Part of iDTSG 25
4.2.6 Dynamic Length of Extra Region 26
4.3 Simulation Results 28
4.3.1 Performance on Reliability 28
4.3.2 Performance on Efficiency 30

5 Effects of Distance-Based Defer Times and Probabilistic Channel to Time-Stable Geocast 37
5.1 Introduction 37
5.2 Deterministic Distance-Based Defer Times 38
5.2.1 Bound on Minimum Deterministic Defer Times 40
5.3 Stochastic Distance-Based Defer Times 41
5.4 Simulation Results 42
5.4.1 Effect of Transmission Range in Probabilistic Channel 42
5.4.2 Effect of Maximum Defer Time 43
5.4.3 Effect of Varying the Shape of Defer Time 48
5.4.4 Minimum Deterministic Defer Times 49
5.4.5 Effect of Stochastic Defer Time 49

6 Estimation of the Average Inter-Vehicle Spacing 52
6.1 Simple Estimation Mechanism 52

6.2 Simulation Results 53

7 Summary and Future Works 56

7.1 Summary 56

7.2 Recommendations and Future Works 57

Appendices 60

A NS-3 code 60

A.1 Controller class 60

A.2 Vehicle class 67

A.3 Highway class 75

References 77

Publications 83

List of Figures

Figure **Title** **Page**

2.1 Illustration of a WAVE system showing the typical locations of the OBUs and RSUs, the general makeup of the WBSSs, and the way a WBSS can connect to a WAN through a portal. 7

2.2 The WAVE Protocol Stack and Its Associated Standards. 8

2.3 Illustration of the directional propagation communication model between cluster groups of vehicles 11

2.4 Geobroadcast, the sender (S) initiate the multihop dissemination of a message within the area of interest. 12

3.1 Wifi Architecture in ns-3 14

3.2 Packet reception probabilities for the deterministic and probabilistic channels. 17

4.1 Problem model and an illustration of the intended, forwarding, and extra regions. 21

4.2 Illustration of the reception probability in one-hop transmission range based on probabilistic channels 22

4.3 Illustration of the distance of opportunistic contact between two vehicles toward each other 23

4.4 iDTSG protocol flow chart 27

4.5 Loss Ratio for dense networks 29

4.6 Loss Ratio for sparse networks 30

4.7 Over head for dense networks 31

4.8 Over head for sparse networks 32

4.9 Number of cumulative helping vehicles for dense networks 33

4.10 Number of cumulative helping vehicles for sparse networks 34

4.11 The relative overhead of iDTSG with DTSG in dense networks 35

4.12 The relative overhead of iDTSG with DTSG in sparse networks 36

5.1 Different defer times T_D: the inverse defer time $T_{D,I}$ in (4.1), T_D in (5.1) with $\epsilon = 2, 1, 0.5$, and 0.2, the probabilistic-channel defer time $T_{D,opt}$ in (5.2), for $R = 300$ and $T_{DMAX} = R = smax = 8.57$ s. For this plot, we assume the receiving node is moving at the maximum speed $(sr = smax)$ and hence from (4.2) we have $T_N = 2R/2s_{max} = T_{DMAX}$. 39

5.2 Loss ratio and overhead for deterministic and probabilistic channels with $R = 180$m and 300m, under dense scenario 43

5.3 Loss ratio for deterministic with $R = 180$m and probabilistic channels with $R = 300$m, under dense scenario 44

5.4 Overhead for deterministic with $R = 180$m and probabilistic channels with $R = 300$m, under dense scenario 45

5.5 Loss ratio for deterministic with $R = 180$m and probabilistic channels with $R = 300$m, under sparse scenario 46

5.6 Loss rario and Overhead for deterministic with $R = 300$m and probabilistic channels with $R = 300$m, under dense scenario 47

5.7 Performances of varying ϵ and shape of T_D for dense scenario. 48

5.8 Performance of critical deterministic defer time T_D for probabilistic channel and dense scenario 50

5.9 Performance of stochastic defer time T_S and deterministic defer time T_D for probabilistic channel and dense scenario 51

6.1 Simple estimation of the average inter-vehicle spacing 52

6.2 Performance metric for dense networks of iDTSG with estimation spacing. 54

6.3 Performance metric for sparse networks of iDTSG with estimation spacing. 55

7.1 Bounded defer times 58

7.2 Performance of bounded deterministic defer time T_D for probabilistic channel and dense scenario 59

A.1 Structure of the simulator 61

List of Tables

Table **Title** **Page**

3.1 Parameters of IEEE 802.11p 15

3.2 IDM parameters used in simulations 18

Chapter 1

Introduction

In this thesis, we propose and evaluate a time-stable geocast protocol which aims to disseminate a message about an emergency situation on a highway. Drivers should receive the message before they encounter to the incident. The proposed protocol works well in both highly and sparsely density networks. An in-depth study of accidents by Thailand Accident Research Center shows that a main factor of road accidents is the driver behavior. The statistics of the National Police indicates that human behavior is the main cause of road accidents when drivers are driving faster than the rate prescribed or high speed access where this is not appropriate for the situation and variety of road conditions. This is the important factor that increases the likelihood of an accident. If the drivers use high speed driving, they will have a short time to make decisions or respond to things around themselves. From the safe driving speed report [1], accidents often happened in the period from 06:00 p.m. to 01:00 a.m., especially late at night with little traffic and low visibility. During that time, many drivers drove with high speed, which increased the opportunity of crashing and reduces the available time needed to avoid a crash. The traffic accidents could be significantly avoided by informing the drivers about the incident or emergency event so the drivers can make a proper reaction on time.

Many researchers have been interested in development vehicular communication systems since IEEE Task Group has recently approved an IEEE 802.11p amendment to enable efficient short range communication for vehicular networks [2]. The IEEE 802.11p is designed for MAC and PHY layers to achieve low latency communication in short coverage areas. A collection of wireless vehicle nodes dynamically forming a temporary network is called Vehicular Ad Hoc Networks (VANETs). A conception of ad hoc networks is incorporating of wireless communication systems and data sharing capabilities. In VANETs, a group of vehicles forms a network providing similar services like mobile ad hoc networks (MANETs). VANETs are also similar to MANETs in the ways of multi-hop mobile networks (e.g., dynamic topology, no central entity, and nodes route data themselves across the network); however, VANETs differ from MANETs in many features. The differences are highly dynamic topology, frequently disconnected networks, sufficient energy and storage, mobility modeling and prediction, various communications environment, and hard delay constraints [3]. Due to the distinct features of the VANETs, different problems have interested many researchers (e.g., mobility modeling, routing protocol, MAC protocol, and its applications).

Conventional ad hoc routing protocols for MANETs (e.g., AODV and DSR) are not suitable for VANETs [4]. Wireless communication among a vehicle-to-vehicle (V2V) or a vehicle-to-infrastructure (V2I) enhance driver's safety and infotainment in the not-too-distant future.

1.1 Motivation and Objective

In data dissemination, broadcasting is common operation to disseminate such emergency information in VANETs. A simple method to disseminate the information is a flooding scheme. However, the flooding scheme results in serious redundant rebroadcasts, contention, and collision problems. These problems refer to a broadcast storm problem [5]. The broadcast storm problem seriously occurs in highly density networks. The broadcasting in VANETs is unreliable in disconnected networks, especially in sparsely density networks during off-peak hours and/or during initial deployment, due to high mobility caused by fast moving vehicles, lacking of acknowledgment mechanisms, and lacking of the typical Request to Sender/Clear to Sender (RTS/CTS) message. Hence, in this thesis, we focus on a design and an evaluation of **Time-Stable Geocast** protocol which disseminates and keeps an alarm message to support post crash notification within a specific area, for a time duration. This protocol is expected to be scalable in highly density networks and to be reliable in sparsely density networks.

1.2 Contribution

In this thesis, a novel time-stable geocast protocol is proposed for the post crash notification in vehicular highway networks. This protocol is called *Improved Dynamic Time Stable Geocast* (iDTSG) which is an improvement of the proposed DTSG protocol in [6]. The aim of iDTSG is to maximize the number of informed nodes in the specific area with a fast and a reliable transmission while network loads are minimized. The proposed protocol compares with DTSG in a ns-3 simulation. The iDTSG protocol improves upon DTSG in several significant ways:

- iDTSG suppresses the broadcast storm further than DTSG at least 20% and 30% in dense density connected networks and sparse density connected networks, respectively.

- iDTSG modifies the length of the extra region (defined later in Chapter 4) which is used in the DTSG. This length is not fixed and depends on the estimate of the vehicle density, while the DTSG assumes that the vehicle density is known a priori.

- Compared to the simulation in [6], the performances of the proposed protocol and the DTSG are simulated in a more realistic car movement model and a more realistic channel model which includes the possibility of packet loss. These more realistic models are expected to provide a better assessment of the performance of the protocols.

• Furthermore the effects of the three design parameters (T_{DMAX}, ϵ, and R) in the popular defer time given in [7], [8], [9], [10], and [11] are studied in both deterministic channel and probabilistic (i.e., fading) channel.

• For deterministic defer times in bi-directional single-lane highway, we analyze and show that there is an optimal parameter design to avoid packet collisions and achieve the best message dissemination time.

• We show that the stochastic defer times can give a better performance, comparing to the deterministic ones. The reason is that the randomness in the defer times introduces a possibility of a closer receiver to rebroadcast the received packet sooner than another further receiver which may not receive the packet due to signal blocking by other vehicles. This behavior should be included when selecting the defer time function.

• Finally, for the protocol to work reliably in any car traffic density, it must be able to estimate the traffic density and adapts its parameters accordingly. We propose a simple estimation algorithm and show by simulation that the protocol performs well with such algorithm.

1.3 Thesis Organization

The structure of this thesis is organized as follows:

• In **Chapter 2**, we present an overview of VANETs. We describe many unique challenges and supported technologies in VANETs. We provide existing broadcast storm suppression mechanisms and dissemination techniques. At the end, we present an existing time-stable geocast protocol.

• In **Chapter 3**, we give the method to set up VANETs simulation on highway with realistic mobility model and probabilistic channel assumption in ns-3 simulator.

• In **Chapter 4** the main topic of this thesis, We describe the iDTSG protocol and make performances comparison.

• In **Chapter 5**, we show the effects of distance-based defer times and probabilistic channel to time-stable geocast. The effects of different defer time shapes and parameters are shown via simulations.

• In **Chapter 6**, we discuss a method to estimate the average inter-vehicle spacing and gives the performance of this mechanism with iDTSG.

• In **Chapter 7**, we give a conclusion of the whole work and possible future research.

• In **Appendix A**, we show the modified codes of highway mobility in ns-3 network simulator which is used in this thesis.

Chapter 2

Background and Related Work

2.1 Vehicular Ad-Hoc Networks (VANETs)

Vehicular Ad-Hoc Networks (VANETs) share a common characteristic with Mobile Ad-Hoc Networks (MANETs) since both are self organized networks and both are lack of centralized control. An idea to develop WLAN communicating among vehicles is to enhance Intelligent Transportation Systems (ITS). VANETs do not necessarily rely on a pure ad hoc networking environment: they may use fixed cellular gateways, WLAN access point, and any available infrastructures for Internet connections. In general, VANETs consist of two networking architectures which are vehicle-to-vehicle (V2V) communication and vehicle-to-infrastructure (V2I) communication. In [3] and [12], the unique challenges of VANETs, that impact on designing of protocols and its security systems, are given as follows:

- **Potentially high number of nodes.** An increasing number of ITS users will be equipped with wireless communication capabilities to participate in their networks. A protocol in those vehicles has to be scalable to avoid high congestion.

- **High mobility and Frequently disconnected network.** Due to a restriction of road pattern, a bi-directional traffic makes high relative mobility between inter-directional vehicles. The vehicles have around 10 seconds to communicate to each other if their speeds are 25 m/s with wireless transmission range of 250 m. In the same direction, the vehicles are likely to move into cluster groups which may cause partitioning networks if the edge of both groups are too far apart.

- **Various communication environment.** In wireless communication networks, a packet reception reliability is affected by surrounding environments. A radio signal does not always have a line-of-sight especially in city. It is often blocked by buildings, trees, and other obstacles. Even in a highway environment, some vehicles like trucks and big cars can obstruct and attenuate a radio signal [13]. These various communication environments directly affect a data link layer and a MAC layer.

- **Privacy and Security.** A vehicle reveals an information about driver's identity, e.g., speed, position, mobility pattern, and destination. This information can be abused by any bad persons. In some cases, the false information may be disseminated in safety application for any bad objectives.

In general, applications in VANETs can be categorized into safety applications and non-safety applications. Some safety applications of VANETs are surveyed in [14] and [15], where safety applications are ranged from low danger to high danger, e.g., approaching emergency vehicle warning, curve speed waring, work zone warning, pre-crash sensing, and post crash notification. These applications are highly desirable in VANETs due to improvement of public road safety. In this applications, drivers are warned by emergency warning messages about vehicles acting out of control due to an accident, mechanical breakdown, or some other failure. We focus on this type of applications, especially in post crash notification.

Such safety applications in VANETs require a broadcast protocol which must have low implementations and low operation cost. The protocol must be self-organized and propagate an alarm message within a small delay. Objectives of post crash notification are to notify vehicles in an approaching area for avoiding the incident and traffic jams. Hence, the alarm message has to be kept in the approaching area until the highway is free. This application must have low latency, high reliability, high scaling, and well-defined scope of receivers. Furthermore, the protocol should be designed to balance trade-off between reliability, message dissemination time, and efficiency.

2.2 Wireless Access in Vehicular Environment (WAVE)

Due to the fact that Wi-Fi systems and cellular systems are designed to operate in well controlled environments. IEEE task group p and IEEE 1609 proposed Wireless Access in Vehicular Environments (WAVE) standards to use in VANETs because VANETs have many applications with various constraints (e.g., extreme Doppler shift, multi-path problems, rapidly changing conditions, real-time exchanging data, and other requirements). In the WAVE standards, there are two units called roadside units (RSUs) and onboard units (OBUs) as shown in Fig. 2.1. The RSUs are usually installed in infrastructures, for example, light poles, traffic lights road signs, and so on. The RSUs might transport to many locations but they cannot work in transit. The OBUs are equipped in vehicles and can operate in moving. A WAVE basic service set (WBSS) is small networks which similarly operate in IEEE 802.11. In particular, the WBSS might consists of only OBUs (vehicle-to-vehicle) or both OBUs and RSUs (vehicle-to-infrastructure). Furthermore, the WBSS is able to connect wide-area networks through the appropriate portals shown in Fig. 2.1 [16].

In open systems interconnection (OSI) models, the WAVE PHY and MAC layers depend on IEEE 802.11p [2] which is an amendment to IEEE802.11-2007 [17] for WAVE applications [18]. The Federal Communications Commission (FCC) allocated 75 MHz of bandwidth for 5.85-5.925 GHz to support Dedicated Short-Range Communication (DSRC) systems in ITS since 1997. IEEE 802.11p divides bandwidths of the DSRC into seven 10-MHz channels and one 5-MHz channel for reserving as the guard band. The seven channels

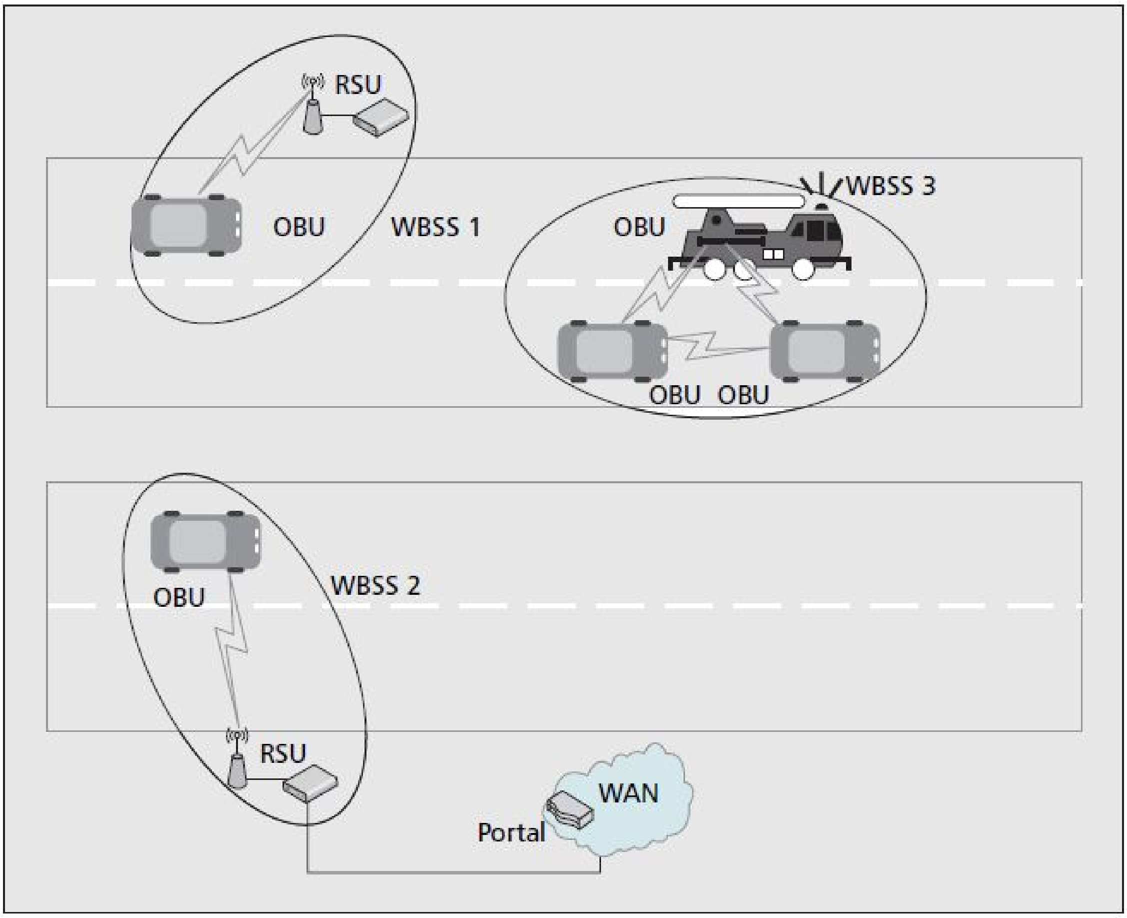


Figure 2.1 Illustration of a WAVE system showing the typical locations of the OBUs and RSUs, the general makeup of the WBSSs, and the way a WBSS can connect to a WAN through a portal.

are configured into 1 control channel (CCH) and 6 service channels (SCHs). The sub-carrier spacing and the supported data rate of OFDM PHY IEEE 802.11p are halved while its symbol interval including cyclic prefix is doubled. The minimum and maximum transmission range vary from 10 meters and 1000 meters. The data rate supports to 6 Mbps - 27 Mbps. To support multichannel operations, IEEE 1609.4 uses a concept of frequency/time division multiple access (FDMA/TDMA). The repetitive periods of 100 ms of the TDMA channel are allocated into 4 ms of guarding interval, 46 ms of CCH, 4 ms of guarding interval, and 46 ms of SCH, respectively. The WAVE systems can either transmit or receive on the CCH and one of six SCH but not simultaneously. A short message for safety applications can be sent in the CCH. A WAVE short message protocol (WSMP) and a WAVE service advertisement (WSA) announce available services on other SCHs.

Unlike traditional wireless LAN stations, the transmission control protocol/user datagram protocol (TCP/UDP) transactions and the WAVE short-message protocol (WSMP) use the Internet protocol version six which accommodates both non-safety and safety applica-

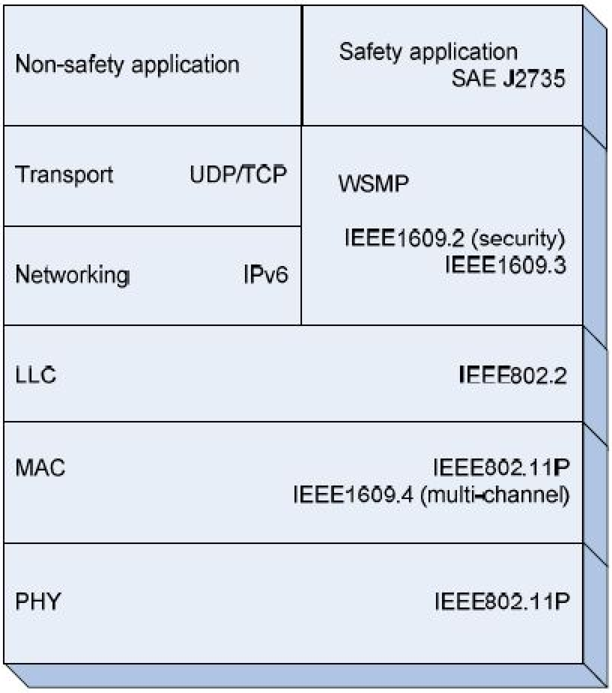


Figure 2.2 The WAVE Protocol Stack and Its Associated Standards.

tions.

As described in [18] and [19],they summarized the following WAVE protocol architectures with its major components as shown in Fig. 2.2:

- IEEE 1609.1 “Trial Use Standard for Wireless Access in Vehicular Environments (WAVE) - Resource Manager.” This standard specifies the services and the interfaces for a WAVE Resource Manager application.

- IEEE 1609.2 “Trial Use Standard for Wireless Access in Vehicular Environments (WAVE) - Security Services for Applications and Management Messages.” This standard defines secure message formats and the circumstances for secure message exchange.

- IEEE 1609.3 “Trial Use Standard for Wireless Access in Vehicular Environments (WAVE) - Networking Services.” This standard defines network service and transport layer services.

- IEEE 1609.4 “Trial Use Standard for Wireless Access in Vehicular Environments (WAVE) - Multi-Channel Operations.” This standard describes enhancements to the IEEE 802.11 MAC layer for multi-channel operations.

- IEEE P1609.11 “Over-the-Air Data Exchange Protocol for Intelligent Transportation Systems (ITS).”

2.3 Broadcast and Dissemination Techniques

For many safety applications, the time to access data through wireless infrastructures or roadside units is too high, comparing to that through ad-hoc networks. Furthermore, wireless infrastructures may be damaged in the event of a disaster; however, vehicle-to-vehicle vehicular ad hoc networks (V2V) can be formed dynamically. A simple dissemination mechanism for V2V is a simple flooding protocol. In the flooding protocol, a node rebroadcasts every received packets which have not been transmitted before. Since each packet contains a unique sequence number, the node drops the packet when it receives the same sequence packet. Several nodes may rebroadcast at the same time and cause collisions, which can immensely high for dense networks. Since the typical request to sender/clear (RTS/CTS) message and an acknowledgment (ACK) mechanism are deactivated in the broadcasting mode. This is known as a broadcast storm problem. While the network density may be very high in urban areas and during rush hours, it is low in rural areas and at night time. Furthermore, VANETs are highly mobile and mostly partitioned into many clusters. The gaps between each cluster prevent a communication path between a source and a destination if these gaps are larger than the transmission range of the vehicles. Although the flooding protocol is highly inefficient in dense networks scenario, the flooding ensures the reliability of broadcast packet. However, the flooding protocol is still unreliable in sparsely connected networks scenario. Hence, many researchers aim to design data dissemination protocols which are scalable, reliable, and efficient for safety applications in VANETs. Due to the dynamic characteristic of the channel in VANETs, to know information about a nodes neighbor requires high overhead especially for networks with high speed nodes. Consequently a neighbor-based scheme, where nodes require neighbor topology information to decide whether to rebroadcast a message, is inadequate in ad hoc networks due to a difficulty of maintaining a neighboring list, a limited bandwidth, and high link failures [4].

In summary, several broadcast storm suppression mechanisms for information dissemination are classified as follows [20]:

• **Probability-based scheme.** Each node makes a decision whether it will forward the packet or not, depending on a probability p after a random back-off period. The node drops the packet with a probability $1 - p$. Note that when $p = 1$, the scheme is equivalent to the simple flooding protocol. The probability-based schemes are simple methods for dense environments broadcasting to mitigate broadcast storms. However, the configuration of parameters for the probabilistic decisions in each node is difficult to achieve when the network configuration dynamically changes.

• **Counter-based scheme.** There is a counter to count a duplicate message in each node. The counter increases when the node overhears any duplicated messages that are forwarded by its neighbor nodes. When each node receives a non-duplicate message, the node initiates a random timer which is decremented afterwards. The node rebroadcasts the received packet if the counter does not exceed a threshold called Max-count when the timer expires. Depending on the sufficient threshold, this scheme works well even in partition networks and dynamic networks. This scheme takes into account the network dynamics in making a decision on the forwarding then the node suppresses the forwarding by silently discarding the packet [10]. Hence the counter-based scheme is more robust in various network-wide broadcasting scenarios, However, it is not optimal in the network efficiency side because of the incomplete elimination of the packet redundancy.

- **Location-based scheme.** The principle of location-based scheme uses its location information and transmission range for deciding to rebroadcast. When a node receives a non-duplicate message, the additional area that can be covered by the transmission range of each node is calculated. If the received node has the maximum additional area that exceeds a predetermined threshold, it forwards the packet otherwise, it drops the packet. For an omnidirectional antenna, the closest node to the destination node is chosen to be the forwarder. This method reduces unnecessary forwarding by choosing the furthest forwarder. However, this method has several shortcomings. One of them is that the neighbor nodes, which have the same threshold, will rebroadcast as they are able to cover the additional area. In very dense networks, this leads to the rebroadcast of hundreds of redundant packets because each node have no global knowledge in which nodes act as the forwarder. Another shortcoming is, due to sparsity of network, the next hop forwarder does not alway present. Thus, both their number and location of the forwarders are the key factors in designing the efficient and reliable location-based protocol.

- **Distance-based scheme.** The distance-based scheme is known as the contention-based forwarding (CBF) [21]. In this scheme, a received node starts a defer time after hearing the first message. The defer time is inversely proportional to a relative distance between itself and a sender. During the defer time, if a received node does not receive any duplicate messages, it rebroadcasts the received message. Due to the fact that further nodes wait shorter than closer nodes to the sender. Hence, the furthest node from the sender rebroadcasts first because its defer time is the shortest. This scheme achieves the lowest message propagation delay due to using minimum hops. CBF requires no information about neighbor positions and beacon exchanging. Since CBF selects nodes for multi-hop relays based on a distance-based defer time, it is suitable for broadcast storm suppression in highly mobile VANETs. However, it suffers from network partitioning problem in sparse networks because the next hop does not always present. The authors in [22] and [23] analyzed the effect of probabilistic channels on CBF in vehicular ad hoc networks. They stated that CBF is beneficial in probabilistic channel, but efficient suppression mechanisms are required to enhance CBF in probabilistic channels. In addition, the node selection for multi-hop relays suffers from a reliability trade-off in probabilistic channels [24].

In sparse density networks, store-carry-forward and directional protocols are employed to deliver messages whenever the multi-hop connectivity among vehicles is not available. As

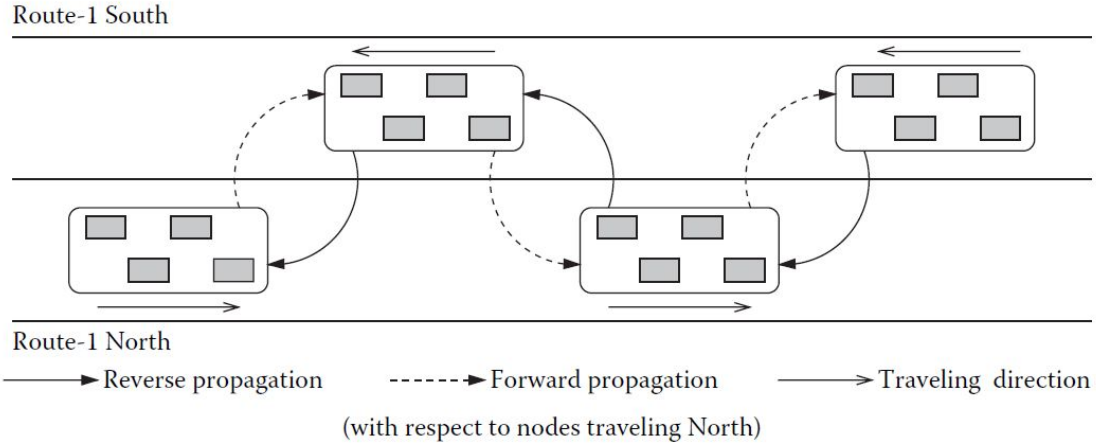


Figure 2.3 Illustration of the directional propagation communication model between cluster groups of vehicles

illustrated in Fig. 2.3 [25], a message have to be propagated to the north direction but the gap between the cluster groups in the north direction is greater than the communication range of each vehicle. The end-to-end path does not accomplish between cluster groups by relying only on the vehicles on the same direction. Assuming the bi-directional road, the message can be disseminated between the cluster groups. Each cluster has a header vehicle and a tailer vehicle which locates at the front and rear of each cluster, respectively. The duty of the header and the tailer is to repeatedly broadcast every constant intervals until the vehicle hears the same message, which is considered an implicit acknowledgment, from the vehicle at the message direction whether it is on the same direction with respect to the received vehicle or not. Helping from the inter-directional vehicle, the message disseminates between the cluster groups. If the message is headed in the direction opposite the direction of the vehicles, the message propagation is called *reverse propagation*. If the message is headed along the direction of the vehicles, the message propagation is called *forward propagation*. The minimum rate of message propagation is the speed of vehicles because the message is traveling along with the vehicles. This mechanism is also called *opportunistic forwarding* because the temporary path occurs due to opportunistic contact with nodes in overlapping clusters. The authors in [26] showed that the opportunistic forwarding in a bi-directional highway successfully delivers a message to target vehicles. The message can be temporarily stored at nodes while the message is waiting for further opportunities to be forwarded.

2.4 Time-Stable Geocast Protocol

With availability of the geographical service or GPS, data dissemination in a specific area (i.e., informing an approaching vehicle about a sudden event or an abnormal road condition) is called geographic broadcast or geocast. As shown in Fig. 2.4 [27], the sender (S)

determines a specific area where only vehicles in the specific area rebroadcast the received message. The position of the vehicles is a factor for deciding to broadcast the message. A broken down vehicle,which marked red on the left side of the figure, starts broadcasting an alarm message to inform about the hazardous situation. The message is disseminated in multiple hops communication to all vehicles within the specific area. For better scalability in a situation with highly density networks, several forwarding schemes as mentioned earlier may also be applied to optimize network loads for reducing the redundancy broadcast. Typically, geobroadcast messages are sent upon a certain external event and need a very low latency of messages to inform vehicles as fast as possible. For example, in case of an accident, the crashed vehicle detects a situation by using its sensors then it sends an appropriate warning message. The inter-vehicle geocast (IVG) [9] protocol is a geocast protocol which uses a concept of the CBF. The role-based multicast protocol [7] uses the similar concept of [9], but the criteria to rebroadcast is different. Distributed Robust Geocast (DRG) is another geocast protocol for 2-dimensional roads. It uses distance-based defer times to select the furthest node as relay. Furthermore, it uses a periodic retransmission to overcome the network fragmentation with implicit acknowledgment [28]. In [29], the authors proposed a light-weight geocast technique in IEEE802.11p. It uses the criteria similar to that in the DRG but it uses a different distance-based defer time and considers only one-dimensional road.

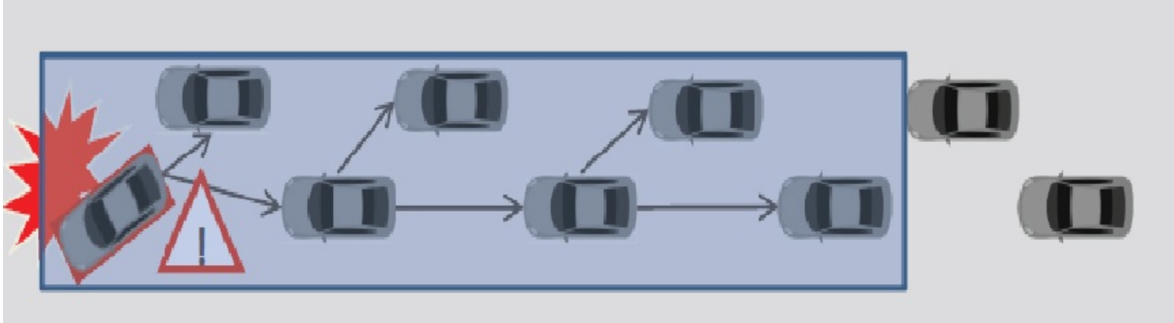


Figure 2.4 Geobroadcast, the sender (S) initiate the multihop dissemination of a message within the area of interest.

However, geocast protocols do not consider how a broadcast message is kept in a specific area for a certain time. This is an important requirement to disseminate an emergency message for post crash notification in vehicular highway networks to prevent chain-collisions or congested traffics until the highway is free. This requirement results to **time-stable geocast** protocols which were first introduced in [30]. The authors in [30] proposed a design space of abiding geocast in three approaches. The first approach is the server approach to store a message within a specific area and a specific time with a periodic rebroadcast from the a server. The second approach is the election approach that selects nodes inside the geocast area to act as a server. The server hands over a role to a suitable node when it leaves from the geocast area. The third approach is the neighbor approach where all nodes inside the geocast area can be a forwarder when a new node enters to the geocast area.

In [31], the authors ensure that a warning message disseminates to all affected vehicles in highway with high reliability and low overhead. The protocol uses the improved predicting interval of vehicles from [32]. It uses the CBF with a distance-based defer time to suppress broadcast storm problems. It periodically rebroadcasts the message to counter with a network fragmentation problems in vehicular ad hoc networks, similarly to [9] and [29]. The dynamic time-stable geocast (DTSG) protocol [6] is a time-stable geocast protocol which our proposed protocol is based on. The DTSG protocol is divided into two periods: pre-stable and stable periods. During the pre-stable period, all vehicles use the CBF to disseminate the message to cover a specific area. When a helping vehicle moves to an extra-region, the helping vehicle rebroadcast again to start the stable period. In the stable period, the message is kept within the specific area, for a time duration via a diagonal dissemination in an extra-region. In DTSG, the defer time is

$$T_D = T_N \times \frac{R}{d}, \tag{2.1}$$

where

$$T_N = \frac{2R}{s_r + s_m}. \tag{2.2}$$

This defer time is called a dynamic sleep time (T_D) because it depends on a speed of a receiver and a distance between a sender and a receiver. This dynamic sleep time is used to suppress broadcast storm problems. To counter the network partitioning problem, a header of each cluster is selected to carry the received message and rebroadcast every T_N intervals in (2.2). This interval T_N considers that two vehicles moving toward to each other is in each other's transmission range. The worst scenario is that a vehicle comes toward a forwarder with a maximum allowance speed, say s_m m/s. Hence, the interval T_N equals to R/s_m to guarantee that the message is successfully delivered to the coming vehicle. The header stops rebroadcasting when it receives three duplicate messages from inter-directional vehicles in the pre-stable period. The DTSG protocol guarantees the message delivery with a low receiving cost without using any roadside units in sparsely density networks. However, the dissemination time is too long and not suitable for the emergency dissemination even in highly density networks.

Chapter 3

VANET Simulation Tools

3.1 Network Simulator (NS-3)

In this section, we describe the network simulator which is used in this thesis. The Network Simulator-3 or ns-3 [33] is used as a discrete event network simulator for the wireless networks in this thesis. When this thesis began, the available version was ns-3.12 which has been used to perform the simulations. We assume that all vehicles are embedded with a wifi device which communicates via the wifi architecture shown in Fig. 3.1

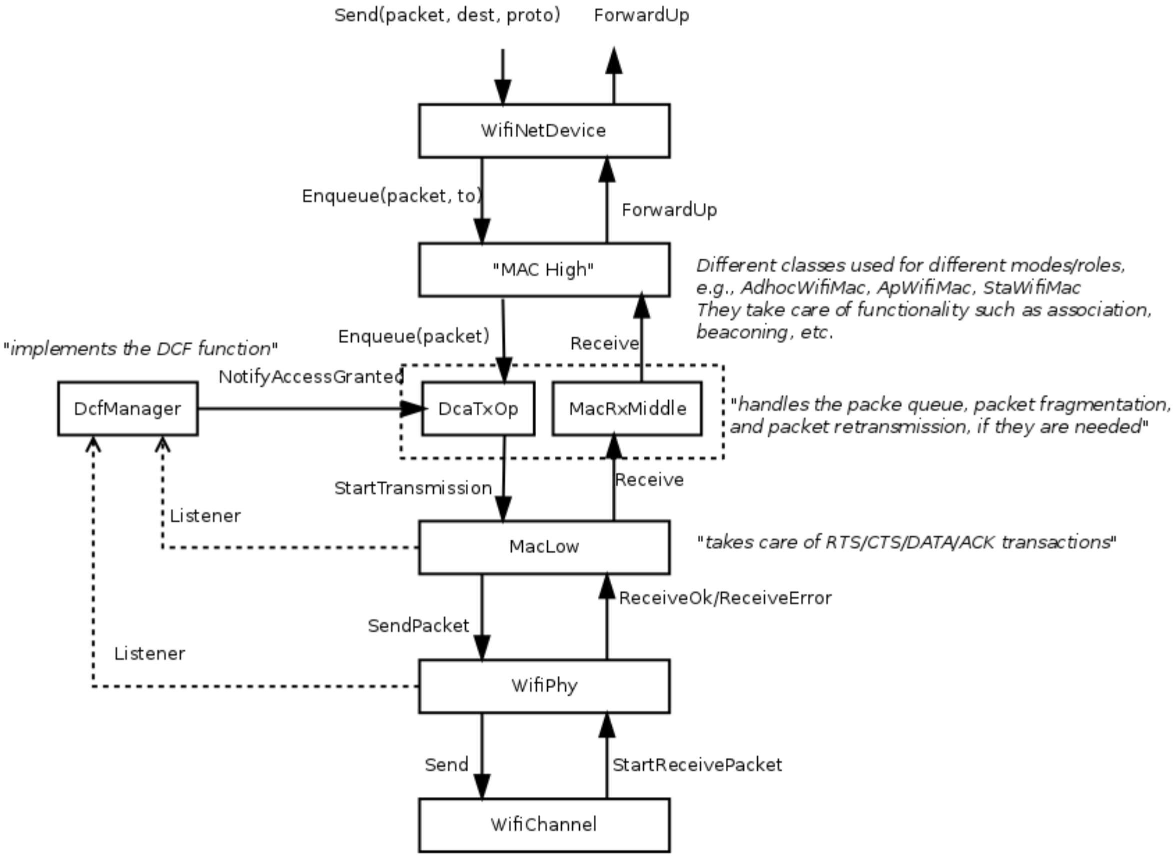


Figure 3.1 Wifi Architecture in ns-3

From Table 3.1, we select a Non-QoS, Wi-Fi MAC high layer which does not perform any kind of beacon generation, probing, or association. This state machine is implemented by the ns3::AdhocWifiMac class. In the Non-Qos, the DcaTxop handles the packet queue, packet fragmentation, and packet retransmissions if needed. It is used for transmission of frames (e.g., management frame) that the node should access the medium using the distributed coordination function (DCF). There are also several rate control algorithms that can be used by the MAC lower layer. The ConstantRateWifiManager is used in this thesis. The WifiPhy layer is described in details in [34]. We use the control channel on IEEE802.11p to perform our protocol. The parameters of IEEE802.11p are shown in Table 3.1.

Table 3.1 Parameters of IEEE 802.11p

Parameter	Value
Standard	IEEE802.11p CCH
Transmit power	5 dBm
Receiver threshold	-87 dBm
Antenna gain	0 dBi
Receiver's Noise Figure	7 dBm
MAC	Non-QoS
OFDM Rate	6 Mbps
Bandwidth	10 MHz

In simulation, a common approach to decide whether the packet is received or not is based on the received power (P_r) of packet: if P_r is greater than the receivers sensitivity threshold (P_{th}), the packet is received. The received power of packet is calculated from the transmission power of packet (P_t) and the channel loss. The channel loss depends on the physical channel model. For wireless communication, there are two popular physical channel models: deterministic and stochastic model. In the deterministic channel model, the packet reception probability is 1 within the transmission range. A famous model of the deterministic channels was established by H.T. Friis [35] in 1946. It is called Friis model. The Friis model is used to predict the received signal strength when transmitter and receiver have a clear, unobstructed line-of-sight path between them so the P_r attenuates quadratically with the distance as follows:

$$P_r(d) = \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d^2 L} \tag{3.1}$$

where G_t and G_r are the antenna gains at transmitter and receiver, respectively, d is the distance between transmitter and receiver while λ is the wavelength corresponding to the carrier frequency, L is the system loss factor not related to propagation loss (when $L = 1$ indicates no loss in the system hardware).

For the stochastic channel models, a popular model is the Nakagami-m distribution [36] which represents fading caused by interference between two or more versions of the transmitted signal. When the receiver receives the two or more versions of the transmitted signal at slightly different times, the combination of those transmitted signals affect the received signal by varying its amplitude and phase. Following the m-Nakagami distribution, the probability density function (PDF) of the received signal power $P_r(x)$ is given as

$$P_r(x; m, \omega) = \frac{2m^m}{\Gamma(m)\omega^m} x^{2m-1} e^{-\frac{m}{\omega}x^2}, \tag{3.2}$$

where m is the fading depth parameter, ω the average received signal power, Γ is the Gamma function. Both m and ω depend on the distance between transmitter and receiver. For the average received signal power ω in VANETs, the authors in [37] recommend to use a log distance path loss propagation model, which calculates the path loss in dB as follows:

$$L = \begin{cases} 0 & d < d_0 \\ L_0 + 10n_0 log_{10}(\frac{d}{d_0}) & d_0 \leq d < d_1 \\ L_0 + 10n_0 log_{10}(\frac{d1}{d_0}) + 10n_1 log_{10}(\frac{d}{d_1}) & d_1 \leq d < d_2 \\ L_0 + 10n_0 log_{10}(\frac{d1}{d_0}) + 10n_1 log_{10}(\frac{d_2}{d_1}) + 10n_2 log_{10}(\frac{d}{d_2}) & d \geq d_2 \end{cases} \tag{3.3}$$

where L is the resulting path loss (dB), d is the distance (m) between transmitter and receiver, L_0 is the path loss at reference distance (dB), and n_0, n_1, and n_2 are the path loss distance exponents for distance d_0, d_1, and d_2, respectively.

For highway environment, the authors in [38] show that the Nakagami-m distribution agrees with empirical propagation data, where $m = 3, 1.5$, and 1 for the transmission range less than 50 meters, between 50 meters and 150 meters and greater than 150 meters, respectively. For the pathloss, the authors in [39] recommend the log-distance pathloss model with the pathloss exponent of 2 and 4, for the range less than 225 meters and between 225 and 1000 meters, respectively. Using ns-3 simulation and the same transmission power of 5 dBm, Fig. 3.2 shows the reception probabilities versus distance for the deterministic channel and the probabilistic channel based on the Nakagami and the log-distance path loss. We denote the reception probability under the probabilistic channel as $P_R(\cdot)$. The result shows that the transmission range in probabilistic channel model is larger than deterministic channel model; however, the reception probability decreases over the distance in probabilistic channel model.

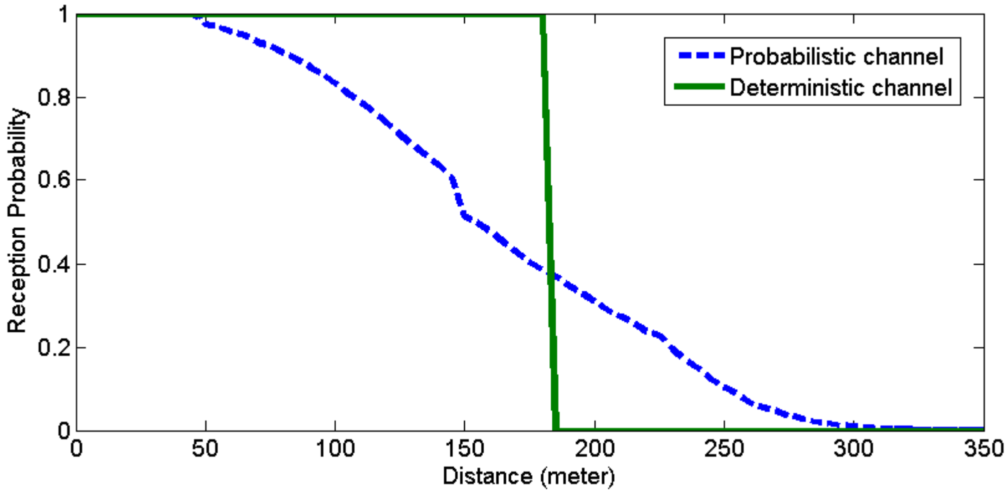


Figure 3.2 Packet reception probabilities for the deterministic and probabilistic channels.

3.2 Highway Mobility

Recently, the authors in [40] proposed an ns-3 highway mobility class for simulation of a realistic vehicle movement in VANETs. The mobility class includes a car-following model based on the Intelligent Driver Model (IDM) [41] and the MOBIL lane-change model [42]. The IDM controls the vehicle by the longitudinal acceleration of vehicle denoted by $\dot{v}_{IDM}(s_\alpha, v_\alpha, \Delta v_\alpha)$ in the following way:

$$\dot{v}_{IDM}(s_\alpha, v_\alpha, \Delta v_\alpha) = a\left[1 - \left(\frac{v_\alpha}{v_0}\right)^4 - \left(\frac{s^*(v_\alpha, \Delta v_\alpha)}{s_\alpha}\right)^2\right], \tag{3.4}$$

$$s^*(v_\alpha, \Delta v_\alpha) = s_0 + v_\alpha T + \frac{v_\alpha \Delta v_\alpha}{2\sqrt{ab}}. \tag{3.5}$$

where s_α: The vehicle spacing between the vehicle α and the ahead vehicle.

s_0: The minimum vehicle spacing that is kept even at a complete stand-still in traffic jam.

v_α: The speed of the vehicle α.

Δv_α: The speed difference between the vehicle α and the ahead vehicle.

v_0: The desired velocity of the driver; the velocity the vehicle would drive at in free traffic.

a: The maximally possible acceleration.

b: The maximum comfortable deceleration.

T: The desired time headway to the ahead vehicle.

$s^*(v_\alpha, \Delta v_\alpha)$: The desired distance to its predecessor.

We apply this model to a bi-directional single lane highway for the realistic vehicle mobility in this highway mobility class. Having the mobility simulator in ns-3, we can simultaneously simulate the vehicular ad-hoc networks and the vehicle mobility with feedback on driver behavior. The authors of [43] shows that the maximum deceleration of IDM is larger than the maximum deceleration of physical limit. Hence, they introduced the following L-IDM equation:

$$a(t) = max(b_{max}, a_{IDM}(t)), \tag{3.6}$$

where $b_{max} < 0$ is the physical maximum deceleration of vehicle and $a_{IDM}(t)$ is the value of acceleration taken from the IDM formula. This modification considers fundamental physical limits hence it gives a more realistic model.

In our simulation, we randomly inject vehicles into the 10-km highway by using random exponential distributions. The mean of the inter-arrival time and the inter-vehicle spacing are taken from the empirical data in [4]. The data was collected from a dual loop detector along the I-80 freeway in southern California during 10 am to 12 pm and 1.00 am to 3.00 am for connected networks and disconnected networks. For connected networks, the data shows that the mean values of the inter-arrival time, the inter-vehicle spacing and the vehicle's speed are 1.37 seconds, 40 meters, and 29.15 m/s, respectively. In disconnected networks or sparse density, the data shows that the mean values of the inter-arrival time, the inter-vehicle spacing and the vehicle's speed are 8.4 seconds, 250 meters, and 30.93 m/s, respectively. The IDM parameters are taken from [43] as shown in Table 3.2.

Table 3.2 IDM parameters used in simulations

Parameter	Value
b_{max}	random{5.9,8.4} m/s^2
a	1.7 m/s
b	-4 m/s^2
T	random{0.1,1.1} s
s_0	4 m

Since emergency notification systems must work in any environment, including any traffic density, here we consider two density scenarios: dense and sparse in probabilistic channel. In the dense scenario, the mean, minimum, and maximum inter-vehicle spacing are 40 m, 7 m, and 150 m, respectively. In the sparse scenario, the mean, minimum and maximum values are 250 m, 80 m, and 600 m, respectively. Note that, from Fig. 3.2, the packet reception probability for the probabilistic channel at 40m and 250m are about 100% and 10%, respectively. This means that in the dense scenario, we would expect several vehicles in the same direction of traffic flow to receive the transmission and in the sparse scenario only one vehicle or none. Hence, in the dense scenario, we have a highly connected network, while in the sparse scenario, a highly disconnected network.

In our simulation, we inject the source vehicle to the 10-km straight highway. After moving for 6.5 km, the source starts broadcasting an alarm message until it receives the same message back from another vehicle. Since there is no any standard to evaluate a protocol for post crash notification, the message in our simulation is required to be within the region D = 3 km or D = 1 km for duration T = 30 minutes in sparse vehicle density networks and T = 10 minutes in high vehicle density networks. The speed limit is s_m = 35 m/s (= 126 km/h) and transmission range parameter R = 300 m. The braking distance is 120 meters. The modified codes are shown in **Appendies A**. For each simulation result, we run 10 independent runs and calculate the average values.

Chapter 4

iDTSG: Time-Stable Geocast for Post Crash Notification on Vehicular Ad Hoc Highway Networks

In this chapter, we describe in details the model of our problem. We also propose a time-stable geocast protocol, called *Improved Dynamic Time Stable Geocast* (iDTSG) [44], for post crash notification in both dense and sparse vehicular highway networks. The iDTSG protocol is an improvement of the DTSG protocol proposed in [6].

4.1 System Assumptions

Consider a portion of a two-way highway with L lanes per direction illustrated in Fig. 4.1 for $L = 3$[1], there is a source vehicle (S), that after having an accident or having encountered an accident, immediately starts broadcasting an alarm message to behind vehicles traveling in the same direction. The alarm message warns about a location of the accident. We assume that all vehicles equip with an omni-directional transceiver and a global positioning system device. We also assume that there is no roadside units available for message dissemination and hence the message must be disseminated only via the vehicular ad hoc network. Due to a variety of network densities, the network densities may range from highly connected network at high densities to disconnected subnetworks at very low densities. As illustrated in Fig. 4.1, we divide all vehicles into intended and helping vehicles except the vehicle S where this model is similar to [6]. We denote this region of D km behind the breaking distance B from the location of the accident as an *intended region*. The *intended vehicles* (I) are the vehicles that are moving toward the accident. They are target recipients of the alarm message. The *helping vehicles* (H) are the vehicles that are moving in the opposite direction on the other lanes, with respect to the source. The helping vehicles can carry the alarm message and disseminate the message to the intended vehicles when the partitioning networks occur. To keep the message within the intended region, we define two additional regions: a forwarding region and an extra region. The intended region and the opposite region in the opposite lane are together called a *forwarding region*. Both ends of the two forwarding regions are extra regions. For simplicity, we assume that there is only one active alarm message to be disseminated. An objective of our protocol is that the alarm

---

[1] For illustration purpose, the figure shows three lanes per direction. In our simulation, we consider, however, one lane per direction.

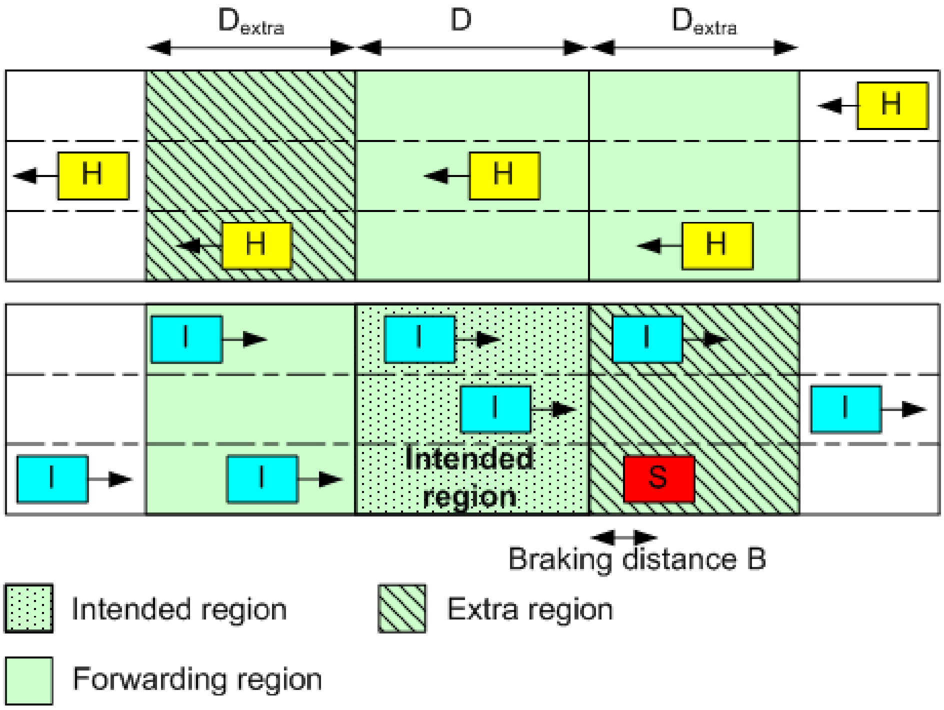


Figure 4.1 Problem model and an illustration of the intended, forwarding, and extra regions.

message has to be kept within the intended region, for a duration of T hours, at minimum transmissions, with minimum dissemination time, and at all vehicular densities.

Each iDTSG message contains a vehicles sender ID or MAC address, a sequence number of each vehicles sender, a location of an event, a date and time of the event, an expected transmission range, an intended region length (D km), an extra region length, an expiry of the message (T hours), a location, a speed, and a direction of a forwarder, and a pre-stable (PS) flag. The source S sends an alarm message only once with the PS flag set to 1 (up). The source stops broadcast the message when it receives the same content message back from any vehicles. Since we need that intended vehicles receive an emergency message and change their lanes to avoid traffic congestions, an intended region has to be large enough to let intended vehicles changing their lane before the bottleneck problem occurs.

4.2 Protocol Description

In probabilistic (i.e, fading and shadowing) channels, an effect of an undesirable shadowing attenuation becomes more significant when a network density becomes dense. Because a transmission range is reduced by the shadowing effect [45]. As illustrated in Fig. 4.2, there is no guarantee that all vehicles between a vehicle $V1$ and a vehicle $V2$ receive a message from the vehicle $V1$ within an expected transmission range. To disseminate a message in a large area within a shortest time, a selecting of a furthest forwarder within one-hop transmission range to broadcast the message is a crucial factor to suppress the broadcast storm problem in highly connected networks.

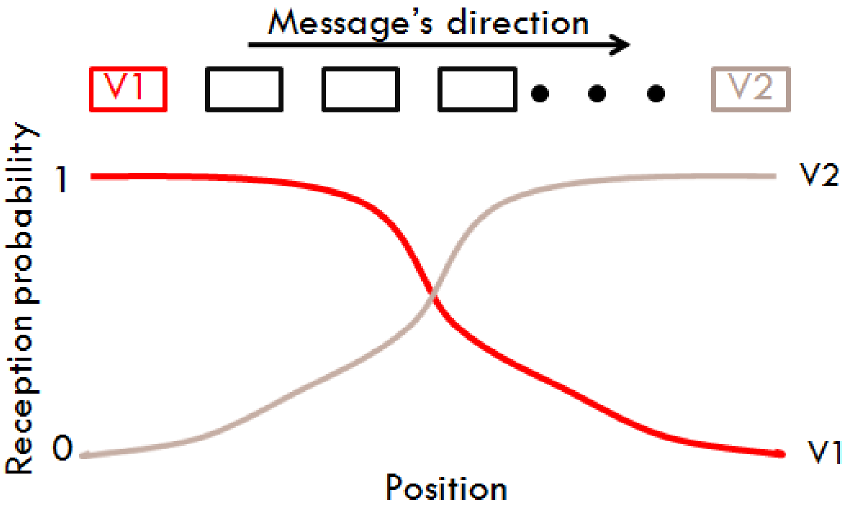


Figure 4.2 Illustration of the reception probability in one-hop transmission range based on probabilistic channels

If all vehicles have a same transceiver characteristic, a distance of opportunistic contact between two vehicles toward each other is twice transmission range ($2R$) shown in Fig. 4.3. A time of two vehicles toward each other is a divided of the $2R$ by a sum of two vehicular speeds. This time is a maximum time of opportunistic contact to receive a message.

To solve the above problems and to satisfy the objective of our protocol, we take the following approaches:

- In dense networks, the iDTSG protocol uses a directional contention-based forwarding (sees Section 4.2.2) to suppress the broadcast storm problem by a message relaying with minimum vehicles.

- In sparse networks, the iDTSG protocol uses a store-carry-forward technique (sees Section 4.2.4) to deliver a message whenever a multi-hop connectivity among nodes is not available. Furthermore, a counter-based scheme uses to stop broadcasting after receiving a certain number of duplicate messages from inter-directional vehicles.

- In time stable geocast, the iDTSG protocol uses a diagonal location-based forwarding (sees Section 4.2.5) to keep a message alive in an intended region by broadcasting in an extra region (sees Section 4.2.6).

The iDTSG protocol is considered as a contention-based forwarding scheme which uses a counter-based scheme and a location-based scheme as a broadcast storm suppression

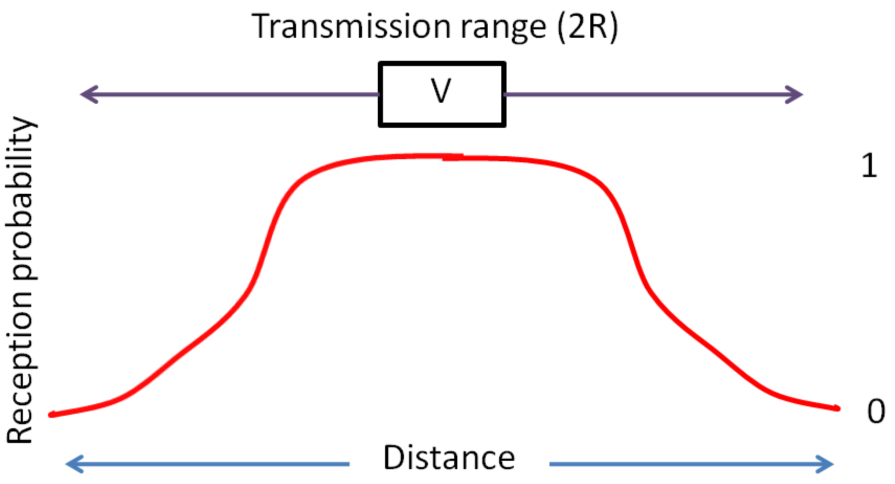


Figure 4.3 Illustration of the distance of opportunistic contact between two vehicles toward each other

mechanism for geocast protocol. It is the location-based scheme because it rebroadcasts a message in different manner depending on where nodes are in a region. It is a counter-based scheme because it stops rebroadcasting a message when a maximum number of received message is satisfied. The aim of the iDTSG protocol is to maximize informed nodes in an intended region with a fast and a reliable transmission while minimizing network loads.

4.2.1 Direction of Propagation of the Warning Message

In our protocol, the alarm message propagates to the vehicles that approach an intended region. Hence, the message's direction should be opposite a direction of intended vehicles. A priority to rebroadcast would be given to back intended vehicles and front helping vehicles of each sender's transmission range. This preference makes the message propagating opposite the direction of intended vehicles as we need.

4.2.2 Broadcast Storm Suppression Part of iDTSG

In the broadcast storm suppression part of iDTSG, a forwarder must be selected without requiring coordination among nodes to suppress broadcast storm problem. Based on probabilistic channels, iDTSG uses a contention-based forwarding scheme instead of a flooding-based scheme. In the contention-based forwarding scheme, a receiver waits for a defer time before deciding to broadcast a received message. For iDTSG, we use a directional contention-based forwarding scheme to select a forwarder node in each hop. In the directional contention-based forwarding scheme, if a receiver does not hear any duplicate messages from nodes in the message's direction from their lanes within a defer time duration, the receiver rebroadcasts the received message. The defer time is inversely proportional to a relative distance between a receiver and a forwarder and hence a furthest node in a message's direction has the shortest defer time. Then, the furthest node rebroadcasts the message first and it is called a *forwarder* node. For example, an intended (or helping) vehicle receives an alarm message at the first time and hence it defers the message broadcasting with the defer time. In the defer time duration, if the intended (or helping) vehicle hears any duplicate messages from another back intended (or front helping) vehicles with respect to itself, the intended (or helping) vehicle drops the received message otherwise it rebroadcasts the received message. Furthermore, the intended (or helping) vehicle drops every received messages at the first time if messages come from another back intended (or front helping) vehicles with respect to their positions. Because those vehicles consider that they are not suitable to disseminate the alarm message. This scheme controls the message's direction. It also limits redundant transmissions and wasting channel utilization in highly density networks. However, as shown in Fig. 4.2, some vehicles that receive a message from the vehicle $V1$ cannot hear any duplicate messages from the furthest node (the vehicle $V2$). On the other hand, some vehicles do not receive a message from the vehicle $V1$ but they receive a message from the vehicle $V2$ at first. Those vehicles will rebroadcast the received message until those receive any duplicate messages from nodes in the message's direction on their lanes. Hence, this is not a optimal scale but it is a quasi-optimal scale for dissemination in probabilistic channels with the distance-based defer time technique. This reduces the redundancy of rebroadcast messages in highly density networks.

4.2.3 Deterministic Distance-Based Defer Time

Due to the fact that the defer time is inversely proportional to the relative distance between a receiver and a forwarder, the greater the distance between a forwarder and a receiver, the shorter the defer time. Hence, only the furthest receiver broadcasts the message. The defer time in (4.1) which is called dynamic sleep time in [6] is applied in our iDTSG's defer time as follows:

$$T_{D,I} = T_N \times \frac{R}{d}, \tag{4.1}$$

where

$$T_N = \frac{2R}{s_r + s_m}. \tag{4.2}$$

R is the expected transmission range, d is the distance between the forwarder and the receiver, s_r is the speed of the receiver, and s_m is the maximum speed limit.

4.2.4 Store-Carry-Forward

Due to the directional contention-based forwarding scheme, if multi-hops connectivity among nodes are not available, forwarders must be a tail vehicle among cluster groups of intended vehicles and a head vehicle among cluster groups of helping vehicles. Since neighbor's information is not known before receiving message, the forwarders must repeatedly rebroadcast the received message to overcome network partitioning problem. The periodic rebroadcast time must be short enough such that there is any contact between a forwarder and an opposite receiver. However, the shorter the interval time, the higher the overhead. On the other hand, the longer the interval time, the higher the loss ratio. To be on a safe side, the forwarders can assume that a next hop node in an opposite direction moves with the maximum speed allowed in the highway and hence the contact time inside the forwarder's transmission range is twice the transmission range divided by the sum of both vehicle's speed. The interval time in (4.2), called a normal sleep time, is applied to repeatedly rebroadcast the received message to ensure that all vehicles receive the alarm message.

4.2.5 Time-Stable Geocast Part of iDTSG

A mechanism to store a message in the intended region is broadcasting in separate regions. Firstly, the area surrounding the intended region names a *forwarding* region shown in Fig. 4.1. The vehicles inside the forwarding area rebroadcast the message to ensure that all vehicles in the intended region receive the message. Secondly, the extra region (D_{extra}) where vehicles exit from the intended region. To keep the message alive in the intended region, the vehicles rebroadcast the message in both extra regions shown in Fig. 4.1. This ensures that all incoming vehicles receive the message before they enter to the intended region. The similar concepts to keep a message alive in a specific area is found in [6] and [31].

As shown in Fig. 4.4, the iDTSG protocol contains two main parts: the broadcast storm suppression part and the time-stable geocast part. In the time-stable geocast part, it is divided into a geocast period and a time-stable period depending on the purpose of PS flag. During the geocast period, if a forwarder is in the forwarding region, it rebroadcasts the received message with the interval T_N given in (4.2) until the stop rebroadcast condition(C1) is satisfied. When a helping vehicle (be a forwarder) moves to the extra region, its PS flag turns down (0). Then, the protocol changes into the time-stable period and it initiates to disseminate the message again only in its extra region. When a receiver sees the turn down PS flag from its received message, the receiver changes into the time-stable period too. During the time stable period, a forwarder repeatedly rebroadcasts the received message with the interval (4.2) in $theD_{extra}$ region on its own lane until the stop rebroadcast condition (C2) is satisfied. The forwarders in two separate extra regions expect to inform the alarm message to other vehicles on the opposite direction before the other vehicles move to the intended region.

Due to this diagonal location-based forwarding mechanism, the message is kept alive within this intended region until an expiration time. This is better than using only a server node to always rebroadcast in the intended region.

Due to a counter-based scheme, the stop-rebroadcast condition C1 is that a forwarder receives any duplicate messages from a same-direction forwarder or three duplicate messages (optimal value in our simulation) from opposite-direction forwarders. The message's direction is opposite a direction of intended vehicles. It does not depend on a leading vehicle on its cluster lane like the DTSG protocol. This is the difference between iDTSG and DTSG in the stop-rebroadcast condition to enhance the broadcast storm suppression. The condition C2 is that a forwarder receive one duplicate message from opposite-direction forwarders or any leading forwarder with respect to its location in an extra region.

4.2.6 Dynamic Length of Extra Region

Recall that the purpose of the extra region is to have a helping (intended) vehicle forwarding a message to an intended (helping) vehicle. All vehicles drop a received message after they exit the extra region. Hence, the length D_{extra} of each extra region is critical to iDTSG performances. Since the extra region is the important region to keep a message alive within the geocast region (D km), for a time duration, at any density networks.

To accommodate different vehicle densities, the DTSG protocol uses $D_{extra} = D/\gamma$ where γ is the pre-determined density of vehicles in one lane (in vehicles/km). Since $1/\gamma$ is the average inter-vehicle spacing, DTSG requires $D_{extra} = D \times$ inter-vehicle spacing. This dependency on D is rather undesirable since if D is small, D_{extra} is too small and the message may not get delivered successfully to the helping or intended vehicles in time. Hence, in iDTSG we propose a more reasonable length of $D_{extra} = \frac{2}{\gamma} = 2\times$average inter-vehicle spacing, which is independent of D.

The factor of 2 is chosen so that on average the vehicle in the exiting extra region sees two receiving vehicles in the opposite lane and has a better chance to carry the message to one of the vehicles entering the forwarding region. We want to make sure that D_{extra} is large enough such that the forwarder is highly successful its delivery. For our simulation, the factor two is chosen to cover variation in the inter-vehicle spacing from the average. Note that a larger factor would give a larger D_{extra} and hence a better delivery ratio but at the cost of more rebroadcast.

Due to the fact that the iDTSG protocol improves from the DTSG protocol. We improve the forwarder selecting mechanism and the stop-rebroadcast condition for the better broadcast storm suppression in both periods. We also define the reasonable length of D_{extra} region in iDTSG protocol for highway networks. We change the condition to decide the period of protocol by using the message's flag instead of the location of vehicles. Furthermore,

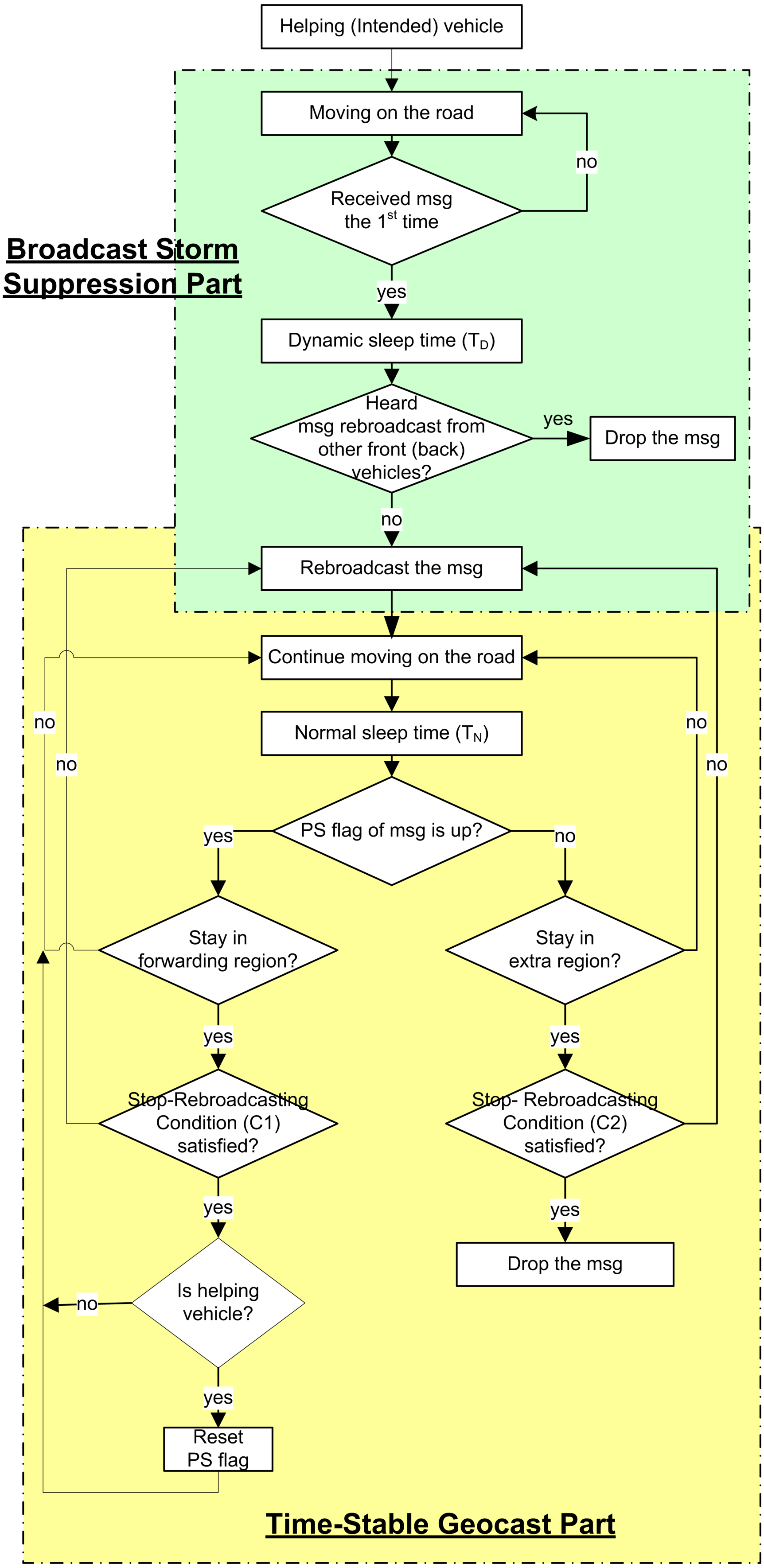


Figure 4.4 iDTSG protocol flow chart

the geocast area (intended region) in iDTSG is larger than the area in [31] to reduce traffic jams because of bottleneck problems .

4.3 Simulation Results

Since the objective of our time stable geocast protocol is informing an alarm message to all vehicles in an intended region as fast as possible and keeping the message alive in the intended region with minimum transmissions, we are interested in reliability and transmission efficiency which generally can be measured in multiple ways in broadcast protocols including time-stable geocast protocols. In our work, the reliability is measured in term of the packet *loss ratio* while the the efficiency is measured via *overhead.*

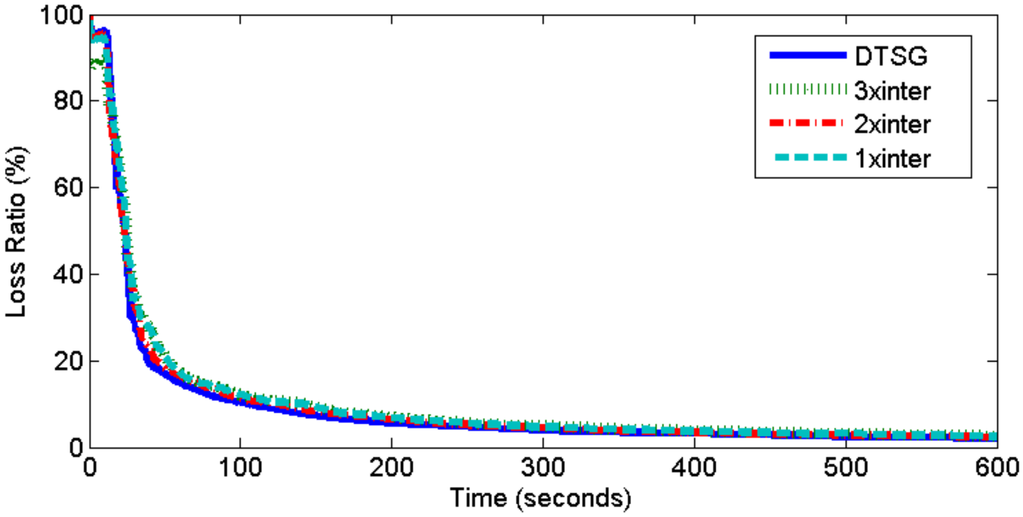
Loss Ratio: Assuming the time of the first broadcast of the message as time $t = 0$, the loss ratio at time t is the ratio between i) the number of those intended nodes that have not received the message up to time t and ii) the total number of intended nodes up to time t. In emergency notification scenario, we are also interested in the *dissemination time*, which is the shortest time that the loss ratio reaches almost 0%.

Overhead: The overhead at time t is the total number of packet rebroadcasts up to time t. This number includes the collided rebroadcasts.

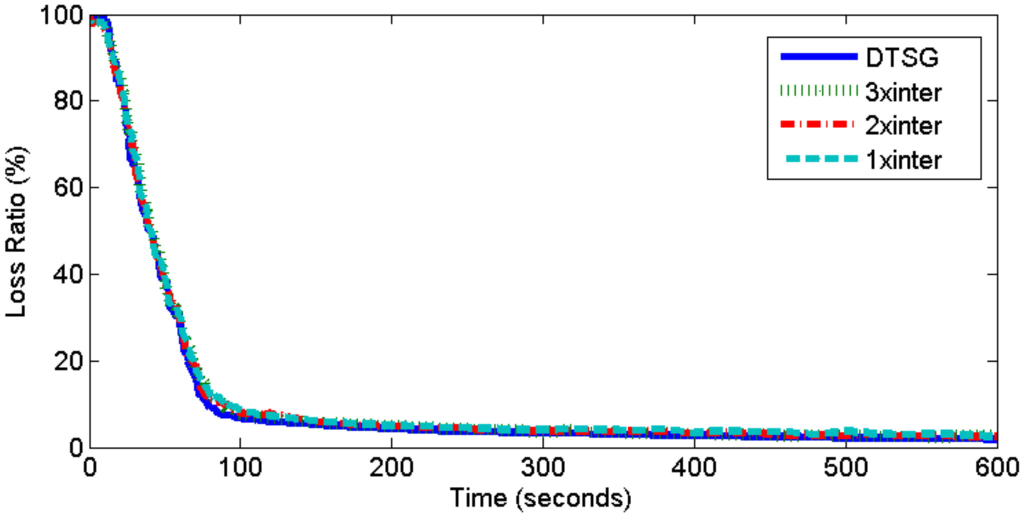
4.3.1 Performance on Reliability

With a priori known inter-vehicle spacing, the iDTSG protocol compares with the DTSG protocol in both dense networks and sparse networks. As mentioned in the previous section, the reasonable length of D_{extra} is 2 × average inter-vehicle spacing. This length should be able to cover the spacing between vehicles. In this work, we believe that our D_{extra} length of iDTSG is more suitable than D_{extra} which depends on the length of D km in [6]. In this section, we vary D_{extra} region from (1 × average inter-vehicle spacing) to (3 × average inter-vehicle spacing) to verify our belief. Fig. 4.5 (a) and (b) illustrate the loss ratio in dense networks where the length D is 1 and 3 km. For reliability, the length of D_{extra} has no affect on the loss ratio in dense networks. Both $D = 1$ and $D = 3$ km, the loss ratio of iDTSG is similarly to the loss ratio of DTSG in dense networks.

However, the length of the D_{extra} affects on the loss ratio of iDTSG in sparse networks. In $D = 1$ km, the loss ratio of the DTSG protocol is better than the loss ratio of the iDTSG protocol when the D_{extra} is $1 \times$ average inter-vehicle spacing and $2 \times$ average inter-vehicle spacing. The loss ratio of DTSG and iDTSG (when the D_{extra} is $3 \times$ average inter-vehicle spacing) are similar and less than 10%.



(a)Loss Ratio in dense networks when D = 1 km

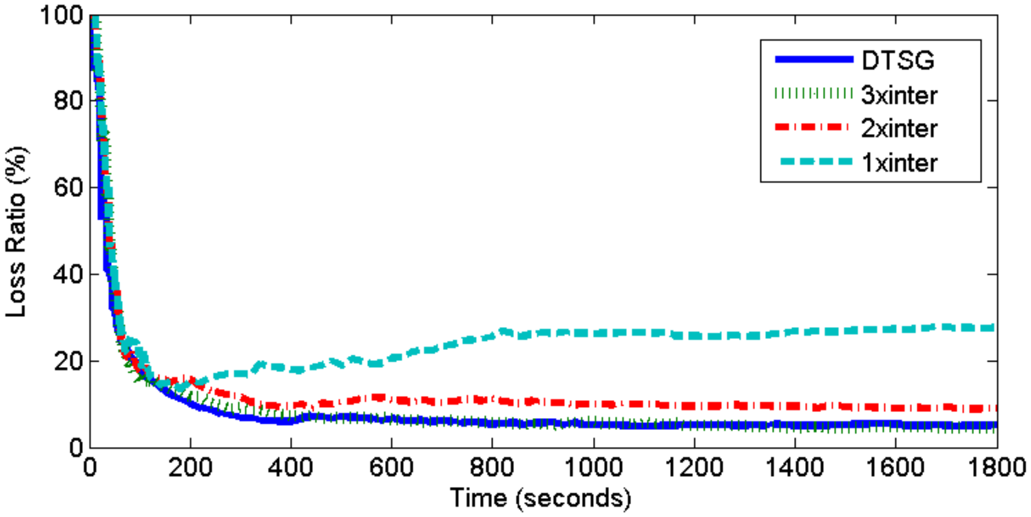


(b)Loss Ratio in dense networks when D = 3 km

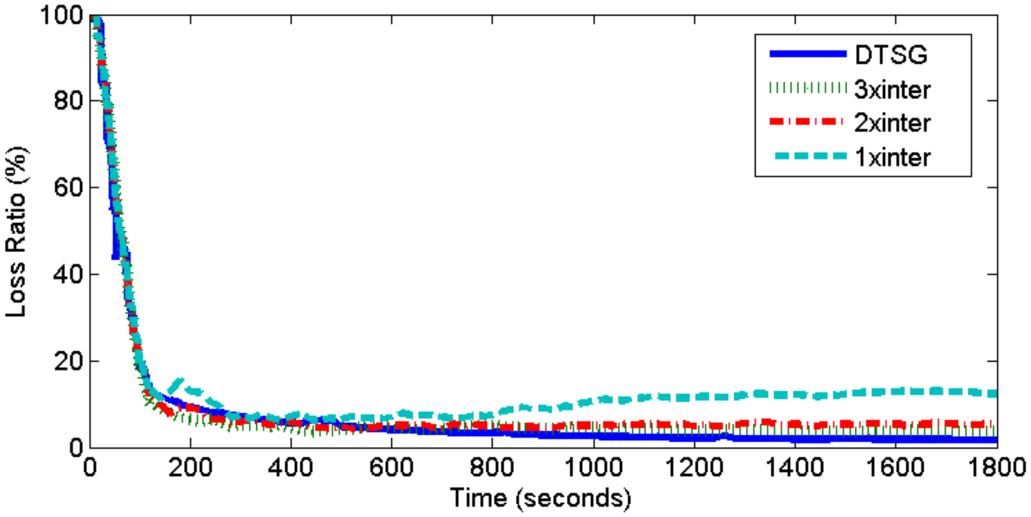
Figure 4.5 Loss Ratio for dense networks

In $D = 3$ km, the loss ratio of iDTSG (when the D_{extra} is $2\times$average inter-vehicle spacing and $3 \times$ average inter-vehicle spacing) and the loss ratio of DTSG are similar. The loss ratio of iDTSG (when the D_{extra} is $1 \times$ average inter-vehicle spacing) is the worst.

As shown in Fig. 4.6 (a) and (b), the results show that the length of D_{extra} is such important region to maintain the message in the intended region for sparsely density networks. Hence, the length of D_{extra} must be independent from the value of the D region and larger to ensure that new coming vehicles receive the message in the D_{extra} region. The iDTSG protocol and the DTSG protocol also depend on the same D km region and the same defer time (T_D) and hence the loss ratios in dense networks are similar. The area of D region has no affect on the reliability of the DTSG protocol and the iDTSG protocol.



(a)Loss Ratio in sparse networks when D = 1 km



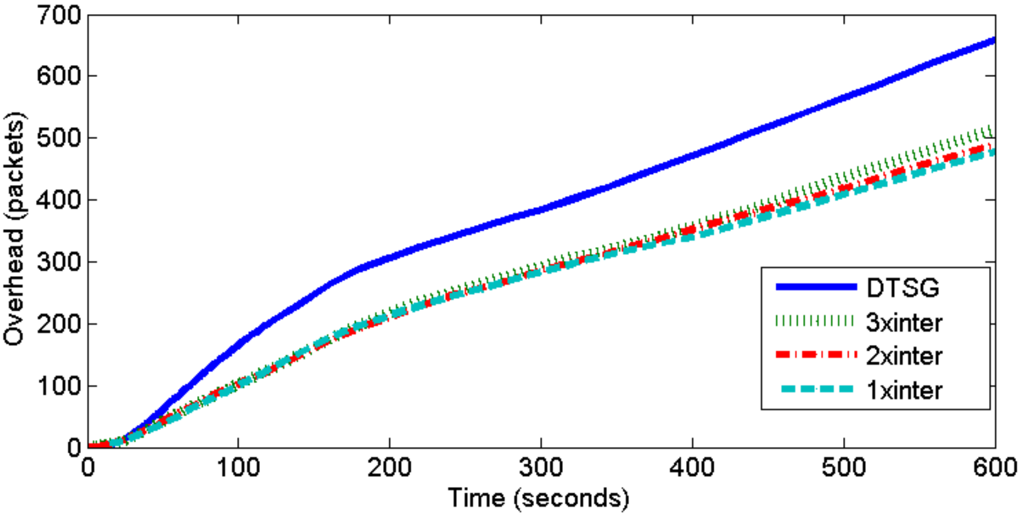
(b)Loss Ratio in sparse networks when D = 3 km

Figure 4.6 Loss Ratio for sparse networks

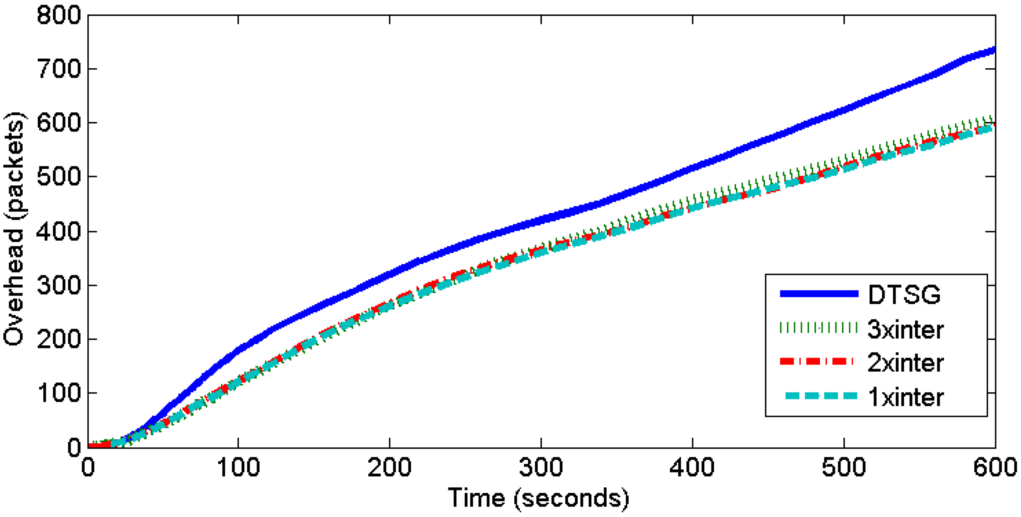
4.3.2 Performance on Efficiency

Since the aim of iDTSG is to maximize informed nodes in an intended region as fast as possible with minimum network loads, we compare the overhead of the iDTSG protocol with the DTSG protocol.

As shown in Fig. 4.7 and Fig. 4.8, all overheads of the iDTSG protocol are less than the overhead of the DTSG protocol in every length of D and D_{extra} regions, in dense and sparse networks. In dense networks, the overheads of the iDTSG protocol are similar. In sparse networks, the larger the D_{extra} region, the larger the overheads. Hence, the length of the D_{extra} region is the crucial factor to maximize the reliability and minimize the overhead



(a)Overhead in dense networks when D = 1 km

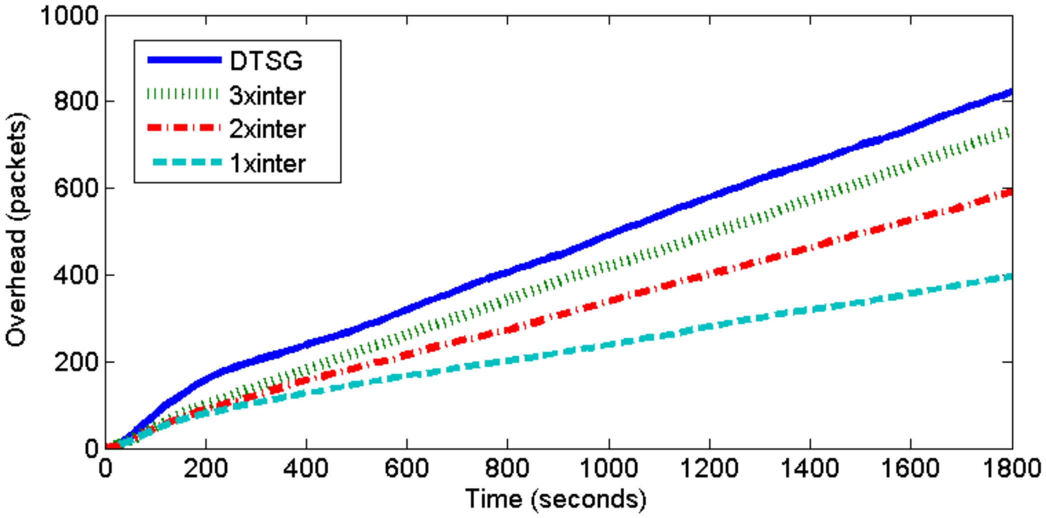


(b)Overhead in dense networks when D = 3 km

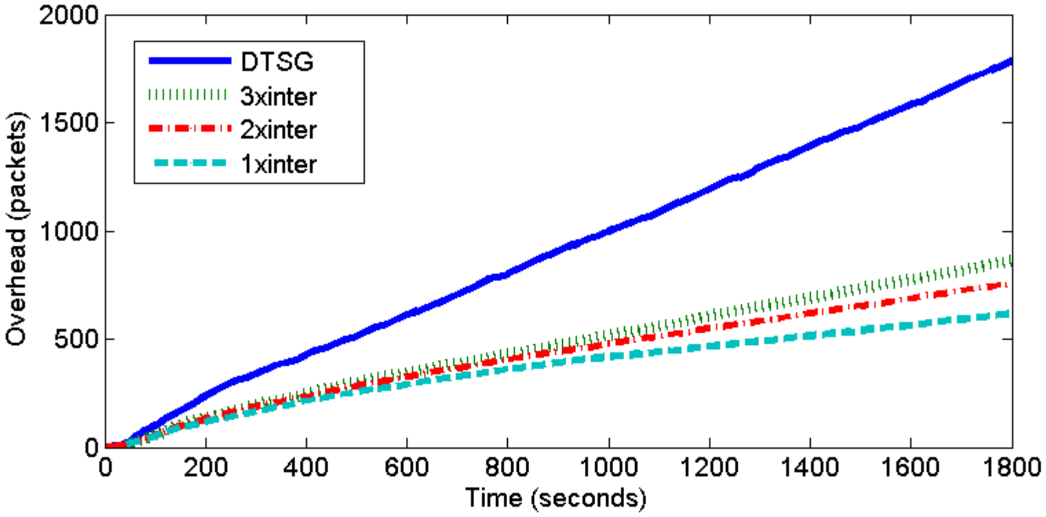
Figure 4.7 Over head for dense networks

in sparse networks.

In Fig. 4.9 and Fig. 4.10, a number of cumulative helping vehicles (i.e., helping vehicles participate in the networks) in every length of D and D_{extra} regions, in dense and sparse networks are shown. The helping vehicles in the iDTSG and DTSG protocols are similar in dense networks. In the not-too-difference helping vehicles of the iDTSG protocol, the overheads are less than the overheads of the DTSG protocol whether what the length of D_{extra} regions is. Hence, the length of D_{extra} regions is no effects on the efficiency in dense networks. In sparse networks, the DTSG protocol requires the helping vehicles to participate in the networks more than the iDTSG protocol. With the similar reliability of iDTSG and DTSG in D_{extra}(3 x inter), the iDTSG protocol incurs the smaller overheads and requires the less helping vehicles to participate in the networks.



(a)Overhead in sparse networks when D = 1 km



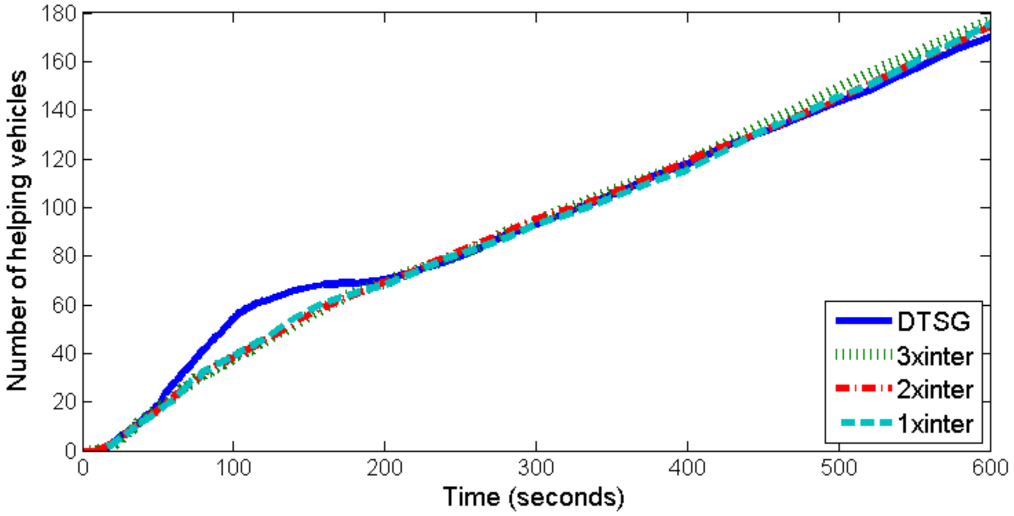
(b)Overhead in sparse networks when D = 3 km

Figure 4.8 Over head for sparse networks

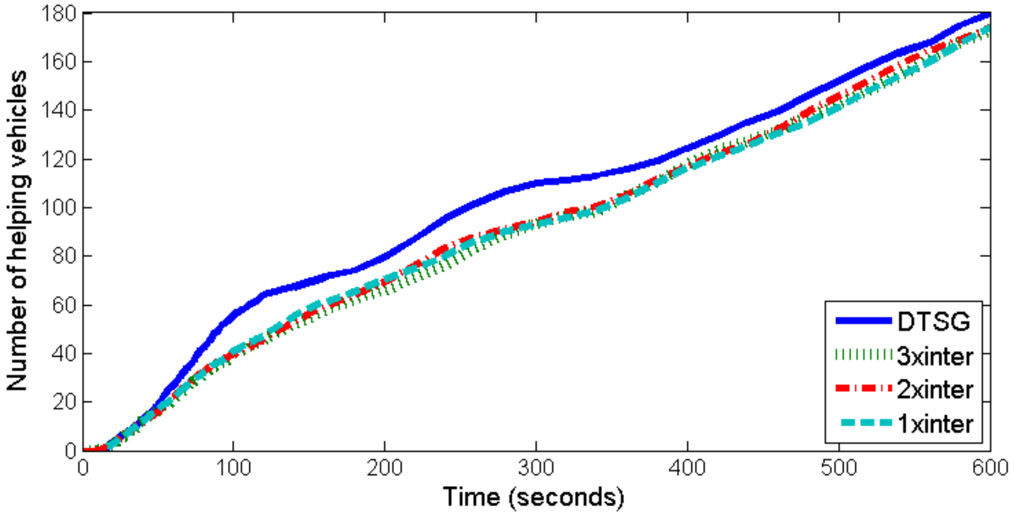
Hence, the well-defined broadcast storm suppression and the reasonable D_{extra} region of the iDTSG protocol result to the better reliability and the better efficiency, comparing to the DTSG protocol.

We compare relative overheads of the iDTSG protocol and the DTSG protocol in both dense networks and sparse networks. As shown in Fig. 4.11 and Fig. 4.12, the overheads of the iDTSG protocol are less than the overheads of the DTSG protocol at least 20% in both the D regions, in dense networks. Furthermore, at the proposed D_{extra}, the overhead of the iDTSG protocol is less than the overhead of the DTSG protocol at least 30% in sparse networks.

The iDTSG protocol incurs the smallest overheads, although the loss ratios of the



(a)Number of helping vehicles in dense networks when D = 1 km

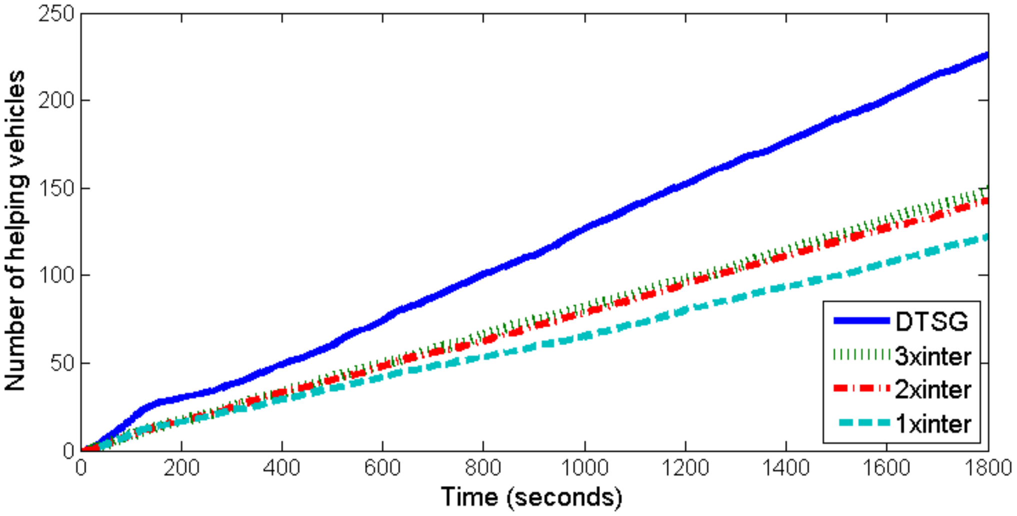


(b)Number of helping vehicles in dense networks when D = 3 km

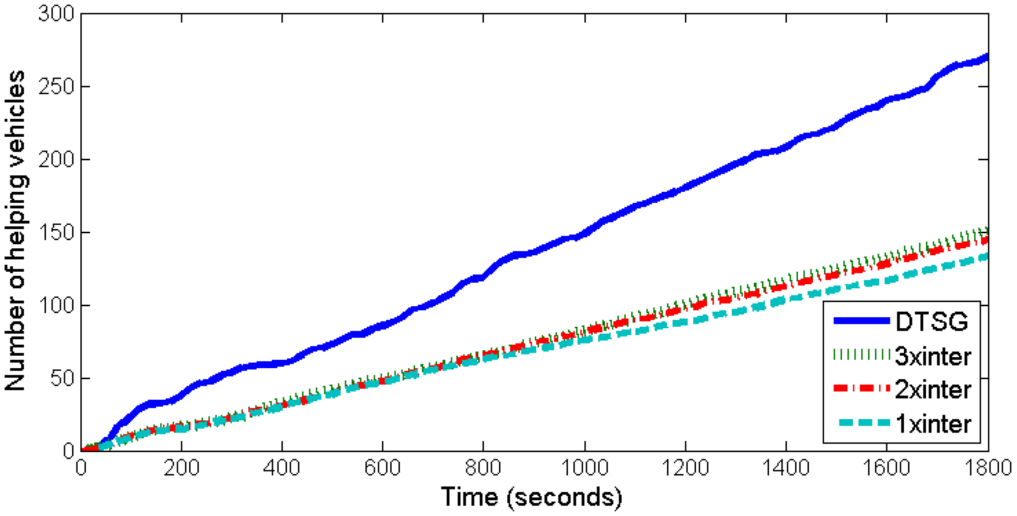
Figure 4.9 Number of cumulative helping vehicles for dense networks

DTSG protocol and the iDTSG protocol are similar. In the iDTSG protocol, the limitation in rebroadcast areas of the geocast period including the independent of D in the D_{extra} region dramatically reduces overheads. Since the iDTSG protocol has less overheads but it has similar loss ratios, comparing to the DTSG protocol. This confirms that our iDTSG protocol is better the DTSG protocol. However, bandwidth should be reserved for comfort applications and hence the iDTSG protocol keeps lower bandwidth utilization. This reduction in the overheads means the less consuming power which has a significant in the green technology of VANETs.

The objectives of the iDTSG protocol are satisfied where the alarm message is successfully disseminated to all intended vehicles and it is kept for 30 minutes in the intended



(a)Number of helping vehicles in sparse networks when D = 1 km

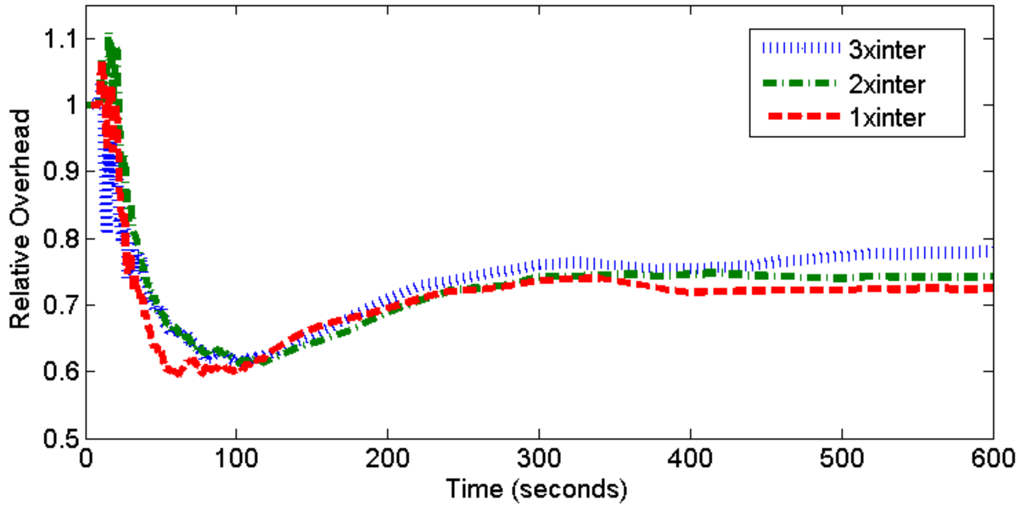


(b)Number of helping vehicles in sparse networks when D = 3 km

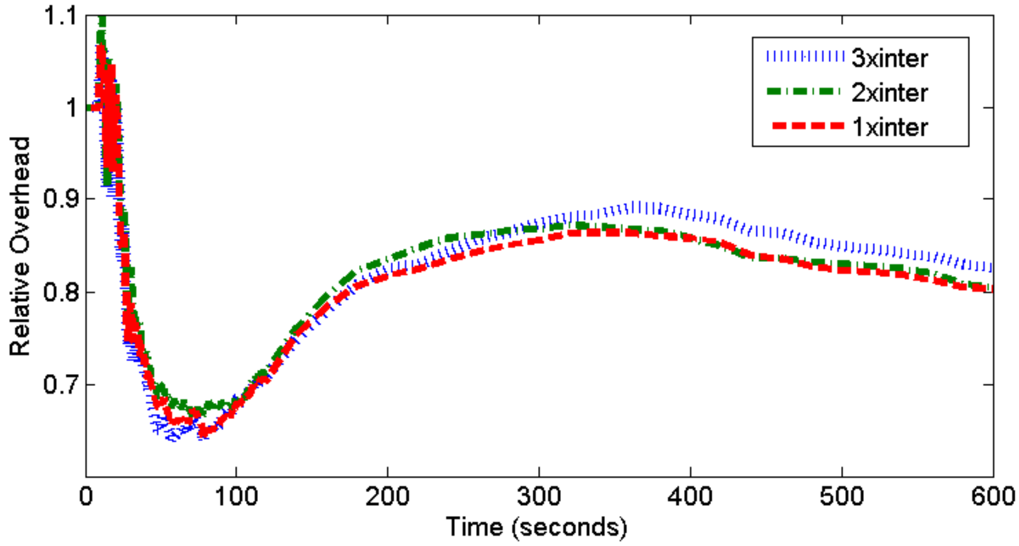
Figure 4.10 Number of cumulative helping vehicles for sparse networks

region. The alarm message disseminates with reliability in sparse networks and efficiency in dense networks. The results show that the iDTSG protocol is high reliability and high efficiency in any density networks, comparing to the DTSG protocol. However, multiple crash vehicles can simultaneously occur in a nearby area, these events lead to multiple sources broadcasting alarm messages. The fundamental limitation of the iDTSG protocol is that it cannot detect redundant information from the different sources to optimize the efficiency.

Hence, the multi-sources time-stable geocast protocol is worth to develop since the reliability and the scalability should be considered in both sparsely and highly density networks as mentioned in [46]. Furthermore, if the performance metrics take the dissemination time into account, the iDTSG protocol is too slow in the message dissemination for safety



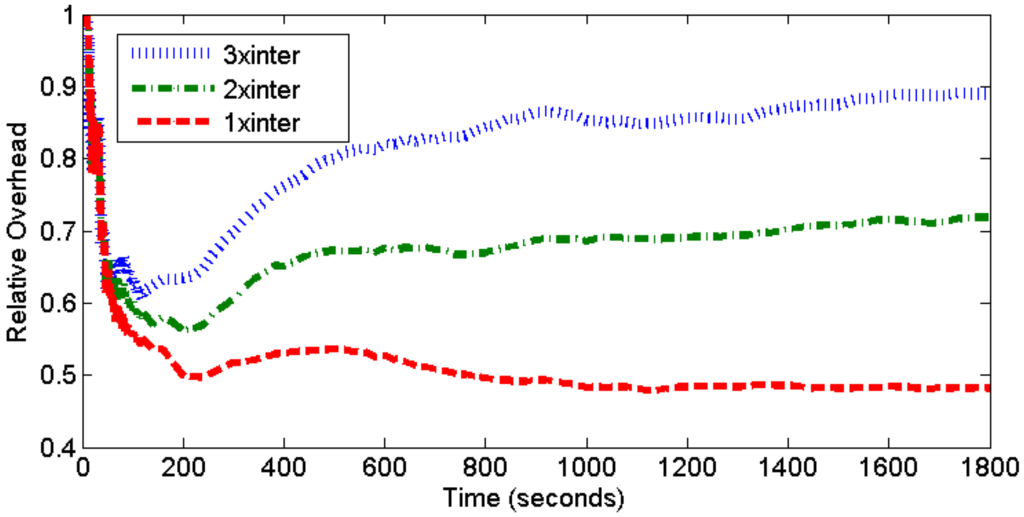
(a)Relative Overhead in dense networks for D=1



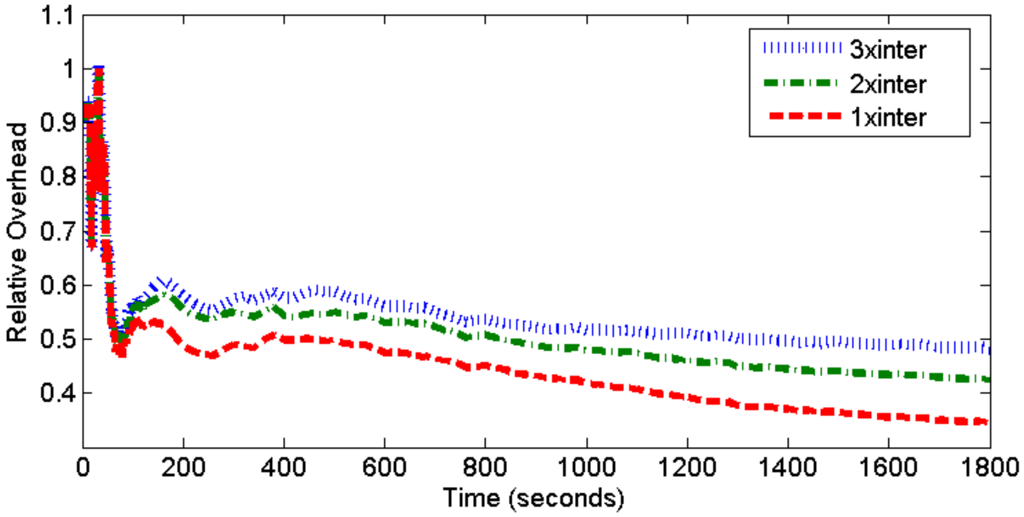
(b)Relative Overhead in dense networks for D=3

Figure 4.11 The relative overhead of iDTSG with DTSG in dense networks

applications. In dense networks, we expect that the loss ratio of iDTSG reaches zero percent as fast as possible with minimum overhead. In dense networks, both protocols take more than 100 seconds to disseminate the message in the intended region. This occurs from using the same defer time in the iDTSG protocol and the DTSG protocol. Hence, we want to investigate effects of varying the T_D in our protocol performance, in the next Chapter.



(a)Relative Overhead in sparse networks for D=1



(b)Relative Overhead in sparse networks for D=3

Figure 4.12 The relative overhead of iDTSG with DTSG in sparse networks

Chapter 5

Effects of Distance-Based Defer Times and Probabilistic Channel to Time-Stable Geocast

This chapter evaluates effects of some existing deterministic and stochastic distance-based defer times in both deterministic and probabilistic (i.e., fading and shadowing) channels. Parts of this chapter appear in [47].

5.1 Introduction

Many articles use contention-based forwarding protocols (e.g., [21]) for forwarder selection in VANETs. The contention-based forwarding protocol requires no information of neighboring vehicles. In this protocol, receiver waits for a time duration, called defer time, before deciding to broadcast the received message. If the receiver does not receive any duplicate messages during its defer time, the receiver broadcasts the received message after the end of its defer time, otherwise it drops the message. The defer time is used in a general approach to suppress the broadcast storm problem in stateless broadcast protocols, where each node does not know or keep track of the locations of its neighbors [48]. The defer time is a crucial factor of contention-based forwarding protocols for broadcast storm suppression in dense networks. It was shown in that contention-based forwarding protocols are better than position-based forwarding protocols in probabilistic channel [22]. However, an influence of the undesirable shadowing attenuation becomes more significant when the traffic becomes dense since the progress of transmission is reduced by the shadowing problem [45]. With the shadowing effect, the message dissemination time may be longer than desired for an emergency notification in VANETs.

Several proposed defer times (e.g., [7], [8], and [9]) depend deterministically on distance. Hence, they are called distance-based defer times. The distance-based defer time is inversely proportional to the distance between sender and receiver. The further the distance from sender, the smaller the defer time of the receiver. If there are more than two receivers at the same distance from the sender, the receiver will broadcast with the same defer time and hence a collision occurs. There are other defer times that depend not only on distance, but also angle [49] and link probabilistic [50]. In addition, stochastic defer times have been proposed such as uniform and Gaussian random defer times [10], and bi-zone random defer time [51] to decrease the dissemination time in non-line-of-sight transmission or probabilistic channel.

In this chapter, we study in more details than what we have done in the Chapter 4 on the importance of the distance-based defer time on the performance of the iDTSG protocol in suppressing the broadcast storm problem. Here, we show by simulation that stochastic defer times are better than the deterministic ones when the protocol is used in real world and faces with probabilistic channels. Furthermore, for deterministic defer time, we show that for a highway of a single lane in each direction, we can determine the minimum defer time such that no collision occurs.

5.2 Deterministic Distance-Based Defer Times

A more popular form of defer times has been used in multiple papers, e.g., [6]-[11]. It is given as

$$T_D(d) = \begin{cases} T_{DMAX}\left[1 - \left(\frac{d}{R}\right)^{\epsilon}\right] & \text{if } 0 \le d \le R, \\ 0 & \text{if } d > R \end{cases} \tag{5.1}$$

where $\epsilon > 0$ is a constant, and T_{DMAX} is a given maximum defer time. With such definition of T_D, the further node from the source waits less and rebroadcasts faster.

Different values of ϵ give different shapes of the defer time. Fig. 5.1 illustrates T_D's for $\epsilon = 2, 1, 0.5$ and 0.2. Without confusion, we denote $T_D(\epsilon = \epsilon_0)$ for T_D with $\epsilon = \epsilon_0$. It can be shown that T_D is linear, convex, and concave for $\epsilon = 1$, $\epsilon < 1$, and $\epsilon > 1$, respectively. Note that when $\epsilon = 0$ or $T_{DMAX} = 0$, we have the simple flooding scheme with no broadcast storm suppression since all nodes use the same defer time of zero.

Several values of ϵ have been used in literature. By assuming that nodes are uniformly distributed over a two-dimensional area, the authors in [8], [9], [11] set $\epsilon = 2$ so that T_D is expected to be uniformly distributed over the range $[0, T_{DMAX}]$. The aim is to have minimum collision, assuming that the vehicles are uniformly distributed on the highway. In [7], the authors used the defer time in (5.1) with $\epsilon = 1$. Although the authors in [10] used random defer times, they also used (5.1) with $\epsilon = 1$ as the mean of the defer time. In summary, it seems that a guideline in choosing ϵ is to have a uniform defer time to avoid collisions. We will see that other values of ϵ actually give better performance.

A few notes on the values of R and T_{DMAX}. It seems that the defer time in (5.1) assumes that the channel is deterministic in the sense that no nodes outside the transmission range R receive the packet transmission, while the nodes inside the range R always does. However, although this channel model is simple, it is not realistic since it ignores multipath

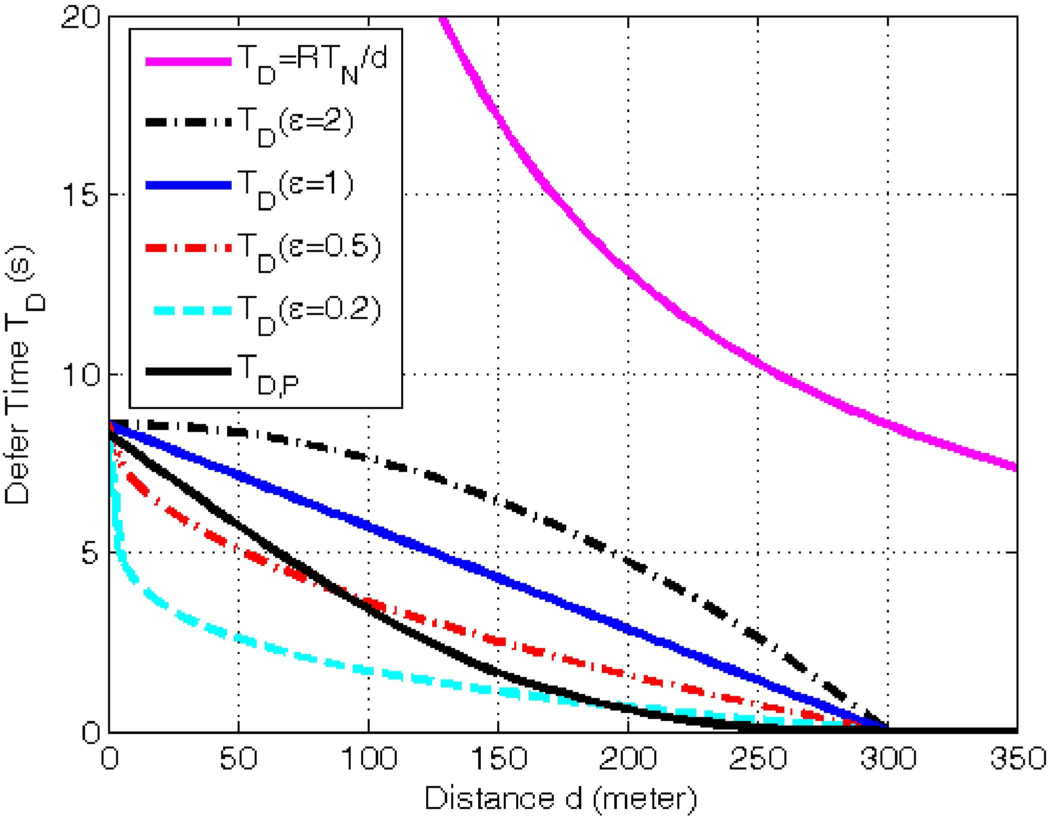


Figure 5.1 Different defer times T_D: the inverse defer time $T_{D,I}$ in (4.1), T_D in (5.1) with $\epsilon = 2, 1, 0.5$, and 0.2, the probabilistic-channel defer time $T_{D,opt}$ in (5.2), for $R = 300$ and $T_{DMAX} = R = smax = 8.57$ s. For this plot, we assume the receiving node is moving at the maximum speed $(sr = smax)$ and hence from (4.2) we have $T_N = 2R/2s_{max} = T_{DMAX}$.

fading. For multipath fading channel, which we call probabilistic channel, the proper value of R is not obvious. For the maximum defer time T_{DMAX}, lowering T_{DMAX} increases the speed that the message is disseminated at the cost of more collisions. Due to the low probability to receive packet at further node, the authors in [52] and [53] recommend to decrease the value of R and to increase the value of T_{DMAX} from (5.1) with $\epsilon = 1$ then the nodes greater than R range do nothing if they receive the packet. With this mechanism, the relay is selected only in R range, it ensures to the higher reliability and the higher efficiency for the probabilistic channel in highly dense networks. Conversely, in sparse networks, it results to the low reliability and efficiency. This problem leads us to design the defer time that is suitable to iDTSG protocol to result the high reliability and the high efficiency on both dense and sparse networks for probabilistic channel.

Using ns-3 simulation and the same transmission power of 5 dBm, Fig. 3.2 in Section 3.1 shows the reception probabilities versus distance for the deterministic channel and the probabilistic channel based on the Nakagami and the log-distance path loss. We denote the reception probability under the probabilistic channel as $P_R(\cdot)$. To include the effect of the probabilistic channel, we propose a new distance-based defer time:

$$T_{D,P} = T_{DMAX}\left[1 - \frac{\int_0^d P_R(x)\,\mathrm{dx}}{\int_0^\infty P_R(x)\,\mathrm{dx}}\right] \tag{5.2}$$

for $d \geq 0$. Specifically, $T_{D,P}$ takes into account the fact that at distance d only fraction $P_R(d)$ of the nodes receive the packet transmission. An example of $T_{D,P}$ for the above reception probability $P_R(\cdot)$ is shown in Fig. 5.1. In the figure, since $P_R(d)$ for $d > R = 300$ is very small, for convenience we truncate $T_{D,P}$ such that the $\int_0^\infty$ is replaced with $\int_0^R$ and $T_{D,P}(d) = 0$ for $d > R$. Note that $T_{D,P}$ requires the knowledge of $P_R(\cdot)$ which, in practice, might be difficult to know and depends on time and location.

The reason to define $T_{D,P}$ as in (5.2) follows the idea in [8], [9] and assumes that the vehicles are distributed uniformly on the highway. The idea is that we should design $T_{D,P}$ such that the defer time is distributed uniformly over its range $[0, T_{DMAX}]$. This criteria would make sure that the collision rate is minimized. Following this design criteria, it can be shown that, after taken into the account the reception probability P_R, the $T_{D,opt}$ function gives the uniformly distributed defer time.

5.2.1 Bound on Minimum Deterministic Defer Times

In this chapter via simulation, we observed that $\epsilon = 0.2$ gave better loss ratio, while having similar overhead, than other higher ϵ. Although $\epsilon = 0$ (corresponding to the flooding scheme) gave the worst performance among all considered ϵ's, the loss ratio for $\epsilon \geq 0.2$ monotonically increases with decreasing ϵ, while having similar overhead. Hence, there must be a minimum ϵ, denoted by ϵ_{min}, that gives the best dissemination time while keeping the overhead as small as possible (i.e., negligible collision). While this ϵ_{min} can be found via repeated simulations, it is better to find at least its approximate bound.

To understand the bound of ϵ_{min}, we observe that in our system no cars that are potential relays in the same direction use the same defer times. This is because (i) the considered highways have a single lane per direction, and hence only two receiving cars at the same distance but opposite from a transmitter wait the same defer time, and (ii) in the same direction as the source car, the message should propagate to the back of the transmitter; hence, only the cars in the back of the transmitter are potential relays.

Hence, if any two neighbor nodes use defer times that differ more than the time (called t_p) to send the message, no collisions occur during the message multihopping, i.e., for any defer time function T_D we need the defer time difference between any neighbors greater than t_p, i.e.,

$$T_D(d - \Delta) - T_D(d) \geq t_p, \tag{5.3}$$

for any distance d from a transmitter and any inter-car spacing Δ. Note that the packet transmission time t_p must include the time due to multiple access protocol as well as the link protocol. Although there is no collision in our situation, there may be time for multiple access protocol such as channel sensing. The defer time difference $T_D(d-\Delta) - T_D$ for d closer to R is an increasing function with ϵ. That is, the best T_D is when (5.3) holds with equality. Taking the popular defer time given in (5.1), no collisions happen when

$$T_{max}\left[\left(\frac{d}{R}\right)^{\epsilon} - \left(\frac{d-\Delta}{R}\right)^{\epsilon}\right] \geq t_p \tag{5.4}$$

In this equation the left term increases with ϵ, as can be observed from Fig. 5.1.

Hence, we are interested in finding the minimum ϵ, ϵ_{min}, given T_{MAX} and R.[2] A physical constraint is that neighboring cars do not get too close, i.e., $\Delta \geq \Delta_{min}$ for some Δ_{min} which may depends on road condition as well as car density. Although d can take almost any value within R, it should be the farthest distance of the node which can still receive the packet transmission with high enough reception probability $P_R(d)$. This node is the typical one which makes a successful rebroadcast. As shown in Section 5.4.4, specific to the parameters in our simulation, ϵ_{min} is between 0.00095 and 0.007.

5.3 Stochastic Distance-Based Defer Times

Since the car positions as well as the channels are stochastic, it might be suboptimal to use deterministic defer times. Consider a simple example where the inter-car spacings are fixed for all cars on the highway. Due to the probabilistic channels, cars further from the transmitters have less chance to receive the packet transmission and hence it is sometimes a waste of time for when there are no the closer nodes to wait a long defer time further nodes to rebroadcast. It is better if the closer nodes can randomly pick their defer times, which might happen to be smaller than those of further nodes. Stochastic defer times allow a breakdown of the deterministic dependence of defer times on distance, to combat against randomness in the locations and channels, possibly at the cost of higher collisions.

In particular, we consider the following version of stochastic defer times: At each reception of a packet, a node at distance d randomly and independently picks a stochastic defer time, denoted by $T_S(d)$, which is uniformly distributed between 0 and $T_D(d)$ where $T_D(d)$ is the popular deterministic defer time in (5.1), i.e.,

$$T_S(d) \in Uniform[0, T_D(d)]. \tag{5.5}$$

---

[2] A simpler protocol parameter design might be based on the linear defer time where $\epsilon = 1$. In this case, (5.4) becomes $T_{max}\Delta/R \geq t_p$, which is independent of d.

5.4 Simulation Results

We study the effects of the defer times to the system performance. Generally, in broadcast protocols including time-stable geocast protocols, we are interested in reliability and transmission efficiency which can be measured in multiple ways. Here the reliability is measured in term of a packet loss ratio while the the efficiency is measured via a overhead. The definition of the loss ratio and the overhead are introduced in Section 4.3.

5.4.1 Effect of Transmission Range in Probabilistic Channel

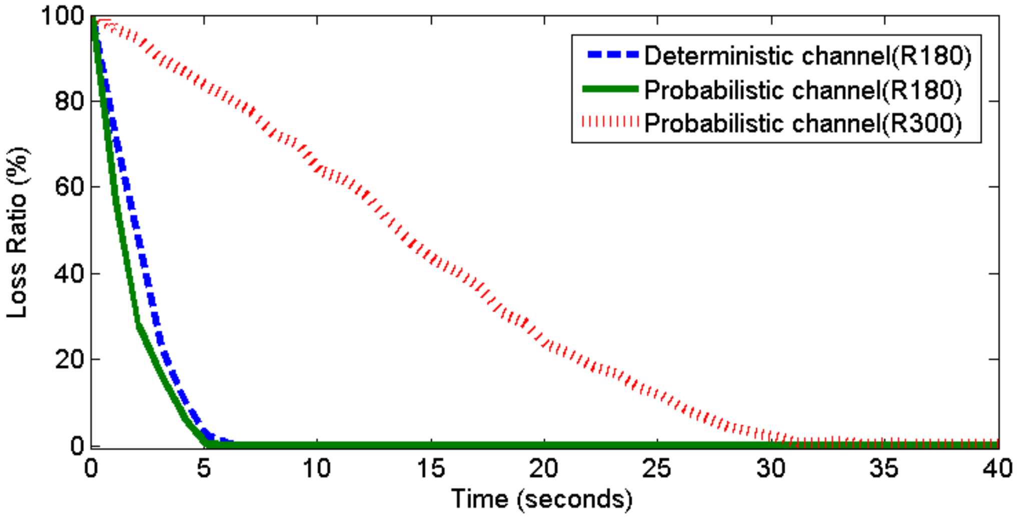
First, we evaluate the effect of varying the value of R in (5.1) on the iDTSG protocol performance for the probabilistic channel. One easily sees that the form of T_D (specially on the part that $T_D(d) = 0$ for $d > R$) is suitable for deterministic channels where R is the transmission range. However, for probabilistic channels, the transmission range where the reception probability is not zero is actually unlimited. Hence, for probabilistic channels the nodes further than R would all use zero defer times and hence collide. This might result in many collisions if R is too small.

To make our evaluation, we consider the case where $\epsilon = 1$ and $T_{DMAX} = R/s_{max}$ and hence, from (5.1),

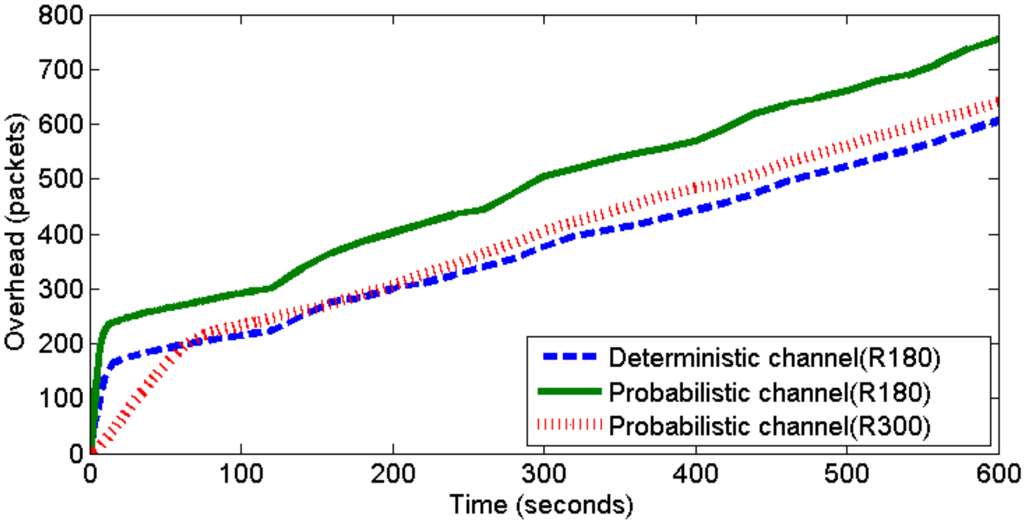
$$T_D(d) = \frac{R}{s_{max}}\Big(1 - \frac{d}{R}\Big) = \frac{R - d}{s_{max}} \tag{5.6}$$

for $d \in [0, R]$ and 0 otherwise. For this T_D and dense traffic scenario, Fig. 5.2 shows the loss ratio, overhead, and collision rate respectively, for 3 cases: i) deterministic channel with $R = 180$, ii) and iii) probabilistic channel with $R = 180$ and $R = 300$, respectively. Fig. 5.2(a) shows that at the same $R = 180$, the probabilistic channel gives a slightly better loss ratio than the other channel. The reason is that under the probabilistic channel nodes further than $R = 180$ can still receive the packet with non-zero probability. For both channels at $R = 180$, all intended vehicles receive the message within 5 seconds. However, it is 30 s for the $R = 300$ is always larger than T_D for $R = 180$ and hence the rebroadcasts happen slower. However, since $T_D(d) = 0$ for $d > R = 180$, as shown in Fig. 5.2(b), there is a higher overhead in the probabilistic channel with $R = 180$ case.

From Fig. 5.2, it is important to evaluate the performance of the defer time in (5.1) with realistic channel model. Furthermore, different R gives different performance tradeoff. Since the probabilistic channel with $R = 300$ gives similar overhead, lower collision rate but worse loss ratio than the deterministic channel, from now on we assume $R = 300$ when we use the probabilistic channel.



(a)Loss ratio in dense networks for D=3



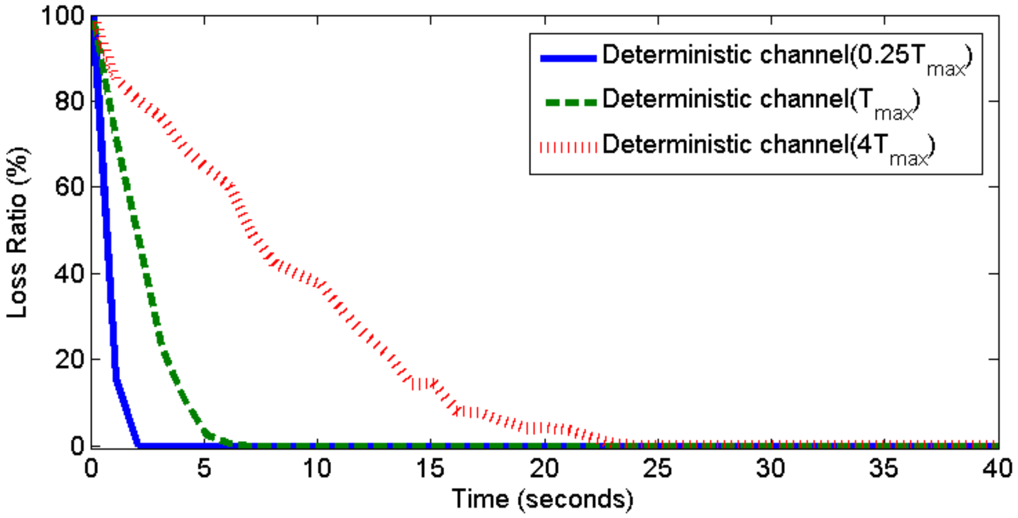
(b)Overhead in dense networks for D=3

Figure 5.2 Loss ratio and overhead for deterministic and probabilistic channels with $R =$ 180m and 300m, under dense scenario

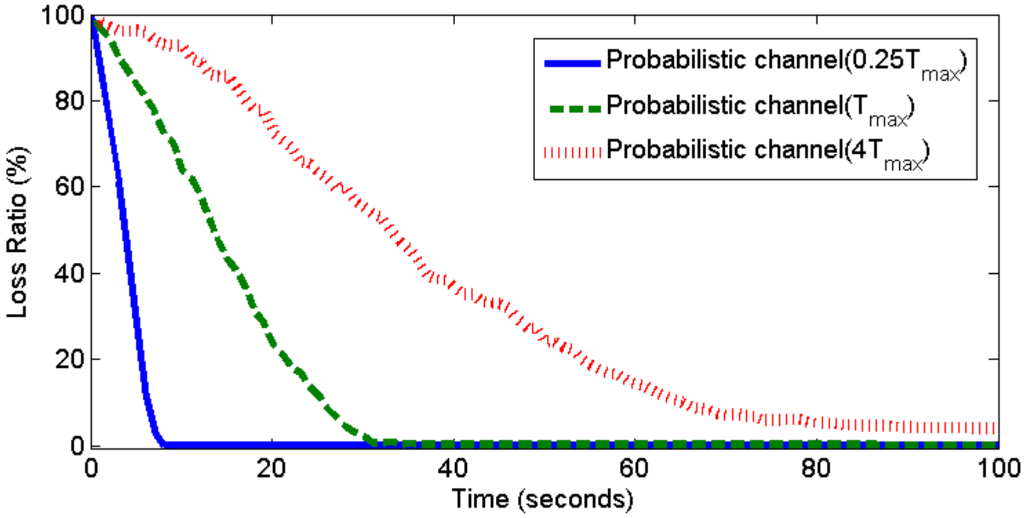
5.4.2 Effect of Maximum Defer Time

Next we study the effect of the maximum defer time T_{DMAX} in (5.1) to the performance of iDTSG. We consider here only the linear defer time case ($\epsilon = 1$) and only three values of the maximum defer times: $T_{DMAX} = 0.25T_{MAX}, T_{MAX}$, and $4T_{MAX}$ where $T_{MAX} = R/s_{max}$. Here we discuss the deterministic channel only. The results for the probabilistic channel are quite similar to the deterministic channel case as shown in Fig. 5.3-Fig. 5.4.

Fig. 5.3-Fig. 5.4 show the performance of varying the maximum defer times T_{DMAX} for dense and sparse scenarios. When the channel is deterministic with $R = 180$ and proba-



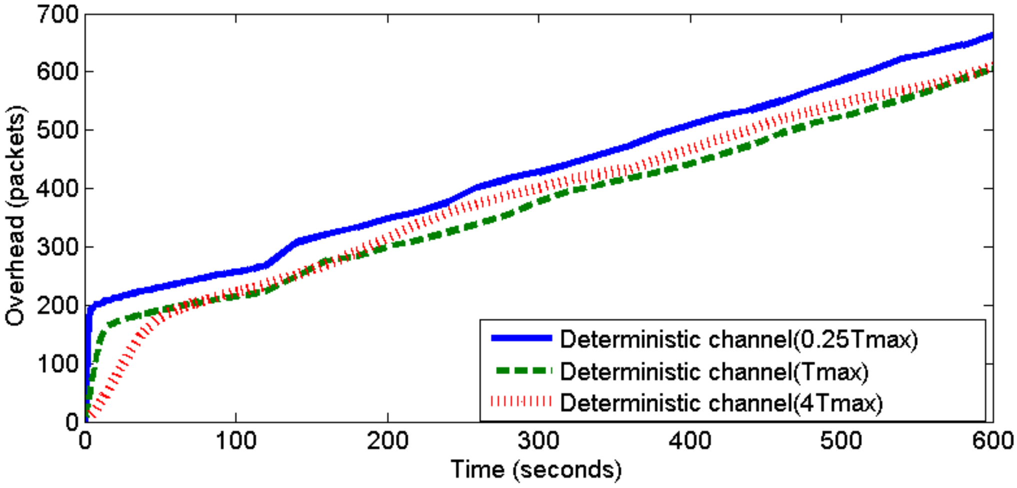
(a)Loss ratio in dense networks for deterministic channel



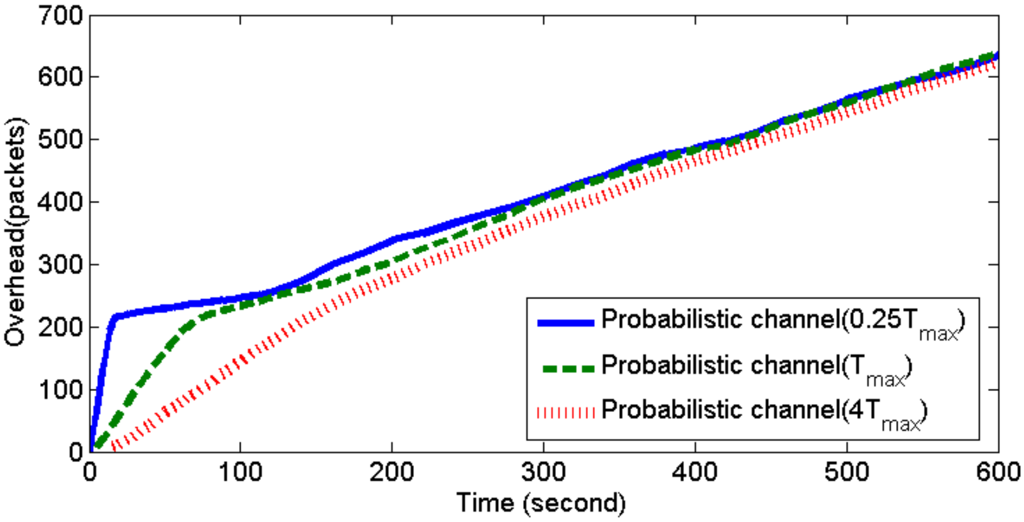
(b)Loss ratio in dense networks for probabilistic channel

Figure 5.3 Loss ratio for deterministic with $R = 180$m and probabilistic channels with $R = 300$m, under dense scenario

bilistic with $R = 300$, T_{DMAX} equals to $180/35 = 5.1$ s and $300/35 = 8.57$ s, respectively. In dense scenario where the average inter-vehicle spacing is 40m, Fig. 5.3 shows that a smaller T_{DMAX} gives a smaller T_D as expected hence a better loss ratio. Since the network is highly connected, the message can propagate to all of the intended vehicles within a few seconds. In deterministic channel, the time of message propagation to all of the intended vehicles is quite guaranteed by T_{DMAX} as shown in Fig. 5.3(a). When T_{DMAX} equals to $1.3, 5.1$, and 20.6 seconds, we require about $2, 6$, and 23 seconds to disseminate the message which it is similar to their T_{DMAX} values. However we require more than $5 - 6$ times of the dissemination time from the deterministic channel in the probabilistic channel as shown in Fig. 5.3(b). Hence we do not have a confidence in T_{DMAX} to guarantee the message propa-



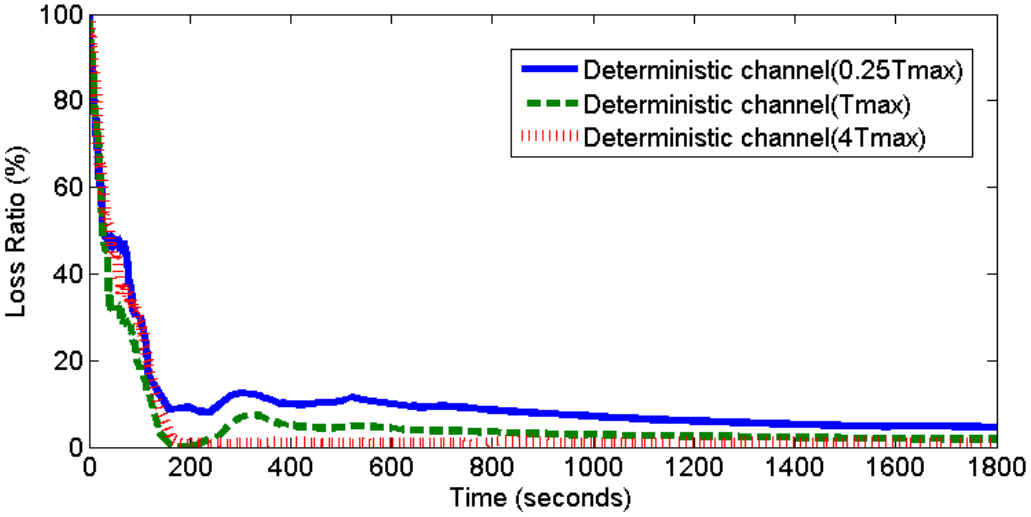
(a)Overhead in dense networks for deterministic channel



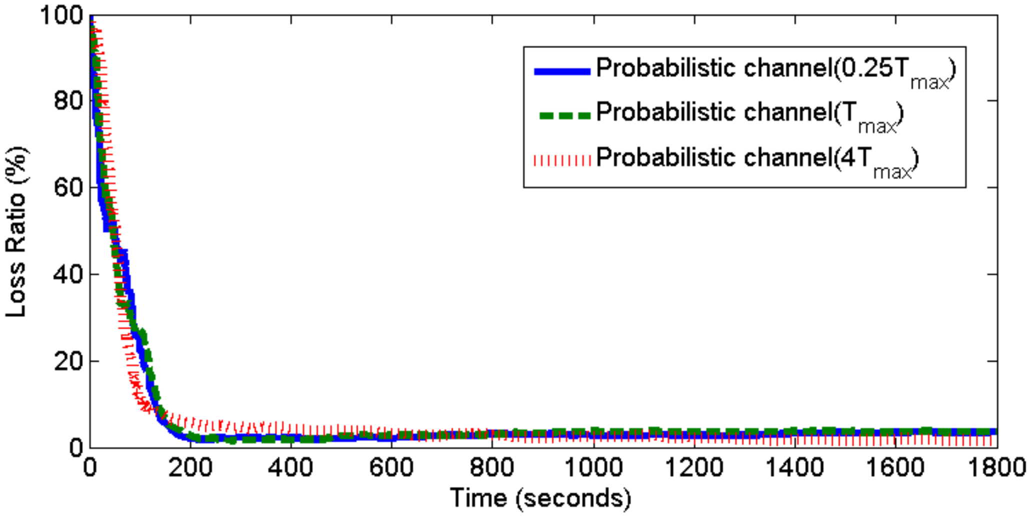
(b)Overhead in dense networks for probabilistic channel

Figure 5.4 Overhead for deterministic with $R = 180$m and probabilistic channels with $R = 300$m, under dense scenario

gation time in the probabilistic channel as it do in deterministic channel. From Fig. 5.4, the iDTSG protocol contains three phases: Phase1 is when the message is being disseminated all intended vehicles in the simulated highway section, Phase 2 is when the message has reached the end of the section but few new cars enter the section, and Phase 3 is the time-stable part when new cars are entering the section and iDTSG needs to inform them of the message. For example, for the $0.25T_{MAX}$ case, Phase 1 happens for the first few seconds, Phase 2 is after that until about 120 s, and Phase 3 is after 120 s. For dense scenario where nodes are highly connected in both directions, in Phase 1, the rate at which the rebroadcasts happen depends on the defer time T_D. Hence, in this phase the smaller T_{DMAX}, the higher the rate at which the message rebroadcasts happen (this is shown as the overhead). In Phase 2, there is a small number of rebroadcasts since all vehicles have received the message. In Phase 3, the rate



(a)Loss ratio in sparse networks for deterministic channel

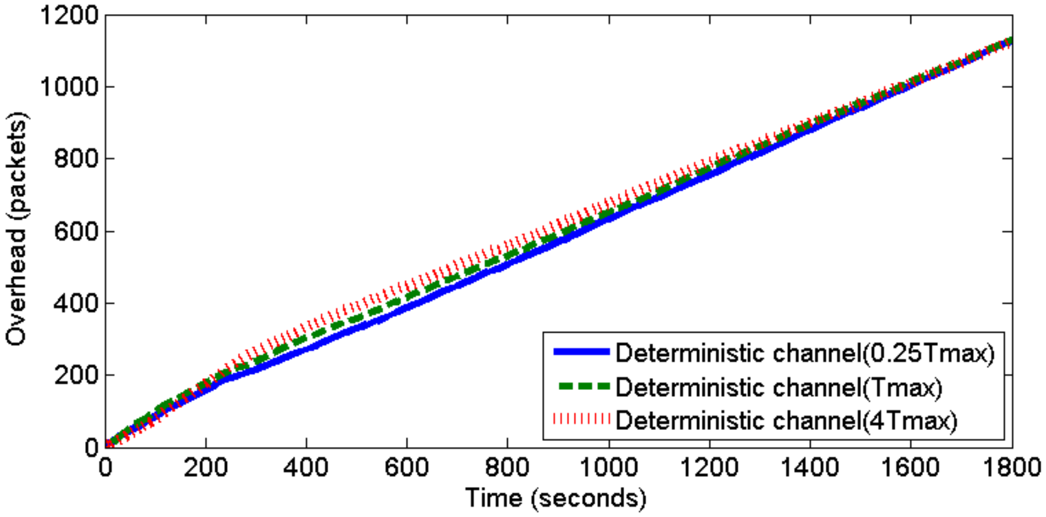


(b)Loss ratio in sparse networks for probabilistic channel

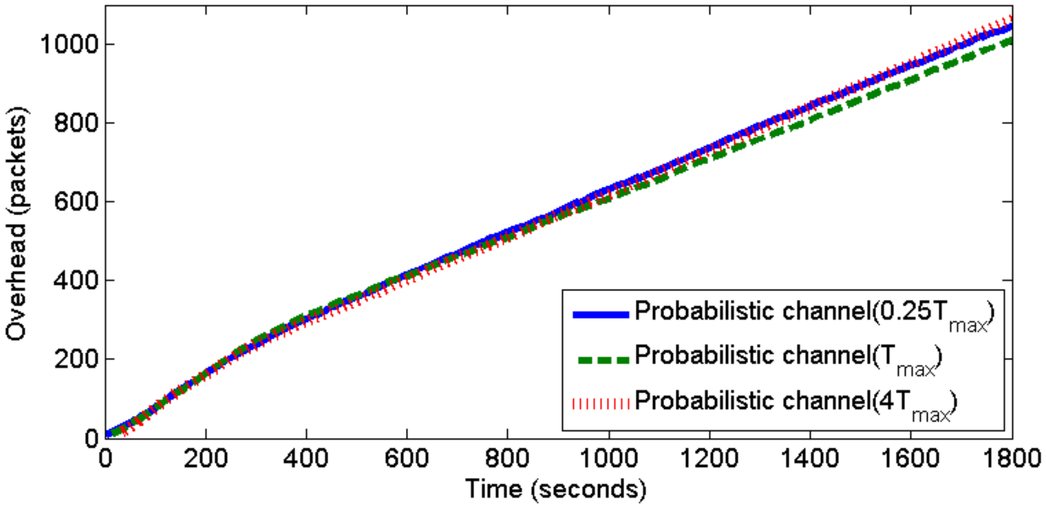
Figure 5.5 Loss ratio for deterministic with $R = 180$m and probabilistic channels with $R = 300$m, under sparse scenario

at which the rebroadcasts happen depends on the normal sleeping time T_N in 4.2 which is independent of T_{DMAX}, as shown in Fig. 5.4.

On the other hand, the results for sparse scenario (Fig. 5.5 and Fig. 5.6) are quite different from the dense scenario. Since the average inter-vehicle spacing is 250m, there are only very few cars or none in transmission range (recall $R = 180$m and $R = 300$m here) and hence the network is highly partitioned in either direction. Reduction T_{DMAX} does not help with the loss ratio much since the network is highly disconnected. In fact the loss rario is reduced via the help from helping vehicles in the opposite direction, carrying the message from one network cluster in the intended direction to another cluster in the same direction.



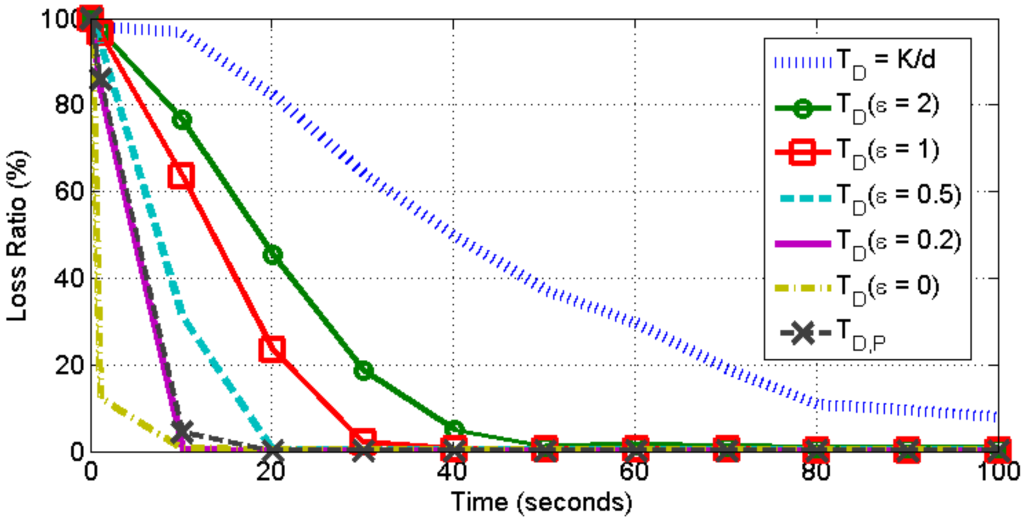
(a)Overhead in sparse networks for deterministic channel



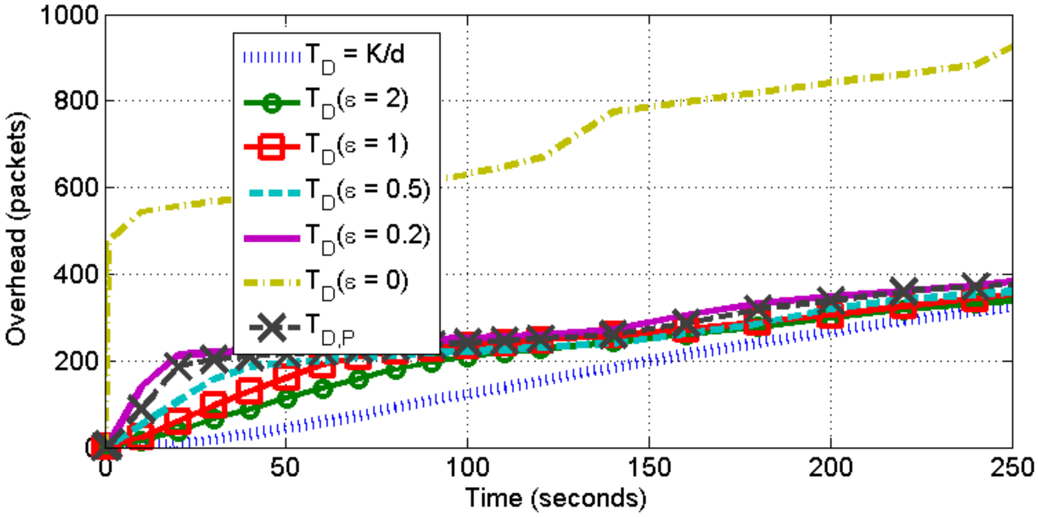
(b)Overhead in sparse networks for probabilistic channel

Figure 5.6 Loss rario and Overhead for deterministic with $R = 300$m and probabilistic channels with $R = 300$m, under dense scenario

Hence, as shown in Fig. 5.5(a) and (b), the longer T_{DMAX} is actually beneficial since this allows the helping vehicles to travel further before theirs rebroadcasts. In addition, the loss ratio for smaller T_{DMAX} does not reach zero even after a long time. For the overhead, the overhead at any time is almost independent of T_{DMAX} as shown in Fig. 5.6(a) and (b). This is also due to the highly partitioned network. For sparse network, there seems to be only two phases (Phase 1 and Phase 3, referred to the dense case) , where the phase changing time is around 200 s.



(a)Loss ratio in dense networks for D=3



(b)Overhead in dense networks for D=3

Figure 5.7 Performances of varying ϵ and shape of T_D for dense scenario.

5.4.3 Effect of Varying the Shape of Defer Time

Here we consider the effect of different shape of the defer times, specifically, the inverse $T_D = RT_N/d$ in the popular T_D in (4.1), the popular T_D in (5.1) with $\epsilon = 2, 1, 0.5, 0.2$ and 0, and the probabilistic channel $T_{D,P}$ in (5.2). Fig. 5.7 shows the performance of the above T_Ds in the dense scenario and the probabilistic channel with $R = 300$. The results show that the smaller the value of ϵ, the better the loss ratio, but at the cost of higher overhead. The flooding scheme $T_D(\epsilon = 0)$ gives the steepest decrease in the loss ratio but at the cost of high collisions and hence some vehicles did not get the message and the overhead is significantly much larger. The inverse T_D has the slowest decay in the loss ratio but the lowest overhead too. $T_D(\epsilon = 1)$ is worse than our proposed probabilistic channel $T_{D,P}$ and

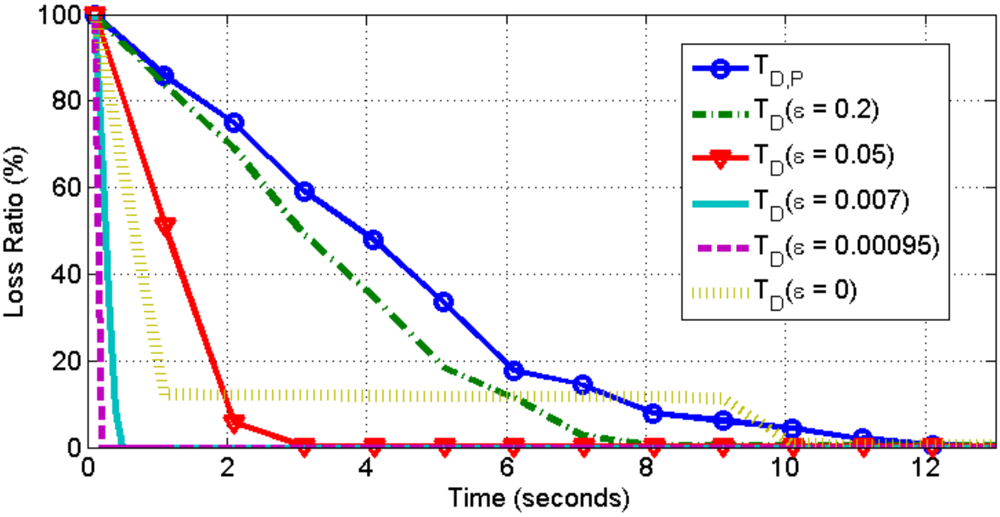
$T_D(\epsilon = 0.5)$ and $T_D(\epsilon = 0.2)$ since the overhead for all these cases are vary similar. Our proposed probabilistic-channel $T_{D,P}$ is also worse than $T_D(\epsilon = 0.2)$ in term of loss ratio.

5.4.4 Minimum Deterministic Defer Times

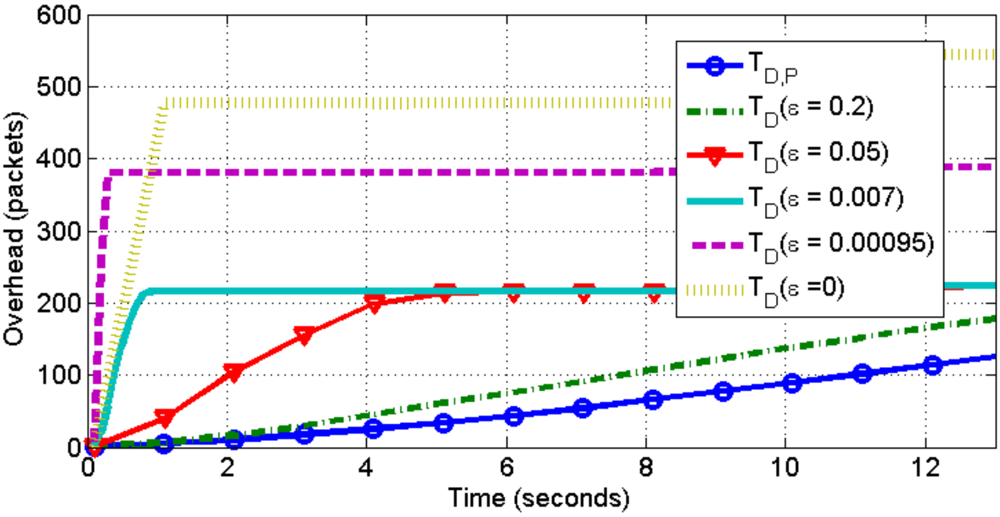
Here we evaluate our proposed a method in (5.4) to find a bound on deterministic defer times. To avoid packet collision as discussed in Section 5.2.1, we consider the minimal inter-vehicle spacing (Δ_{min}) to be 6 m, which gives ϵ_{min} to be 0.007 which is the minimum value to prevent flooding behavior. However, if Δ_{min} is changed to 40 m (i.e., the average inter-vehicle spacing in our simulation), the value of ϵ_{min} becomes 0.00095 which leads to flooding, as shown in Fig. 5.8(b). Since vehicles sometime get closer than the average spacing of 40 m, a vehicle may not be able to hear its neighbor's rebroadcast within its defer time and hence starts its own rebroadcast which results in collision. However, in probabilistic channel, each transmission range is not exactly the same; the packet collisions occur only in the overlap transmission ranges. This is why the loss ratio of $T_D(\epsilon = 0.00095)$ rapidly decreases to 0% within 0.1 s. Since the disparity of defer times between two closest vehicles in $T_D(\epsilon = 0.007)$ and $T_D(\epsilon = 0.05)$ exceeds the minimum packet transmission time, the loss ratio of $\epsilon = 0.007$ rapidly decreases in 0.2 s which it is faster than the $\epsilon = 0.05$ while the overhead is similar as shown in Fig. 5.8(a). For $T_{D,P}$ and $T_D(\epsilon = 0.2)$, the longer the defer times, the smaller the overhead and hence the slower the decline rate of the loss ratio. For the purpose of choosing the value of Δ, we see that if the value of Δ greater than the minimal inter-vehicle spacing is selected, there are many vehicles in this range and the defer time of each vehicle in this range is less than t_p which leads to the packet collisions. The parameter ϵ is inversely proportional to Δ. The distance d is proportional to ϵ. This criteria can be applied to calculate other design parameters (e.g., T_{DMAX}, R) while ϵ is fixed. T_{DMAX} can decrease until the slope or the disparity between defer times of two closest neighbors is equal to t_p.

5.4.5 Effect of Stochastic Defer Time

Finally, we compare the performance of stochastic defer times and deterministic defer times. We select deterministic $T_D(\epsilon = 1)$ and $T_D = 0$ to compare with stochastic $T_D(\epsilon = 1)$ and $T_S = T_{max}$, respectively. As shown in Fig. 5.9(a), the stochastic defer times outperform the deterministic defer times in reliability. The loss ratio of the stochastic defer time $T_D(\epsilon = 1)$ decreases faster than the deterministic defer time $T_D(\epsilon = 1)$. The overhead of the stochastic T_{max} is less than the deterministic $T_D = 0$ and is similar to the others. The dissemination time of T_{max} is quite longer than the stochastic $T_D(\epsilon = 1)$ because the variance of the defer times for stochastic T_{max} is much higher. Note that the shown plot is a result of averaging 10 independent runs. It happens that in some runs, many nodes pick small defer times and hence the dissemination time is small, while in other runs, larger defer times are



(a)Loss Ratio for deterministic defer time

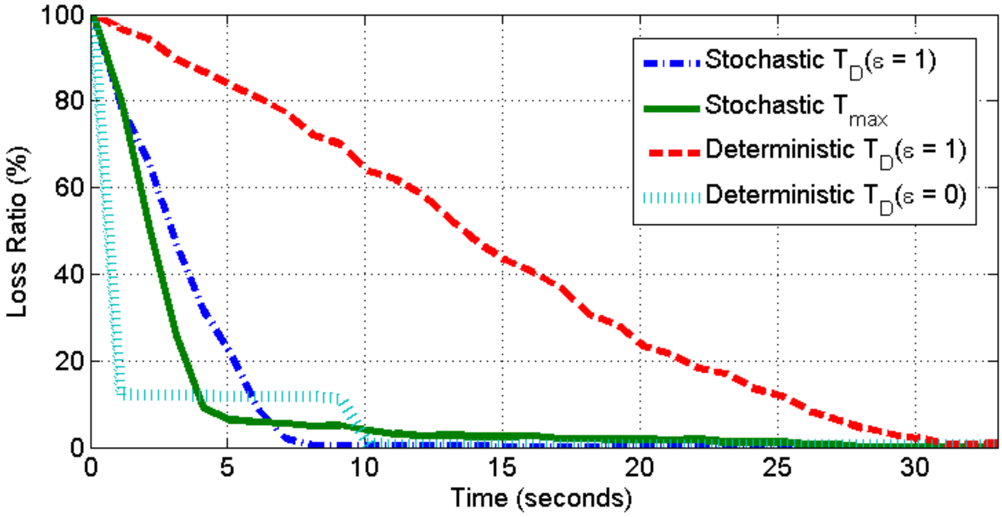


(b)Overhead for deterministic defer time

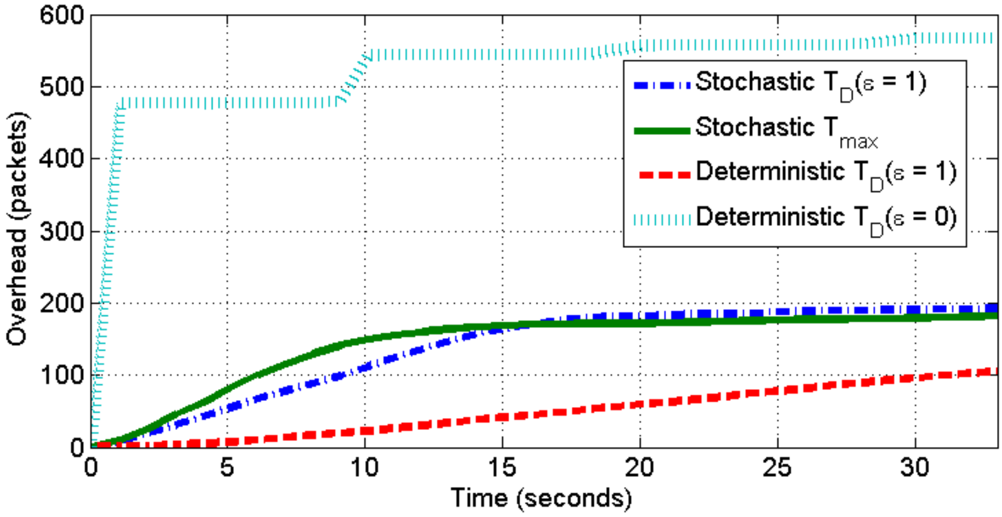
Figure 5.8 Performance of critical deterministic defer time T_D for probabilistic channel and dense scenario

picked more often and hence the dissemination time is larger. Hence as shown in Fig. 5.9(a), the average of loss ratio of stochastic T_{max} does not sharply decrease to 0%.

In the flooding scheme ($\epsilon = 0$), the packet collisions occur in the first broadcast and all nodes rebroadcast again after expiring of the normal sleep time of iDTSG which depends on the speed of senders. Since speed of senders is usually different, the second retransmission packet is likely to succeed. This is why the loss ratio drops to 18% but stable afterward for a long duration. The loss ratio of $T_D(\epsilon = 0)$ becomes 0% after 10 sec. However, the overhead of $T_D(\epsilon = 0)$ is the highest. Hence, we have seen that the stochastic defer times increase reliability while they barely incur any extra overhead. Hence, applying the stochastic defer time is beneficial in our iDTSG protocol. A caution is that the stochastic defer times increase



(a)Loss Ratio for stochastic defer time



(b)Overhead for stochastic defer time

Figure 5.9 Performance of stochastic defer time T_S and deterministic defer time T_D for probabilistic channel and dense scenario

the reliability if only T_{DMAX} is much longer than the minimum defer time. If the disparity of defer times between two closest neighbors is close to the packet transmission time, the stochastic defer times actually can be worse than the deterministic defer times. Due to the fact that all vehicles have a chance to rebroadcast when nodes receive a broadcasting packet, we believe that only simple suppression mechanism (i.e, the vehicles stop rebroadcast when they receive duplicate messages) is not sufficient for broadcast storm suppression in the first stage of iDTSG protocol. We will look into combining this mechanism with geo-broadcast or limiting the maximum number of hops to enhance broadcast storm suppression in the future.

Chapter 6

Estimation of the Average Inter-Vehicle Spacing

6.1 Simple Estimation Mechanism

Since in practice the average inter-vehicle spacing is difficult to be determined a priori and furthermore this value varies over the highway, the iDTSG protocol makes a simple estimation of the average inter-vehicle spacing in the extra region, by taking a sample from two consecutive messages a node received from the other two opposite-lane nodes. The density of the vehicles in one lane is introduced in this work. Specifically, Fig. 6.1 illustrates this estimation procedure performed by an intended node (I1) moving to the right. If I1 receives a message from a helping vehicle (H1) moving to the left at time t_1. I1 then knows from the message content, the location x_1 and speed v_1 of H1 at time t_1. If at time t_2, I1 receives another message from another helping node H2. After three of those vehicles continue moving along the road, I1 receive the alarm message from the helping vehicle (H2) moving to the left at time t_2. From the message of H2, I1 then knows the location x_3 and

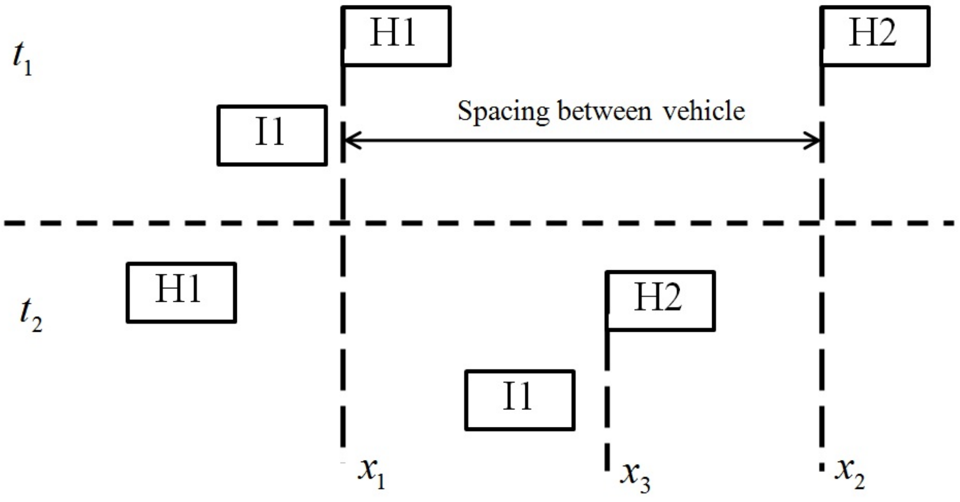


Figure 6.1 Simple estimation of the average inter-vehicle spacing

speed v_2 of H2 at time t_2. By assuming that H2 drives with a constant speed during time t_1 and t_2, I1 can estimate the spacing between H1 and H2 as follows:

$$\text{inter-vehicle spacing} \approx |x_1 - x_3 + v_2(t_2 - t_1)|. \tag{6.1}$$

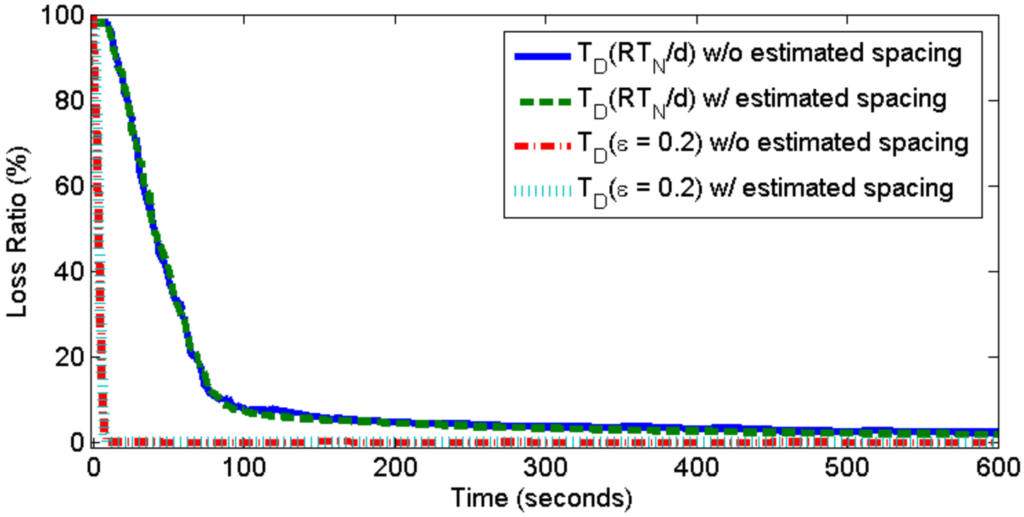
Hence, the spacing between H1 and H2 can be calculated and taken as the estimate of the average inter-vehicle spacing. The average inter-vehicle spacing updates value as follows: inter-vehicle spacing $= \alpha(\text{inter-vehicle spacing})_{old} + (1-\alpha)(\text{inter-vehicle spacing})_{new}$. The α is the parameter to balance the previous inter-vehicle spacing and the new inter-vehicle spacing information. This is likely to be valid for sparse scenarios. However, for dense scenarios, there would likely be some other vehicles between H1 and H2 and hence the spacing between H1 and H2 would be larger than the actual inter-vehicle spacing. This is not harmful to the loss ratio of the protocol since the extra region would just be longer than necessary. In iDTSG, the initial value of D_{extra} is set to 20 m then an spacing estimation is done. D_{extra} is updated as $2 \times inter - vehicle\ spacing$. [3] If no estimation can be done does not happen, the extra region will be the same as that in the message content.

6.2 Simulation Results

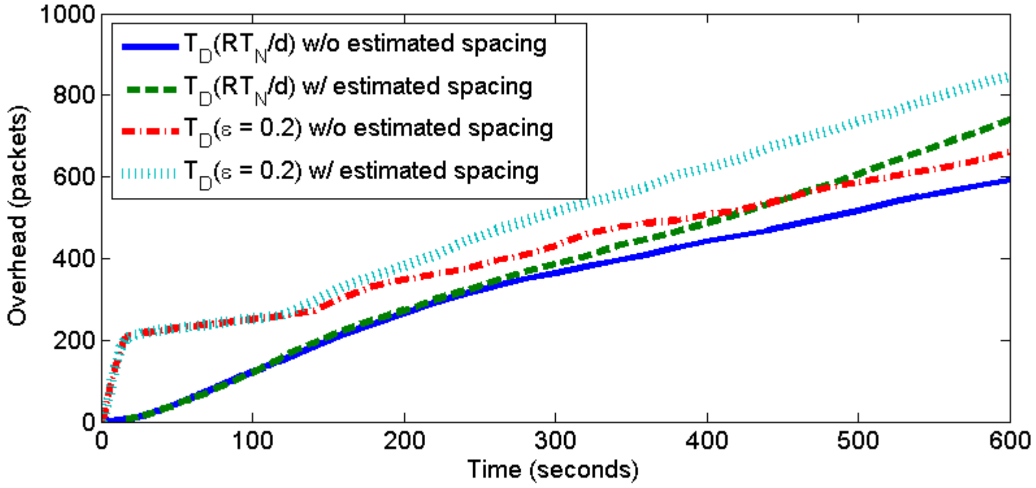
After applying the simple estimation of the average inter-vehicle spacing to the iDTSG protocol, we evaluate the performance by using the $T_D(RT_N/d)$ and $T_D(\epsilon = 0.2)$ in dense and sparse networks for $D = 3$ km and $\alpha = 0.5$. The performance metrics are the loss ratio and the overhead which are stated in the Chapter 5. As shown in Fig. 6.2 (a), the loss ratio of the $T_D(RT_N/d)$ does not reach 0% within 100 seconds in dense networks. The loss ratio of the $T_D(\epsilon = 0.2)$ reaches 0% within 10 seconds in dense networks. On the other hand, the overhead of the $T_D(\epsilon = 0.2)$ is greater than the overhead of $T_D(RT_N/d)$ by 12% at 600 seconds. The overhead of the $T_D(\epsilon = 0.2)$ is greater than the overhead of the $T_D(RT_N/d)$ in dense networks, although the overhead of the $T_D(\epsilon = 0.2)$ is less than the overhead of the $T_D(RT_N/d)$ in sparse networks. Fig. 6.3 illustrates that the iDTSG loss ratio of $T_D(\epsilon = 0.2)$ saturates at 7% while the loss ratio of $T_D(RT_N/d)$ keeps lower than 5%. The simple estimation of the average inter-vehicle spacing is usable in both dense and sparse networks. Sometimes the estimation are too small so the loss ratio increases, while the overhead decreases. However, the overhead (when the protocol uses the simple estimation of the average inter-vehicle spacing) is too high because the average inter-vehicle spacing is mostly overestimated. This simulation does not take an error from GPS into account and hence the average inter-vehicle spacing is not accurate. Since the iDTSG protocol uses a position info from GPS, the accuracy of GPS would affect the performance of the iDTSG protocol. Our simple estimation of the average inter-vehicle spacing has to be improved to tolerate GPS errors and to be more accurate. This is left as a future work.

---

[3] A more accurate but more complicated estimate of the average inter-vehicle spacing would be a running average of all the values of the inter-vehicle spacing that a node has observed so far.

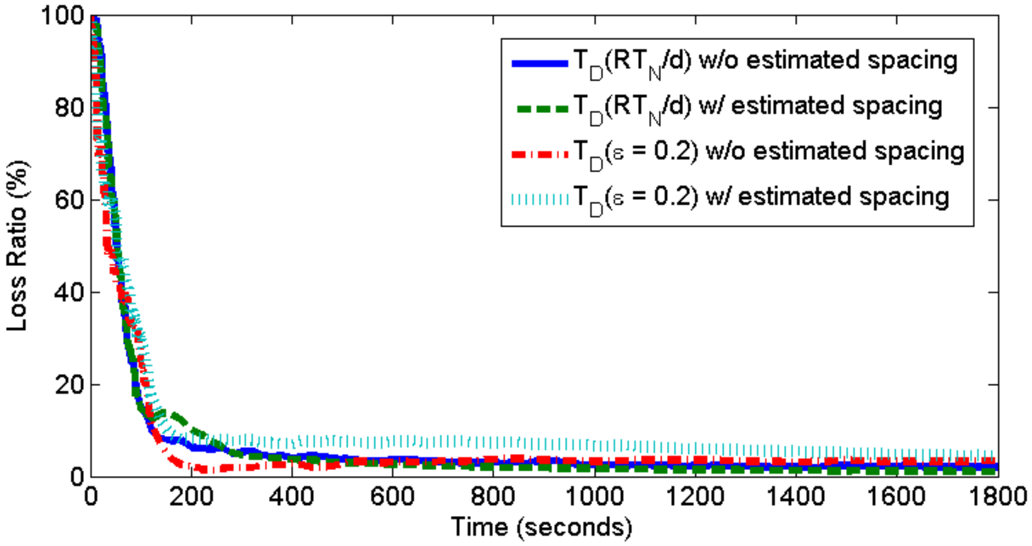


(a)Loss Ratio

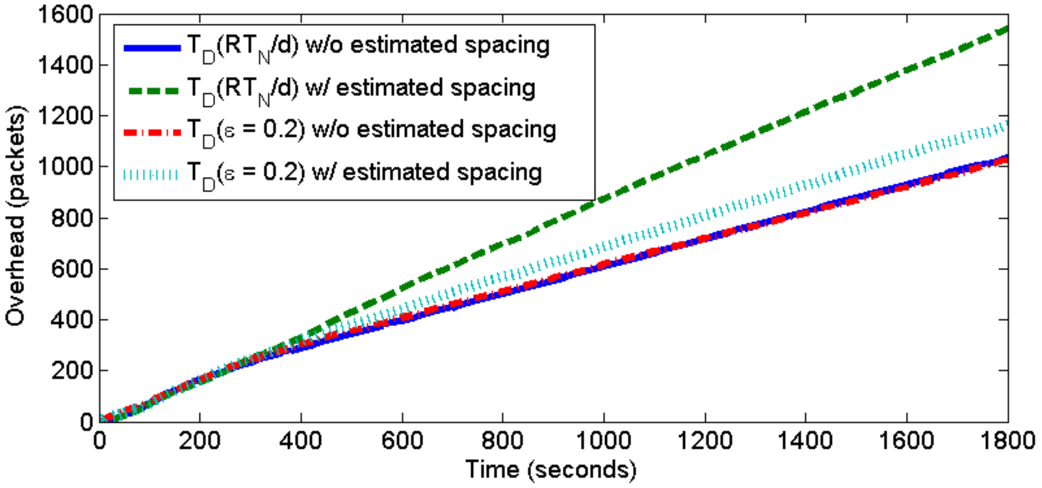


(b)Overhead

Figure 6.2 Performance metric for dense networks of iDTSG with estimation spacing.



(a)Loss ratio



(b)Overhead

Figure 6.3 Performance metric for sparse networks of iDTSG with estimation spacing.

Chapter 7

Summary and Future Works

Vehicle ad-hoc networks enhance transportation safety at low implementation and operation cost. In such high speed movement of vehicles in highway environment, the vehicles are equipped with WAVE transceivers and GPS. Each driver make decision about the emergency events which have been informed by neighbor vehicles along the road in VANETs. In the post crash notification, the emergency or warning message is disseminated to prevent chain-collision or congested traffic until the accident has been cleared from the highway. To meet this objective, the time-stable geocast protocol is designed to disseminate and to maintain the warning message for the sake of the new approaching vehicles in the specific area for a certain time.

7.1 Summary

In this thesis, we propose a new time-stable geocast protocol, called iDTSG. The iDTSG protocol is the multi-hop broadcast protocol which considers the trade-off between reliability, dissemination time, and efficiency or redundancy of the message. The aim of iDTSG is to maximize the number of informed nodes in the intended region with a fast and a reliable transmission while network loads are minimized. iDTSG uses the contention-based forwarding scheme to select forwarders, the counter-based scheme to suppress broadcast storm problem, and the location-based scheme to keep the message alive in the intended region.

By simulation in ns-3, we compared the performance of the iDTSG protocol with the DTSG protocol in realistic channel and highway mobility models. Furthermore, we evaluated the effects of the design parameters (T_{DMAX}, ϵ, and R) for the distance-based defer times and the shapes of the defer times with the iDTSG protocol. In addition to the evaluation of the previously proposed defer times, we proposed and evaluated a new defer time which is more appropriate with probabilistic channels. We also proposed the method for calculating the deterministic defer time to avoid collisions for a bi-directional single lane highway. We considered both dense and sparse traffic scenarios and both deterministic and probabilistic channels. The simulation results show the following:

1.) Our iDTSG achieved almost 0% for the loss ratio, same as DTSG but at a much lower cost of rebroadcast packets at least 20% and 30% in dense and sparse density connected networks, respectively. The reliability of iDTSG did not depend on the geocast region,

2.) Trade-offs between reliability and efficiency.

3.) Reducing the deterministic defer time (by reducing T_{Dmax} or ϵ) increases the reliability at the cost of the efficiency;

4.) The deterministic distance-based defer time has a optimal design value of ϵ to avoid collision.

5.) Although our proposed defer time takes into account the different reception probability at different distant and hence performs well, our proposed defer time can still be improved in the future by considering the fact that nodes suppress their rebroadcasts if they hear prior rebroadcasts.

6.) Furthermore, the stochastic distance-based defer time is more appropriate than the deterministic distance-based defer time in case of increasing reliability if the parameter T_{max} is larger than necessary.

7.) iDTSG gives high reliability and high efficiency. Since iDTSG requires knowledge of the local traffic density, we provided a simple density estimation algorithm. The results showed that the estimation performed well.

7.2 Recommendations and Future Works

In the Chapter 5, we proposed the method to find a bound on minimum deterministic defer times in a bi-directional single-lane highway. We showed that the $\epsilon = 0.007$ and the $\epsilon = 0.00095$ are the bounded value for $T_{max} = R/s_{max}$. In this section, we apply the same method to find a bounded T_{max} for $\epsilon = 1$. The bounded T_{max} are 0.09 and 0.02 seconds. Fig. 7.1 illustrates defer times for $T_{max} = 0.09$ seconds with varying ϵ values and $T_{max} = 0.02$ seconds.

The results in Fig. 7.2 show the trade-off between loss ratios and overheads. The loss ratios of convex defer times are better than the linear defer time. The overheads of the $T_{D,P}$ and the $T_{max=0.09}$ are similar. Due to the fact that the $T_D(\epsilon = 0.2)$ has a similar defer time of the $T_{max} = 0.02$ in a far distance. Hence, the $T_D(\epsilon = 0.2)$ incurs the extra overhead the same as the $T_{max} = 0.02$. This means that the $T_D(\epsilon = 1)$ with $T_{max} = 0.09$ is not an optimal value as we think from the calculation. However, we are able to determine the bounded defer time in the bi-directional single-lane highway from the calculation.

Since the protocol has to work in bi-directional multiple-lane highways, our protocol has to consider following issues:

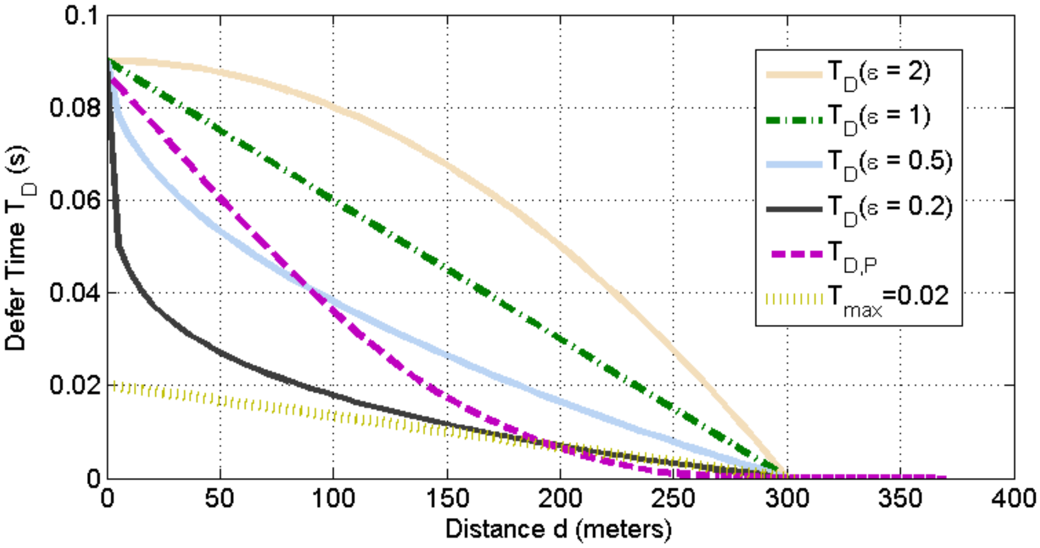


Figure 7.1 Bounded defer times

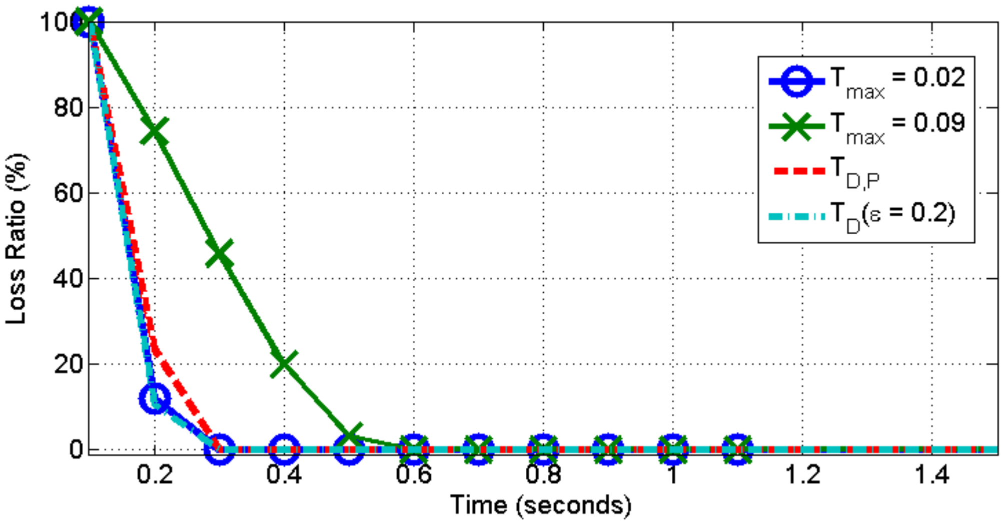
1.) Since the protocol parameter depends on the location from GPS, the accuracy of GPS would effect the performance of the protocol.

2.) Multiple crash vehicles can simultaneously occur and hence the protocol has to work with multi-sources broadcasting an alarm message.

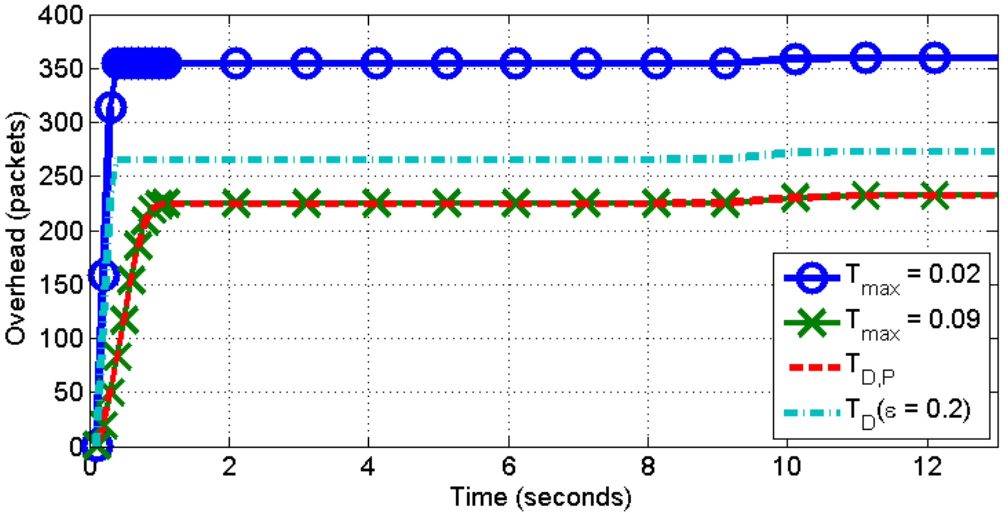
3.) The calculation of the bounded minimum defer times can be wrong in a bi-directional multiple-lane highway.

4.) Our simple estimation of the average inter-vehicle spacing has to be improved to tolerate GPS errors and to update the average inter-vehicle spacing.

5.) By considering the fact that all vehicles have a chance to broadcast if they hear the first message in broadcast storm suppression part of iDTSG protocol. Efficiency can still be improved in the future by combining this mechanisms with geo-broadcast or limiting the maximum number of hops to enhance broadcast storm suppression.



(a)Loss Ratio for bounded defer time



(b)Overhead for bounded defer time

Figure 7.2 Performance of bounded deterministic defer time T_D for probabilistic channel and dense scenario

Appendix A

NS-3 code

Since H. Arbabi [40] proposed the highway mobility in ns-3, the simulations of vehicular ad hoc networks do not require complicated interfaces between the network simulator and the mobility simulator. A simplified schematization of the whole simulator is shown in Fig. A.1 which we used to perform evaluations of our work. The highway mobility moves the nodes and changes their states following the rules given by the IDM mobility model and the MOBIL lane-change model, while user-defined and a controller class are created to handle events and to create special vehicles. For example, after the vehicle is injected to the highway to act as the source which initiates the disseminated message, the receivers call the iDTSG protocol in vehicle class. The following sections show in details C++ codes which are modified from [40].

A.1 Controller class

```
    #include <iostream>
#include <sstream>
#include “Controller.h”
using namespace std
using namespace ns3
namespace ns3{
int all =0;
int success = 0;
int broadcast=0;
int packeterror =0;
int packetreceive=0;
int packetsent=0;
bool ch = false;
int bID = 0;
double fi =0;
bool h = true;
double nnn =0;
```

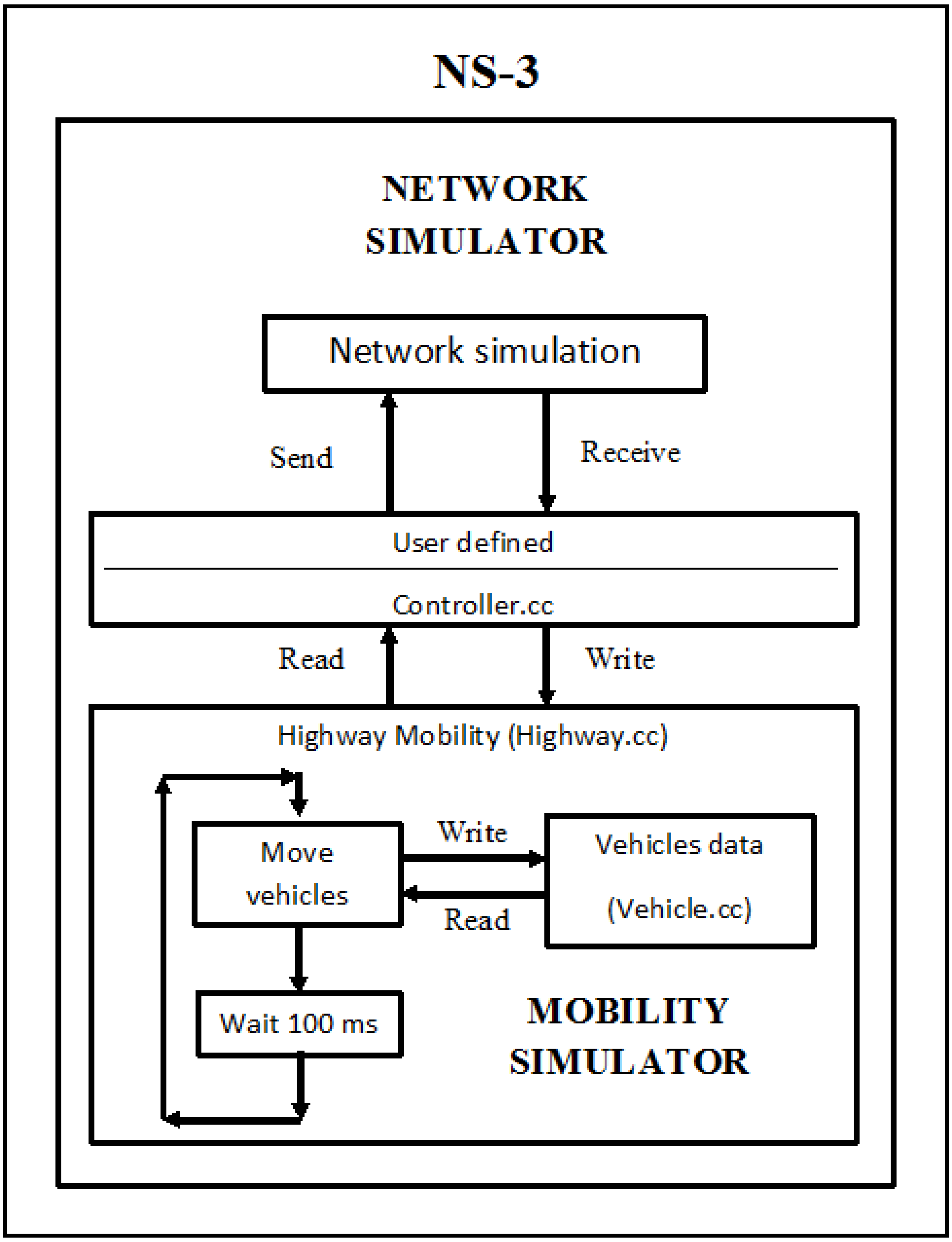


Figure A.1 Structure of the simulator

```
    Controller::Controller(){
T= −1.0;
Plot=false; }
Controller::Controller(Ptr<Highway> highway) {
this->highway=highway; }
void Controller::SetHighway(Ptr<Highway> highway) {
this->highway=highway; }
Ptr<Highway> Controller::GetHighway(){
return this->highway; }
bool Controller::Injectbroken(Ptr<Highway> highway, int& VID)
{ YansWifiPhyHelper sourcePhyHelper=YansWifiPhyHelper::Default();
sourcePhyHelper.SetChannel(highway->GetWifiChannel());
sourcePhyHelper.Set(“TxPowerStart”,DoubleValue(5));
sourcePhyHelper.Set(“TxPowerEnd”,DoubleValue(5));
sourcePhyHelper.Set(“TxPowerLevels”,UintegerValue(1));
sourcePhyHelper.Set(“TxGain”,DoubleValue(0));
sourcePhyHelper.Set(“RxGain”,DoubleValue(0));
sourcePhyHelper.Set(“EnergyDetectionThreshold”, DoubleValue(-87.0));
sourcePhyHelper.Set(“CcaMode1Threshold”, DoubleValue(-83));
sourcePhyHelper.Set(“RxNoiseFigure”, DoubleValue(7));

Ptr<Vehicle> source = CreateObject<Vehicle>();
source->SetupWifi(highway->GetWifiHelper(),
sourcePhyHelper, highway->GetNqosWifiMacHelper());
source->SetVehicleId(VID++);
source->SetDirection(1);
source->SetPosition(Vector(0.0, highway->GetYForLane(0,1), 0));
source->SetVelocity(0.0);
source->SetAcceleration(0.0);
Ptr<Model> sourceModel=highway->CreateSedanModel();
sourceModel->SetDesiredVelocity(30.93);
source->SetModel(sourceModel);
source->SetLaneChange(highway->GetSedanLaneChange());
source->SetLength(4);
source->SetWidth(2);
source->SetReceiveCallback(highway->GetReceiveDataCallback());
highway->AddVehicle(source);
bID = source->GetVehicleId();
highway->SetAutoInject(true);
return false;
}
```

```
bool Controller::InitVehicle(Ptr<Highway> highway, int& VID) {
Simulator::Schedule(Seconds(500.0), &Controller::Injectbroken, this,highway, VID);
return false;
}


bool Controller::ControlVehicle(Ptr<Highway> highway, Ptr<Vehicle> vehicle, double dt)
{

    if(Plot==true)
{ bool newStep=false;
double now=Simulator::Now().GetDouble();

    if(now > T)
{ T = now;
newStep=true; }

    if(newStep==true && ch == true ) { newStep=false; }

    if(newStep==false && ch == true )
{ Controller::DeliveryRatio(vehicle);
}
}

    if(vehicle->GetVehicleId()==bID && vehicle->GetPosition().x >=6500 && ch ==
false)
{ double t = Simulator::Now().GetSeconds();

    if (ch == false)
{ ch = true;
Controller::BroadcastWarning(vehicle);
Controller::DeliveryRatio(vehicle);
Controller::Printresult(t) ;
}
return false;
}
return false;
}


    void Controller::BroadcastWarning(Ptr<Vehicle> veh)
static double now=0;
static int crash_pos = veh->GetPosition().x ;
```

```
    if (veh->GetSequencenumber() < 1)
{ veh->SetFlag(1);
now=Simulator::Now().GetSeconds();
}

    if (veh->GetAckfirst(0) == true ){ f = veh->Getfirst(2); }
if (veh->GetCount() < 1 && veh->GetPosition().x < 9600)
{ stringstream msg;
msg << veh->GetVehicleId()
<< “ ” << veh->GetSequencenumber()
<< “ ” << veh->GetPosition().x
<< “ ” << veh->GetDirection()
<< “ ” << crash_pos
<< “ ” << veh->GetDirection()
<< “ ” << veh->GetFlag()
<< “ ” << 3000 // The D region is 3 km. << “ ” << now
<< “ ” << 20 // The initial value of inter-vehicle spacing << “ ” << veh->GetVelocity();
uint16_t packetSize = 800;
Ptr<Packet> packet = Create<Packet>((uint8_t*) msg.str().c_str(), packetSize);
veh->SendTo(veh->GetBroadcastAddress(), packet);
veh->SetSequencenumber();
Simulator::Schedule(Seconds(veh->GetWaitingTime(true)),&Controller::BroadcastWarning,
this,veh);
}
}


    void Controller::ReceiveData(Ptr<Vehicle> veh, Ptr<const Packet> pk, Address address)
{ int direction = veh->GetDirection();
int vid=veh->GetVehicleId();

    if (veh->IsEquipped==false) {return;}
uint8_t *buffer = new uint8_t[pk->GetSize()];
pk->CopyData (buffer, pk->GetSize());
stringstream ss (stringstream::in | stringstream::out);
string data = string((char*)buffer);
int sender_id, message_sequence, sender_direction, event_direction, period_flag, D_region ;
double sender_pos, event_pos, event_time, speed, inter_spacing;
ss << data;
ss >> sender_id;
ss >> message_sequence;
```

```
ss >> sender_pos;
ss >> sender_direction;
ss >> event_pos;
ss >> event_direction;
ss >> period_flag;
ss >> D_region;
ss >> event_time;
ss >> inter_spacing;
ss >> speed;

    if(veh->GetDirection() == 1)
{ density1 = (density1*0.5)+(dense*0.5); }
else if(veh->GetDirection() == -1){ density2 = (density2*0.5)+(dense*0.5); }
fi = event_pos − D_region;
veh->SetCount(veh ,pk);
if (period_flag==2){veh->SetFlag(2); }

    if (direction == −1&& sender_direction == 1 )
{

    if(veh->Checkccc() == 0)
{ veh->SetintiX(sender_pos, now, sender_id); veh->Setccc();}

    if (veh->Checkccc() == 1 )
{ veh->SetExtraRangehelping (sender_pos, now, speed, sender_id);
veh->Setccc(); }
} else if( direction == 1 && sender_direction2 == −1 )
{ if(veh->Checkccc() == 0) veh->SetintiX(sender_pos, now, sender_id);
veh->Setccc(); }

    if (veh->Checkccc() == 1 )
{ veh->SetExtraRangeintended(sender_pos, now, speed, sender_id );
veh->Setccc(); }
}

    if (vid=bID && veh->GetCount() ==1)
{ veh->SetWaitingTime(veh->GetVelocity(), sender_x2, veh->GetPosition().x );
if (period_flag == 1)veh->SetFlag(1);
veh->SetExtra(inter_spacing);
veh->iDTSG(veh, pk, address); }

    if (veh->GetPosition().x > event_pos && veh->GetDirection() == 1)
{
if(sender_direction2 == −1){veh->stableOpcount();}
```

```
else if (sender_direction== 1 and space < 0){veh->FrontStableCount();}
}
else if(veh->GetPosition().x < fi && veh->GetDirection() == −1)
{ if(sender_direction == 1){veh->stableOpcount();}
else if (sender_direction == −1 and space > 0){veh->FrontStableCount();}
}
}

    void Controller::DeliveryRatio(Ptr<Vehicle> veh)
{ double veh_pos = veh->GetPosition().x;

    if(veh->IsEquipped==false){ return;}
broadcast= veh->Broadcastcount();
packeterror = veh->errorcount();
packetreceive = veh->Macreceivecount();
packetsent = veh->Macsentcount();

    if (x <= 6381 && x >= 3501 && veh->GetDirection()==1 && veh->GetRange() ==
false)
{ all = all+1;
veh->SetRange(); }

    if ( veh->GetCount() >=1 && x <= 6381 && x >= 3501 && veh->GetDirection()==1
&& veh->GetRange() == true && veh-> GetSuccess() == false)
{ success = success+1;
veh->SetSuccess(); }
}

    void Controller::Printresult(double Time)
{ double now=Simulator::Now().GetSeconds();
now = (now-Time);
int lost = all−success;

    if(now >= 120){nnn = 20;}
else if ( now >= 0.1 ){nnn = 1;}
else{nnn = 0.1;}

    std::cout<< now << “ ”<< lost << “ ”<< broadcast << “ ” << all << “ ” <<
success << “ ” << packeterror << “ ” << packetreceive << “ ” << packetsent << ”;”
<< endl;
Simulator::Schedule(Seconds(nnn),&Controller::Printresult, this, Time);
}
}
```

A.2 Vehicle class

The following functions are added into the vehicle class of the [40]’ work.

```
namespace ns3 { int bc =0;
int packeterror =0;
int m_packetsSent1 =0;
int m_packetReceive=0;

    void Vehicle::SetSequencenumber(void) { sequencenum++; }

    int Vehicle::GetSequencenumber(void) { return sequencenum; }

    void Vehicle::SetFlag(int value1) { period_flag = value1; }

    int Vehicle::GetFlag(void) { return period_flag; }

    void Vehicle::SetRange(void) { range = true; }

    void Vehicle:: SetSuccess(void) { success = true; }

    bool Vehicle::GetRange(void) { return range; }

    bool Vehicle::GetSuccess(void) { return success; }

    bool Vehicle:: Checkccc(void) { return ccc; }

    void Vehicle:: Setccc(void) { ccc = true; }

    bool Vehicle::Getyyy(void) { return yyy; }

    void Vehicle::SetExtraRangeintended(double x2, double t, double speed, int iid)
{

    if(send_id= iid)
{ int delta = t-ti;
int dense = (inti_x - (x2+(speed*delta)) ) ;
if (dense < 0){dense = (0 - dense);}
density = (0.5*dense)+(0.5*density) ;
yyy=true;
inti_x = x2;
ti = t;
}
}

    void Vehicle::SetExtraRangehelping (double x2, double t, double speed, int iid) {
```

```
    if (send_id= iid)
{ int delta = t-ti;
int dense = inti_x - (x2-(speed*delta)) ;
if (dense < 0){dense = (0 - dense);
density = (0.5*dense)+(0.5*density) ; yyy=true;
inti_x = x2;
ti = t;
}
}

    void Vehicle::SetExtra(double duu) { density = duu; }

    void Vehicle::SetintiX(double x,double t, int iid) { inti_x = x;
ti = t;
send_id = iid;
}

    int Vehicle::checkdensity(void) { return density ; }

    void Vehicle::SetWaitingTime(void)
{ double speed = m_velocity ;
repeat_Time = (600(speed+35)); }

    double Vehicle::GetWaitingTime(bool value)
{ switch (value)
{ case true:
return repeat_Time;
break;
case false:
return defer_Time;
break;
default:
NS_FATAL_ERROR (“No such value:” << value); }
}

    void Vehicle::SetWaitingTime(double speed, double position_source,
double position_receiver)
{ int diff = position_source - position_receive;

    if(diff <){diff = 0-diff;}
if(diff > 10) { diff = diff; }
else if (diff < 10) { diff = 10; } defer_Time = ((600(speed+35))*(300((double)diff))); }

    int Vehicle::GetBackCount(void) { return back_count; }
```

```
    int Vehicle::GetOpCount(void) { return op_count; }

    int Vehicle::GetLeaderCount(void) { return leader_count; }

    int Vehicle::GetCount(void) { count =leader_count+op_count+back_count;
return count; }

    void Vehicle::SetCount(Ptr<Vehicle> veh, Ptr<const Packet> packet) { uint8_t *buffer
= new uint8_t[packet->GetSize()];
packet->CopyData (buffer, packet->GetSize());
stringstream ss (stringstream::in | stringstream::out);
string data = string((char*)buffer);
int sender_id, message_sequence, sender_direction, event_direction, period_flag, D_region ;
double sender_pos, event_pos, event_time;
ss << data;
ss >> sender_id;
ss >> message_sequence;
ss >> sender_pos;
ss >> sender_direction;
ss >> event_pos;
ss >> event_direction;
ss >> period_flag;
ss >> D_region;
ss >> event_time;
int direction = veh->GetDirection();
double space = ( veh->GetPosition().x - sender_x);

    if(direction == 1 and direction == sender_direction and space < 0) { leader_count++; }
else if ( direction == 1 and direction == sender_direction and space > 0 ) { back_count++; }
else if ( direction= sender_direction) { op_count++; }
else if(direction == -1 and direction == sender_direction and space < 0) { back_count++; }
else if ( direction == -1 and direction == sender_direction and space > 0 ) { leader_count++;
}
}

    void Vehicle::helpingbroadcast(Ptr<Vehicle> v1, Ptr<const Packet> pk,Address address3) { uint8_t *buffer = new uint8_t[pk->GetSize()];
pk->CopyData (buffer, pk->GetSize());
stringstream ss (stringstream::in | stringstream::out);
string data = string((char*)buffer);
static int sender_id, sequence_number, sender_direction, event_direction, prestableflag, D_region
;
static double sender_pos, event_pos, event_time, dense, speed;
```

```
ss << data;
ss >> sender_id;
ss >> message_sequence;
ss >> sender_pos;
ss >> sender_direction;
ss >> event_pos;
ss >> event_direction;
ss >> period_flag;
ss >> D_region;
ss >> event_time;
ss >> dense;
ss >> speed;
double D_extra = event_pos - D_region - (2*dense);
double finale = event_pos - D_region ;
double bound = event_pos + (2*dense);
Vehicle::SetWaitingTime();

    if (v1->GetPosition().x >250 && v1->GetPosition().x < 9600){ v1->Setprob();

    if (v1->GetSequencenumber()==0 && v1->GetLeaderCount() < 1 )
{ v1->SetSequencenumber();
stringstream msgx;
msgx << “ ” << v1->GetVehicleId()
<< “ ” << v1->GetSequencenumber()
<< “ ” << v1->GetPosition().x
<< “ ” << v1->GetDirection()
<< “ ” << event_pos
<< “ ” << event_direction
<< “ ” << v1->GetFlag()
<< “ ” << D_region
<< “ ” << event_time
<< “ ” << v1->checkdensity()
<< “ ” << v1->GetVelocity();
uint16_t packetSize = 800;
Ptr<Packet> packet = Create<Packet>((uint8_t*) msgx.str().c_str(), packetSize);
v1->SendTo(v1->GetBroadcastAddress(), packet);

    Simulator::Schedule(Seconds(v1->GetWaitingTime(true)),&Vehicle::helpingbroadcast,
this,v1,packet,address3);
return;
}

    else if (v1->GetFlag()== 1 && v1-GetLeaderCount() < 1 && v1->GetOpCount() <3
```

```
&& v1->GetSequencenumber() > 0 && v1->GetPosition().x > D_extra && v1->GetPosition().x
< bound)
{ v1->SetSequencenumber();
stringstream msgx;
msgx << “ ” << v1->GetVehicleId()
<< “ ” << v1->GetSequencenumber()
<< “ ” << v1->GetPosition().x
<< “ ” << v1->GetDirection()
<< “ ” << event_pos
<< “ ” << event_direction
<< “ ” << v1->GetFlag()
<< “ ” << D_region
<< “ ” << event_time
<< “ ” << v1->checkdensity()
<< “ ” << v1->GetVelocity();
uint16_t packetSize = 800;
Ptr<Packet> packet = Create<Packet>((uint8_t*) msgx.str().c_str(), packetSize);
v1->SendTo(v1->GetBroadcastAddress(), packet);

    Simulator::Schedule(Seconds(v1->GetWaitingTime(true)),&Vehicle::helpingbroadcast,
this,v1,packet,address3);
}

    else if (v1->GetSequencenumber() >0)
{ if ( v1->GetPosition().x > finale) {

    Simulator::Schedule(Seconds(v1->GetWaitingTime(true)),&Vehicle::helpingbroadcast,
this,v1,pk,address3);
return;
}

    else if ( v1->GetPosition().x <= finale && v1->GetPosition().x > D_extra && v1-
>GetstableOpcount() < 1 && v1->GetFrontStableCount() < 1)
{ v1->SetFlag(2);
v1->SetSequencenumber();
stringstream msgx;
msgx << “ ” << v1->GetVehicleId()
<< “ ” << v1->GetSequencenumber()
<< “ ” << v1->GetPosition().x
<< “ ” << v1->GetDirection()
<< “ ” << event_pos
<< “ ” << event_direction
<< “ ” << v1->GetFlag()
```

```
<< “ ” << D_region
<< “ ” << event_time
<< “ ” << v1->checkdensity()
<< “ ” << v1->GetVelocity();
uint16_t packetSize = 800;
Ptr<Packet> packet = Create<Packet>((uint8_t*) msgx.str().c_str(), packetSize);
v1->SendTo(v1->GetBroadcastAddress(), packet);

    Simulator::Schedule(Seconds(v1->GetWaitingTime(true)),&Vehicle::helpingbroadcast,
this,v1,pack,address3); }
}
else return;
} else return;
}

    void Vehicle::intendedbroadcast(Ptr<Vehicle> v17, Ptr<const Packet> pkv,Address
address4)
{ uint8_t *buffers = new uint8_t[pkv->GetSize()];
pkv->CopyData (buffers, pkv->GetSize());
stringstream ss (stringstream::in | stringstream::out);
string data = string((char*)buffers);
static int sender_id, message_sequence, sender_direction, event_direction, period_flag,D_region
;
static double sender_pos, event_pos,event_time,dense,speed;
ss << data;
ss >> sender_id;
ss >> message_sequence;
ss >> sender_pos;
ss >> sender_direction;
ss >> event_pos;
ss >> event_direction;
ss >> period_flag;
ss >> D_region;
ss >> event_time;
ss >> dense;
ss >> speed;
double D_extra = event_pos +(2*dense);
double fin = event_pos-D_region;
double bound = event_pos-D_region -(2*dens);
Vehicle::SetWaitingTime();

    if (v17->GetPosition().x >250 && v17->GetPosition().x < 9600)
```

```
{ if (v17->GetSequencenumber()==0 && v17->GetBackCount() < 1 )
{ v17->SetSequencenumber();
stringstream msgx;
msgx << “ ” << v17->GetVehicleId()
<< “ ” << v17->GetSequencenumber()
<< “ ” << v17->GetPosition().x
<< “ ” << v17->GetDirection()
<< “ ” << event_pos
<< “ ” << event_direction
<< “ ” << v17->GetFlag()
<< “ ” << D_region
<< “ ” << event_time
<< “ ” << v17->checkdensity()
<< “ ” << v17->GetVelocity();
uint16_t packetSize = 800;
Ptr<Packet> pacr = Create<Packet>((uint8_t*) msgp.str().c_str(), packetSize);
v17->SendTo(v17->GetBroadcastAddress(), pacr);

    Simulator::Schedule(Seconds(v17->GetWaitingTime(true)),
&Vehicle::intendedbroadcast, this,v17,pacr,address4);
return;
}

    else if ( v17->GetFlag() == 1 && v17->GetBackCount() ¡ 1 && v17->GetOpCount()
<3 && v17->GetSequencenumber() > 0 && v17->GetPosition().x < D_extra && v17-
>GetPosition().x > bound)
{ v17->SetSequencenumber();
stringstream msgp;
msgp << “ ” << v17->GetVehicleId()
<< “ ” << v17->GetSequencenumber()
<< “ ” << v17->GetPosition().x
<< “ ” << v17->GetDirection()
<< “ ” << event_pos
<< “ ” << event_direction
<< “ ” << v17->GetFlag()
<< “ ” << D_region
<< “ ” << event_time
<< “ ” << v17->checkdensity()
<< “ ” << v17->GetVelocity();
uint16_t packetSize = 800;
Ptr<Packet> pacr = Create<Packet>((uint8_t*) msgp.str().c_str(), packetSize);
v17->SendTo(v17->GetBroadcastAddress(), pacr);
```

```
    Simulator::Schedule(Seconds(v17->GetWaitingTime(true)),&Vehicle::intendedbroadcast,
this,v17,pacr,address4);
}

    else if ( v17->GetSequencenumber() ¿ 0)
{ if (v17->GetPosition().x < event_pos)
{

    Simulator::Schedule(Seconds(v17-GetWaitingTime(true)),&Vehicle::intendedbroadcast,
this,v17,pkv,address4);
return;
}

    else if (v17->GetFlag() == 2 && v17->GetPosition().x >= event_pos && v17->GetPosition().x
< D_extra && v17->GetstableOpcount() < 1 && v17->GetFrontStableCount() < 1 )
{ v17->SetSequencenumber();
stringstream msgc;
msgc << " " << v17->GetVehicleId()
<< " " << v17->GetSequencenumber()
<< " " << v17->GetPosition().x
<< " " << v17->GetDirection()
<< " " << event_pos
<< " " << event_direction
<< " " << v17->GetFlag()
<< " " << D_region
<< " " << event_time
<< " " << v17->checkdensity()
<< " " << v17->GetVelocity();
uint16_t packetSize = 800;
Ptr<Packet> packq = Create<Packet>((uint8_t*) msgc.str().c_str(), packetSize);
v17->SendTo(v17->GetBroadcastAddress(), packq);

    Simulator::Schedule(Seconds(v17->GetWaitingTime(true)),&Vehicle::intendedbroadcast,
this,v17,packq,address4);
}
} else return;
} else return;
}

    void Vehicle::iDTSG(Ptr<Vehicle> v6, Ptr<const Packet> pkt, Address address1)
{ if (v6->GetDirection() == -1)
{

    Simulator::Schedule(Seconds(v6->GetWaitingTime(false)),&Vehicle::helpingbroadcast,
```

```
this,v6,pkt,address1);
return; }

    else if (v6->GetDirection() == 1)
{

    Simulator::Schedule(Seconds(v6->GetWaitingTime(false)),&Vehicle::intendedbroadcast,
this,v6,pkt,address1);
return; }
}

    int Vehicle::Broadcastcount(void) { return bc; }

    void Vehicle::stableOpcount(void) { Opstablecount++; }

    int Vehicle::GetstableOpcount(void) { return Opstablecount; }

    void Vehicle::FrontStableCount(void) { frontstablecount++; }

    int Vehicle::GetFrontStableCount(void) { return frontstablecount; }
```

A.3 Highway class

The following configurations are modified in the Highway function.

```
Highway::Highway()
{
// Setup Wifi

    m_wifiHelper = WifiHelper::Default();

    m_wifiHelper.SetStandard (WIFI_PHY_STANDARD_80211p_CCH);

    m_wifiMacHelper = NqosWifiMacHelper::Default();

    m_wifiPhyHelper = YansWifiPhyHelper::Default();

    m_wifiMacHelper.SetType (”ns3::AdhocWifiMac”);

    m_wifiHelper.SetRemoteStationManager (“ns3::ConstantRateWifiManager”,
“DataMode”, StringValue (“OfdmRate6MbpsBW10MHz”),
“ControlMode”, StringValue(“OfdmRate6MbpsBW10MHz”));

    m_wifiChannelHelper.SetPropagationDelay (“ns3::ConstantSpeedPropagationDelayModel”);

    m_wifiChannelHelper.AddPropagationLoss(“ns3::ThreeLogDistancePropagationLossModel”,
```

```
“Distance0”,DoubleValue(1.0), “Distance1”,DoubleValue(225.0), “Distance2”,DoubleValue(1000.0), “Exponent0”,DoubleValue(2) , “Exponent1”,DoubleValue(4), “Exponent2”,DoubleValue(4));

    m_wifiChannelHelper.AddPropagationLoss(“ns3::NakagamiPropagationLossModel”, “Distance1”, DoubleValue(50.0) , “Distance2”,DoubleValue(150.0), “m0”, DoubleValue(3.0), “m1”, DoubleValue(1.5), “m2”, DoubleValue(1.0));

    m_wifiChannel = m_wifiChannelHelper.Create();

    m_wifiPhyHelper.SetChannel (m_wifiChannel);

    m_wifiPhyHelper.Set(“TxPowerStart”,DoubleValue(5));

    m_wifiPhyHelper.Set(“TxPowerEnd”,DoubleValue(5));

    m_wifiPhyHelper.Set(“TxPowerLevels”,UintegerValue(1));

    m_wifiPhyHelper.Set(“TxGain”,DoubleValue(0));

    m_wifiPhyHelper.Set(“RxGain”,DoubleValue(0));

    m_wifiPhyHelper.Set(“EnergyDetectionThreshold”, DoubleValue(-87.0));

    m_wifiPhyHelper.Set(“CcaMode1Threshold”, DoubleValue(-83));

    m_wifiPhyHelper.Set(“RxNoiseFigure”, DoubleValue(7)); }
```

References

[1] K. Kanitpong, Sirilan, S. Kachathong, S. Ponboon, N. Boontob, and P. Aniwattakulchai, "Safe driving speed report," tech. rep., Thailand Accident Research Center (TARC) Transportation Engineering, School of Engineering and Technology, Asian Institute of Technology (AIT), July 2008.

[2] "Ieee standard for information technology– local and metropolitan area networks– specific requirements– part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications amendment 6: Wireless access in vehicular environments," *IEEE Std 802.11p-2010 (Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008, IEEE Std 802.11r-2008, IEEE Std 802.11y-2008, IEEE Std 802.11n-2009, and IEEE Std 802.11w-2009)*, pp. 1 –51, 15 2010.

[3] F. Li and Y. Wang, "Routing in vehicular ad hoc networks: A survey," *Vehicular Technology Magazine, IEEE*, vol. 2, pp. 12 –22, June 2007.

[4] N. Wisitpongphan, F. Bai, P. Mudalige, and O. Tonguz, "On the routing problem in disconnected vehicular ad-hoc networks," in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pp. 2291 –2295, May 2007.

[5] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu, "The broadcast storm problem in a mobile ad hoc network," in *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, MobiCom '99, (New York, NY, USA), pp. 151–162, ACM, 1999.

[6] H. Rahbar, K. Naik, and A. Nayak, "Dtsg: Dynamic time-stable geocast routing in vehicular ad hoc networks," in *Ad Hoc Networking Workshop (Med-Hoc-Net), 2010 The 9th IFIP Annual Mediterranean*, pp. 1 –7, june 2010.

[7] L. Briesemeister and G. Hommel, "Role-based multicast in highly mobile but sparsely connected ad hoc networks," in *Mobile and Ad Hoc Networking and Computing, 2000. MobiHOC. 2000 First Annual Workshop on*, pp. 45 –50, 2000.

[8] M.-T. Sun, W. Feng, and T.-H. Lai, "Location aided broadcast in wireless ad hoc

networks," in *Global Telecommunications Conference, 2001. GLOBECOM '01. IEEE*, vol. 5, pp. 2842 –2846 vol.5, 2001.

[9] A. Bachir and A. Benslimane, "A multicast protocol in ad hoc networks inter-vehicle geocast," in *Vehicular Technology Conference, 2003. VTC 2003-Spring. The 57th IEEE Semiannual*, vol. 4, pp. 2456 – 2460 vol.4, april 2003.

[10] S. Oh, J. Kang, and M. Gruteser, "Location-based flooding techniques for vehicular emergency messaging," in *Mobile and Ubiquitous Systems: Networking Services, 2006 Third Annual International Conference on*, pp. 1 –9, july 2006.

[11] K. Ibrahim, M. Weigle, and M. Abuelela, "p-ivg: Probabilistic inter-vehicle geocast for dense vehicular networks," in *Vehicular Technology Conference, 2009. VTC Spring 2009. IEEE 69th*, pp. 1 –5, april 2009.

[12] A. Dahiya and R. K. Chauhan, "A comparative study of manet and vanet environment," *Journal of computing*, vol. 2, pp. 87–92, July 2010.

[13] M. Boban, T. Vinhoza, M. Ferreira, J. Barros, and O. Tonguz, "Impact of vehicles as obstacles in vehicular ad hoc networks," *Selected Areas in Communications, IEEE Journal on*, vol. 29, pp. 15 –28, january 2011.

[14] E. Schoch, F. Kargl, M. Weber, and T. Leinmuller, "Communication patterns in vanets," *Communications Magazine, IEEE*, vol. 46, pp. 119 –125, november 2008.

[15] T. Willke, P. Tientrakool, and N. Maxemchuk, "A survey of inter-vehicle communication protocols and their applications," *Communications Surveys Tutorials, IEEE*, vol. 11, pp. 3 –20, quarter 2009.

[16] R. Uzcategui and G. Acosta-Marum, "Wave: A tutorial," *Communications Magazine, IEEE*, vol. 47, pp. 126 –133, may 2009.

[17] "Ieee standard for information technology - telecommunications and information exchange between systems - local and metropolitan area networks - specific requirements - part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications," *IEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999)*, pp. 1 –1076, 12 2007.

[18] Y. Li, "An overview of the dsrc/wave technology," in *Quality, Reliability, Security and Robustness in Heterogeneous Networks* (X. Zhang and D. Qiao, eds.), vol. 74 of

Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, pp. 544–558, Springer Berlin Heidelberg, 2012.

[19] "http://www.standards.its.dot.gov/fact_sheet.asp?$f = 80$."

[20] M. Bakhouya, J. Gaber, and P. Lorenz, "An adaptive approach for information dissemination in vehicular ad hoc networks," *Journal of Network and Computer Applications*, vol. 34, no. 6, pp. 1971 – 1978, 2011.

[21] H. Füßler, J. Widmer, M. Käsemann, M. Mauve, and H. Hartenstein, "Contention-based forwarding for mobile ad hoc networks," *Ad Hoc Networks*, vol. 1, no. 4, pp. 351–369, 2003.

[22] M. Torrent-Moreno, F. Schmidt-Eisenlohr, H. Fubler, and H. Hartenstein, "Effects of a realistic channel model on packet forwarding in vehicular ad hoc networks," in *Wireless Communications and Networking Conference, 2006. WCNC 2006. IEEE*, vol. 1, pp. 385 –391, april 2006.

[23] W. Zhang, "Analysis of packet forwarding in vanets using probabilistic channel model," in *Vehicular Technology Conference, 2009. VTC Spring 2009. IEEE 69th*, pp. 1 –5, april 2009.

[24] R. Chen, W.-L. Jin, and A. Regan, "Broadcasting safety information in vehicular networks: issues and approaches," *Network, IEEE*, vol. 24, pp. 20 –25, jan.-feb. 2010.

[25] Y. Shao, C. Liu, and J. Wu, *Handbook on Vehicular Networks*, ch. 10 Delay-Tolerant Networks in VANETs. M. Weigle and S. Olariu: Taylor & Frances, 2009.

[26] Z. D. Chen, H. Kung, and D. Vlah, "Ad hoc relay wireless networks over moving vehicles on highways," in *Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*, MobiHoc '01, (New York, NY, USA), pp. 247–250, ACM, 2001.

[27] M. Almeida, ed., *Advances in Vehicular Networking Technologies*. InTech, 2011.

[28] H. P. J. Maria Kihl, Mihail Sichitiu, "Design and evaluation of two geocast protocols for vehicular ad-hoc networks," *Journal of Internet Engineering*, vol. 2, no. 1, pp. 127–135, 2008.

[29] M. Javed and J. Khan, “A geocasting technique in an ieee802.11p based vehicular ad hoc network for road traffic management,” in *Australasian Telecommunication Networks and Applications Conference (ATNAC), 2011*, pp. 1 –6, nov. 2011.

[30] C. Maihöfer, T. Leinmüller, and E. Schoch, “Abiding geocast: time–stable geocast for ad hoc networks,” in *Proceedings of the 2nd ACM international workshop on Vehicular ad hoc networks*, VANET ’05, (New York, NY, USA), pp. 20–29, ACM, 2005.

[31] Q. Yu and D. Liu, “Disseminate warning message in vanets based on predicting the interval of vehicles,” in *Frontier of Computer Science and Technology (FCST), 2010 Fifth International Conference on*, pp. 559 –564, aug. 2010.

[32] Q. Yu and G. Heijenk, “Abiding geocast for warning message dissemination in vehicular ad hoc networks,” in *Communications Workshops, 2008. ICC Workshops ’08. IEEE International Conference on*, pp. 400 –404, may 2008.

[33] “http://www.nsnam.org/.”

[34] M. Lacage and T. R. Henderson, “Yet another network simulator,” in *Proceeding from the 2006 workshop on ns-2: the IP network simulator*, WNS2 ’06, (New York, NY, USA), ACM, 2006.

[35] H. Friis, “A note on a simple transmission formula,” *Proceedings of the IRE*, vol. 34, pp. 254 – 256, may 1946.

[36] M. Nakagami, “The m-distribution- a general formula of intensity distribution of rapid fading,” in *Statistical Methods in Radio Wave Propagation*, pp. 3–36, Pergamon Press, Oxford, U. K., 1960.

[37] Q. Chen, F. Schmidt-Eisenlohr, D. Jiang, M. Torrent-Moreno, L. Delgrossi, and H. Hartenstein, “Overhaul of ieee 802.11 modeling and simulation in ns-2,” in *Proceedings of the 10th ACM Symposium on Modeling, analysis, and simulation of wireless and mobile systems*, MSWiM ’07, (New York, NY, USA), pp. 159–168, ACM, 2007.

[38] M. Torrent-Moreno, D. Jiang, and H. Hartenstein, “Broadcast reception rates and effects of priority access in 802.11-based vehicular ad-hoc networks,” in *Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks*, VANET ’04, (New York, NY, USA), pp. 10–18, ACM, 2004.

[39] L. Cheng, B. Henty, F. Bai, and D. Stancil, “Highway and rural propagation channel modeling for vehicle-to-vehicle communications at 5.9 ghz,” in *Antennas and Propagation Society International Symposium, 2008. AP-S 2008. IEEE*, pp. 1 –4, july 2008.

[40] H. Arbabi and M. Weigle, “Highway mobility and vehicular ad-hoc networks in ns-3,” in *Simulation Conference (WSC), Proceedings of the 2010 Winter*, pp. 2991 –3003, dec. 2010.

[41] M. Treiber, A. Hennecke, and D. Helbing, “Congested traffic states in empirical observations and microscopic simulations,” *Phys. Rev. E*, vol. 62, pp. 1805–1824, Aug 2000.

[42] M. Treiber and D. Helbing, “Realistische mikrosimulation von straβenverkehr mit einem einfachen modell,” in *16th Symposium "Simulationstechnik ASIM 2002" Rostock* (D. Tavangarian and R. Grützner, eds.), pp. 514–520, 2002.

[43] M. Segata and R. L. Cigno, “Emergency braking: a study of network and application performance,” in *Proceedings of the Eighth International Workshop on Vehicular Ad Hoc Networks, VANET 2011, Las Vegas, NV, USA, September 1, 2011*, pp. 1–10, ACM, 2011.

[44] P. Kheawchaoom, S. Kittipiyakul, and K. Ayutayay, “idtsg time-stable geocast for post crash notification in vehicular highway networks,” in *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2012 9th International Conference on*, (Phetchaburi, Thailand), pp. 1 –4, may 2012.

[45] P.-J. Wang, C.-M. Li, and H.-J. Li, “Influence of the shadowing on the information transmission distance in inter-vehicle communications,” in *Personal, Indoor and Mobile Radio Communications, 2009 IEEE 20th International Symposium on*, pp. 3015 –3019, sept. 2009.

[46] Y.-W. Lin, Y.-S. Chen, and S.-L. Lee, “Routing protocols in vehicular ad hoc networks: A survey and future perspectives,” *J. Inf. Sci. Eng.*, vol. 26, pp. 913–932, May 2010.

[47] P. Kheawchaoom and S. Kittipiyakul, “Effects of distance-based defer times and probabilistic channel to time-stable geocast,” in *International Conference on Communications Systems and Technologies (ICCST’12)*, (San Francisco, USA), October 2012.

[48] B. T. . W. M. . B. T. Heissenbuttel, M., "Optimized stateless broadcasting in wireless multi-hop networks," in *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, (Barcelona, Spain), pp. 1–12, April 2006.

[49] H. Sadeghian, A. Farahani, and M. Abbaspour, "Dynamic forwarding delay in contention-based forwarding protocols," in *Computational Problem-Solving (ICCP), 2011 International Conference on*, pp. 688 –691, oct. 2011.

[50] Q. Yang and L. Shen, "A multi-hop broadcast scheme for propagation of emergency messages in vanet," in *Communication Technology (ICCT), 2010 12th IEEE International Conference on*, pp. 1072 –1075, nov. 2010.

[51] F. Hrizi, C. Bonnet, J. Härri, and F. Filali, "Adapting contention-based forwarding to urban vehicular topologies for active safety applications," *Annals of Telecommunications*, pp. 1–19, 2012.

[52] M. Torrent-Moreno, "Inter-vehicle communications: assessing information dissemination under safety constraints," in *Wireless on Demand Network Systems and Services, 2007. WONS '07. Fourth Annual Conference on*, pp. 59 –64, jan. 2007.

[53] L. Anchora, L. Casone, G. Ciccarese, M. De Blasi, P. Marra, and C. Palazzo, "An optimal setting for the parameters of an intelligent flooding scheme in vanets," in *Wireless Conference (EW), 2010 European*, pp. 332 –338, april 2010.

Publications

[1] P. Kheawchaoom and S. Kittipiyakul, “Effects of Stochastic and Deterministic Defer Times on Time-Stable Geocast in VANETs,” *IAENG International Journal of Applied Mathematics,* Volume 43, Issue 1, pp. 30-36, 2013.

[2] P. Kheawchaoom and S. Kittipiyakul, “Effects of Distance-Based Defer Time and Probabilistic Channel to Time-Stable Geocast,” *International Conference on Communications Systems and Technologies (ICCST12)*, USA, Oct, 2012.

[3] P. Kheawchaoom, S. Kittipiyakul, and K. Siriwong Na Ayutaya, “iDTSG: Time-Stable Geocast for Post Crash Notification in Vehicular Highway Networks,” *9th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON 2012)*, Thailand, May, 2012.