Analytical Investigation and Implementation of Carry and Forward based Routing Protocol for Vehicular Ad hoc Network

by

© Aslinda Hassan

A thesis submitted to the

School of Graduate Studies

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Faculty of Engineering and Applied Science

Memorial University of Newfoundland

May 2014

St. John's Newfoundland

Abstract

Recent research studies have recognized the applicability of carry and forward based routing in a vehicular ad hoc network (VANET), where packets are stored and carried by a moving vehicle until another vehicle comes into its transmission range and the packets are transmitted via wireless channel. This thesis explores several research topics concerning the use of a carry and forward approach in a vehicular network. In the first part of our research, we develop an end-to-end delay model in a unidirectional highway using vehicle-to-vehicle connectivity parameters that include the carry and forward approach which extends an existing catch-up time delay model for two disconnected vehicle clusters to multiple disconnected clusters on a unidirectional highway. Consequently, two distributions are newly derived to represent the number of clusters on a highway using a vehicular traffic model. The analytical results obtained from the end-to-end distribution model are then validated through simulation results. In the second part of our research, we present a fuzzy logic based beaconing system where beacon intervals are adjusted based on packet carried time, number of single-hop neighbors, and vehicles speed. It is common for vehicles in a VANET to exchange information by broadcasting beacon messages periodically. This information is required not only for routing protocols when making routing decisions, but also for safety applications. Choosing a suitable interval for broadcasting beacon messages has been considered a communication challenge since there will be a trade-off between information accuracy and channel usage. Therefore, an adaptive

beaconing approach is needed so that vehicles can regulate their beacon rate based on traffic condition. Through simulation in a grid model and a realistic scenario, we are able to show that the fuzzy logic based beaconing system is not only able to reduce routing overhead and packet collision, but also decrease the average end-to-end delay and increase the delivery rate as well. The last issue of this thesis focuses on developing a proactive multi-copy routing protocol with carry and forward mechanism that is able to deliver packets from a source vehicle to a destination vehicle at a small delivery delay. It has been ascertained by the majority of researches in VANET that the carry and forward procedure can significantly affect an end-to-end delivery delay. Our approach is to replicate data packets and distribute them to different relays. The proposed protocol creates enough diversity to reach the destination vehicle with a small end-to-end delivery delay while keeping low routing overhead by routing multiple copies independently. The simulation results in an urban grid model show that the proposed multi-copy forwarding protocol is able to deliver packets at small delivery delay compared to a single-copy forwarding algorithm without having to rely on real time traffic data or flooding mechanism.

Acknowledgements

I would like to express my sincere gratitude and appreciation to my supervisors Professor M. Azizur Rahman and Dr. Mohamed Hossam Ahmed for their continuous guidance, advice and encouragement towards the completion of the PhD program.

Special thanks to the member of my supervising committee, Dr. Saleh A. Saleh, and Dr. Glyn George for their useful comments and suggestions for my thesis.

I am very grateful to University Technical Melaka Malaysia (UTeM) for providing me an opportunity and scholarship to further my Doctoral study at Memorial University of Newfoundland Canada.

I also would like to express my deepest gratitude and sincere appreciation to my husband Mr. Azziddin Razali and my three wonderful daughters, my dearest friend Fatemeh M. Kiaie, my father Mr. Hassan Adam, my sisters, as well as other family members, relatives and friends for their understanding and everything they have done for me. They have supported and encouraged me besides showing their great patience during all my period of studies at MUN. Their existence in my life reminds me that there are things more important than this work.

Special thanks to all MUN Faculty members and School of Graduate Studies for giving full support and assistantship regarding the university policy and management aspects.

Contents

List of Tables

List of Figures

List of Abbreviations

ABS	Adaptive Beaconing System
AODV	Ad-hoc On-demand Distance Vector
ASTM	American Society for Testing and Materials
ATB	Adaptive Traffic Beacon
BBR	Border Node based Routing
CDF	Cumulative Distribution Function
COG	Center of Gravity or Centroid
CPU	Central processing unit
DSR	Dynamic Source Routing
DSRC	Dedicated Short Range Communications
DV-CAST	Distributed Vehicular Broadcast
ETSI	European Telecommunications Standard Institute

FBI	Fixed Beacon Interval
FCC	United States Federal Communications Commission
FLS	Fuzzy Logic System
GPCR	Greedy Perimeter Coordinator Routing
GPS	Global Positioning System
GPSR	Greedy Perimeter Stateless Routing
GyTAR	Greedy Traffic-Aware Routing
IDM_LC	Intelligent Driver Model with Lane Changing
IEEE	Institute of Electrical and Electronics Engineers Inc.
ITS	Intelligent Transportation Systems
IVC	Inter-Vehicle Communication
MAC	Media Access Control
MANET	Mobile Ad hoc Network
MMSE	Minimum Mean Square Error
MOPR	Movement Prediction-based Routing
MURU	Multi-hop routing protocol for Urban Vehicular Ad hoc Network

NS-2	Network Simulator version 2
PDF	Probability Density Function
PHY	Physical
PMC	Proactive Multi-copy Routing
SCF	Single-copy Forwarding
SNR	Signal to Noise Ratio
SSE	Sum of Square Errors
UMB	Urban Multi-hop Broadcast protocol
V2I	Vehicle-to-Infrastructure
V2V	Vehicle-to-Vehicle
VADD	Vehicle-Assisted Data Delivery
VANET	Vehicular Ad hoc Network
VC	Vehicular Communication
WAVE	Wireless Access in the Vehicular Environment
WLAN	Wireless Local Area Network

List of Symbols

C_L	the length between the first vehicle and the last vehicle in a cluster
D	The length of a road
$DPkt_n$	Data packets with id $n = 1 \cdots m$
$F_Y(y)$	Cumulative distribution function of a random variable Y
$f_Y(y)$	Probability density function of a random variable Y
I_i	Intersections available in a city map
I_c	The current intersection
I_{min}	Minimum beacon interval
I_{max}	Maximum beacon interval
L	Gap of between two neighboring vehicles
L_c	Gap between two neighboring connected vehicles
L_{uc}	Gap between two neighboring disconnected vehicles
λ	Traffic flow rate (vehicles/unit time)
λ_s	Vehicles density (vehicles/unit distance)
λ_{T_C}	Minimum mean square error between the exact distribution of T_C

and an exponential distribution

M The number of vehicle gaps in a cluster

nb_table Table for the local neighborhood state information

m_copy Table for packet replication information

N_c Number of vehicle clusters on a highway

$N(t)$ Number of vehicles arrive at the highway during interval [0,t]

$P[y]$ Probability of an event y

$\Phi_Y(s)$ Characteristic function of a random variable Y

r Vehicle radio range

R^2 Coefficient of determination

r_{I_i} Road segments at Intersection i

$\overrightarrow{R}$ Road direction

S_{inter} The spacing between the last vehicle of the leading cluster and the first vehicle of the following cluster

T The arrival time of a vehicle at the highway and it is uniformly distributed at interval $(0, t]$

T_c Time duration of a catch-up phase

T_f Time duration of a forward phase

T_D The sum of multiple catch-up times, T_c

V the vehicles' speed and it is uniformly distributed at interval $[v_{min}, v_{max}]$

v_s A source vehicle

v_d	A destination vehicle
v_k	Neighboring vehicles
v_c	A current forwarding vehicle
v_{nexth}	A next hop vehicle
w_C	Channel quality weighting
w_I	Interval weighting
X_f	The distance traveled by messages during a forwarding phase
X_c	The distance traveled by messages during a catch-up phase
$X(t)$	Message propagation distance during $(0, t]$
$X'(t)$	Distance that the partition tail moves during [0,t]
Y	The size of a cluster

Chapter 1

Introduction

1.1. Background

Vehicular Ad hoc Network (VANET) is *vehicle to vehicle* (V2V) and *vehicle to infrastructure* (V2I) communications using wireless local area network technologies. The main idea of VANET is to provide continuous connectivity to mobile users while on the road, and to provide efficient vehicle-to-vehicle communications [1, 2]. In recent years, the research and development in this area has intensified due to several factors. One of the contributing factors is the potential advantages of VANET applications. V2V and V2I communications have enabled the development and implementation of a variety of applications, as well as providing a broad range of information to drivers and travelers. By integrating a vehicle's on-board devices with the network interface, various types of sensors and Global Positioning System (GPS) devices, the vehicle has the capability to aggregate, process and disseminate information about itself and its environment to other vehicles in the immediate vicinity that can be used for enhancing road safety and providing passenger comfort [3–5].

In addition, the advancements in computing and wireless communication technologies have increased interest in "smart" vehicles, resulting in more vehicle manufacturers beginning to adopt the use of information and technology to tackle the issue of safety and the environment, as well as the comfort of their vehicles. A *smart vehicle* should at least be equipped with on-board units, also known as in-vehicle equipments, that are needed for communication. For inter-vehicle communication, it is assumed that a vehicle should have a central processing unit (CPU) that implements applications and communication protocols; a wireless transceiver for transmitting and receiving data packets or wireless signals; a GPS receiver for location and time synchronization information, and a human interface between the driver and the system [6–8]. In [8, 9], the authors described computing platforms for *vehicular communication* (VC) that are dedicated to VC functionality and independent from car processors and controllers. Car processors and controllers are normally used for tasks such as fuel injection, braking, transmission and car charging [8, 9]. However, VC computing platforms are independent from these vehicle power systems and are responsible for V2V and V2I communication protocols and applications. The VC computing platforms usually use information provided by the vehicle processors and controller and forward them to safety and driving efficiency applications [8].

Another contributing factor in the increment of VANET studies is the commitment of national and regional governments to assign wireless spectrum and the wide implementation of wireless access technologies that provide the required radio interface to facilitate V2V and V2I communications between vehicles [5, 10, 11]. In 1999, the United States Federal Communications Commission (FCC) assigned the 75 MHz band of Dedicated Short Range Communications (DSRC) at the 5.850 - 5.925 GHz frequency for Intelligent Transportation Systems (ITS) application in North America, which is used for variety of services such as safety applications, real-time traffic management, traveler information and many more [3, 5, 11, 12]. In Europe, ETSI (European Telecommunications Standard Institute), which is responsible for the standardization in the telecommunication industry, has designated the frequency band between 5.885 – 5.905 GHz for ITS applications in year 2008 [3, 5, 11]. DSRC radio technology is built based on the IEEE 802.11p standard, which is modified from the IEEE 802.11a standard since the latter is not sufficient enough to support inter-vehicle communication. The American Society for Testing and Materials (ASTM) modified the 802.11a standard to match the vehicular environment, and from this effort, IEEE standardized a new standard specifically for wireless access in the vehicular environment (WAVE) which is IEEE 802.11p with higher tolerance to multi-path propagation and Doppler spread effects for moving vehicles [5, 13].

1.2. Research Motivations

As mentioned in Section 2.1, one of VANET key features that differentiates itself from traditional ad hoc network is its dynamic topology, due to the vehicles' speed and movement. Therefore, it is natural for VANET's topology to have two extreme cases of network density, which are very high network density where each node can have hundreds of neighbors within its transmission range and very low network density in which transmission via wireless channel might be impossible. Because of VANETs' dynamic topologies and highly variable vehicle densities, it is difficult for topology-based or on-demand routing protocols to establish an end-to-end route between the source and destination vehicle, although it is not impossible. Examples of VANET topology based routing protocols are *multi-hop routing protocol for urban vehicular ad hoc network* (MURU), which is proposed in [14]; and *movement prediction-based routing* (MOPR) proposed in [15].

Since the topology of a vehicular network can change rapidly and discovering new routes not only can incur large overhead but large delays as well, most routing protocols in VANETs are shifted towards making forwarding decisions based on information of vehicle's immediate neighbors only. The set of all neighbors and their respective information are determined using periodic beacon messages that are exchanged among nearby nodes. Using this method, vehicles do not have global knowledge of the network topology and forwarding decisions are made locally. Consequently, these protocols often experience local optimum problems, in which a vehicle cannot find the next forwarding node. This problem normally occurs in a sparse network that has intermittent network connectivity and it can cause routing failures. Therefore, it is important for a protocol to have an alternative method for packet dissemination such as a *carry and forward* feature to increase the chance of the packets arriving at the destination. Basically, packets are stored and carried temporarily in a moving vehicle while waiting for opportunities to forward them via wireless channel. Majority of VANET routing protocols is now designed to include a *store-carry and forward* feature as part of their recovery method. Some examples that use this feature are *Vehicle-Assisted Data Delivery* (VADD) protocol [16], *Greedy Traffic-Aware Routing* (GyTAR) protocol [17], and *Border Node based Routing* (BBR) protocol [18].

Choosing forwarding path with high density paths may not be the optimal solutions in minimizing delivery delay. If all vehicles have the same idea in using high density paths, channel utilizations along these paths will increase, and as a result, packets may either get dropped or incur higher delays. Relying on preloaded traffic information such as vehicle speed limits and traffic density at different times of the day to estimate delay is not a good solution either since the information may not be accurately adapted to the dynamic change in the environment, and it can lead to selecting a forwarding path with high delivery delay instead.

Obtaining accurate information on local topology such as neighboring vehicles' locations, directions, and speed for forwarding purposes is crucial especially when small delivery delay is required by a protocol or an application. Although the use of beacon packet is beneficial in maintaining the information of the network topology, beacon broadcasting in the wireless channel with limited bandwidth can affect the efficiency of the communication. Since beacon messages are broadcast in an interval time using a single communication channel that is basically shared by all nodes, it can potentially increase the wireless channel load which in turn can reduce the performance of routing protocols and challenge a reliable and successful delivery. Nevertheless, decreasing the beacon interval rate is not considered to be a suitable solution, since it will reduce the information freshness at the same time [19]. Stale neighbor information can cause packets to be delivered to unsuitable neighbor; or packet drop if the next hop neighbor has already left the transmission range. These problems can reduce the reliability and efficiency of a routing protocol.

1.3. Research Objectives

Our main focus in this research is to investigate a carry and forward based routing protocol in both sparse and dense vehicular networks.

1. To investigate end-to-end delay in packet propagation via a carry and forward method for a random number of vehicle clusters in a one directional highway scenario

2. To create an adaptive beaconing system using a fuzzy logic system according to the vehicle's own movements, number of neighbors and packets carried time to help reducing routing overhead and collision rate especially in a dense network.

3. To use proactive multi-copy routing with controlled forwarding to increase the chance of finding the destination node and reduce end-to-end delay.

4. To incorporate the elements of geographic routing and direction to enable efficient packet forwarding with small delivery delay in a vehicular ad hoc network.

1.4. Thesis Contributions

In the context of our research, we made the following contributions:

(a) We investigate and derive two new distribution models to represent the number of disconnected vehicle clusters on a unidirectional highway, as given in Section 3.3.1.1 and 3.3.1.2. Using a Poisson process as our basic assumption for the arrival of vehicles and two distinct vehicle-to-vehicle (V2V) connectivity models, we formulate the distribution model for number of vehicle clusters in a certain length of a highway. Using this framework, we are able to estimate the number of vehicle clusters in both sparse and dense networks. The analytical models for number of disconnected clusters are then utilized in our investigation to develop a distribution model for end-to-end delay on a highway using V2V connectivity parameters that include the carry and forward mechanism. By producing a closed form solution for the catch-up time delay model between two disconnected clusters, we are able to ascertain the distribution of the catch-up time delay model as an exponential distribution. We are able to further determine the probability distribution model for the end-to-end delay between the multiple disconnected clusters as an Erlang-n distribution by using an exponential approximation for the catch-up delay between two disconnected clusters and the models for the multiple number of clusters. This framework enable us to assess the end-to-end delivery delay between a source and destination moving on a highway for both disconnected and well-connected vehicular networks.

(b) We design and develop a beaconing system that can adjust its beacon rate based on the current condition of the network. The beacon rate adaptation is achieved through the use of a fuzzy logic system. In general, a fuzzy logic system can adjust its output value based on the input values and its knowledge base. In our implementation, we set the beacon interval as the fuzzy logic system's output and its value is adjusted based on the input values, which are packet carried time, number of single-hop neighbors, and the vehicle current speed. The developed fuzzy logic system is able to determine the current network condition whether the network is sparse or well-connected based on the above input values and the knowledge base. In a sparse network, packets are being carried most of the time since the number of vehicles is small and can approach to zero, and vehicles are normally moved with high speed. In this situation, the fuzzy system sets the interval to low value so that the frequency of broadcasting the beacon packet is increased. In a well-connected network where the number of neighboring vehicles is usually large and vehicles are moving at a slow speed, packets are often transmitted via wireless channel instead of being carried by vehicles. In this type of network, the beacon frequency is reduced since vehicles have low mobility and neighboring vehicles do not change rapidly. By adjusting beacon intervals to the current network condition, the adaptive beaconing system is able to reduce the unnecessary routing overhead in a dense network and obtain accurate information rapidly in a disconnected network.

(c) We construct and implement a proactive multi-copy routing (PMC) protocol that is able to reduce end-to-end delay without the use of real time traffic data or a flooding mechanism. The main component of the developed PMC protocol is the controlled replication of data packets. Fundamentally, each time a vehicle arrives at an intersection, the protocol generates multiple copies of a data packet and distributes them to different relays. Subsequently, each of these relays forwards or carries its copy to the destination vehicles or other intermediate vehicles. The selection of relays begins with the process of the selection of the road segments with direction that can bring the packets closer to the destination vehicles. Once the road segments have been selected, PMC selects a suitable next hop vehicle for each road segment by using a greedy forwarding algorithm with direction awareness. In this algorithm, vehicles that are closest to the destination vehicle and move towards the latter will be chosen. At a regular road segment, packets are also forwarded by utilizing greedy forwarding with direction awareness algorithm. To reduce redundant replications, we add additional fields to the beacon packet that include the acknowledgment identification, the identification of replicated packet, the destination identification, and intersection location where the current packet is being replicated. The information is saved by the beacon's receivers and will be used to avoid redundant replication of a data packet at a particular intersection.

1.5. Thesis Organization

In this section, we outline the organization of this thesis and give a brief overview of each chapter.

Chapter 2: In this chapter, we discuss the challenges faced by common MANET routing protocols in a vehicular network and VANET routing solutions that employs carry and forward approach. We also include our examination on adaptive beaconing mechanisms proposed in the literature, and analyses on VANET connectivity and end-to-end delay model.

Chapter 3: An analytical framework on end-to-end delay based on multiple disconnected clusters in a unidirectional highway is presented in this chapter.

Chapter 4: In this chapter, we describe a fuzzy logic approach for an adaptive beaconing system in broadcasting single-hop beacon packets.

Chapter 5: The design and implementation a routing protocol that proactively replicates data packets at road intersections is given in this chapter.

Chapter 6: We conclude our thesis in this chapter by summarizing the contributions of this thesis and possible future research directions

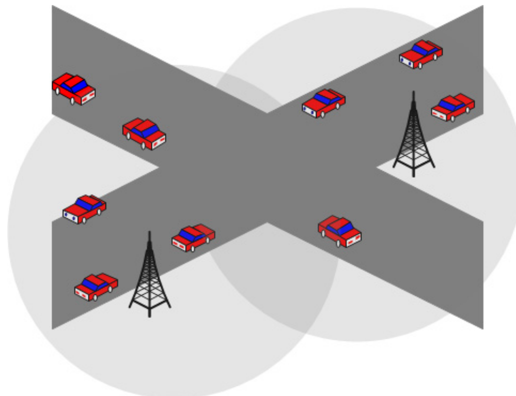
Chapter 2

Literature Review

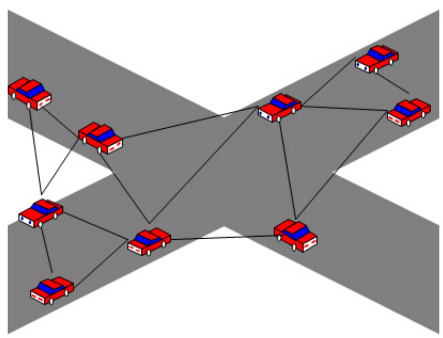
2.1. Overview on VANET

In general, VANET is formed between nodes an as-needed basis. To create a VANET, vehicles need to have wireless transceivers and computerized modules that enable the vehicles to act as network nodes. There are three types of VANETs architecture namely; pure cellular/wireless local area network (WLAN), pure ad hoc, and hybrid [1], as shown in Figure 2.1. In pure cellular/WLAN architecture, VANETs use fixed cellular gateways and WLAN access points either to gain access to the Internet, gather information on traffic or find information for routing purposes.

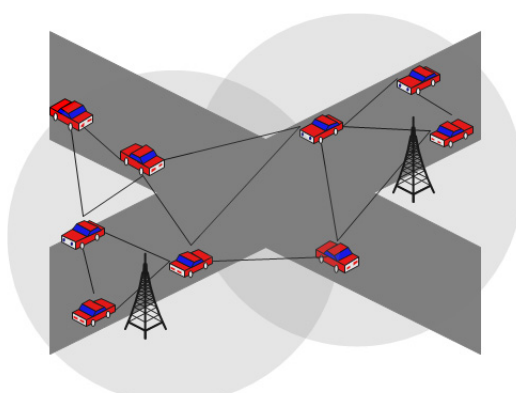
Similar to mobile ad hoc networks (MANET), VANETs consist of radio-enabled vehicles which act as mobile nodes and routers for other nodes. Although VANETs share a few common MANETs characteristics such as self-organization and self-management, short radio transmission range and limited bandwidth, they also have



(a) WLAN/cellular



(b) ad hoc



(c) hybrid

Figure 2.1. Three types of VANET network architecture

significant features that differentiate them from other types of ad hoc networks [1,2,8]. The features of VANETs are as follows:

1. *Highly dynamic topology*

   VANETs' topology is considered very dynamic due to the high speed and movement of the vehicles

2. *Frequently disconnected network*

   Due to the same reason, VANETs connectivity is also often changing, especially when the number of vehicles is low, which usually leads to disconnected networks. In [20], the authors proposed a vehicular connectivity model based on empirical data collected at highway I-80 of San Francisco-Oakland Bay Bridge between Emeryville, CA and Berkeley, CA; by Berkeley Highway Laboratory (BHL). From the data collected, it can be seen that there are times when the traffic volume is low, which is around 1 am to 3 am [20]. The low traffic volume can create partition in the network and can hinder packet transmission vehicles.

3. *Sufficient energy and storage*

   Since the mobile nodes in VANETs are mostly cars, they should have sufficient energy and computing power (including both storage and processing) compared to mobile nodes in regular ad hoc networks.

4. *Geographical type of communication*

   In addition to a typical unicast or multicast communication where the end points are normally defined by ID or group ID; VANETs use a new type of communication, which uses geographical areas to determine where packets need to be forwarded.

5. *Mobility modeling and predictions*

   Mobility modeling and predictions are considered as important factors in network protocol design in VANETs since MANET's mobility models may not be suitable for VANET. One reason is due to dynamic topology and mobile node's high mobibility. Another reason is the significant cost of implementing a VANET system in real world. Therefore, it is important to have a realistic vehicular mobility model in VANET simulation to make sure any conclusion derives from simulation will carry through to real deployment [21]. In addition, the future position of mobile nodes can often be predicted given the nodes' speed and street maps since the nodes are constrained by highways, roads, and streets.

6. *Various communication environments*

   There are two typical communication scenarios operated by mobile nodes in VANETs. The first scenario is highway traffic scenario where the environment is relatively simple and straightforward. The second scenario is a city streets scenario, in which the streets are often separated by buildings, trees, and other obstacles. Therefore, there is not always a direct line of communications in the direction of intended data communication.

7. *Hard delay constraints*

   Some VANETs applications emphasize more hard delay constraints more than high data rates, such as accident avoidance applications where certain events such as air bag ignition or brake event happened. The message must be delivered in a certain amount of time to avoid more accidents.

8. *Interaction with on-board sensors*

   It can be assumed that vehicles in VANETs are equipped with on-board sensors such as GPS devices which can provide information that can be used for routing purposes or to form communication links. Other sensors such as sensors for air-bag ignition or brake events can also be used to provide information for safety applications.

The authors in [21] also defines several VANET characteristics that separate VANET from MANET. One of the attributes that are discussed is the node velocity, which can range from zero for vehicles that are caught in traffic jam or stationary road side units, to over 150 km/h for vehicles that are on highways [21]. The authors in [21] discussed how both extreme cases can create special challenges in vehicular ad hoc communication systems. Vehicles with high velocity have very short wireless communications window due to short encounters between vehicles. The short communication window can cause link failure since the link layer is unable to predict when a connection will be disrupted. Furthermore, brief encounters between vehicles can cause the topology to be highly unstable, which makes topology-based routing protocols inadequate for packet dissemination [21]. On the other hand, vehicles with low or zero velocity indicates the network has a very high vehicle density, which can lead to high interference, medium access problems, and packet collisions.

Another VANET characteristic defined in [21] is the pattern movement. Unlike in MANET, mobile nodes in VANET do not move in random manner, instead, they move in predefined locations, specifically in streets with normally two directions. Unpredictable changes in traffic directions usually happen at the intersections of roads. The roads can be categorized in three main locations, which are city, rural areas, and highways. Node density is another element that differentiate VANET from MANET [21]. The density of mobile nodes in VANET can range from zero or dozens in rural areas to hundreds in city area.

2.2. VANET Routing Protocols

As mentioned in the previous chapter, VANET's topology is considered very dynamic due to the high speed and mobility of the vehicles. Due to its dynamic nature, finding and maintaining routes is considered a challenge in VANET. Since VANET and MANET share the basic principle in terms of architecture and communication, most of the ad hoc routing protocols such as *Ad-hoc On-demand Distance Vector* (AODV) protocol [22] and *Dynamic Source Routing* (DSR) protocol [23] can still be applied to VANETs. Both AODV and DSR are on-demand routing protocol, and they do not maintain routes unless routes are needed for forwarding packets, which is very helpful in reducing network overhead especially in small networks. On the other hand, simulations done by [24–27] show that most of the general purpose ad hoc routing protocols have poor route convergence and low communication throughput in VANET because of the nodes' high mobility. Nevertheless, a number of researchers have proposed modified versions of MANET protocols to accommodate VANET's unique properties. One of these protocols is called *multi-hop routing protocol for urban vehicular ad hoc network* (MURU), which is proposed in [14]. In the protocol, each intermediate node approximates the quality of the wireless link between itself and its downlink vehicle; and updates the value of metric *expected disconnection degree*. The metric is used to assess the probability that a path would be broken in a predefined time period [14]. Subsequently, the protocol selects the route with the lowest breakage probability for packets transmission. Another MANET routing protocol modified for VANET is proposed by Menouar et al. in [15] and it is called *movement prediction-based routing* (MOPR). The protocol ascertains the most stable route based on the lifetime and links in the route by calculating the speeds of two neighboring vehicles.

Position based routing is one of the routing strategies that are often used in ad hoc network research studies. In this strategy, the protocol makes routing decision by using geographic information obtained from street maps, traffic models, or on-board navigational systems. One of the best known position based routing protocols is *Greedy Perimeter Stateless Routing* (GPSR) [28] protocol, which combines greedy routing with face routing. In greedy forwarding, GPSR tries to bring packets closer to the destination in each hop by using geographic information. However, in many cases, greedy forwarding can lead to a local maximum problem where there is no neighbor closer to the destination node except for the current forwarding node. In this case, GPSR uses face routing to recover from the *local maximum* problem by forwarding the packet along the interior faces of the communication graph. Unfortunately, GPSR performs best in a free open space scenario with evenly distributed nodes, and simulations in [25, 29, 30] show GPSR's performance degrades when it is applied to city scenarios. One of the main reasons is that in these scenarios, greedy forwarding is restricted as it is not easy to get direct communications between mobile nodes due to obstacles such as buildings and trees.

To mitigate the problem faced by GPSR in a city environment, the authors in [29] propose a solution called *Greedy Perimeter Coordinator Routing* (GPCR) which does not use either source routing or street maps. This protocol uses a restricted greedy forwarding instead of regular greedy forwarding when transmitting packets on a street. In restricted greedy forwarding, GPCR prefers to choose a node on a junction as the next hop even though the node is not geographically closest to the destination node [11, 29]. This protocol uses the fact that the nodes at a junction in the street follow a natural planar graph. Consequently, using the restricted greedy algorithm, packets will be always forwarded to a node on a junction instead of being forwarded across the junction.

Aside from unicast routing, broadcast routing is also used in VANETs for disseminating information among vehicles such as sharing traffic information, weather, emergency news, and road condition. The most straightforward approach to implement a broadcast routing protocol is to flood the whole network where each node re-broadcasts a message to its entire neighboring vehicles excluding the node that sent the message [1,31]. One of the benefits of flooding the network is that it ensures all nodes in the network will receive the message ultimately. It is a simple method to implement, and this method operates quite well in a small network. However, broadcast service can cause large overhead for a large network. Flooding can cause contentions and collisions, broadcast storms, and high bandwidth consumption, since all nodes will be receiving and broadcasting messages nearly at the same time. According to the authors in [1,31], these congestions can be avoided by using selective forwarding instead of flooding.

An example of modified broadcast service is the *Urban Multi-hop Broadcast protocol* (UMB) [32]. According to the authors in [32], this protocol is designed to overcome interference, packet collisions, and hidden nodes problems during multihop broadcast communication. In this protocol, the sending nodes attempt to select the furthest node in the broadcast direction to assign the duty of disseminating the messages without any prior knowledge of the network topology.

The above routing protocols are able to attain a high delivery rate in a dense network. However, these protocols are unable to achieve the same delivery rate in a sparse network where a node usually does not have any neighbors closer to the destination node than itself. To increase the chance of packet delivery in a sparse network, a number of research studies begin to include a *carry and forward* [33] strategy in their proposed protocol. A notable example is a protocol called *Vehicle-Assisted Data Delivery* (VADD), which is proposed in [16]. VADD protocol incorporates the carry and forward feature that buffers packets if the network is partitioned. The carry-and-forward strategy is used to transfer those packets along vehicles on the fastest roads available if packets are unable to be delivered via wireless channel. A vehicle makes a decision at an intersection and selects the next forwarding road with the smallest packet delivery delay. The VADD protocol uses traffic information such as density, speed limits and traffic signal to select a path dynamically for the forwarding process [16]. The protocol prefers high traffic density path to geographically shortest

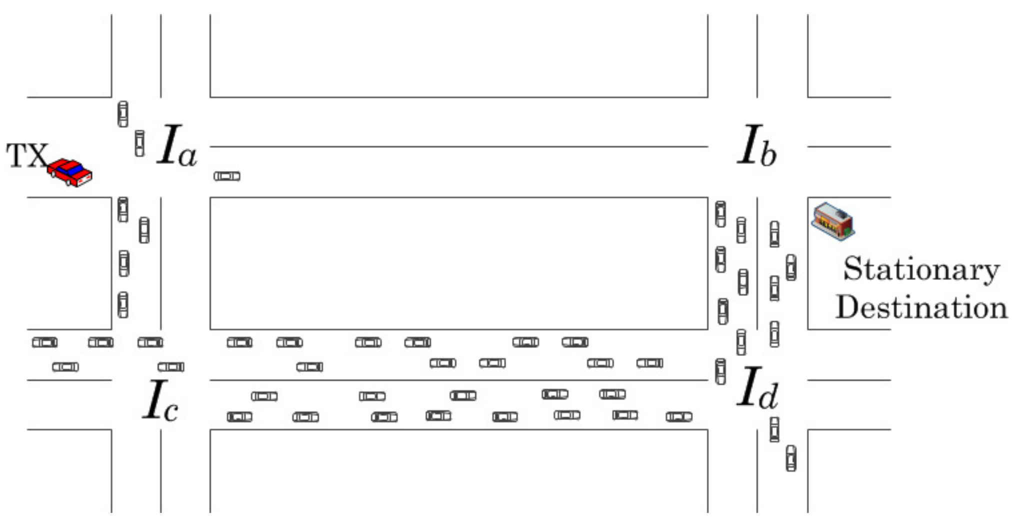


Figure 2.2. VADD scenario between a mobile node and a stationary destination

path especially if the geographically shortest path contains partitioned network and packets have to be carried by vehicles instead of transmitted via wireless channel. In a scenario outlined in [16], which is shown in Figure 2.2, even though path $I_a \rightarrow I_b$ is geographically the shortest path, VADD forwards packets through path $I_a \rightarrow I_c$, $I_c \rightarrow I_d$, $I_d \rightarrow I_b$ since the intermediate node in path $I_a I_b$ will suffer network disconnection, whereas packets in the latter path can be transmitted via wireless channel and with small delivery delay.

Aside from VADD, the *Greedy Traffic-Aware Routing* (GyTAR) protocol, which is proposed in [11, 17], also uses the *carry and forward* approach as one of its routing strategies in solving the network disconnection problem. The protocol consists of two parts, which are *dynamic junction selection procedure* and *forwarding strategy* between between two involved intersections. GyTAR protocol uses digital maps to identify the position of intersections as well as finding the shortest path towards the destination. When choosing the next intersection, the relay node looks for the location of the nearby intersections using the digital city map. For each candidate intersection, the protocol calculates a score using the traffic density between the current and the candidate intersections; and the curvemetric distance of the candidate junction to the destination. The term *curvemetric* refers to the distance measured when following the geometric shape of a road. The candidate intersection with the highest score will be selected. After the next junction is selected, the protocol then implements the improved greedy strategy to forward the packets toward the selected junction.

While GyTAR offers a more intricate solution in selecting the next intersection by taking into account the progression of different intersections towards the destination as well as the vehicles density of each junction; it does not, however, take the vehicles direction into consideration. The intersection selection can be ineffective if most of the vehicles between the current and the selected intersections are moving the opposite direction of the forwarding vehicle.

Zhang, M. and Wolff, R. propose a protocol that uses the carry and forward approach which is called the *Border Node based Routing* (BBR) protocol [18]. The protocol is an infrastructure-less protocol, and it combines a single-hop beaconing service and the carry-and-forward approach to transmit packets to the destination node with minimum delivery delay. BBR uses periodic single-hop beacon packets for *neighbor discovery process* and *border node selection algorithm.* The *neighbor discovery process* is responsible for collecting the current single-hop neighbor information. This information is then used in the *border node selection algorithm.* A *border node* is defined as a node that is closest to the edge of the transmission range and is responsible for saving and forwarding any received packets. However, unlike VADD, BBR does not use any geographical information in selecting the border node. Instead, the algorithm uses the single-hop neighbor information and selects the border node using the *minimum common neighbor approach.* Through a distributed process, this approach selects the node with the least number of common neighbors as the border node. Henceforth, the border node is responsible for broadcasting any received data packets or carrying the packets if there is no node within its transmission range.

Tonguz et al. (2010) [34] propose a broadcast protocol called *Distributed Vehicular Broadcast* (DV-CAST) protocol that is suitable for both sparse and dense network. The protocol is able to handle a broadcast storm problem in a dense network and a disconnected network problem in a sparse network while incurring small amount of overhead. Similar to VADD and BBR, DV-CAST also uses periodic single-hop beacon messages to maintain one-hop neighbor list and employs the carry and forward approach when forwarding packets in a disconnected network. Information exchanged via beacon messages is used by DV-CAST in the neighbor detection mechanism to determine neighbors' connectivity. A neighbor is considered out of range if the vehicle does not hear from that particular neighbor for more than twice the duration of the beacon update interval. Through this mechanism, DV-CAST is able to determine whether a vehicle is in a sparse or well-connected network. In a well-connected network, a vehicle uses one of the DV-CAST broadcast suppression schemes in order to reduce the broadcast storm problem when it receives a broadcast message [34].

2.3. Adaptive Beaconing System in Vehicular Ad hoc Network

The utilization of periodic information exchange using such as *beacons packets* in safety applications has been analyzed in extensive simulations in [35]. It is shown in [35] that with increasing distance, the success ratio decreases quickly. However, this approach could be improved when combined with a position based forwarding strategy. The authors in [36] shows that network load can be significantly reduced by selectively suppressing *broadcasts* based on single-hop neighbor information and the reliability can be increased via the use of explicit acknowledgments. The main challenge for all such beacon systems is that they are very sensitive to environmental conditions such as vehicle density and network load. Many researchers have started to propose an *adaptive beaconing system*, in which the interval rate can be adapted to certain parameters or information that are available in the network.

Sommer et al. [37] propose a method called the *Adaptive Traffic Beacon* (ATB) scheme which allows information to be exchanged as frequently as possible and at the same time, maintain a congestion-free wireless channel. The adaptation in [37] is achieved by using the *channel quality metric* and the *importance of the message* in generating the next *beacon interval* I. The proposed method can also dynamically use infrastructure networks in the vicinity if needed. The interval parameter I, which is in the range of $(0, 1)$, is calculated with the channel quality C, the message utility P and the weighting factor w_I in accordance with the following equation:

$$I = (1 - w_I) \times P^2 + \left(w_I \times C^2\right) \tag{2.1}$$

From parameter I, the protocol determines the *beacon interval* ΔI using the following equation:

$$\Delta I = I_{min} + (I_{max} - I_{min}) \times I \tag{2.2}$$

The channel quality metric in ATB measures the channel availability for the *beacon* *transmission*. The authors in [37] uses number of collisions with parameter K, congestion probability with parameter N, and Signal to Noise ratio (SNR) with parameter S to determine the *channel quality* metric. The three parameters are determined using the equations below:

$$K = 1 - \frac{1}{1 + \#\text{collision}} \tag{2.3}$$

$$N = \min\left\{\left(\frac{\#\text{neighbors}}{\max\ \#\text{neighbor}}\right)^2 ;1\right\} \tag{2.4}$$

$$S = \max\left\{0; \left(\frac{\text{SNR}}{\max\ \text{SNR}}\right)^2\right\} \tag{2.5}$$

Subsequently, using the calculated parameters in Equations (2.3), (2.4) and (2.5), the metric C is formulated as follows:

$$C = \frac{N + w_C \times \dfrac{S + K}{2}}{1 + w_C} \tag{2.6}$$

The message priority P specifies the significance of each message in the current network environment [37]. Basically, it allows message with higher priority to be transmitted first. Equation (2.7), (2.8), (2.9) and (2.10) show four additional parameters that are used to determine the metric P. The parameter A in Equation (2.7) is used to calculate the message age. Parameters D_e and D_r in Equations (2.8) and (2.9), respectively, represents a node's proximity to an event and its proximity to the next road-side unit. Both parameters use the current speed of the vehicle, v to measure the proximity in the form of an approximated travel time. The parameter B in Equation (2.10) is used to discover how much of the information to be sent was not received over a road-side unit [37].

$$A = \min\left\{\left(\frac{\text{message age}}{I_{max}}\right)^2; 1\right\} \tag{2.7}$$

$$D_e = \min\left\{\left(\frac{\text{distance to event}/v}{I_{max}}\right)^2; 1\right\} \tag{2.8}$$

$$D_r = \max\left\{0; 1 - \sqrt{\frac{\text{distance to RSU}/v}{I_{max}}}\right\} \tag{2.9}$$

$$B = \frac{1}{1 + \#\text{unknown entries}} \tag{2.10}$$

Finally, using Equations (2.7), (2.8), (2.9) and (2.10), the metric P is found using the following equation [37]:

$$P = B \times \frac{A + D_e + D_r}{3} \tag{2.11}$$

Thaina et al. proposed two solutions for an *adaptive beaconing system* in [38], which are based on statistical and machine learning techniques. Both solutions generate *beacon intervals* subject to the condition of vehicular networks' topology. The first method is based on linear regression analysis and represented as follows:

$$\widehat{Y} = a + bX, \tag{2.12}$$

where a and b are regression coefficients and expressed using the following equations:

$$a = \overline{y} - b\overline{x} \tag{2.13}$$

$$b = \frac{\sum_{i=1}^{n} (x_i - \overline{x}) \times (y_i - \overline{y})}{\sum_{i=1}^{n} (x_i - \overline{x})^2} \tag{2.14}$$

where:

x_i = the *ith* value of the independent variable X.

y_i = the *ith* value of the dependent variable Y.

$\overline{x}$ = the average value of x.

$\overline{y}$ = the average value of y.

In this first method, the *vehicle's environment parameters* are set as the independent variable whereas the *beacon interval* is the dependent variable [38]. Each node in a vehicular network uses Equation (2.12) to determine the next interval by counting the *number of neighbor nodes* and the *number of buffered messages*. The sum of these two values is indicated as the node's parameter environment, which is X. The value of X is then substituted in Equation (2.12) to obtain the Y value, which is the next *beacon interval*.

The second method proposed in [38] is based on a machine learning technique, which is called as the *k-nearest neighbor algorithm*. The second method still uses the *number of neighboring nodes* and the *number of buffered messages* as its environment parameters, but using a different method of calculation. In *k-nearest neighbor learning*, the training examples are collected in the form of $(x_i, f(x_i))$ with the assumption that each pair of training examples correspond to a point in n-dimensional space. The training examples are collected as a pair of a node's *environment parameter* and the *beacon interval*. The method calculates the number of neighboring nodes and the number of buffered messages and sets the summation of the two parameters as the *environment parameter*, x_q. Equations (2.15) and (2.16) are used to determine all of the nearest neighbors and weight of each nearest neighbor (w_i), respectively. In Equation (2.15), if instance x consists of attribute $\langle a_1(x), a_2(x), \cdots, a_n(x)\rangle$, then attribute $a_r(x)$ denotes the value of the rth attribute of instance x [38].

$$d\left(x_q, x_i\right) \equiv \sqrt{\sum_{r=1}^{n}\left(a_r\left(x_q\right) - a_r\left(x_i\right)\right)^2} \tag{2.15}$$

$$w_i \equiv \frac{1}{d\left(x_q, x_i\right)^2} \tag{2.16}$$

Finally, the next beacon interval is calculated using the following equation:

$$\hat{f}\left(x_q\right) \longleftarrow \frac{\sum_{i=1}^{k} w_i f\left(x_i\right)}{\sum_{i=1}^{k} w_i} \tag{2.17}$$

where k denote the number of nearest neighbors.

A further example on an *adaptive beaconing system* is given by Boukerche et al. [39], which adapt the beaconing rate by predicting the position of a neighbor in the near future. By using a neighbor node's previous location, speed and direction that are stored in a received beacon, a vehicle predicts the location of that particular neighbor node and compares the predicted location with the current location that exists in the received beacon. The beacon packet will be sent out only if the difference between the predicted position and the actual position of a neighbor is greater than the protocol pre-determined threshold value.

2.4. End-to-end Delay Analysis for Carry and Forward based Routing in VANET

Although many research studies have been carried out to incorporate the *carry and forward* approach in their proposed routing protocols, not many studies have been done in analyzing the end-to-end delay when the carry and forward approach is employed during the packet forwarding in VANET. Wu et al. [40] have presented an analytical study on the *information propagation speed* when the *carry and forward* approach is used in both one- and two-way highway scenarios where vehicle arrivals are based on Poisson Process and the vehicle speeds are uniformly distributed in a designated range. The authors provide numerical results on information propagation speed under two network models, which are low density network and high density network.

A number of researches have developed analytical models for studying vehicular network characteristics and performance metrics [41–45]. However, these studies are focusing mainly on information propagation speed model, connectivity model, mobility model and link reliability model. They do not present any probability distribution model on end-to-end delay and the information propagation speed model is normally based on the expected values of the end-to-end delivery delay. The authors in [43] has modified the information propagation speed model from Wu et al. [40] by using a traffic intensity for Poisson arrival model and truncated Gaussian distribution for vehicles' speed. However, the study done by [43] only includes the result on the information propagation speed model and the study does not show any development on the distribution model for end-to-end delay.

Wisitpongphan et al. [20] proposed a similar analytical model as [40] for VANET connectivity in a sparse network. Using empirical traffic data, the authors study and formulate VANET parameters such as *inter-arrival time* and *inter-vehicle spacing*. The authors also derive a comprehensive analytical framework that can be used to characterize a sparse vehicular network for one- and two-directional highways. Furthermore, the authors did an analysis on a parameter similar to the *catch-up time* in [40] which is referred to as the *re-healing time*. However, the study on re-healing time in [20] is focused on the two-directional highways and between adjacent vehicles. Through simulation, the authors are able to validate their analytical framework and analyze and-to-end delay for packet transmission with distance between source and destination varies from 1 to 30 km [20]. Nonetheless, the study on the average end-to-end delay in [20] is purely based on simulation results. There is no analytical model derived for the end-to-end parameter.

A study on end-to-end delay model is done in [46] where the authors analyze the total delay time needed by a relay to carry a packet from a source to a destination using the *carry and forward* system. The main goal of this study is to find the relay's optimal location that minimizes the total delay while taking into account the effect of channel fading, path loss, and forward error correction. However, the study is based on a mobile ad hoc network scenario with only one relay between the source and destination.

In [47], the author also proposes a similar study on the end-to-end model, where the author uses the ergodic Markov chain to model the vehicle's mobility, the exponential distribution for the initial vehicle density, and the normal distribution for the average vehicle speed. The author creates the model for vehicles that are sparsely arranged on a one-directional straight road. Using these assumptions, the author is able to obtain expressions for the exact delay time and delivery ratio. Nevertheless, the model in [47] only considered transmission between two vehicles, not between clusters of vehicles. In addition, the model does not consider the carry and forward approach during packet transmissions. Instead, the authors use T-seconds-wait rule where the packets are discarded if vehicles are unable to transmit them within T seconds.

2.5. Conclusion

In this chapter, we have reviewed a number of works related to the carry and forward approach which done either analytically or through the development of routing protocols. These researches have confirmed that the diversity of network density in a VANET topology can interfere with the transmission of the data packets. The carry and forward approach has allowed researchers to find an alternative in transmitting packets in a disconnected network. However, analytical studies also have shown that this approach can cause information propagation speed to be reduced to the maximum speed of vehicles on a highway. Therefore, this chapter has supported the thesis motivation to find a distribution model for estimating end-to-end delivery delay as well as an alternative approach of packet dissemination with small delivery delay.

Chapter 3

Analytical Framework for End-to-End Delay based on Unidirectional Highway Scenario

3.1. Introduction

In a VANET, when the distance between two vehicles is less than the communication range of the vehicles, these vehicles are able to communicate with each other using the wireless channel. Nevertheless, in a sparse vehicular network, a vehicle normally employs a *store-carry-and-forward* approach, where it holds the message it wants to transmit until the vehicle meets other vehicles or other roadside units. Consequently, the end-to-end delay in VANET is usually high. The study of the *end-to-end delay* in VANET can be considered as one of the most important investigations in vehicular network because of extensive applications such as active safety and emergency response applications [21] require the messages to be transmitted with minimal delay.

Wu et al. [40] have determine an *information propagation speed model* in a VANET that is based on the *carry and forward* scheme. However, the derivation of the *catch-up delay* model is limited to only between a cluster with *informed vehicles* and another cluster with *uninformed vehicles*, and the closed form distribution of the catch-up model is not presented in the numerical results. Therefore, we are not able to ascertain the type of distribution for the *catch-up model.*

In our analysis, we derive the closed form distribution of the *catch-up model* and extend the *catch-up model* to multiple disconnected clusters in a highway with one-way vehicle traffic; and this leads to the derivation of an end-to-end delay model for a unidirectional highway. The extension of the analytical model still uses the same definitions and assumption as provided in [40]. Two assumptions are used to define the vehicle traffic model in the study are:

- ***Poisson arrival***

  Vehicle traffic that pass a random point on the road follows a Poisson process with an average rate equal to the *traffic flow rate* (vehicles per unit time). Empirical studies have shown that Poisson arrival model can be used to model vehicle arrival rate in free flow phase [20] and it is commonly used model in the studies of VANETs [43–45]

- ***Random and independent vehicle mobility***

  Based on the studies done in [40,44,48], vehicles' speed can be represented with uniform distribution with interval $[v_{min}, v_{max}]$, where each vehicle freely moves at its chosen velocity. Each vehicle is then assumed to move along the highway at a constant speed, v m/s, such that the distance between the vehicle and its neighbors remains unchanged.

3.2. System Model of the Time Duration for the Catch-up Phase

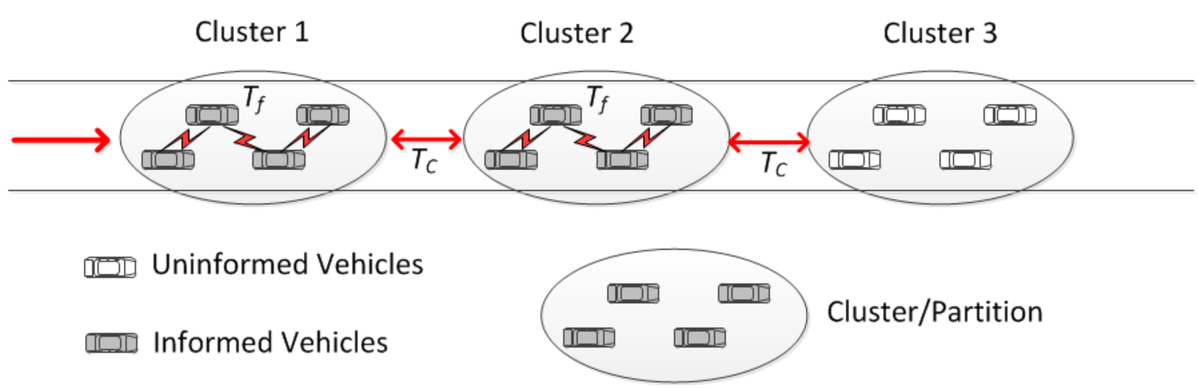


Figure 3.1. Example of message propagation scenario

Consider a scenario presented in Figure 3.1 where a number of vehicles independently travel along on a unidirectional highway of length D meters. The speed of each vehicle, V, is modeled using uniform distribution over $[v_{min}, v_{max}]$. The model assumes vehicles arrive to the highway following a Poisson process and partition into a number of clusters. In this model, a *cluster* can be defined as a group of vehicles that are able to propagate messages using multi-hop forwarding via wireless channel. Road traffic statistics in [49] have shown that vehicles tend to travel in clusters on a highway. The clusters, which are formed in the highway, are split and merged over time due to the mobility of the vehicles. If the gap between two clusters is larger than the transmission range, r, then the *carry and forward* strategy is used to forward messages. A vehicle is considered as an *informed vehicle* if the vehicle has the message that needs to be transmitted. In this scenario, we also assume the source of the messages is located in a cluster of *informed vehicles* and the receiver of the messages is found in an *uninformed cluster* located at the end of the highway. The message is transmitted via one of two ways, either through the forward process or the catch-up process. In the *forward process*, the message is forwarded to other neighboring vehicles within a partition via the wireless channel, where the message rapidly propagates hop by hop until it reaches the farthest vehicle of that partition. In the *catch-up process*, the message travels along with the carrying vehicle until the carrying vehicle arrives within the communication range of the last uninformed vehicle in the partition ahead of it. Once the carrying vehicle is inside the partition with a group of uninformed vehicles, the message will be again propagated via *forward process*. Both processes alternate with each other as the message propagates along the road.

The term T_C in Figure 3.1 is the time duration for the *catch-up process*, where packets are being carried by the carrying vehicle until the vehicle is able to forward the packets via wireless transmission to the last uninformed vehicle in the partition ahead of it. Although, the open form cumulative distribution function (CDF) of T_C has been extensively studied and derived in [40], as shown in Equation (3.1); the authors do not present the CDF or probability density function (PDF) of T_C in their numerical results. Based on the assumptions, expressions, notations, and model parameters provided by the authors in [40], we are able to produce the CDF and PDF of T_C via numerical integration. Table 3.1 lists the notations and parameters use in this analysis.

Table 3.1: Notations and Model Parameters needed for the derivation of T_C distribution

D	The length of a road
L	Gap between two neighboring vehicles
L_c	Gap between two neighboring connected vehicles
L_{uc}	Gap between two neighboring disconnected vehicles
λ	Traffic flow rate (vehicles/unit time)
λ_s	Vehicles density (vehicles/unit distance)
$N(t)$	Number of vehicles arrive at the highway during interval $[0, t]$
N_c	Number of vehicle clusters
r	Vehicle radio range
T_c	Time duration of a catch-up phase
T_f	Time duration of a forward phase
V_i	Average speed of vehicle $i, i = 0, 1, \cdots n$; a random variable in the interval $[v_min, v_max]$
$X(t)$	Message propagation distance during $(0, t]$
$X'(t)$	Distance that the partition tail moves during [0,t]

The cumulative distribution function (CDF) of T_C is presented as:

$$F_{T_c}(t) = P[T_c \leq t] = \int_r^\infty P(T_c \leq t | L_{UC} = l) f_{L_{UC}}(l) dl$$

$$= \int_r^\infty P[X^{'}(t) \leq X(t) + r - l] f_{L_{UC}}(l) dl$$

$$F_{T_c}(t) = \int_r^\infty [\int_r^\infty F_{X'(t)}(x + r - l) f_{X(t)}(x) dx] f_{L_{UC}}(l) dl \qquad (3.1)$$

Based on Equation (3.1), we conclude that we need to derive closed form solutions for $f\,(x;t)$, $F(x', t)$, and $f_{L_{UC}}\,(l)$, which are not presented in [40].

3.2.1. The Derivation of Closed Form Solution for CDF and PDF of Message Propagation Distance, $X\,(t)$

Let $X(t)$ denote the distance traveled by a first vehicle in the front most informed cluster after passing a random location, H, during the time interval $[0, t]$ (refer to Figure 3.1). The CDF of $X(t)$ is expressed as:

$$F(x, t) = \sum_{n=0}^{\infty} P\,[X\,(t) < x | N(t) = n]\,.P\,[N\,(t) = n] \qquad (3.2)$$

In Equation (3.2), $P\,[X\,(t) < x | N(t) = n]$ is given as:

$$P\,[X\,(t) < x | N(t) = n] = P\,[V_0 t < x, V_i\,(t - T_i) < x \text{ for each } i = 1, 2, ..., n] \qquad (3.3)$$

Let V_0 in Equation (3.3) denote the speed of the source vehicle located at the location H at time, $t = 0$. Let V_i denote the speed of the vehicle i and T_i denote the arrival time of a vehicle i at location H after $t = 0$. Based on the previously mentioned assumption of a Poisson arrival in Section 3.1, $N(t)$ is defined as the number of vehicles arrive at the highway during $(0,t]$ and expressed as:

$$P\big[N(t)=n\big]=\frac{e^{\lambda t}(\lambda t)^n}{n!} \tag{3.4}$$

Using the Poisson process theorem [50], given that n vehicles have passed the location H between time $(0,t]$, the arrival times $T_1,\cdots,T_n$ at which the events occur, considered as unordered random variables, are distributed independently and uniformly in the interval (0, t). Therefore, with this theorem, T_i can be presented as $T_i \sim \text{uniform}\,(0,t)$. With the assumptions that $T_1,T_2,\cdots,T_n$ and $V_1,V_2,\cdots,V_n$ are independent identically distributed (i.i.d) and uniformly distributed at the interval $(0,t]$ and $[v_{min},v_{max}]$ respectively, Equation (3.3) can be expressed as:

$$P\left[X\left(t\right)<x|N(t)=n\right]=P\left[Vt<x\right]P\left[V\left(t-T\right)<x\right]^n \tag{3.5}$$

Using Equations (3.5) and (3.4), the CDF of $X(t)$ in Equation (3.2) can be written as:

$$\begin{aligned}
F(x,t)&=\sum_{n=0}^{\infty}P\left[X\left(t\right)|N\left(t\right)=n\right]P\left[N\left(t\right)=n\right]\\
&=\sum_{n=0}^{\infty}P\left[Vt<x\right]P\left[V\left(t-T\right)<x\right]^n\frac{e^{\lambda t}(\lambda t)^n}{n!}\\
&=P\left[Vt<x\right]\sum_{n=0}^{\infty}\left[\left(P\left[V(t-T)<x\right]\right)^n\frac{e^{\lambda t}(\lambda t)^n}{n!}\right]
\end{aligned} \tag{3.6}$$

To solve Equation 3.6, we need to solve $P\left[Vt<x\right]$ and $P\left[V\left(t-T\right)<x\right]$.

Let V denote the vehicles' speed and it is uniformly distributed at interval $[v_{min},v_{max}]$.

Therefore, $P[Vt < x]$ can be expressed as:

$$P\big[Vt < x\big] = P\left[V < \frac{x}{t}\right] = \begin{cases} 0, & \frac{x}{t} < v_{\min} \\ \frac{\left(\frac{x}{t}\right) - v_{min}}{v_{max} - v_{min}}, & v_{min} \leq \frac{x}{t} \leq v_{max} \\ 1, & \frac{x}{t} > v_{max} \end{cases} \tag{3.7}$$

The probability $P[V(t-T) < x]$ in Equation (3.6) denote the probability of the distance traveled by n vehicles after passing the location H at the speed between the interval $[v_{min}, v_{max}]$ and at the time in the interval $(0, t)$. The probability can be computed as:

$$\begin{aligned} P\left[V(t-T) < x\right] &= \int_{v_{min}}^{v_{max}} P\left[t - T \leq \frac{x}{V} \,|V = v\right] f_V(v)dv \\ &= \int_{v_{min}}^{v_{vmax}} P\left[t - T \leq \frac{x}{v}\right] f_V(v)dv \end{aligned}$$

$$P\left[V(t-T) < x\right] = \int_{v_{min}}^{v_{max}} P\left[T > t - \frac{x}{v}\right] f_V(v)dv \tag{3.8}$$

With the assumption that the random variable T is independent identically distributed (i.i.d.) and uniformly distributed in the interval $(0, t]$, $P\left[T > t - \frac{x}{v}\right]$ in Equation (3.8) can be expressed as:

$$\begin{aligned} P\left[T > \left(t - \frac{x}{v}\right)\right] &= 1 - P\left[T < \left(t - \frac{x}{v}\right)\right] \\ &= 1 - \frac{1}{t}\left(t - \frac{x}{v}\right) \end{aligned}$$

$$P\left[T > \left(t - \frac{x}{v}\right)\right] = \begin{cases} \frac{x}{vt} & x < vt; v > \frac{x}{t} \\ 1 & x > vt; v < \frac{x}{t} \end{cases} \tag{3.9}$$

Substituting Equation (3.9) in Equation (3.8), we arrive at:

$$P\left[V\left(t-T\right)<x\right]=\int_{v_{min}}^{x/t}1\cdot f_v\left(v\right)dv+\int_{x/t}^{v_{max}}\frac{x}{vt}\cdot f_v\left(v\right)dv$$

$$P\left[V\left(t-T\right)<x\right]=\frac{x-tv_{min}+x\ln\left(\dfrac{v_{max}}{x/t}\right)}{t\left(v_{max}-v_{min}\right)} \tag{3.10}$$

Using Equations (3.4),(3.7) and (3.10), we are able to formulate closed form solution for $F\left(x,t\right)$, which is shown below.

$$F\left(x,t\right)=P\left[Vt<x\right]\sum_{n=0}^{\infty}\left[\left(P\left[V(t-T)<x\right]\right)^n\frac{e^{\lambda t}(\lambda t)^n}{n!}\right]$$

$$=\frac{(x/t)-v_{min}}{v_{max}-v_{min}}\sum_{n=0}^{\infty}\left(\left(\frac{x-tv_{min}+x\ln\left(\dfrac{v_{max}}{x/t}\right)}{t}\right)^n\frac{e^{-\lambda t}\left(\lambda t\right)^n}{n!}\right)$$

$$F\left(x,t\right)=\begin{cases}\dfrac{(x/t)-v_{min}}{e^{\lambda t}\left(v_{max}-v_{min}\right)}\cdot\sigma_1 & ;v_{min}\leq\frac{x}{t}\leq v_{max}\\ 1 & ;\frac{x}{t}>v_{max}\end{cases} \tag{3.11}$$

where

$$\sigma_1=e^{\frac{\lambda\left(x-tv_{min}+x\ln\left(\frac{tv_{max}}{x}\right)\right)}{v_{max}-v_{min}}}$$

Next, we take the derivative of $F_{X(t)}\left(x\right)$ to derive the PDF of $X\left(t\right)$:

$$f\left(x;t\right)=\frac{dF_{X(t)}}{dx}$$

$$=\frac{e^{\frac{\lambda\sigma_1}{v_{max}-v_{min}}}}{te^{\lambda t}\left(v_{max}-v_{min}\right)}-\frac{\lambda\sigma_1\ln\left(\dfrac{v_{max}}{x/t}\right)\left(v_{min}-\frac{x}{t}\right)}{e^{\lambda t}\left(v_{max}-v_{min}\right)^2} \tag{3.12}$$

3.2.2. The Derivation of Closed Form Solution for Cumulative Distribution Function of $X'(t)$ - Distance that the Partition Tail Moves

Let $X'(t)$ denote the distance traveled by the last vehicle in an uninformed cluster that is in front of an informed cluster during the time interval $[0, t]$. The CDF of $X'(t)$ is expressed as:

$$F(x', t) = 1 - \sum_{n=0}^{\infty} P\left[X'(t) > x | N'(t) = n\right] P\left[N'(t) = n\right] \tag{3.13}$$

where

$$P\left[X'(t) > x | N'(t) = n\right] P\left[N'(t) = n\right] = P\left[V_0 t > x, V_i(t - T_i) \text{ for each } i = 1, 2, ..., n\right] \tag{3.14}$$

In Equation (3.14), V_0 is defined as the speed of the uninformed vehicle at location J at time $t = 0$ and $N'(t)$ denote the number of vehicles that pass location J. As shown in Figure 3.1, we conclude there are no other uninformed vehicles pass location J after the last uninformed vehicle during a catch-up phrase, and hence, the earliest time for the last uninformed vehicle to pass location J is $-\left(\frac{v_{max}}{v_{min}} - 1\right) t$, with the condition that a vehicle is in location J at the time 0 and the vehicle speed is in the interval $[v_{min}, v_{max}]$.

Using the same properties of the Poisson process, the distribution of T in this analysis is based on uniform distribution in the interval $\left[-\left(\frac{v_{max}}{v_{min}} - 1\right) t, 0\right]$. Thus, Equation (3.4) has to be rewritten according the the new distribution of T:

$$P\left[N^{'}(t) = n\right] = \frac{e^{-\lambda\left(\frac{v_{max}}{v_{min}} - 1\right) t} \left(\lambda\left(\frac{v_{max}}{v_{min}} - 1\right) t\right)^n}{n!} \tag{3.15}$$

Using the same properties of independent identically distributed for T_i as in Section 3.2.1, Equation (3.14) can be rewritten as:

$$P\left[X'\left(t\right) > x|N'\left(t\right) = n\right]P\left[N'\left(t\right) = n\right] = P[Vt > x]P[V\left(t-T\right) > x]^n \tag{3.16}$$

Substituting Equations (3.15) and (3.16), the CDF of $X'\left(t\right)$ in Equation (3.13) can be expressed as follows:

$$\begin{aligned} F\left(x',t\right) &= 1-\sum_{n=0}^{\infty}P\left[X'\left(t\right) > x|N'(t)=n\right]P\left[N'(t)=n\right] \\ &= 1-P[Vt > x]\sum_{n=0}^{\infty}\left[P[V\left(t-T\right) > x]^n\frac{e^{-\lambda\left(\frac{v_{max}}{v_{min}}-1\right)t}\left(\lambda\left(\frac{v_{max}}{v_{min}}-1\right)t\right)^n}{n!}\right] \end{aligned} \tag{3.17}$$

In Equation (3.17), $P\left[Vt > x\right]$ is expressed as:

$$\begin{aligned} P\left(Vt > x\right) &= 1-P\left(Vt < x\right) \\ P\left(Vt > x\right) &= \frac{v_{max}-\frac{x}{t}}{v_{max}-v_{min}} \end{aligned} \tag{3.18}$$

In this section, $P[V\left(t-T\right) > x]$ denote the probability of the distance traveled by n vehicles after passing location J in the interval of $\left[-\left(\frac{v_{max}}{v_{min}}-1\right)t,0\right]$ with a speed between $[v_{min},v_{max}]$. Therefore, the probability $P\left[T < \left(t-\frac{x}{v}\right)\right]$ in Equation (3.9) can be rewritten as:

$$P\left[T < \left(t-\frac{x}{v}\right)\right] = \frac{v_{min}}{\left(v_{max}-v_{min}\right)}-\frac{xv_{min}}{tv\left(v_{max}-v_{min}\right)}+1 \tag{3.19}$$

Using Equation (3.19), Equation (3.10) can be rewritten as:

$$\begin{aligned} P\Big[V\left(t-T\right) > x\Big] &= \int_{v_{min}}^{x/t}P\left[T < t-\frac{x}{v}\right]f_V(v)dv+\int_{x/t}^{v_{max}}P\left[T < t-\frac{x}{v}\right]f_V(v)dv \\ &= \frac{v_{min}\left(\frac{x}{t}-v_{min}\right)}{\left(v_{max}-v_{min}\right)^2}-\frac{xv_{min}}{tv\left(v_{max}-v_{min}\right)^2}\left[\ln\left(\frac{x}{t}\right)-\ln\left(v_{min}\right)\right]+1 \end{aligned} \tag{3.20}$$

Finally, the closed form solution for $F(x', t)$ can be expressed as:

$$F(x',t) = 1 - \sum_{n=0}^{\infty} P\left[X'(t) > x | N'(t) = n\right] P\left[N'(t) = n\right]$$

$$= 1 - P\left[Vt > x\right] \sum_{n=0}^{\infty} P\left[V(t-T) > x\right]^n P\left[N'(t) = n\right]$$

$$= 1 - \frac{v_{max} - \frac{x}{t}}{v_{max} - v_{min}} \times$$

$$\sum_{n=0}^{\infty} \left[\left(\frac{v_{min}\left(\frac{x}{t} - v_{min}\right)}{\left(v_{max} - v_{min}\right)^2} - \frac{x v_{min}}{tv\left(v_{max} - v_{min}\right)^2} \left[\ln\left(\frac{x}{t}\right) - \ln\left(v_{min}\right)\right] + 1 \right)^n \times \frac{e^{-\lambda\left(\frac{v_{max}}{v_{min}} - 1\right)t} \left(\lambda\left(\frac{v_{max}}{v_{min}} - 1\right)t\right)^n}{n!} \right]$$

$$F(x',t) = \begin{cases} 0 & ; v_{max} < \frac{x}{t} \\ 1 - \dfrac{e^{\sigma_2}\left(v_{max} - \left(\frac{x}{t}\right)\right)}{e^{\lambda\left(\frac{v_{max}}{v_{min}} - 1\right)t}\left(v_{max} - v_{min}\right)} & ; v_{min} \leq \frac{x}{t} \leq v_{max} \\ 1 & ; v_{min} > \frac{x}{t} \end{cases} \tag{3.21}$$

where

$$\sigma_2 = \frac{\lambda\left(tv_{max}^2 - 2tv_{max}v_{min} + v_{min}x + v_{min}x\ln(v_{min}) - v_{min}x\ln\left(\frac{x}{t}\right)\right)}{v_{min}\left(v_{max} - v_{min}\right)}$$

3.2.3. Derivation of PDF for $f_{L_{UC}}$ - the Distribution of Disconnected Vehicles Gap

As stated in Table 3.1, L is the gap between two neighboring vehicles and L_{UC} is the gap between two neighboring disconnected vehicles. Two neighboring vehicles are considered disconnected if the gap between the vehicles is larger than communication range, r. If the spacing between two neighboring vehicles, L follows an exponential distribution, then the derivation of the PDF of $f_{L_{UC}}(l)$ is as follows:

$$\begin{aligned} f_{L_{UC}}(l) &= f_L\left(l \mid L > r\right) \\ &= \frac{f_L(l)}{1 - F_L(r)} \quad ; l < r \end{aligned} \tag{3.22}$$

According to Wisitpongphan et al. in [20], the spacing between two vehicles can be expressed by an exponential distribution and the validity of assumption for VANETs has been confirmed by the empirical measurement reported in [20]. Therefore, the inter-vehicle spacing is denoted as:

$$f_s\left(s\right) = \lambda_s e^{-\lambda_s s}, \tag{3.23}$$

where the parameter λ_s, which is the vehicle density, can be estimated as:

$$\lambda_s = \frac{\lambda}{E[V]}$$

If we use Equation (3.23) in $f_L(l)$ and $F_L(r)$, we are able to formulate the vehicle gap distribution for $l > r$:

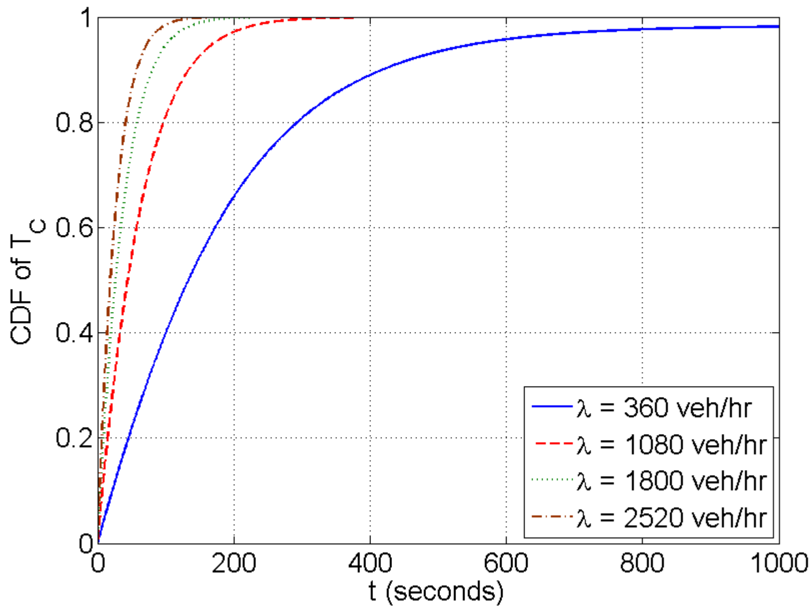
$$f_{L_{UC}}(l) = \frac{\lambda_s e^{-\lambda_s l}}{1 - (1 - e^{-\lambda_s r})}$$

$$f_{L_{UC}}(l) = \lambda_s e^{-\lambda_s l} e^{\lambda_s r} \tag{3.24}$$

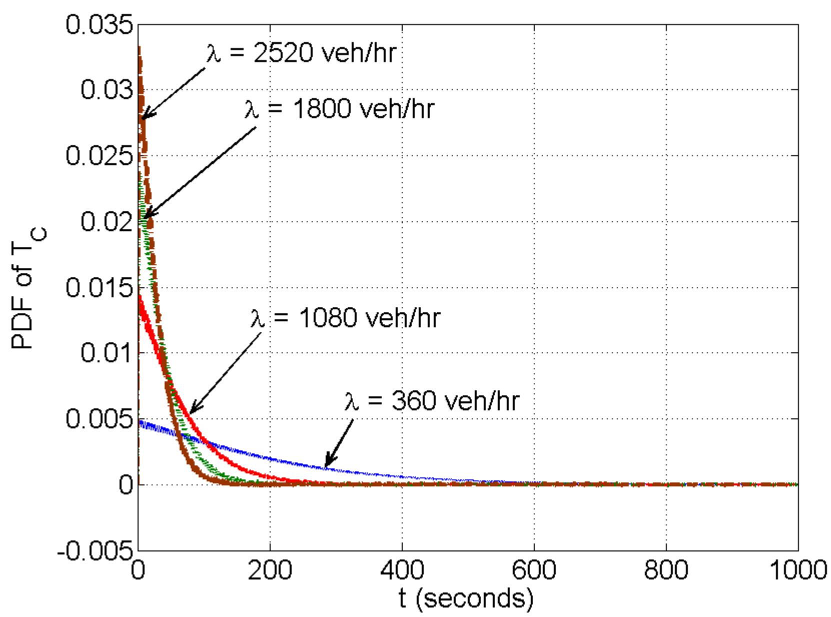
3.2.4. Derivation of CDF and PDF of T_C via Numerical Integration and Approximation of T_C Distribution

It should be noted that the CDF of T_C given in Equation (3.1) does not have a closed form solution; but has to be evaluated via numerical integration. Plots in Figures 3.2 present the probability of T_C distribution against time in seconds for different values of flow rate.

In general, Figures 3.2 indicate with the increase in vehicle flow rates, the distance between vehicles decreases and, therefore, decreases the catch up delay. The x-axis in Figures 3.2 denote the catch-up delay between two disconnected clusters of vehicles. A large value of t indicates that the data packets are carried by an informed vehicle in a *catch-up phase* most of the time and a small value of t indicates the *catch-up phase* occurs in a short time and the data packets are transmitted using *forwarding phase* most of the time. As shown in Figure 3.2a, in a sparse network, the *catch-up phase* occurs at a high delay. At $\lambda = 360$ veh/hr, the frontmost vehicle of an informed cluster of vehicles has a probability of 70% to catch up with the last vehicle in an uninformed cluster of vehicles at a time delay t larger or equal to 200 seconds. However, in a dense network with $\lambda = 2520$ veh/hr, the catch-up process is highly likely to happen at approximately 100 seconds or less.



(a) CDF of T_C



(b) PDF of T_C

Figure 3.2. F_{T_C} distribution with different values of flow rates (vehicle/hour)

Figures 3.2 show that the T_C distribution has the shape of an exponential distribution. Therefore, we use exponential regression analysis to approximate the T_C distribution with an exponential distribution using the following equation [51,52]:

$$g(x) = ae^{bx} \tag{3.25}$$

where a and b are constant called the *model regression coefficients* [51,52]. An exponential regression analysis is performed by applying the logarithm to the base of e of both sides of Equation (3.25). Subsequently, Equation (3.25) can be written by:

$$\log_e g(x) = \log_e \left(ae^{bx}\right)$$

$$\ln g(x) = \ln a + bx \tag{3.26}$$

By substituting $y = \ln g(x)$, a linear regression analysis equation, which is expressed as $y = a + bx$, can be rewritten as:

$$y = \ln a + bx, \tag{3.27}$$

where the *regression coefficients* a and b are expressed as:

$$b = \frac{\sum xy - n\bar{x}\bar{y}}{\sum x^2 - n\bar{x}^2} \tag{3.28}$$

$$\ln a = \bar{y} - b\bar{x} \tag{3.29}$$

where

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i, \qquad \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i, \qquad \sum xy = \sum_{i=1}^{n} x_i y_i, \qquad \text{and} \qquad \sum x^2 = \sum_{i=1}^{n} x_i^2$$

Table 3.2: Exponential Regression Parameters

Vehicle Flow Rate	SSE	R^2	coefficient a	coefficient b
360	0.0000399	0.9760	0.005067	-0.00498
1080	0.0002648	0.9619	0.01465	-0.01414
1800	0.0006772	0.9448	0.02371	-0.02283
2520	0.0013000	0.9282	0.03227	-0.03101

There are two main parameters in regression analysis that can indicate an exponential distribution is a good fit for T_C distribution [51, 52]:

i. **Sum of square errors (SSE)**

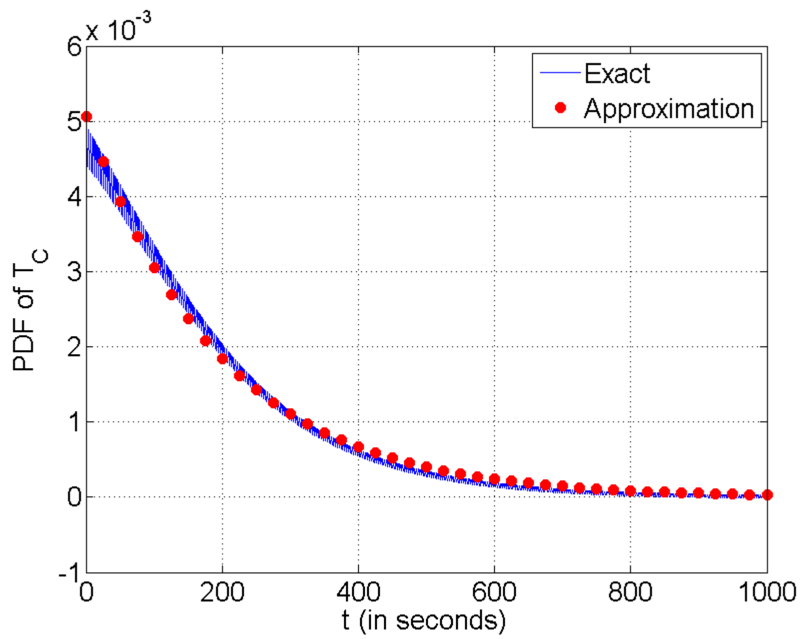
   In general, this parameter measures the difference between data points and an estimation model with a value closer to zero to indicate a good fit.

ii. **Coefficient of determination (R^2)**

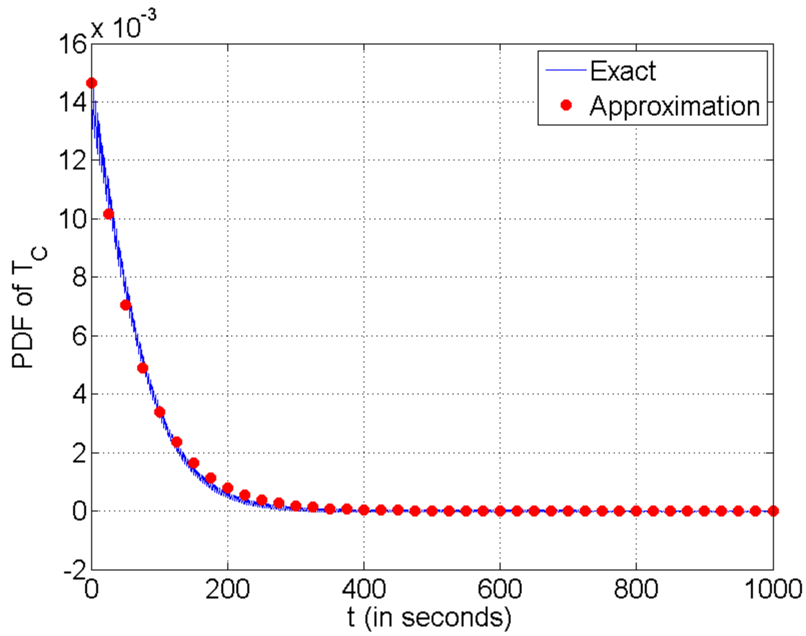
   This parameter indicates how well data points fit an approximation curve with a value approaching to one to demonstrate a good fit.

Table 3.2 shows the output of the exponential regression analysis from MATLAB Curve Fitting Tool for T_C distributions for vehicle flow rates 360, 1080, 1800 and 2520 veh/hr. For each of the traffic flow rates, the exponential regression yields a high value for R^2 parameter, which is larger than 0.9. Table 3.2 also indicates the parameter SSE yields values that are very close to zero. With the parameter R^2 yield values close to one and the SSE values produces values near to zero in the exponential regression analysis, we can ascertain that our T_C distribution can be approximated with an exponential distribution.

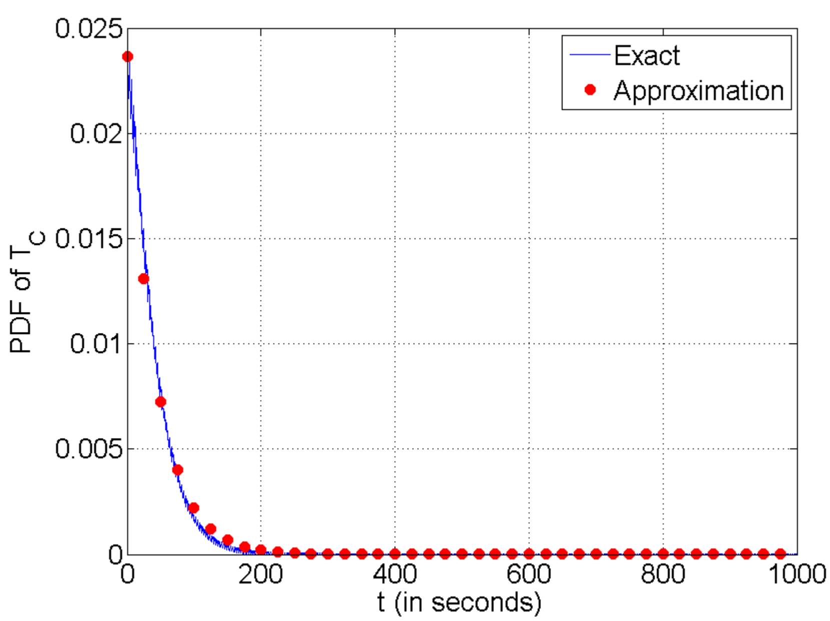
Figures 3.3a - 3.3d plot the exact distribution function of T_C with its approximation counterpart, given in Equation (3.25) for the respective vehicle flow rate. From these figures, we can establish the high accuracy of the approximation. Based on Figures 3.3, we determine that the exact distribution function of T_C can be approximated using an exponential distribution expression.



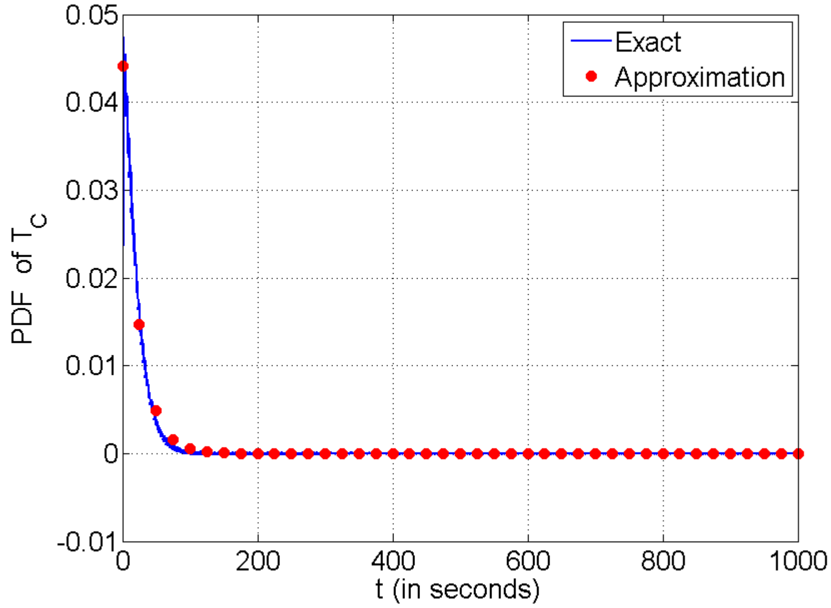
(a) Exact f_{T_C} versus approximate f_{T_C} for $\lambda = 360$ veh/hr



(b) Exact f_{T_C} versus approximate f_{T_C} for $\lambda = 1080$ veh/hr



(c) Exact f_{T_C} versus approximate f_{T_C} for $\lambda = 1800$ veh/hr



(d) Exact f_{T_C} versus approximate f_{T_C} for $\lambda = 2520$ veh/hr

Figure 3.3. The comparison between the exact distribution of T_C and its approximation

3.3. Derivation of the Sum of Multiple Catch-up Time, T_D, for Unidirectional Highway

Using Figure 3.1 as an example, total catch-up time, T_D, can be expressed as:

$$T_D = T_{f_1} + T_{c_1} + T_{f_2} + T_{c_2} + T_{f_3} + T_{c_3} + \cdots + T_{f_n} + T_{c_n} \tag{3.30}$$

where:

T_{f_k}= *forwarding time* in Cluster k

T_{c_k}= *catching up time* from Cluster k to Cluster $k+1$

It is assumed that low vehicle density in a vehicular network causes the communication range to become smaller than the average inter-vehicle gap. Based on this assumption, the message transmission time can be approximated using entirely on the vehicle movement while ignoring the message transmission time within a cluster, i.e., T_{f_1} as it is very small, since the packets are transferred via wireless channel within the cluster, resulting in $T_{f_1} \ll T_{C_1}$. Therefore, Equation (3.30) can be re-written as:

$$T_D = T_{c_1} + T_{c_2} + T_{c_3} + \cdots + T_{c_n} \tag{3.31}$$

Generalizing Equation (3.31) for N_c clusters, we are able to derive T_D as:

$$T_D = \sum_{k=1}^{N_c} T_{c_k} \tag{3.32}$$

Let N_c denote the number of vehicle clusters on a highway with the assumption that N_c is a random variable that is independent of T_C's where N_c derivation will be explained later in Section 3.3.1. Hence, we can find the conditional PDF of T_D given that $N_c = n$ using the conditional characteristic function of T_D. From Equation (3.31), the conditional characteristic function of T_D given that $N_c = n$ can be expressed as:

$$\begin{aligned} T_D &= T_{C_1} + T_{C_2} + \cdots + T_{C_n} \\ \Phi_{T_D|N_c=n}(s) &= E\left[e^{sT_D}\right] \\ &= E\left[e^{s\left(T_{C_1}+T_{C_2}+\cdots+T_{C_N}\right)}\right] \\ \Phi_{T_D|N_c=n}(s) &= \left\{\Phi_{T_C}(s)\right\}^n \end{aligned} \tag{3.33}$$

The conditional PDF of T_D can be found by taking the inverse transform of $\Phi_{T_D|N_c=n}$. However, since the distribution of T_C in Equation (3.1) is found using numerical integration, a closed form solution is not feasible for Equation (3.33). In

Section 3.2.4, using exponential regression analysis, we have determined the catch-up time (T_C) distribution in Figures 3.2 can be approximated with an exponential distribution. Henceforth, f_{T_C} can be denoted with the following expression:

$$f_{T_C}(t) \approx \lambda_{T_C} e^{-\lambda_{T_C} t}, \tag{3.34}$$

where the value of λ_{T_C} should be found using minimum mean square error (MMSE) between the exact distribution function of T_C in Equation (3.1) and an exponential distribution. Using the approximation of T_C as an exponential distribution, the conditional PDF of T_D given that $N_c = n$ can be found using characteristic function expressions of T_C and sums of T_C's. The characteristic function of T_C can be expressed as:

$$\begin{aligned}
\Phi_{T_C}(s) &= E\left[e^{sT_C}\right] \\
&= \int_0^\infty e^{st} f_{T_C} dt \\
&= \int_0^\infty e^{st} \lambda_{T_C} e^{-\lambda_{T_C} t} dt \\
\Phi_{T_C}(s) &= \frac{\lambda_{T_C}}{(\lambda_{T_C} - s)}
\end{aligned} \tag{3.35}$$

From Equation (3.35), we can find the expression for the characteristic function of T_D given that $N_C = n$:

$$
\begin{aligned}
T_D &= T_{C_1} + T_{C_2} + \cdots + T_{C_n} \\
\Phi_{T_D|N_c=n}(s) &= E\left[e^{sT_D}\right] \\
&= E\left[e^{s\left(T_{C_1}+T_{C_2}+\cdots+T_{C_n}\right)}\right] \\
&= \Phi_{T_{C_1}}(s)\cdots\Phi_{T_{C_N}}(s) \\
\Phi_{T_D|N_c=n}(s) &= \{\Phi_{T_C}(s)\}^n \\
\Phi_{T_D|N_c=n}(s) &= \left\{\frac{\lambda_{T_C}}{(\lambda_{T_C}-s)}\right\}^n
\end{aligned}
\tag{3.36}
$$

Thus, the conditional PDF of T_D can be found using inverse Laplace Transform of the conditional characteristic function of T_D in Equation (3.36) which is expressed in Equation (3.37):

$$
\begin{aligned}
f_{T_D|N_c=n}(t) &= \mathcal{L}^{-1}\{\Phi_{T_D}(s)\} \\
&= \mathcal{L}^{-1}\left\{\left\{\frac{\lambda_{T_C}}{(\lambda_{T_C}-s)}\right\}^n\right\} \\
f_{T_D|N_c=n}(t) &= \frac{\lambda_{T_C}e^{-\lambda_{T_C}t}\left(\lambda_{T_C}t\right)^{n-1}}{(n-1)!}
\end{aligned}
\tag{3.37}
$$

From Equation (3.37), we ascertain that the conditional PDF of T_D given that $N_C = n$ follows an Erlang(n, λ_{T_C}) distribution.

3.3.1. Analysis on the Distribution of Number of Clusters

To find the PDF of T_D, we have to derive the distribution model for the number of clusters, N_C. Based on the message propagation scenario in Figure 3.1 of Section 3.2, we consider a unidirectional highway of length D meters. The source vehicle is located in the first informed cluster and the destination vehicle is located at the end of the highway. Therefore, a message from the source vehicle has to be propagated over multiple clusters of vehicles in order to be transmitted to the destination vehicle. In addition, we assume the vehicles enter the highway according to a Poisson process with traffic flow rate of λ. Therefore, by employing the Poisson process assumption, we consider the number of clusters can be modeled by using the Poisson distribution. Subsequently, by applying the vehicular network analytical framework provided in [20] and [40], we are able to formulate two different models for number of clusters for a unidirectional highway of length D. The derivation of the two models will be explained in detail in the following sections.

3.3.1.1. Analysis of N_{C_1}

We denote N_{C_1} as the number of cluster derivation in the first analysis. Hence, we derive the N_{c_1} distribution using the parameters $E[S_{inter}]$ and $E[C_L]$ provided in [20]:

(a) **Average inter-cluster spacing,** $E\left[S_{inter}\right]$

In accordance with the concept of cluster, inter-cluster spacing is defined as the spacing between the last vehicle of the leading cluster and the first vehicle of the following cluster given that the spacing is larger than the transmission range r [20]. Since it has been established in [20] that the inter-vehicle spacing is exponentially distributed with parameter λ_s, the average inter-cluster spacing can be expressed as:

$$E\left[S_{inter}\right] = r + \frac{1}{\lambda_s}$$

(b) **Average cluster length,** $E\left[C_L\right]$

Cluster length is defined as the length between the first vehicle and the last vehicle in a cluster. The average cluster length is expressed as:

$$E\left[C_L\right] = \left(\frac{1}{e^{-\lambda_s r}}\right)\left(\frac{1}{\lambda_s} - \frac{re^{-\lambda_s r}}{1-e^{\lambda_s r}}\right)$$

From Figure 3.4, we derive the probability mass function of N_{C_1} as:

$$P\left(N_c = n\right) = \frac{\alpha_1^n}{n!}e^{\alpha_1}, \qquad \text{with } \alpha_1 = \frac{D}{E\left[S_{inter}\right] + E\left[C_L\right]} \tag{3.38}$$

where:

D is the length of the highway

$E\left[S_{inter}\right]$ is the average inter-cluster spacing,

$E\left[C_L\right]$ is the average cluster length

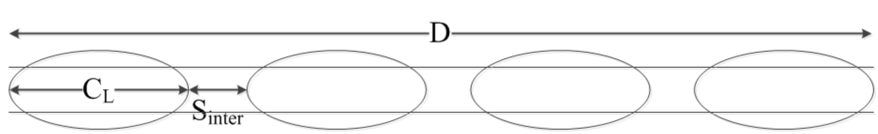


Figure 3.4. Scenario for N_{C_1} distribution

3.3.1.2. Analysis of N_{C_2}

N_{C_2} denote number of clusters in the second analysis and its derivation is based on vehicular network paramters $E[X_f]$ and $E[X_c]$ provided in [40].

(a) **Average distance during forwarding phase,** $E\left[X_f\right]$

X_f represents the distance traveled by messages during a *forwarding phase* where the expected value is given as:

$$E\left[X_f\right] = E\left[Y\right] + r, \tag{3.39}$$

where

r is the transmission range, and

$E\left[Y\right]$ is the average cluster size and it is expressed as:

$$E\left[Y\right] = E\left[M\right] E\left[L_c\right], \tag{3.40}$$

where

$E\left[M\right]$ is the average number of vehicle gaps in a cluster.

$E\left[L_c\right]$ is the average gap between two connected neighboring vehicles.

Using Equation (3.23), we derive the following closed form expressions for Equations (3.41) and (3.42), respectively:

$$\begin{aligned} E\left[M\right] &= \frac{F_L\left(r\right)}{1 - F_L\left(r\right)} \\ &= \frac{1 - e^{-\lambda_s r}}{e^{-\lambda_s r}} \end{aligned} \tag{3.41}$$

$$\begin{aligned} E\left[L_c\right] &= \int\limits_0^r l \frac{f_L\left(l\right)}{f_L\left(r\right)} dl \\ &= \frac{1 - \left(e^{-\lambda_s r}\left(\lambda_s r + 1\right)\right)}{\lambda_s\left(1 - e^{-\lambda_s r}\right)} \end{aligned} \tag{3.42}$$

If we substitute Equations (3.41) and (3.42) in Equation (3.40), we arrive at:

$$\begin{aligned} E\left[Y\right] &= E\left[M\right] E\left[L_c\right] \\ &= \frac{e^{-\lambda_s r}\left(1-\left(e^{-\lambda_s r}\left(\lambda_s r+1\right)\right)\right)}{\lambda_s}, \end{aligned}$$

Therefore:

$$E\left[X_f\right] = r + \frac{e^{-\lambda_s r}\left(1-\left(e^{-\lambda_s r}\left(\lambda_s r+1\right)\right)\right)}{\lambda_s}$$

(b) **Average distance during catch-up phase,** $E\left[X_c\right]$

X_c denotes distance traveled by messages during a catch-up phase and the expected value is expressed as:

$$\begin{aligned} E\left[X_c\right] &= \int_0^\infty E\left[X_c | T_c = t\right] f_{T_C}\left(t\right) dt \\ &= \int_0^\infty E\left[X\left(t\right)\right] f_{T_C}\left(t\right) dt \end{aligned} \tag{3.43}$$

It should be noted that the closed form expression is not feasible for Equation (3.43) since $E\left[X\left(t\right)\right]$ is solved using numerical integration. Therefore, Equation (3.43) is solved through numerical integration using the exact distribution of T_C in Equation (3.1).

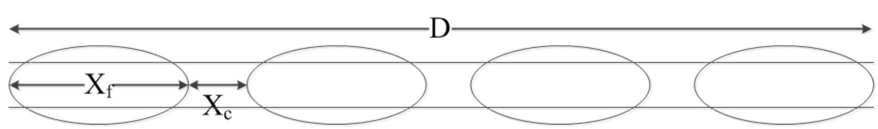


Figure 3.5. Scenario for N_{C_2} distribution

Based on Figure 3.5, the distribution of N_{C_2} can be expressed as:

$$P\left(N=n\right)=\frac{\alpha_2^n}{n!}e^{\alpha_2}, \qquad \text{with } \alpha_2=\frac{D}{E\left[X_f\right]+E\left[X_C\right]} \tag{3.44}$$

where:

D is the length of the highway,

$E\left[X_f\right]$ is the expected value of the forwarding distance,

$E\left[X_C\right]$ is the expected value of the catch-up distance

3.3.2. Derivation of T_D Distribution based on T_C and N_c distributions

Using N_C distributions, we are able to formulate $f\left(T_D;t\right)$ using *Law of Total Probability* [53] as:

$$\begin{aligned} f\left(T_D;t\right) &= f_{T_D}\left(t|N_c=n_1\right)P\left(N_c=n_1\right)+f_{T_D}\left(t|N_c=n_2\right)P\left(N_c=n_2\right)+ \\ &\quad f_{T_D}\left(t|N_c=n_3\right)P\left(N_c=n_3\right)\cdots \\ f\left(T_D;t\right) &= \sum_{n=1}^{\infty}\frac{\lambda_{T_C}e^{-\lambda_{T_C}t}\left(\lambda_{T_C}t\right)^{n-1}}{(n-1)!}\cdot\frac{\alpha_i^n}{n!}e^{\alpha}, \end{aligned} \tag{3.45}$$

where i can be either 1 or 2.

3.4. Results and Analysis for Distribution of Total Catch-up Time T_D in One Way Street

3.4.1. Numerical Results

In this section, we present some pertinent numerical results regarding the analysis done in this chapter. Figures 3.6 and 3.7 show the probability mass functions for N_{C_1} from Equation (3.38) and N_{C_2} from Equation (3.44), respectively, for traffic rate $\lambda = 360, 1080, 1800$ and 2520 veh/hr. Both Figures 3.6 and 3.7 reveal that N_{C_1} and N_{C_2} distributions have different peak in low traffic rates less than 1800 veh/hr. For $\lambda = 360$ veh/hr, the highest peaks for the N_{C_1} distribution is at approximately 15 clusters indicating large number of clusters are formed at low traffic rate; whereas the N_{C_2} distribution has the highest peak at less than 10 clusters. It is also interesting to note that the N_{C2} distribution does not show much difference for traffic rate 360 and 1080 veh/hr compared to the N_{C1} distribution.

Furthermore, there is a correlation between number of clusters and traffic density where the number of clusters decrease as traffic density increases. The relationship is shown in Figure 3.8, which exhibits average cluster length, $E[C_L]$; and average inter-vehicle spacing, $E[S_{inter}]$. Figure 3.8 shows the average cluster length increases as vehicle traffic flow rate increases which indicates that as traffic density increases, the inter-vehicle spacing reduces until the gap is less than the transmission range. Consequently, clusters in the highway merge to form a larger cluster, thereby reducing number of clusters in the highway. Figure 3.9 presents the average distance traveled

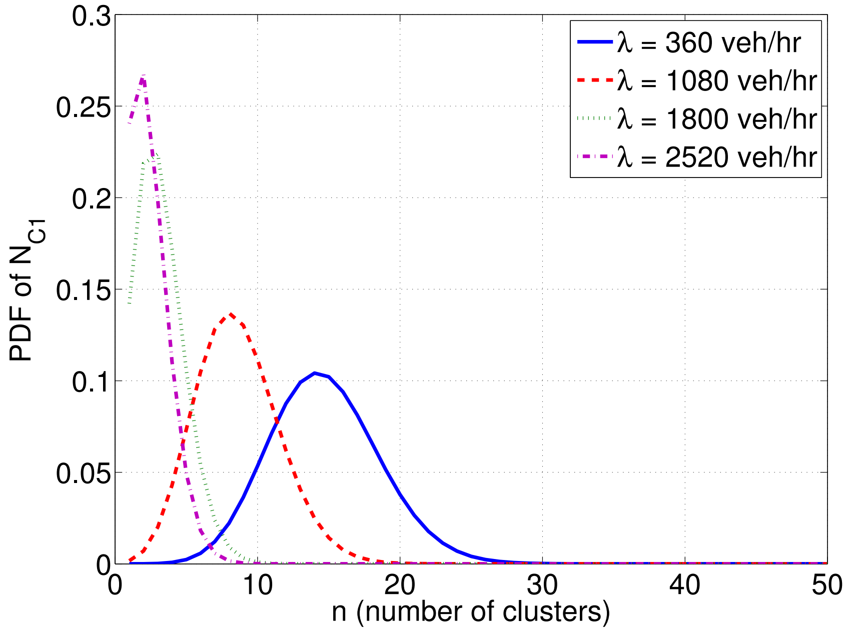


Figure 3.6. PDF of N_{C_1}

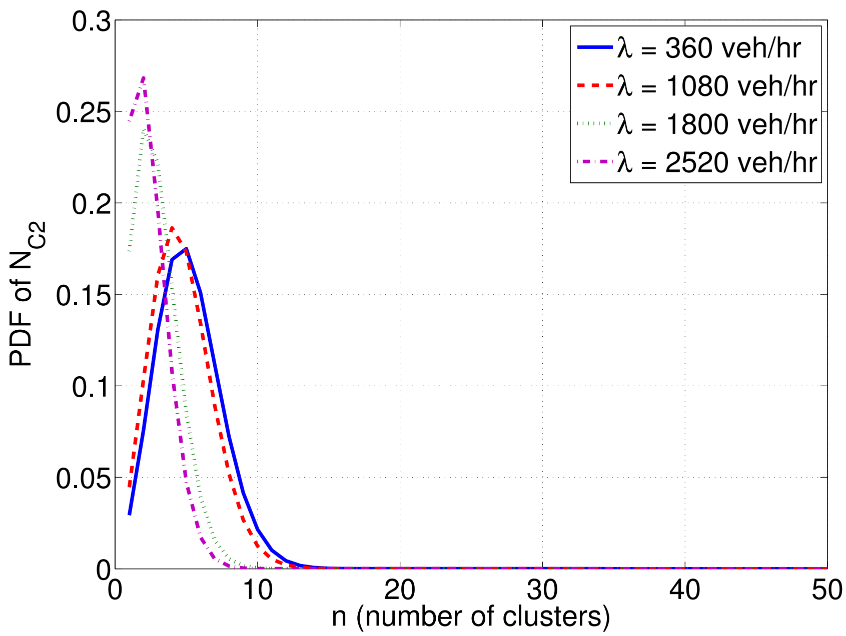


Figure 3.7. PDF of N_{C_2}

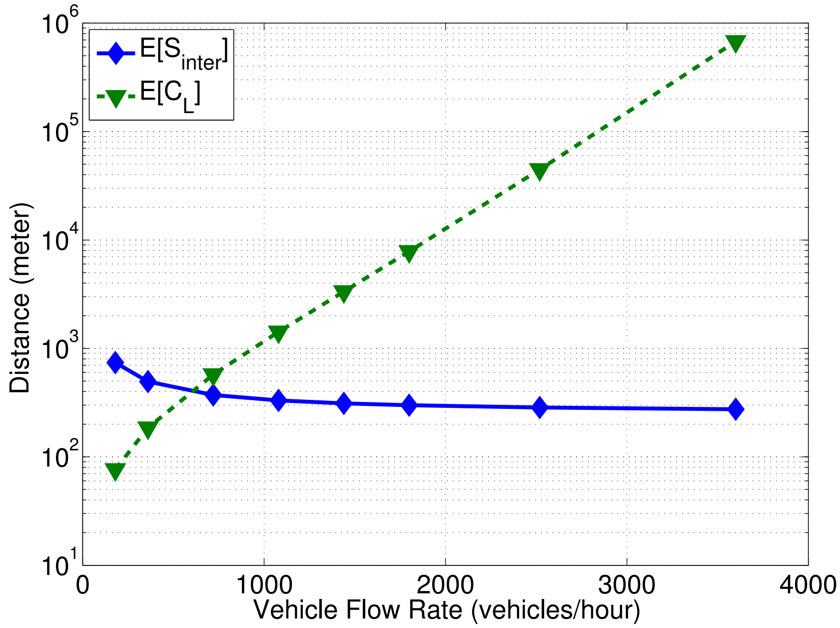


Figure 3.8. Expected values of C_L and S_{inter} for N_{C_1} distribution

by messages during forward phase, $E[X_f]$; and catch-up phase, $E[X_c]$. Figure 3.9 shows that the value of $E[X_f]$ also increases with the increment of traffic flow rate, confirming that as the traffic density increases, messages are mostly transmitted via wireless channel rather than being carried by vehicles.

Furthermore, the trend in the plots of $E[X_f]$ and $E[X_c]$ in Figure 3.9 derived from our analysis in Section 3.3.1.2 show an exact match with the numerical results of the original $E[X_f]$ and $E[X_c]$ from [40], which is shown in Figure 3.10. The same trend displayed by Figures 3.9 and 3.10 validated the accuracy of our work on the distribution of T_C from [40]. Figure 3.11 displayed the information propagation speed based on the T_{D_1} and T_{D_2} distributions, which further validates the accuracy of our analysis as Figure 3.11 shows similar trend with the information propagation speed from [40]. Therefore, we are able to arrive with the same conclusion as [40] which higher vehicle density leads to a larger partition size and shorter inter-cluster distance, and henceforth, reduce the catch-up time. Figure 3.11 shows a sharp increase as the vehicle flow rate increases and the propagation speed is shown much faster than the vehicle movement.

Figures 3.13 - 3.14 display the PDF of T_{D_1} and T_{D_2} distributions, respectively. Although both distributions exhibit similarities to an $Erlang - n$ distribution, the two distributions show different peaks for traffic rate lower than 2000 veh/hr. The variations of N_{C1} and N_{C2} at low traffic rate mentioned in preceding paragraph are also displayed in T_{D_1} and T_{D_2} distributions in Figures 3.13 and 3.14, respectively. At $\lambda = 360$ veh/hr, the T_{D_1} distribution in Figure 3.13 shows high end-to-end delay since N_{C1} distribution in Figure 3.4 displays high number of disconnected clusters of vehicles. In addition, Figures 3.6 and 3.7 show that N_{C1} and N_{C2} distributions have the similar peaks for traffic rate larger than 1500 veh/hr. This trend is also shown in T_{D_1} and T_{D_2} distribution at Figures 3.13 and 3.14, respectively. Based on empirical traffic data, it has been established that a vehicular network on a highway is fully connected when the traffic volume exceeds 1000 veh/hr [20]. Therefore, we ascertain at $\lambda \geq 1500$ veh/hr, a vehicular network on a highway has almost full connectivity and most of the clusters are merging into larger clusters. Subsequently, the packets are forwarded using via wireless channel most of the time, resulting a small variation on the end-to-end delay between the two distributions.

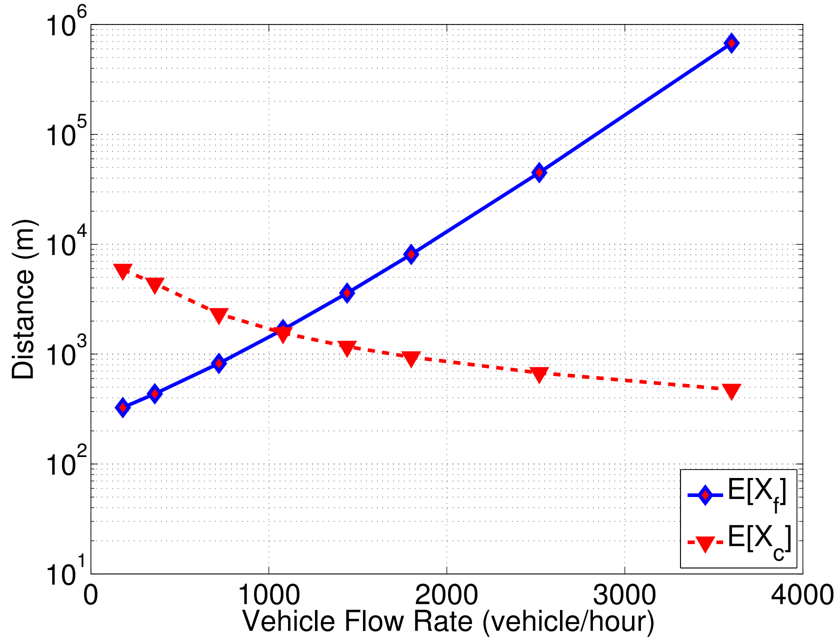


Figure 3.9. Expected values of X_f and X_c for N_{C_2} analysis

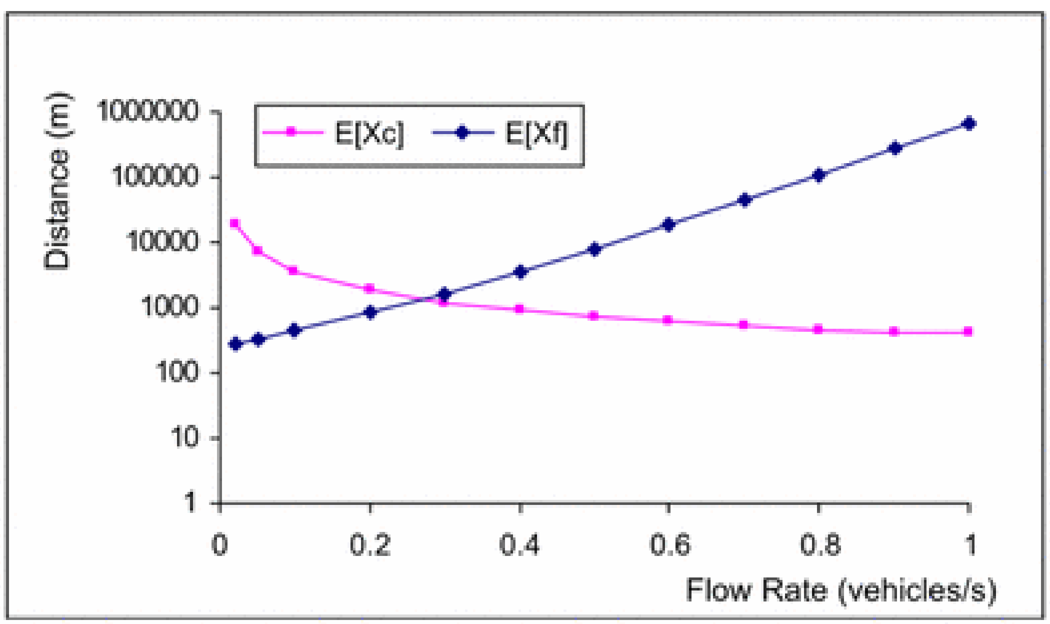


Figure 3.10. $E[X_f]$ and $E[X_c]$ numerical results from [40]

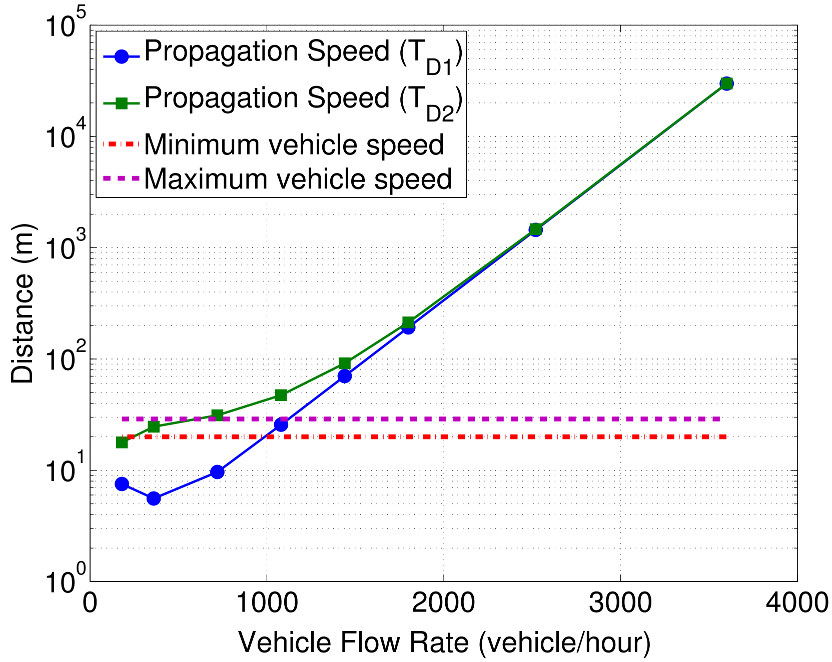


Figure 3.11. Information propagation speed from T_{D_1} and T_{D_2} analyses

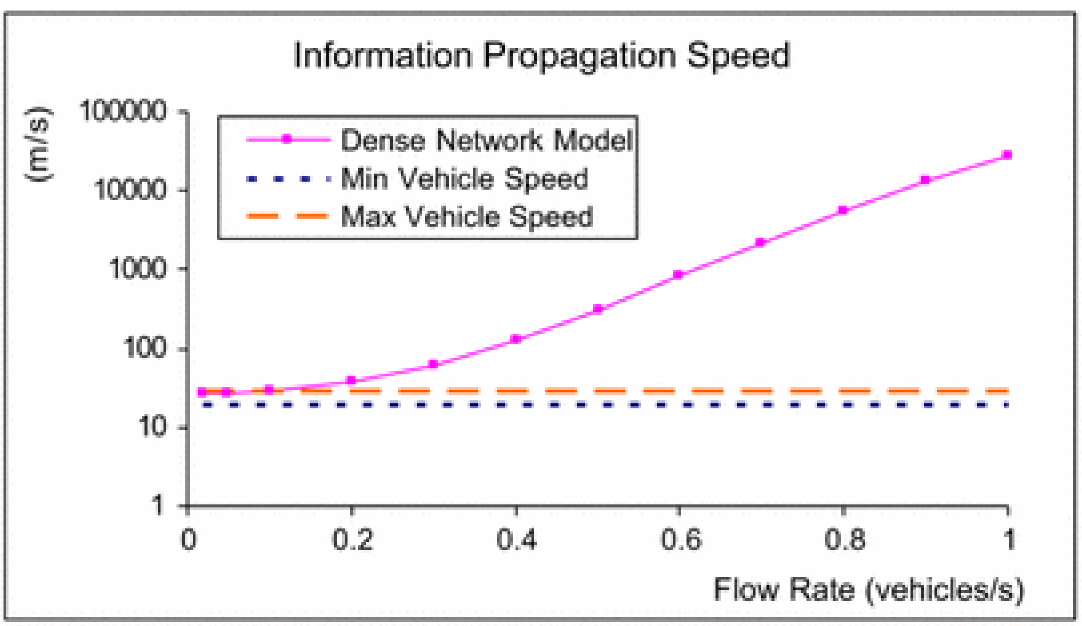


Figure 3.12. Information propagation speed from [40]

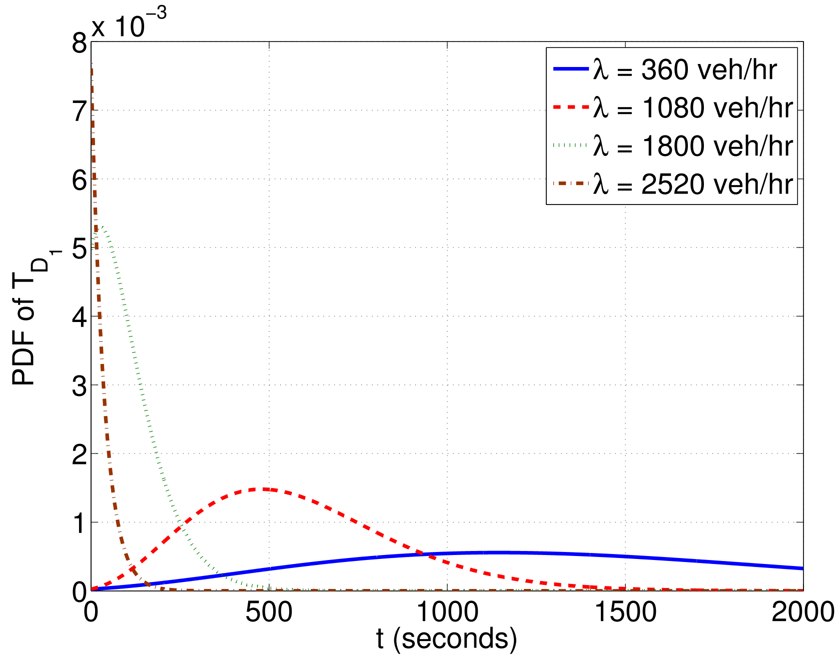


Figure 3.13. PDF of T_{D_1}

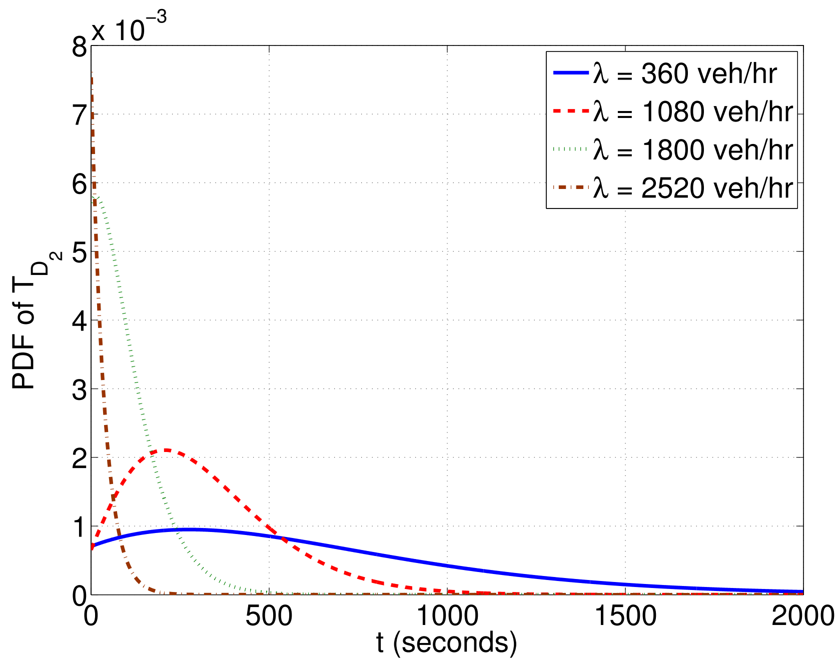


Figure 3.14. PDF of T_{D_2}

3.4.2. Simulation Results

In this section, we present simulation results in Figures 3.16a -3.16f for our proposed analytical model using the network simulator NS-2 [54, 55]. Our simulation scenario, which is displayed in Figure 3.15, is based on a one-directional highway with the length of 15 km. The highway is assumed to have multiple one-directional lanes, where vehicles can overtake each other without changing their lane or maneuvering. Vehicles are generated using a Poisson process with flow rates of 360, 1080, 1800 and 3600 veh/hour. Each vehicle is assigned a random speed based on a uniform distribution between the interval $v_{min} = 20$ m/s and $v_{max} = 28.89$ m/s and the assigned speed does not change over the simulation time. We perform the simulation for 1200 seconds and repeat the simulation for 1000 iterations. Since NS-2 is built to simulate a network environment, we configure media access control (MAC) and physical (PHY) layers in NS-2 to retain the assumptions of ideal MAC and PHY layers for the model so that the simulation is executed under ideal communication channel. The packets are generated using Poisson traffic with mean of 0.1 second and the transmission range is set to 250 meter. In addition, we configure the source vehicle to be the only *informed vehicle* at time t and located at located at position H; and the destination vehicle is the first vehicle to pass location H at time t.

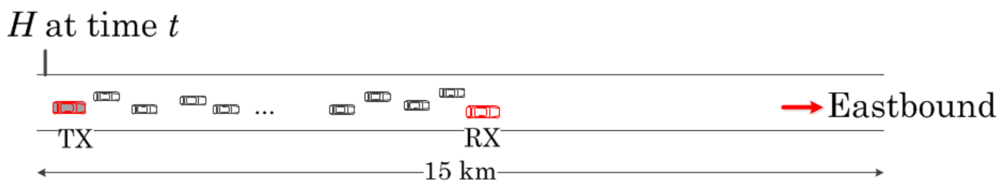
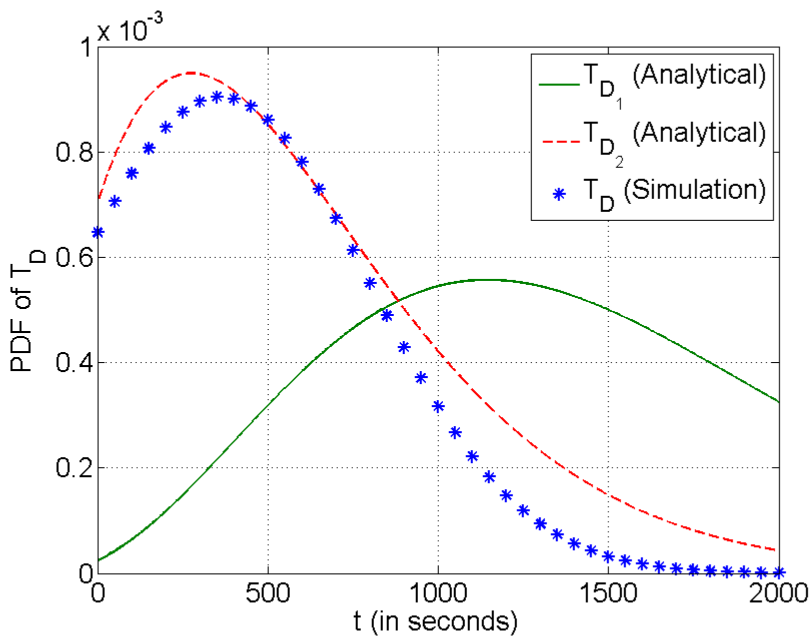


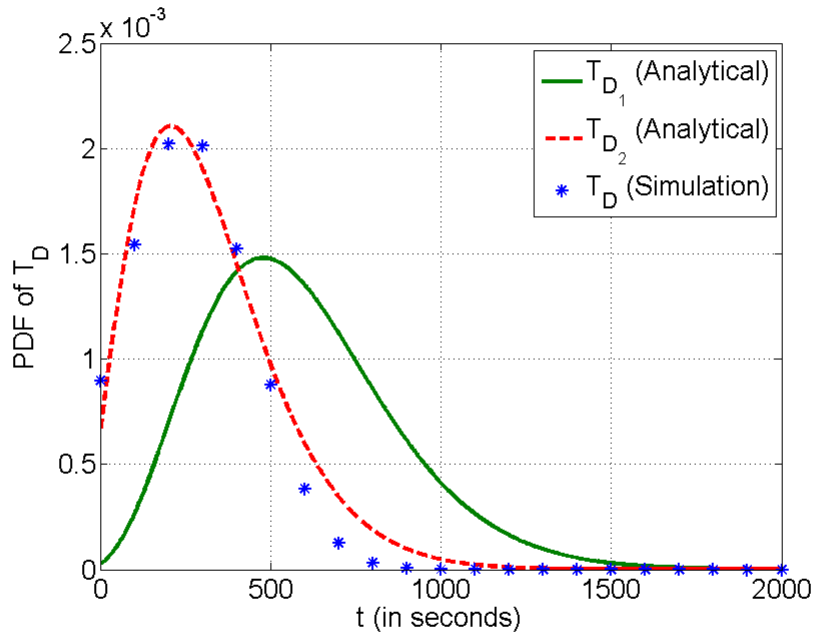
Figure 3.15. Simulation Scenario

Figures 3.16 display the comparison between the numerical results of the T_{D_1} and T_{D_2} analyses with the simulation results. It should be noted that the simulation results in Figures 3.16 are consistent with N_{c2} and T_{D_2} numerical results. Although the analytical results for both $E\left[S_{inter}\right]$ and $E\left[C_L\right]$ parameters in [20] are supported by simulation results, the simulation topology in [20] is based on a straight two-directional highway. On the other hand, the analyses on the average distance during a forwarding phase, $E\left[X_f\right]$; and a catch-up phase, $E\left[X_C\right]$ in [40] is based on a one-directional highway, although the analyses are not supported by simulation results. Henceforth, we argue that NS-2 simulation results are consistent with the T_{D_2} distribution because our simulation topology is based on a one-directional highway.

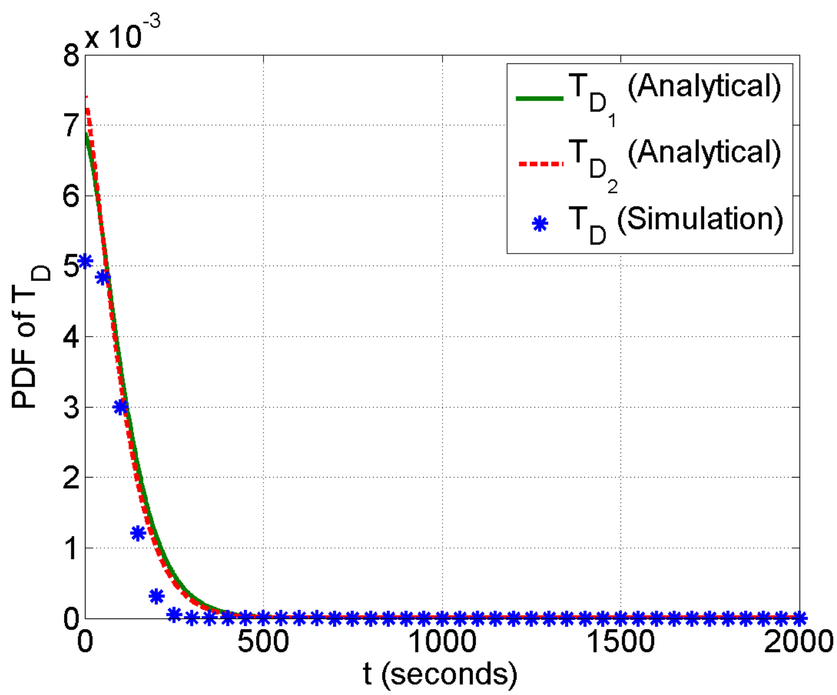
In addition, the plot for T_{D_2} distribution in Figure 3.16a, which is for $\lambda = 360$ veh/hr, has a small discrepancy with the simulation plot of T_D, although both plots show the same trend of Erlang-n distribution model. As the value of λ increases, the plots for T_{D_2} distributions displayed in Figures 3.16b-3.16d show an exact match with the simulation plots of T_D. From Figure 3.16, we ascertain that the small discrepancy between the plot of T_{D_2} distribution and the plot of T_D simulation is caused by the Poisson process properties which it assumes a large number of arrivals. We also conclude that even though NS-2 has been configured with the ideal communication channel, MAC and PHY conditions in NS-2 still affect the simulation results. Furthermore, we ascertain that the use of Poisson traffic for data packets generation in the simulation cause a small deviation between analytical and simulation results. The end-to-end delay result shown in Figure 3.16f also does not show a definite match with the analytical results, even though the simulation results shown consistency with the analytical results. Figures 3.16a and 3.16b clearly show that at $\lambda \leq 1500$ veh/hr, NS-2 simulation results are consistent with T_{D_2} analytical results. At $\lambda \geq 1500$ veh/hr, Figures 3.16c and 3.16d display consistency between the simulation and analytical; results for both T_{D_1} and T_{D_2}.



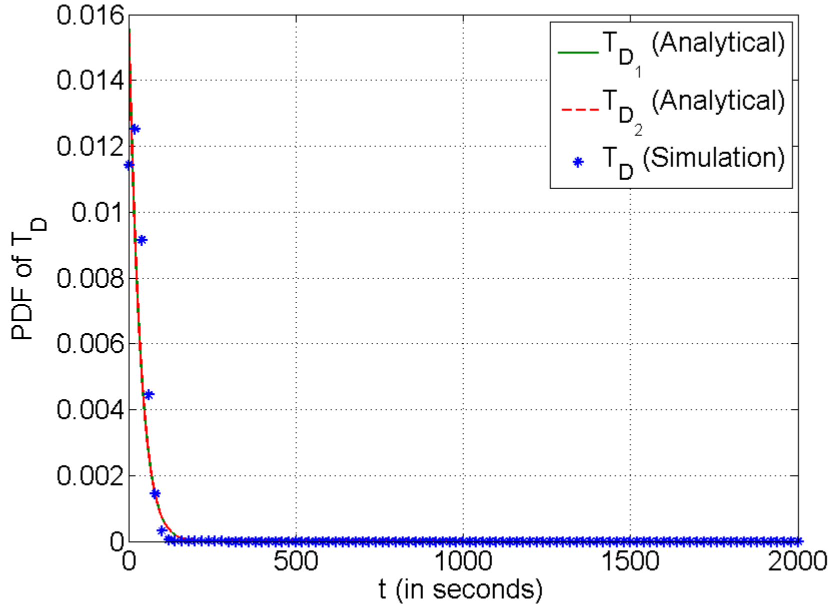
(a) $\lambda = 360$ veh/hr



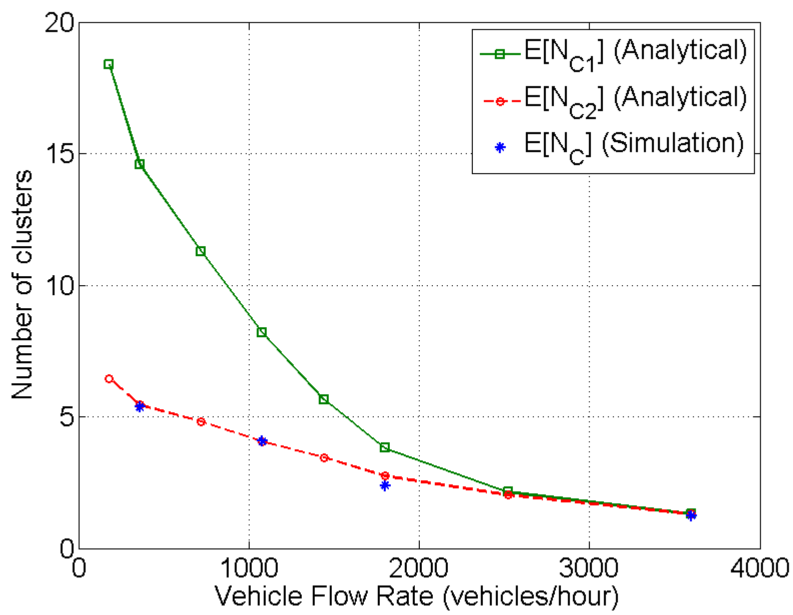
(b) $\lambda = 1080$ veh/hr



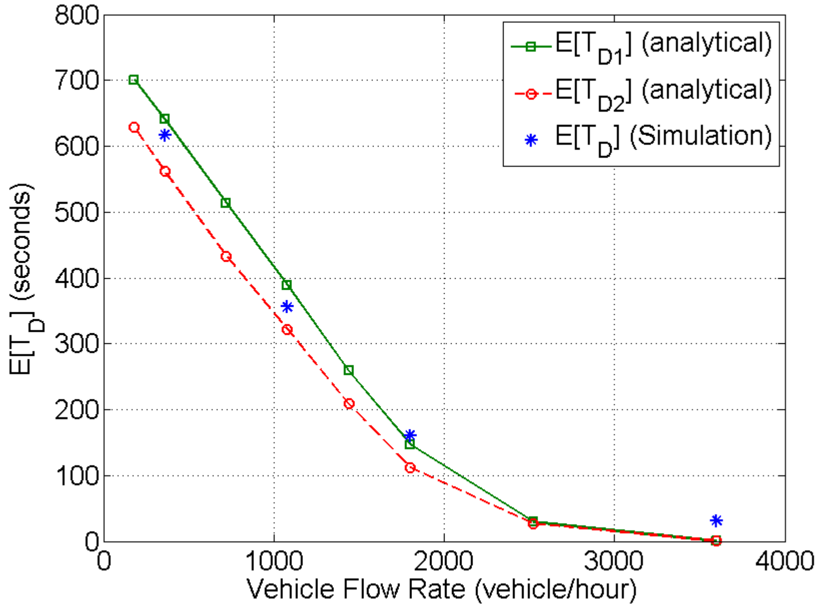
(c) $\lambda = 1800$ veh/hr



(d) $\lambda = 3600$ veh/hr



(e) Expected number of clusters



(f) Expected end-to-end delay

Figure 3.16. Comparison between analytical results of T_{D_1} and T_{D_2} and simulation results of T_D

3.5. Conclusion

In this section, we have proposed a probability distribution for the end-to-end delay model for a vehicular network on a unidirectional highway by extending the catch-up delay model between two adjacent vehicle clusters to multiple vehicle clusters as well as using traffic characteristics models to determine the distribution for number of clusters. We have approximated the distribution of the catch-up delay model between two disconnected clusters using the exponential regression analysis to derive the catch-up delay model for multiple clusters. Using the approximation, we have established that the total catch-up delay model for multiple clusters follow an $Erlang - n$ distribution. We also have validated our analytical results through simulation. From the results, we have established that our simulation results are consistent to the T_{D_2} analytical result since the parameters used in the N_{C2} distribution are based on a two-directional highway.

Chapter 4

Adaptive Beaconing System Based on Fuzzy Logic Approach

4.1. Introduction

As mentioned in Chapter 1, VANETs are exposed to frequent topology changes; intermittent communication due to high velocity on highways and the impact of traffic control systems in urban areas; and different type of network densities. For example, VANETs on highways or in urban areas are more likely to establish highly dense networks during rush hour traffic, whereas the same networks can experience frequent network partition because of sparsely populated highways or during late night hours.

Beaconing, or *single-hop broadcast*, is an essential feature of many of *vehicle-to-vehicle communication* systems. One of the main utilizations of beaconing is to collect neighbors information for packet forwarding [56]. *Beacon packets* are periodically and locally broadcasted to announce vehicles' current status in the network and the messages normally contain vehicles' identifier; their geographical position and velocity; and the time when the beacon is broadcasted into the network [19].

Broadcasting beacon messages is still considered as a communication challenge in data dissemination since all communicating vehicles must broadcast messages at a constant rate. The majority of VANET routing protocols is designed to include constant beaconing as part of their routing methods, such as the *Vehicle-Assisted Data Delivery* (VADD) protocol proposed in [16] and the *Distributed Vehicular Broadcast* (DV-CAST) protocol proposed in [34], where both protocols require vehicles in the network to transmit beacon messages every 0.5 sec and 1 sec, respectively.

VANET has a highly dynamic topology where the traffic density can rapidly change from sparse to heavy. In a situation where the number of vehicles is small and vehicle mobility is high, the dependency of accurate location information is intensified as vehicle locations change very quickly. In order to acquire accurate information in such networks, the most logical solution is to increase the frequency of emitting beacon messages. Unfortunately, this solution can lead to a high number of beacon messages being exchanged, which also leads to high channel occupancy and a high number of packet collisions. On the other hand, choosing not to increase beaconing frequency can cause the stale neighbor information problem to occur.

Based on the above scenarios, it is important to consider an adaptive beaconing approach where a vehicle can adjust its beacon rate based on the changes in traffic characteristics and vehicle mobility. In this chapter, we propose an *adaptive beaconing system* that adapts beacon frequency by using the fuzzy inference engine and utilizing three main parameters. These are *packet carried time*, *number of single-hop neighbor* and *current vehicle speed.* Using the *fuzzy logic* approach, the system will adjust beacon frequency by decreasing the rate in dense traffic condition to reduce the overloading of the channel and increasing the rate in spare network to increase neighbor information accuracy.

4.2. Overview of Fuzzy Logic

Fuzzy logic is based on the theory of fuzzy sets, where an object's membership of a set is gradual rather than just based on extreme values of conventional logic that uses precise data such as 0 or 1; true or false; or on or off [57, 58]. The representation of extreme values may not always be sufficient in describing some aspects of the real world.

Formally, *fuzzy logic* is a mathematical representation of vague real world information. Instead of using just two extreme data points as in binary logic, there can be multiple *in-between* data points [59]. Both conventional and fuzzy data can be modeled using mathematical sets. *Conventional* or *crisp sets* contain data that satisfy specific requirements for membership. The degree to which an element belongs to a set is called the *degree of membership* in the set. Figure 4.1 and Figure 4.2 illustrate

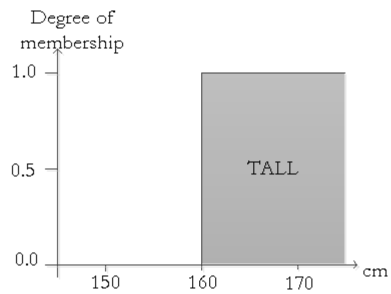


Figure 4.1. Crisp set for *tall*

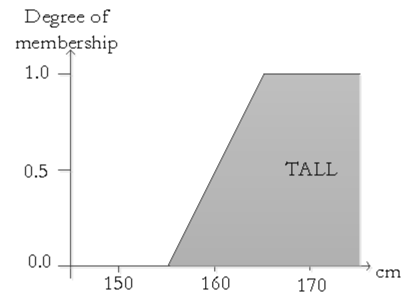


Figure 4.2. Fuzzy set for *tall*

the difference between the crisp set and fuzzy set for the tall definition .

Using the crisp definition of tall In Figure 4.1, a person with height of 157 cm would not be tall at all, whereas using the fuzzy definition in Figure 4.2, the same person would be considered tall with a degree of 0.375. Mathematically, the degree of membership for both crisp definition of tall can be mathematically described as:

$$\text{Degree of tallness} = \begin{cases} 0, & x < 160 \\ 1, & x \geq 160 \end{cases}$$

whereas the fuzzy definition can mathematically be described as:

$$\text{Degree of tallness} = \begin{cases} 0, & x < 150 \\ \dfrac{x - 150}{15}, & 165 \geq x \geq 150 \\ 1, & x > 160 \end{cases}$$

These functions that describe the degree of membership in a given set are known as *membership functions.*

4.2.1. Fuzzy Logic Membership Function

A *membership function* is used to describe a fuzzy set. A *membership function* for a fuzzy set A can be formally defined as $\mu_A : X \rightarrow [0, 1]$, where each element of X is mapped to a value between 0 and 1 [57]. This value, which is referred to as degree of membership, quantifies the grade of membership of the element in X to the fuzzy set A. There are different forms of membership functions that have been discussed in the literature such as triangular, trapezoidal, piecewise linear and Gaussian [60]. However, this section will present types of membership functions that are used in this research.

- Triangular Membership Function

  A triangular function is defined by a lower limit a, and upper limit b, and a value m, where $a < m < b$ (Refer to Figure 4.3).

$$\mu_A(x) = \begin{cases} 0, & x \leq a \\ \dfrac{x-a}{m-a}, & a < x \leq m \\ \dfrac{b-x}{b-m}. & m < x \leq b \\ 1, & x \geq b \end{cases}$$

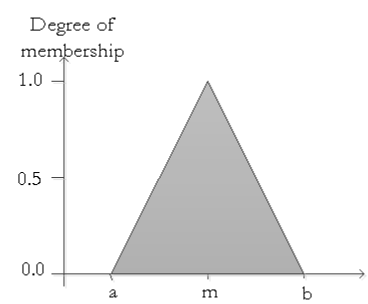


Figure 4.3. Triangular Membership Function

- Trapezoidal Membership Function

  This function is defined by a lower limit a, and upper limit d, a lower support limit b, and an upper support limit c, where $a < b < c < d$ (Refer to Figure

4.4). Figures 4.5 and 4.6 display two special cases of a trapezoidal function, which are called R-function and L-function.

$$\mu_A(x) = \begin{cases} 0, & (x < a) \text{ or } (x > d) \\ \dfrac{x-a}{b-a}, & a \le x \le m \\ 1, & b \le x \le c \\ \dfrac{d-x}{d-c}. & c \le x \le d \end{cases}$$

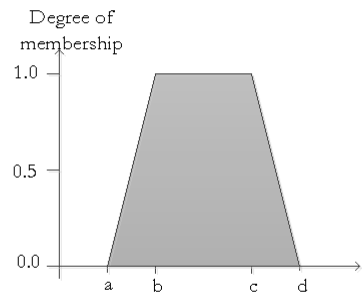


Figure 4.4. Trapezoid Membership Function

$$\mu_A(x) = \begin{cases} 0, & x > d \\ \dfrac{d-x}{d-c}, & c \le x \le d \\ 1, & x < c \end{cases}$$

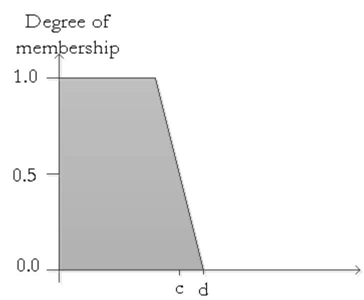


Figure 4.5. R-Function of a Trapezoidal Function

$$\mu_A(x) = \begin{cases} 0, & x < a \\ \dfrac{x-a}{b-a}, & a \le x \le b \\ 1, & x > b \end{cases}$$

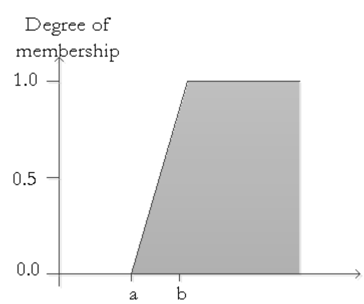


Figure 4.6. L-Function of a Trapezoidal Function

4.2.2. Fuzzy Logic System (FLS)

According to Mendel in [59], a *fuzzy logic system* (FLS) can be defined as a nonlinear mapping of an input data vector into a scalar output. The strength of a fuzzy logic system is that there are huge numbers of possibilities that lead to lots of different mappings [59]. On the other hand, this strength does require a cautious understanding of fuzzy logic as well as its elements that comprise a fuzzy logic system. In general, a fuzzy logic system comprises of four components [59]:

1. *Fuzzification module* –in this module, the system inputs, which are crisp values, are transformed into fuzzy sets by using fuzzy linguistic variables and term, and membership functions.

2. *Knowledge base* –this module stores IF-THEN rules provided by experts.

3. *Inference engine* –this module implements the fuzzy inference process on the inputs based on IF-THEN rules from the knowledge base.

4. *Defuzzification module* –this module transforms the fuzzy set obtained by the inference engine into a crisp value.

The process of the fuzzy logic system in the proposed adaptive beaconing system is explained in Algorithm 4.1

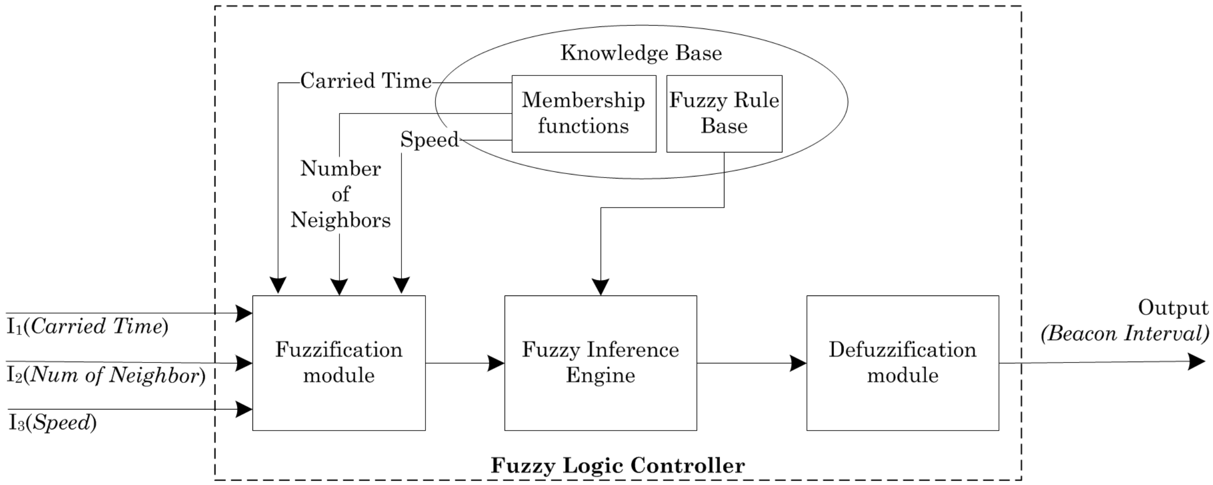


Figure 4.7. Fuzzy logic system components

Algorithm 4.1 Fuzzy logic system

```
1: function INITIALIZATION
2:     Define the system inputs: Carried Time, Number of Neighbor, Speed
3:     Construct the membership functions
4:     Construct the rule base
5: end function
6: function FUZZIFICATION
7:     Convert crisp input data to fuzzy values using membership functions
8: end function
9: function INFERENCE
10:    Evaluate the rules in the rule base
11:    Combine the results of each rule
12: end function
13: function DEFUZZIFICATION
14:    Convert the beacon interval fuzzy set to a single beacon interval
15: end function
```

During FLS initialization, which is shown in Line 1-5 of Algorithm 4.1, the proposed system defines three parameters as its system inputs. The three parameters are

Carried Time, *Number of Neighbour* and *Speed*. The main reasons for the selection of the three parameters as the FLS inputs can be summarized as follows. First, Wisitpongphan et al. conduct an empirical study in [20] to characterize key parameters for intermittently connected or disconnected VANET. The study shows the characteristic of the inter-vehicle spacing during rush hour and off-peak hour. From the empirical study, we ascertain that the number of neighbouring vehicles within the transmission range of 250 meters can be used to indicate whether the vehicle traffic in VANET is heavy or sparse. The parameter *Speed* is used as another indicator for the network condition because the velocity of vehicles and the vehicular traffic density are implicitly interrelated with each other. The relationship are clearly shown in [61, 62] using the traffic flow theory, where the average vehicles velocity decreases when the vehicle traffic density increases. The input *Carried Time* is the time duration when packets undergo the Recovery Mode via carry and forward technique. Since this method is used only when the network is partitioned or disconnected in [16, 17, 34], we establish that the parameter *Carried Time* can also be used to indicate whether the network is dense or sparse.

4.3. Adaptive Beaconing System based on Fuzzy Logic System for Vehicular Network

4.3.1. Key Routing Parameters and Design

The proposed *Adaptive Beaconing System* is built upon a carry and forward based routing protocol. In this section, we present the principle operations of the *carry and forward* based routing protocol. Figure 4.8 illustrates the operations of the routing protocol.

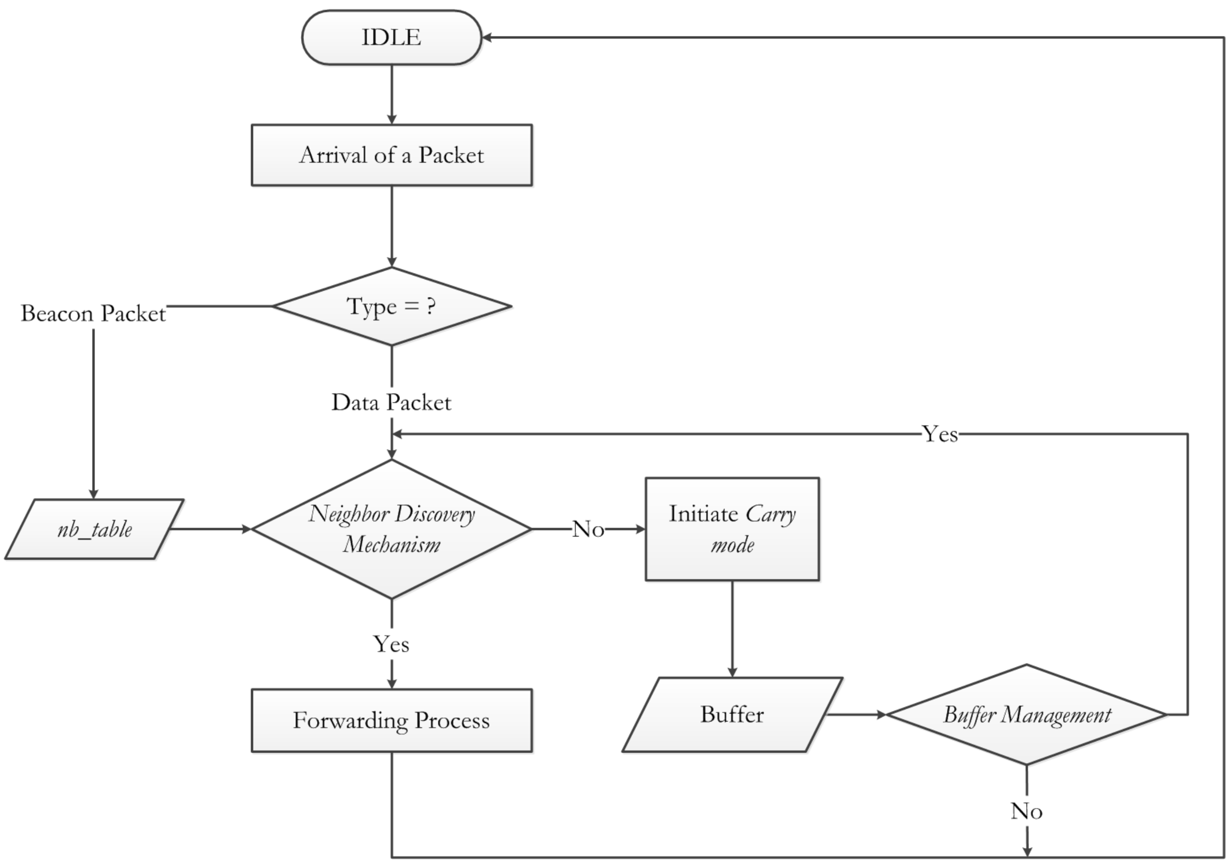


Figure 4.8. Flowchart of the carry and forward based routing protocol

In the current protocol design, each vehicle maintains a local neighborhood state information, called *nb_table*. The *nb_table* stores local neighboring vehicles information such as the vehicle id, the time of the beacon transmission, the current position, the speed, and the velocity of the vehicle at the time when the beacon is transmitted. Figure 4.9 shows the beacon packet format which contains the information necessary to run the carry and forward based routing protocol.

1. **Source ID** is a unique vehicle identification for the vehicle that broadcasts the beacon packets. The size of this field is currently set to 2 bytes.

2. **Source Location** is the geographical position of the vehicle that broadcasts the beacon packets. The size of this field is 8 bytes.

3. **Speed** is the current speed of the vehicle that broadcasts the beacon packets. The size of this field is set to 4 bytes.

4. **Velocity** is the current direction of the vehicle that broadcasts the beacon packets. In an urban model, the direction is based on the cardinal directions, which are *North*, *South*, *East*, *West*. The size of this field is set to 8 bytes.

5. **TimeStamp** is the time when the beacon is broadcasted by the source vehicle. This size of this field is 4 bytes.

Source ID	Source Location	Speed	Velocity	TimeStamp

Figure 4.9. Beacon Packet Format

Moreover, each entry is maintained based on their age using a timer, which also known as the neighbor age threshold. The neighbor information is considered as latest if the NAT is below the threshold, whereas any member of the list older than the neighbor age threshold is discarded.

As illustrated in Figure 4.8, the decision in forwarding process is made through greedy forwarding in the *Neighbor Discovery Mechanism*, where an intermediate node that is closest to the destination nodes become the next relay in the packet forwarding process based on the information stored in the *nb_table*. When a current forwarder enters a partitioned network where there is no suitable neighbor to forward packets towards the destination node, the current forwarder will use the *Carry Mode* as its recovery mechanism until it moves into other vehicles' transmission range.

4.3.2. The Design of Fuzzy Logic System (FLS) for Adaptive Beaconing System

The process of the proposed *Adaptive Beaconing System* is illustrated in Figure 4.10. In the proposed adaptive beaconing fuzzy logic system, each vehicle has the same initial broadcast interval. The next beacon interval is then determined by feeding the three inputs which are *Carried Time*, *Number of Neighbor*, and *Speed* into the fuzzy inference engine. The *fuzzy inference engine* is a group of rules that are developed using expert knowledge. The *knowledge based rules* that connect the inputs and the output is based on a careful understanding of the philosophy behind vehicular network behavior. The fuzzy inference engine for the proposed adaptive beaconing

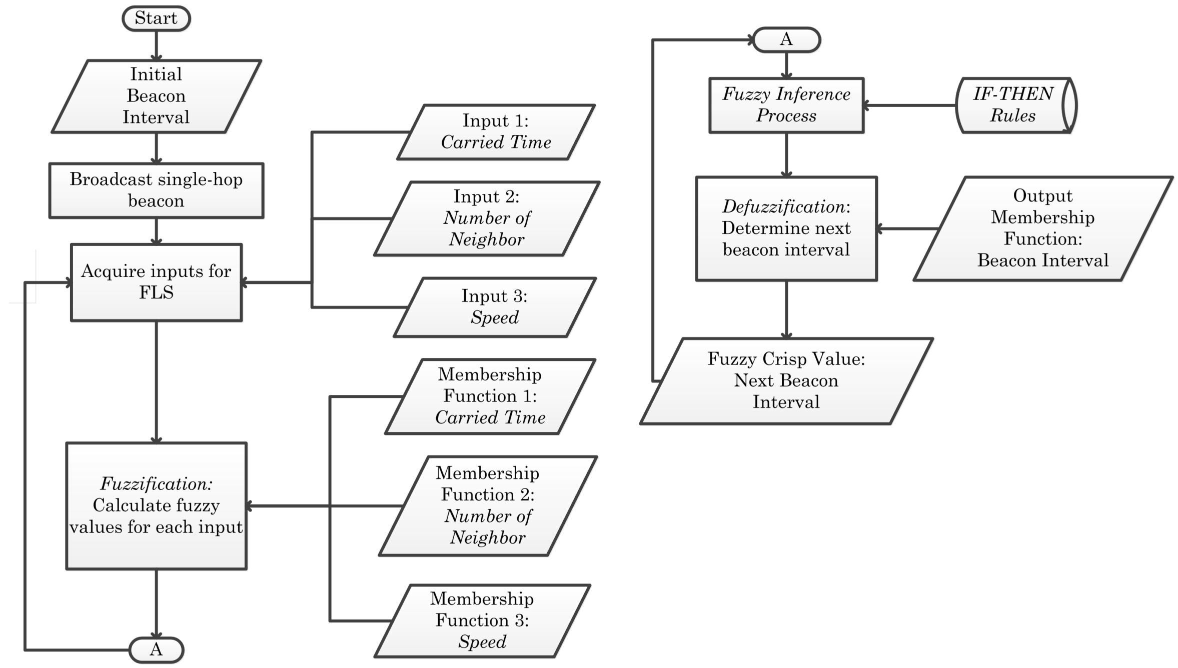


Figure 4.10. Fuzzy logic components in the Adaptive Beaconing System

system is based on Mamdani's fuzzy inference method [58,59] and is constructed using IF-THEN rules based on three main rules that are presented in Section 4.3.4

4.3.3. Fuzzification of Inputs and Output

In the proposed system, each vehicle determines its beacon interval based on three inputs:

1. ***Carried Time***: Measures how long packets have to be carried by a current forwarder/vehicle before being forwarded via wireless transmission

2. ***Number of neighbors***: The number of neighbors that are within the vehicle's transmission range.

3. The vehicle's current speed.

The above three inputs will be used in the *fuzzy inference engine* to determine the fuzzy output, which is the *beacon interval*. Therefore, the *membership functions* need to be developed for the three input parameters as well as the output parameter. In fuzzy logic, an object's membership is extended to accommodate various degrees of membership on the continuous interval of 0 and 1 where the endpoints normally correspond to no membership and full membership, respectively [57, 58]. The selection of the *membership functions* for the three inputs and output are derived based on previous simulation results of the fixed beaconing system. To design the fuzzy logic system, we use MATLAB Fuzzy Logic Toolbox. Figures 4.11–4.14 present the membership functions of the three inputs which are used in the proposed FLS system.

As shown in Figure 4.11, membership functions *Low*, *MediumLow, Medium*,

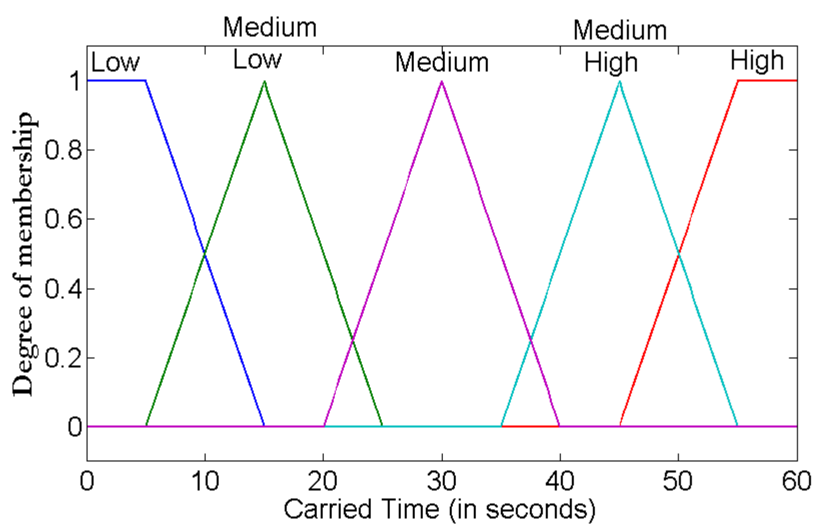


Figure 4.11. Membership Functions for Adaptive Beaconing FLS Input *Carried Time*

MediumHigh, and *High* are used to represent input *Carried Time*. Packets can be carried by a vehicle with a minimum time of zero second, in which packets are immediately forwarded, to a maximum of 60 seconds. However, in a sparse network, it is possible that packets have to stay in the buffer longer than 60 seconds. In such cases, membership function *High* is used to represent such time delay.

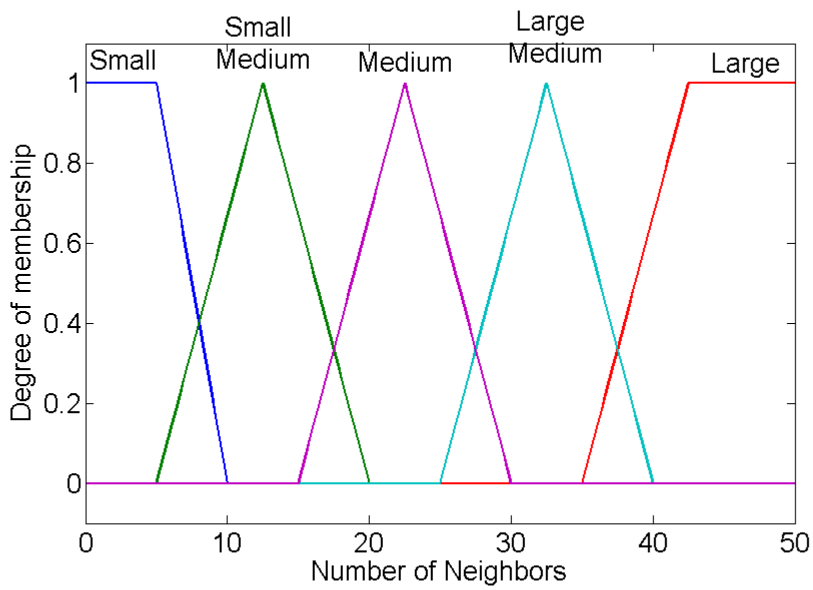


Figure 4.12. Membership Functions for Adaptive Beaconing FLS Input *Number of Neighbors*

The input *Number of Neighbors* which is represented by membership functions; *Small*, *SmallMedium*, *Medium*, *LargeMedium*, and *Large* are presented in Figure 4.12. The range of values that are used in the membership functions represents number of vehicles in a single-hop transmission in both sparse and dense networks. The range of values that are used in the membership functions represents the number of vehicles in a single-hop transmission in both sparse and dense networks. Nevertheless, in this input, only neighbors that are moving in the same direction and in front of the current vehicle are counted. Figure 4.13 illustrates how neighbors are chosen. In Figure 4.13, vehicles V_1 until V_{10} are single-hop neighbors to the current forwarder, V_C. However, for our FLS input, only V_1, V_2, and V_3 are counted as neighbors to V_C since they are moving in the same direction as V_C and located in front of V_C. Similar to input *Carried Time*, if the number of neighbors is larger than 50 vehicles, the membership function *Large* is used to represent such a value.

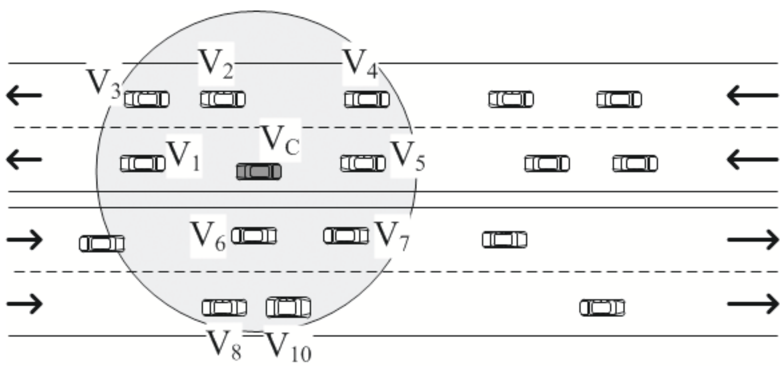


Figure 4.13. Selection of neighbors for input *Number of neighbors*

As displayed in Figure 4.14, membership functions *Slow*, *Medium*, and *Fast* are used to represent input *Speed*. The unit used in the parameter is meter/sec. The range of speed in the membership functions are based on the normal speed that is set in an urban model.

Figure 4.15 presents the membership functions for output *Beacon Interval*. The output *Beacon Interval* is configured to range between 0.05 to 5 seconds. The higher

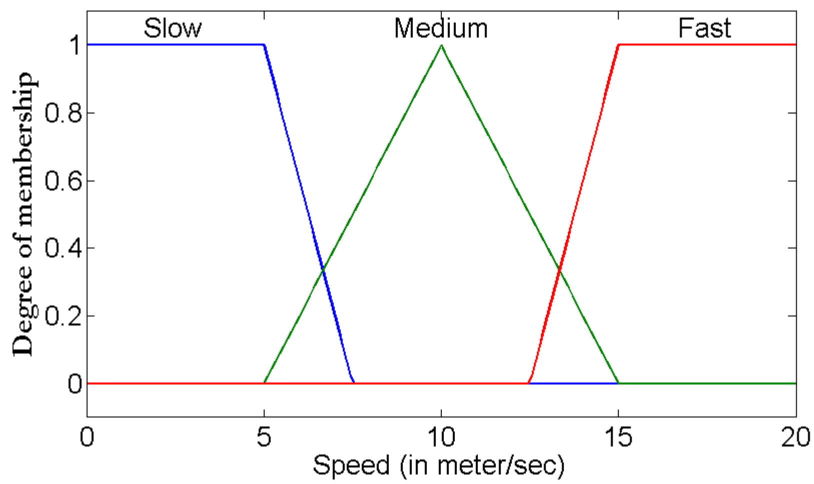


Figure 4.14. Membership Functions for Adaptive Beaconing FLS Input *Speed*

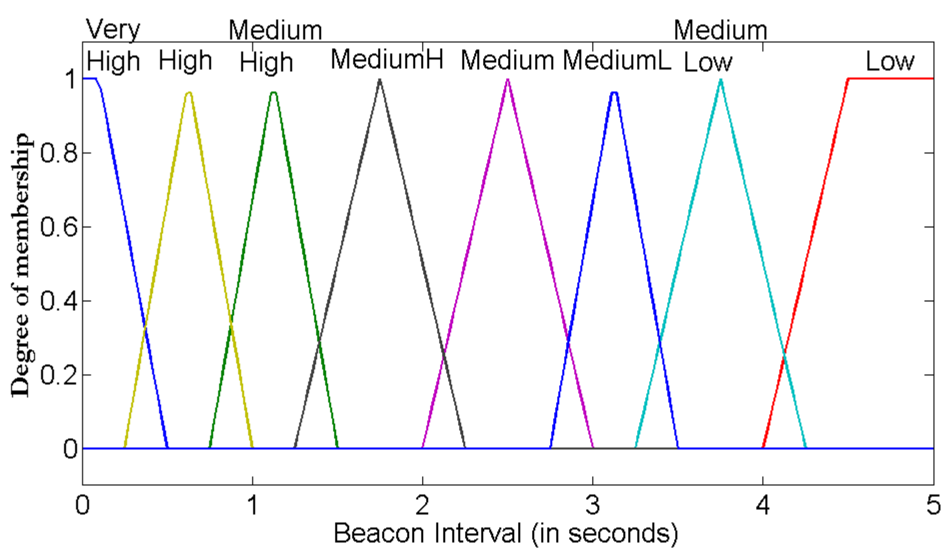


Figure 4.15. Membership Functions for Adaptive Beaconing FLS Output *Beacon Interval*

the value, the lower is the frequency for beacon generation. The range of values between 0.05 to 1 second is often used in a sparse network. In a sparse network, vehicles move with high mobility, and therefore, they usually move out of transmission range fairly quickly. *High* interval rate would ensure that a vehicle can forward packets to its neighbor that is still within the transmission range. Interval values in membership functions *MediumLow* and *Low* are normally used in a dense network, since vehicles have low mobility in this network and neighboring vehicles do not change rapidly.

4.3.4. Fuzzy Inference Engine

The *fuzzy inference engine* (Line 9-12 of Algorithm 4.1) for the proposed adaptive beaconing system is based on Mamdani's fuzzy inference method [57, 58] and is constructed using IF-THEN rules from the *knowledge base*. The group of IF-THEN rules for the proposed FLS is based on main rules that determine the condition of a vehicular network, which are shown in Table 4.1. The knowledge base is normally determined during the FLS initialization (Line 4 of Algorithm 4.1).

Referring to Table 4.1, Main Rule 1 is used to describe a sparse network, where packets are being carried most of the time; the number of neighbors is small and there is a possibility that a vehicle has no neighbor within the transmission range; and the

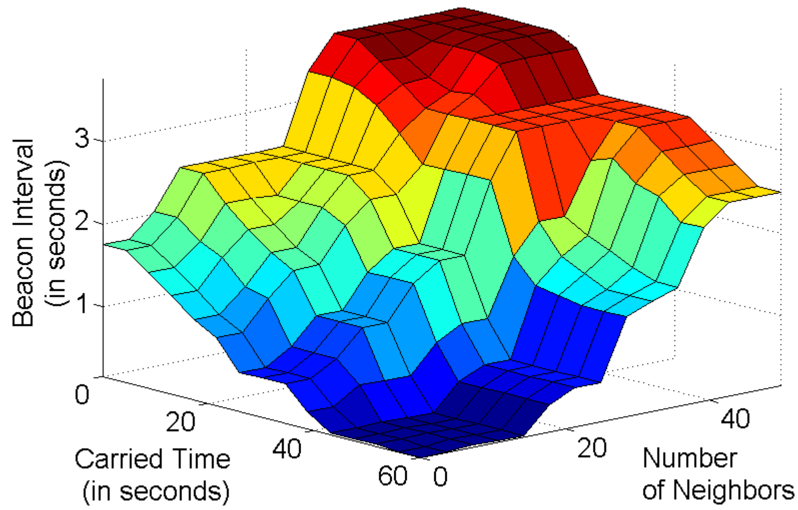
Table 4.1: The main rules for the fuzzy inference engine

Main Rule	IF			THEN
	Carried Time	Number of Neighbors	Speed	Interval Frequency
1	High	Small	Fast	High
2	Low	Large	Slow	Low
3	High	Large	Slow	Low

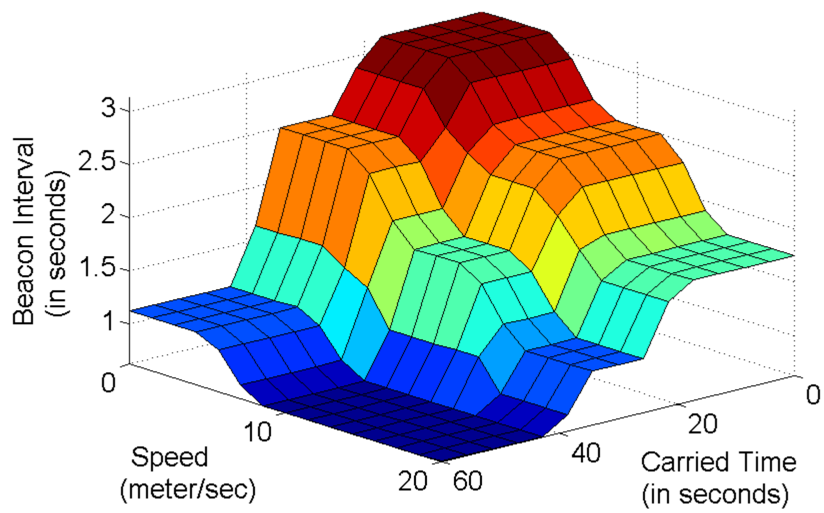
vehicle would move with high speed. Main Rules 2 and 3 are used to portray a dense network where the number of neighbors is high and vehicles move at slow speed. Carried time for packets can be high or low. Low carried time means packets are immediately forwarded to the next hop or destination node via the wireless channel. However, carried time in a dense network can be high if the collision rate is high and the wireless channel is congested. This situation may happen if most of the vehicles in the network are trying to forward packets via the wireless channel and broadcast beacon messages at the same time.

Using the main rules in Table 4.1 and the input membership functions shown in Figures 4.11, 4.12 and 4.14, we construct 75 IF-THEN rules for the knowledge base of the proposed fuzzy logic system.

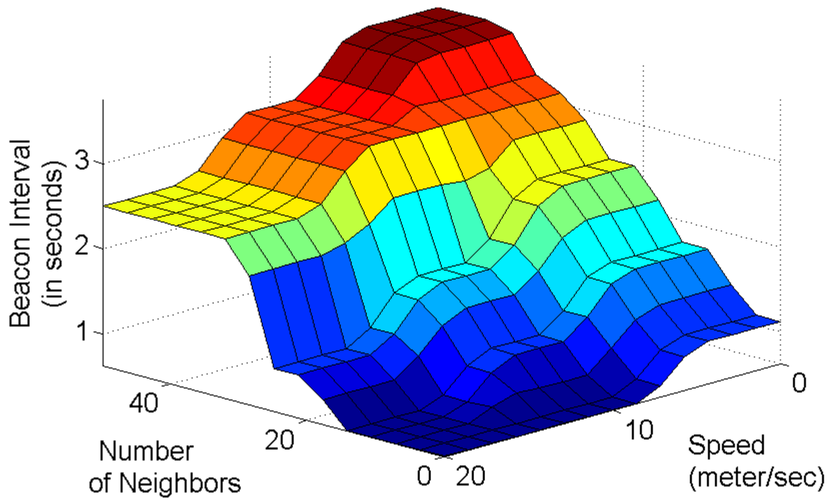
Figures 4.16a –4.16c show the correlation behavior between input and output variables. The trend shown in Figures 4.16a, 4.16b, and 4.16c match the three main rules that are used to build our FLS knowledge base. The three figures demonstrate that the beacon frequency increases as the value of *Carried Time* increases, the value of *Number of Neighbor* decreases and the value of *Speed* increases. In addition, Figures 4.16a, 4.16b, and 4.16c show that for both high and low *Carried Time* values, the beacon frequency decreases as the value of *Number of Neighbor* increases and the value of *Speed* decreases.



(a) Correlation between input *Carried Time*, *Number of Neighbor* and output *Interval Rate*



(b) Correlation between input *Carried Time*, *Speed* and output *Interval Rate*



(c) Correlation between input *Number of Neighbor*, *Speed* and output *Interval Rate*

Figure 4.16. Correlation between FLS inputs and output

4.3.5. Defuzzification

As illustrated in Figure 4.10, once the FLS receives the fuzzy values from the fuzzy inference engine, the values are then used in the defuzzification process to determine the next beacon interval, as shown in Line 13-15 of Algorithm 4.1. *Defuzzification* is a process of generating a crisp result based on the output membership function, which is defined in Figure 4.15, and corresponding membership degrees. In the proposed system, the *center of gravity or centroid* (COG) [57–59] is used for the defuzzification process, The centroid method is equivalent to finding the center of mass of the output composition [58, 59].

As shown in Figure 4.17, the output of the defuzzification can be seen as logical union of two or more fuzzy membership functions defined on the output variable [58, 59]. In Figure 4.17, if the degree for Interval *VeryHigh* is 0.4, the degree for Interval *High* is 0.2, and the degree for Interval *MediumHigh* is 0.1, the resulting result function forms a shape as shown in the figure. The x-coordinate, which is the defuzzified value, of the centroid of the shape is then calculated. With this method, every piece contributing to the output composition is accounted for in the final defuzzified value.

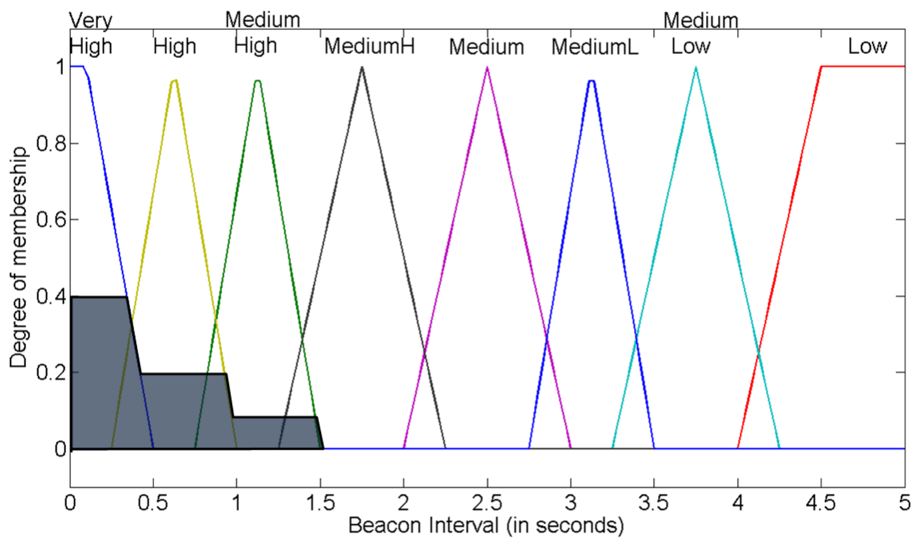


Figure 4.17. Defuzzification using center of gravity

4.4. Simulation Framework and Evaluation for Manhattan Grid Model

4.4.1. Simulation Assumptions

All vehicles are assumed to be equipped with wireless transceivers that allow the vehicles to act as mobile network nodes, and a Global Positioning System (GPS) device which provides the current vehicles' location within the network. It is also assumed that each vehicle has its own on-board unit, which has high capacity for storing packets during the forwarding process.

4.4.2. Mobility and Network Model

The VANET simulation as shown in Figure 4.18 is implemented in a 4000 x 3200 meters grid model of the city environment. The model is based on the *Manhattan Grid mobility mode*l, also known as the *City Section Mobility Model.* The vehicles' mobility is generated based on the VanetMobiSim *Intelligent Driver Model with Lane Changing* (IDM_LC) where the model adjusts the vehicles' speed based on the neighboring vehicles' movements [63]. Vehicles' movements in this model also support smart intersection management, where vehicles slow down and stop at intersections, or they act according to traffic lights. In addition, the model allows the vehicles to change lane and overtake other vehicles in the presence of multi-lane roads.

The network model is simulated using NS-2 [54,55] using the mobility trace that is generated by the VanetMobiSim [63] engine. The simulation scenarios are configured

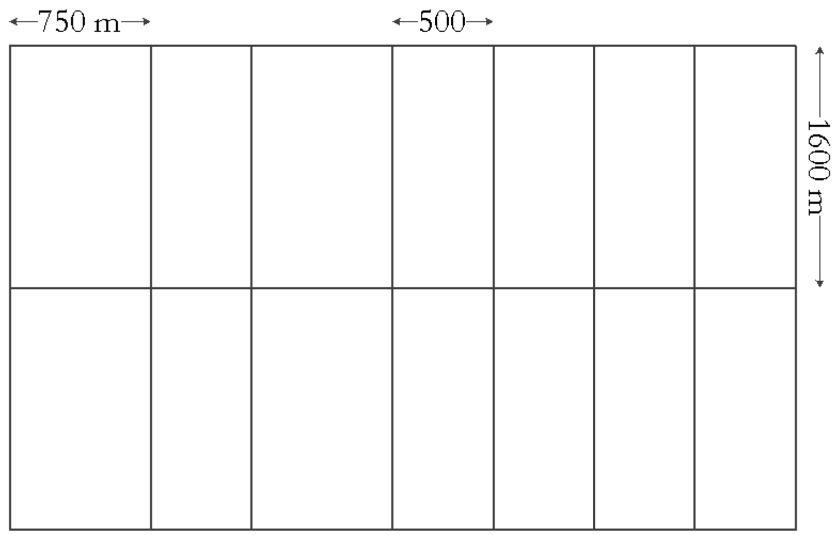


Figure 4.18. Manhattan Grid Model used in the simulation

using common parameters that are shown in Table 4.2. Since NS-2 cannot interact directly with Fuzzy Logic Toolbox in MATLAB, we implement the fuzzy logic system for the *Adaptive Beaconing System* in NS-2 using C++ based on the membership functions and the rules that we have created in MATLAB.

4.4.3. Performance Metrics

For each scenario, we execute multiple simulation runs that are no less than 60 iterations to attain a 95% confidence interval. The performance assessment is based on four metrics:

1. ***Packet Delivery Ratio***: Measures the fraction of data packets that are successfully received by destination to those generated by traffic source.

2. ***Average End-to-End Delay***: Measures the average difference between the

Table 4.2: Network Model Configuration

Area	4 km by 3.2 km
Number of vehicles	40, 80, 120, 160, 200
Speed (meter/sec)	Between 5 and 20 m/s
Simulation time	1000 seconds
Number of connections	15
Traffic pattern	CBR Traffic
CBR Rate	0.5 packets/sec
Packet size	256 bytes
Propagation Model	Two Ray Ground Model
MAC Layer	IEEE 802.11b
Transmission range	250 meters
Forwarding and recovery strategy	Greedy Forwarding with Carry and Forward strategy

time a data packet is originated by an application and the time the same packet arrives at its destination

3. ***Routing Overhead Ratio***: Measures the fraction of total beacon packets emitted to total number packets transmitted in the network

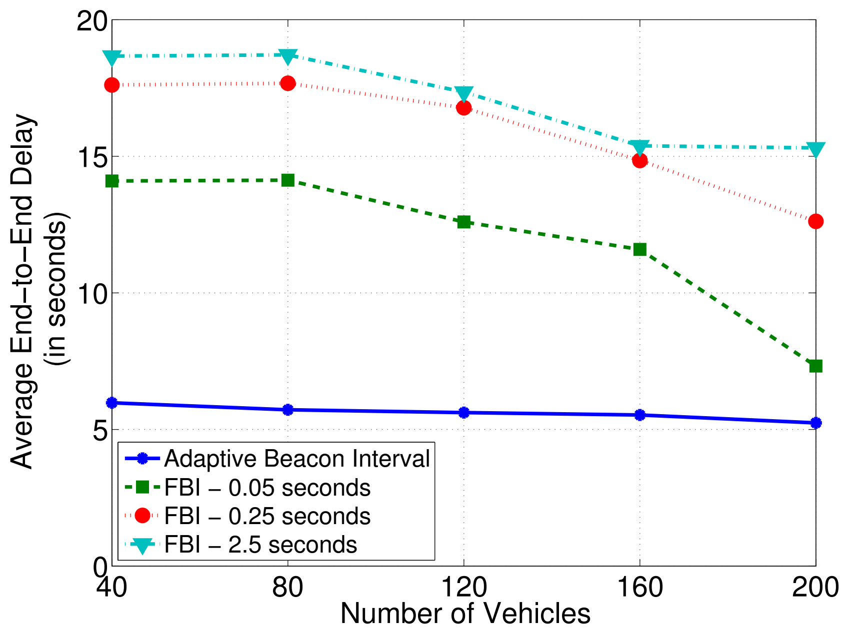
4. ***Total Collision Ratio***: Measure the ratio of total number of collisions to total number of packets transmitted in the network

4.4.4. Simulation Results and Analysis

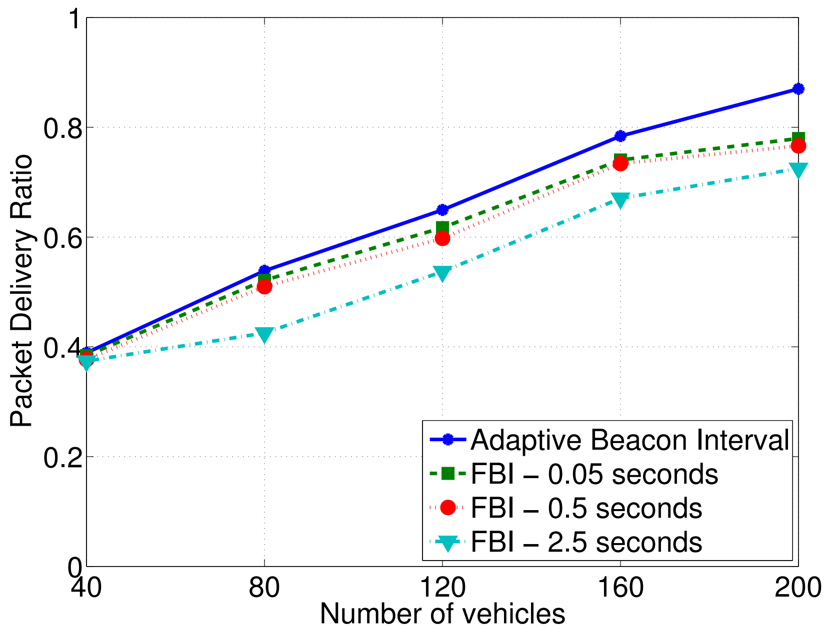
As shown in Figure 4.18, the performance assessment is made on a 4 km by 3.2 km Manhattan grid model with five different vehicle densities. For the first scenario, 40 vehicles are configured to move in a random behavior on the grid with maximum speed of 20 meter/sec. The number of vehicles is increased in the subsequent scenarios up to 200 vehicles. In addition, each scenario is configured with 15 random sources transmitting data packets in every two seconds to 15 random receivers. The performance comparison is made with *fixed beacon interval* (FBI) rate of 0.05, 0.5 and 2.5 seconds.

Plots displayed in Figures 4.19 confirm our argument that increasing the frequency of beacon emission is not the best solution in getting accurate information on neighboring vehicles as it can raise other routing issues. In Figure 4.19b, both adaptive and fixed intervals are able to achieve approximately the same delivery rate for a sparse network. However, Figure 4.19a shows that fixed intervals incur higher average end-to-end delay, compare to the adaptive beacon interval that able to maintain a low average end-to-end delay. It is confirmed that vehicles are unable to forward data packets because the channel is frequently being used to broadcast beacon messages.

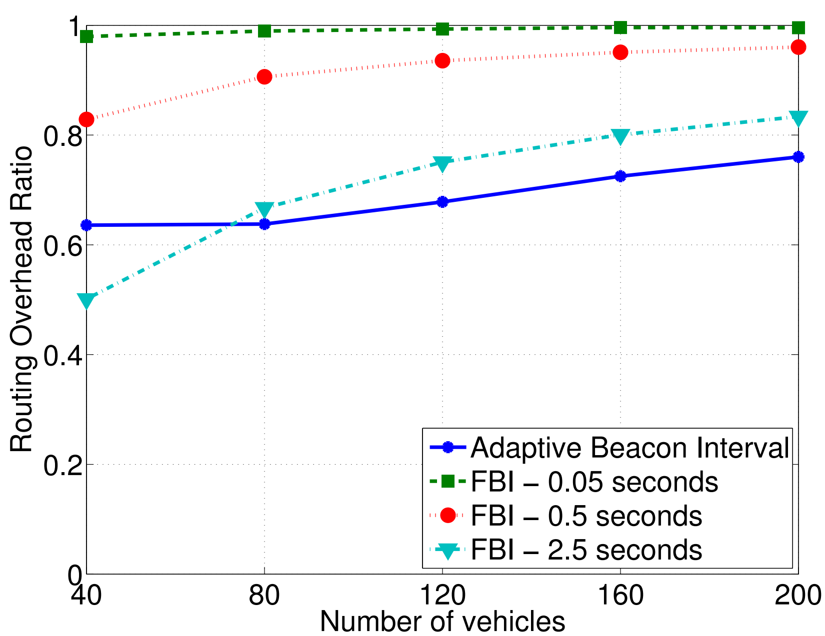
Figure 4.19a also demonstrates that the fixed interval of 2.5 seconds has the highest average end-to-end delay compare to other fixed intervals. This result also affirms our contention that reducing beacon frequency can also reduce the information accuracy and cause packets to be forwarded to unsuitable neighbors or dropped if the neighbor is already out of range. In Figure 4.19b and 4.19a, in spite of the fact that it shows the adaptive beacon interval has a small increase in term of delivery rate, the adaptive beacon interval has significant reduction in term of average end-to-end delay. This is because our adaptive beaconing system able to reduce the routing overhead approximately 25% to 35% compare to the shortest fixed interval which is 0.05 seconds, which is shown in Figure 4.19c. By reducing routing overhead, the proposed beaconing system also is able to reduce packet collision in the network as presented in Figure 4.19d. With congestion-free channel and small number of packet collision, vehicles are able to forward packets at small forwarding delay.



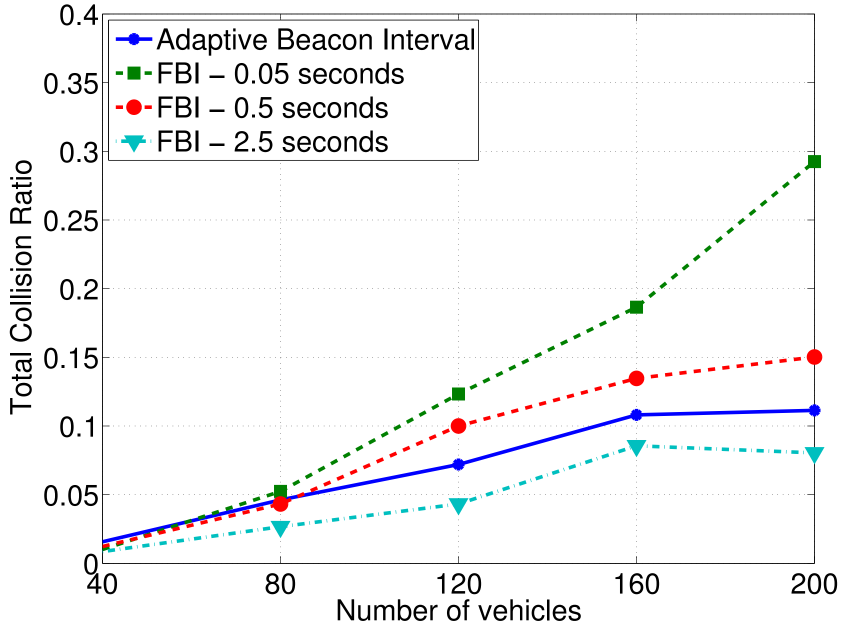
(a) Average end-to-end delay (in seconds)



(b) Packet Delivery Ratio



(c) Routing overhead ratio



(d) Collision ratio

Figure 4.19. Simulation results comparison between Adaptive Beacon Interval and FBI

4.5. Simulation Framework and Evaluation for Realistic Scenario

In this section, we present our evaluation framework in comparing the performance of our proposed *Adaptive Beaconing System* (ABS) with the *Adaptive Traffic Beacon* (ATB) protocol [37]. The details on ATB has been discussed in Section 2.3. The assumptions made in Section 4.4.1 are still hold for the realistic scenario and the same metrics detailed in Section 4.4.3 are still used in this evaluation. In addition, both ABS and ATB protocols use the same forwarding process, which is through greedy forwarding with carry and forward method as its recovery strategy. For this comparison, we implement the ATB protocol as the beaconing system as closely as proposed in [37].

4.5.1. Mobility and Network Model

To evaluate and compare the performance of the proposed system in a realistic scenario, we chose a road network based on OpenStreetMap [64] data of the area of the University of Toronto, Canada. The area section of the city is approximately 8.69 km by 3.65 km and includes the University of Toronto campus and a small part of Toronto downtown area. Figures 4.20 and 4.21 show the map of Toronto city from OpenStreetMap and the map generated by the VanetMobisim [63], respectively. The simulation scenarios are configured using common parameters that are shown in Table 4.3. ATB parameters in Table 4.4 are configured based on the configuration in [37].

Similar to the simulation in Section 4.4, each simulation scenario is repeated with at least 60 iterations to attain a 95% confidence interval.

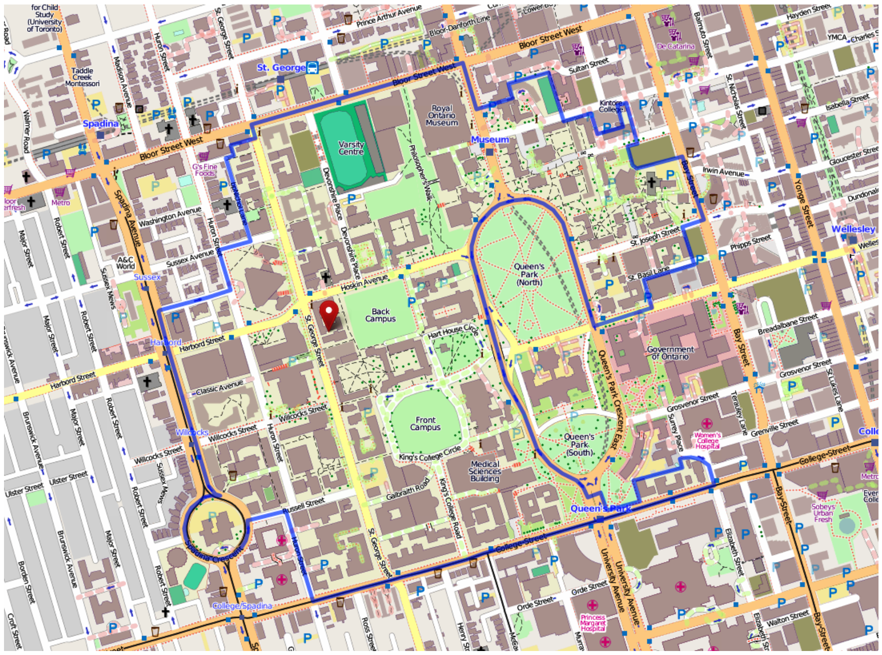


Figure 4.20. Map of Toronto city based on OpenStreetMap

Figure 4.21. Map of Toronto city generated by VanetMobisim

Table 4.3: Network Model Configuration

Area	8.69 km by 3.65 km
Number of vehicles	25, 50, 75, 100, 125, 150, 175, 200,
Speed (meter/sec)	Between 5 to 15 m/s
Maximum acceleration	2.6 m/s^2
Maximum deceleration	4.5 m/s^2
Simulation time	500 seconds
Number of connections	20
Traffic pattern	CBR Traffic
CBR Rate	0.5 packets/sec
Packet size	256 bytes
Propagation Model	Two Ray Ground Model
MAC Layer	IEEE 802.11b
Transmission range	250 meters
Forwarding and recovery strategy	Greedy Forwarding with Carry and Forward strategy

Table 4.4: ATB Simulation Parameters [37]

Parameter	Value
Minimum beacon interval I_{min}	30 ms
Maximum beacon interval I_{max}	60 s
channel quality weighting w_C	2
interval weighting w_I	0.75
Number of neighbors for $N = 1$	50
SNR for $S = 1$	50 dB
Neighborship data expiry	60 s

4.5.2. Simulation Results and Analysis

This section presents simulation results for performance comparison between ABS and ATB. The simulations are implemented in a realistic map using eight different densities (refer to Table 4.3). In each scenario, vehicles are configured to move in a random behavior with maximum speed of 15 m/s. Box plots in Figure 4.22 show the statistic of beacon intervals used in the simulation. For each of the data set, a box is used to represent the first and third quartiles, with the median marked in the middle of the box. Whiskers, which are the lines that extend from the edges of the box, show the minimum and maximum beacon interval of each protocol.

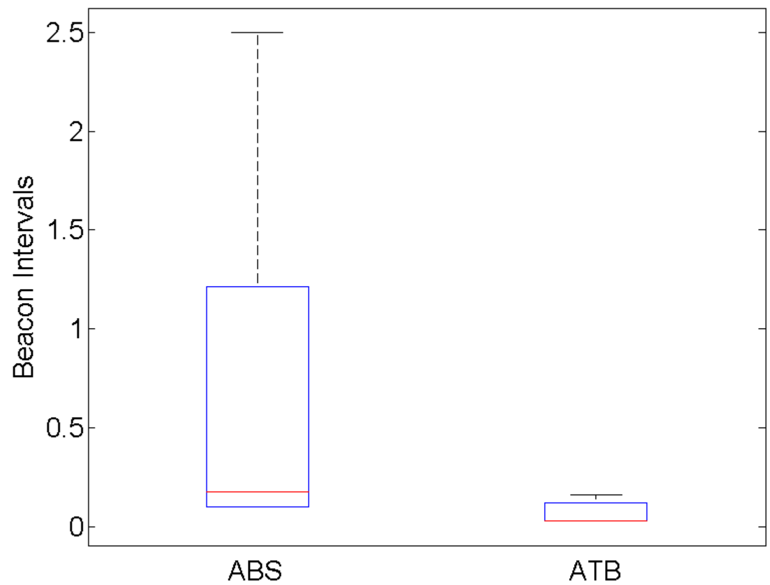
In a sparse network, Figure 4.22a shows that ABS uses interval range approximately between 0.1 and 2.5 seconds. The interquartile range for ABS is between 0.1 and 1.2 seconds with the median of 0.15 second. The box plot for ABS in Figure 4.22a skews toward values less than 1 second, which indicates ABS generates intervals ranging from 0.1 to 0.2 seconds most of the simulation time. The ATB box plot in Figure 4.22a shows a small distribution of interval values that are used in the sparse network simulations. ATB has interquartile range between 0.02 to 1 second with the median of 0.02 second. ATB statistics confirm that when ATB detects a congestion free channel, it tries to exchange information as frequently as possible.

Figure 4.22b shows the statistics of ABS and ATB intervals in a dense network. The ABS box plot in Figure 4.22b indicates the range of intervals used throughout the simulation is between approximately 1.5 to 4.5 seconds. The interquartile range for ABS is approximately between 2.5 to 4.0 seconds with the median value of 3.75 seconds. The skewness of the ABS box plot in Figure 4.22b reveals that ABS uses interval values between 3.5 to 4.0 seconds. On the other hand, ATB box plot in Figure 4.22b shows a small distribution of beacon intervals in which ATB uses interval values ranging from approximately 0.3 to 0.4 second throughout the simulation.

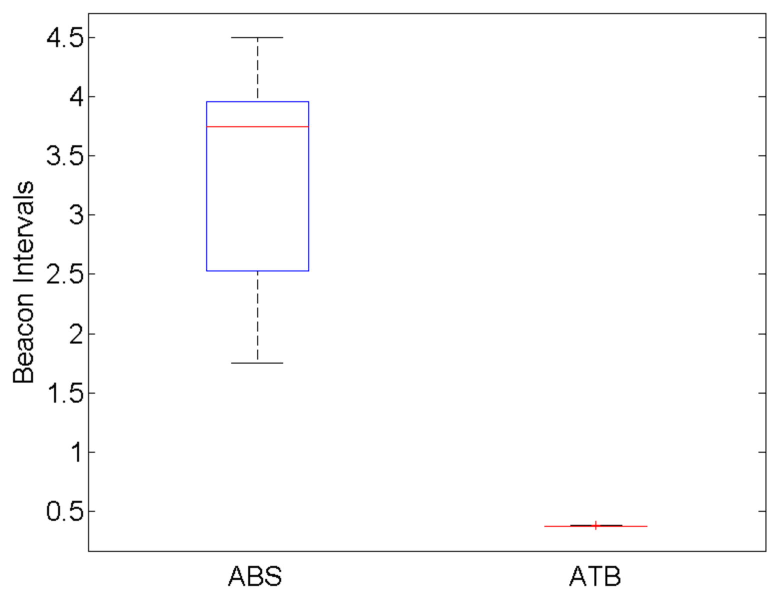
Both ABS box plots in Figures 4.22 indicates that ABS is capable of adjusting the intervals based on the current network's condition. ABS tries to exchange information as frequently as possible in a sparse network and lower the beacon frequency in a dense network to reduce channel overhead and packet collisions. Based on Figures 4.22, we conclude that ATB is more fitting for broadcasting current events such as accidents or road congestion compared to our scenario of transmitting data packets with carry and forward strategy. A small distribution of ATB intervals in Figures 4.22 can be confirmed with [37] results in their realistic scenario simulation, as shown in Table 4.5.

Table 4.5: ATB results in from [37]

Minimum	First Quartile	Median	Third Quartile	Maximum
0.03	0.05	0.06	0.14	15.55



(a) Sparse network

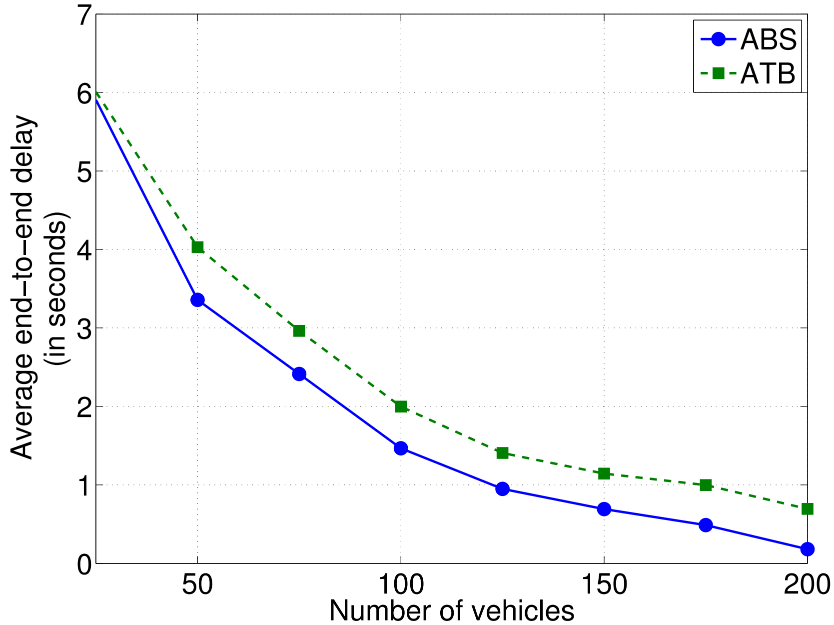


(b) Dense network

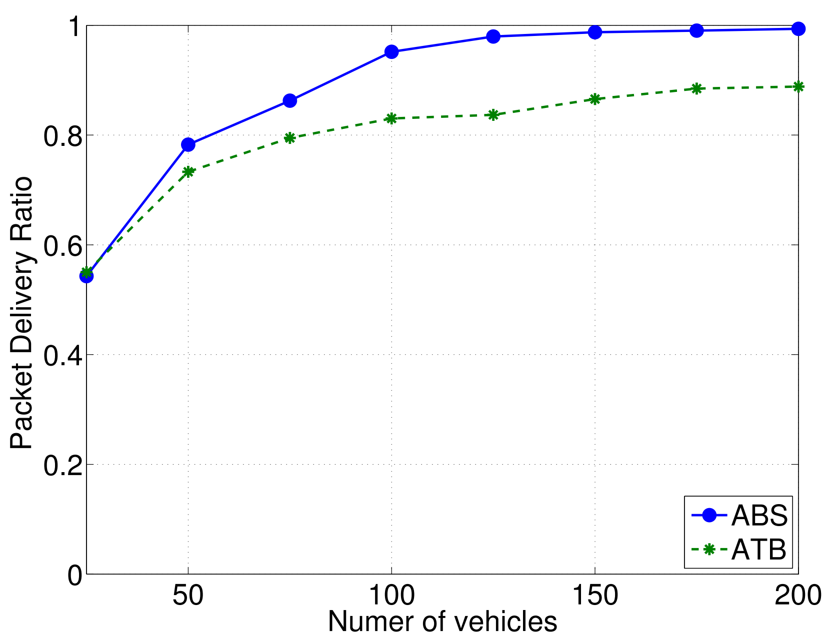
Figure 4.22. Interval rate comparison between ABS and ATB in two different densities

Plots displayed in Figures 4.23 show consistency with our results for the Manhattan grid model in Section 4.4.4. In Figures 4.23a and 4.23b, in a sparse network, both ABS and ATB show consistent decrease in average end-to-end delay and increase in packet delivery ratio. However, as shown in Figures 4.23a and 4.23b, ABS has successfully lowered end-to-end delay by relatively 0.5 to 1 second and increased delivery ratio by approximately 5% - 20% compared to ATB.

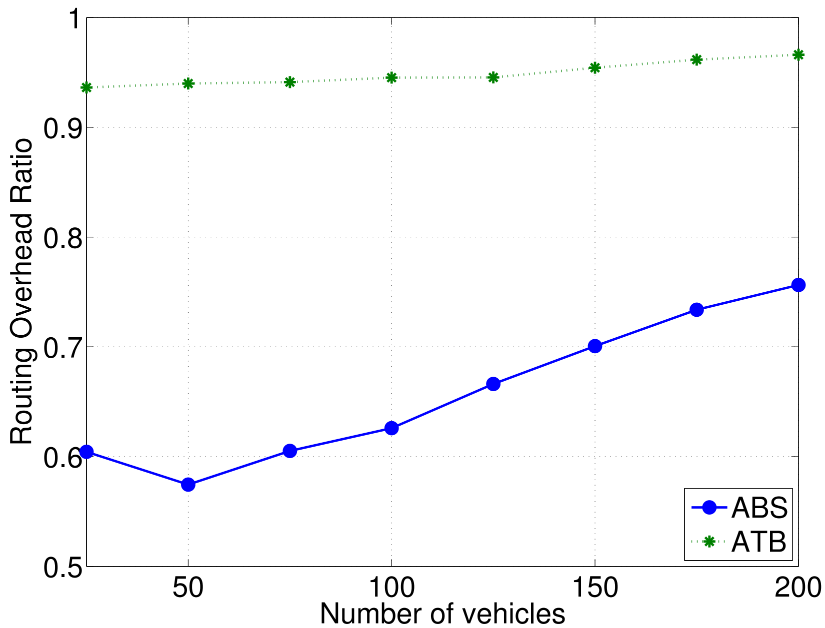
Figure 4.23c reveals that 95% of ATB overall transmissions in the network are dominated by beacon packets whereas the highest overhead ratio for ABS is at 200 vehicles with approximately 75%. The trend can be confirmed by Figures 4.22 where ATB uses interval values less than 0.5 second in broadcasting the beacon packets throughout the simulation. As displayed in Figures 4.23c and 4.23d, ABS is able to reduce the overhead ratio by 20% - 40% of ATB overhead ratio and maintain low collision ratio at 10% while the ATB collision ratio steadily increases with the increment of vehicle density. Results in Figures 8 and 9 indicates that by adapting beacon intervals to the current network's condition, ABS is able to reduce routing overhead and packet collisions, and, therefore, reduce overall congestion in the wireless channel. With minimum congestion, data packets are able to be forwarded at small delivery delay and high delivery ratio.



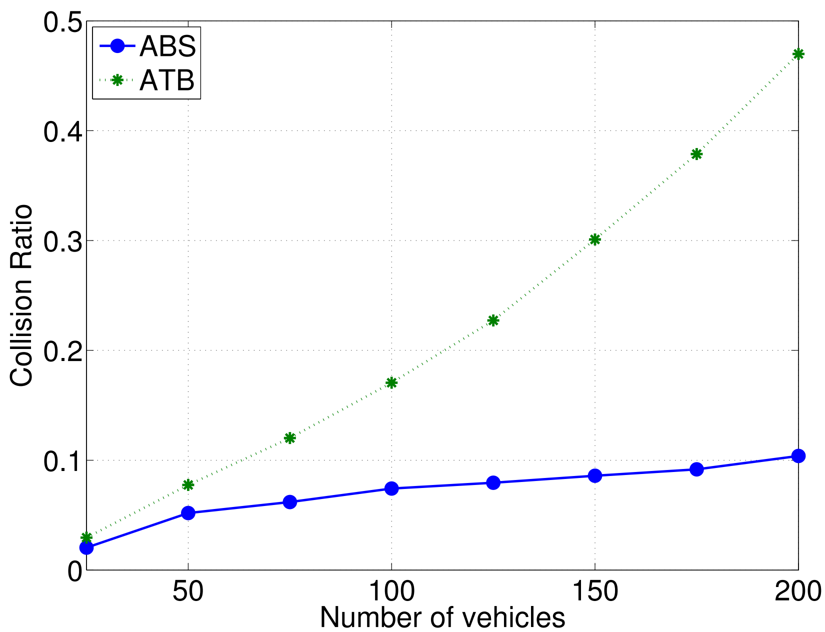
(a) Average end-to-end delay



(b) Packet delivery ratio



(c) Routing overhead ratio



(d) Collision ratio

Figure 4.23. Performance comparison between ABS and ATB

Results in Figures 4.23 indicates that by adapting beacon intervals to the current network's condition, ABS is able to reduce routing overhead and packet collisions, and, therefore, reduce overall congestion in the wireless channel. With minimum congestion, data packets are able to be forwarded at small delivery delay and high delivery ratio.

4.6. Conclusion

In vehicular network, sending beacon packets at constant rate can overload the channel and affect both routing protocols and application performances. Thus, adaptation in the beaconing system is necessary to maintain congestion free channel and at the same time, exchange information as frequently as possible. In this chapter of the thesis, we have presented an adaptive beaconing system based on fuzzy logic approach using three conditions. The three conditions are *packet carried time*, *number of single-hop neighbors* and *vehicles' speed*. The performance evaluation of the proposed beaconing system and the fixed beacon intervals have been carried out using NS2. The performance of the proposed system has also been compared with ATB protocol [37]. The results have shown that the proposed *adaptive beaconing system* is able to adjust the beacon interval based on vehicles mobility and current traffic condition. The results also have confirmed that the adaptive beaconing system is able to reduce overhead and collision significantly compared to fixed interval rates and the ATB protocol.

Chapter 5

The Implementation of Proactive Multi-copy Routing (PMC) Protocol

In this section, we present the design for *proactive multi-copy routing protocol* (PMC), which able to reduce the end-to-end delivery delay in a vehicular network. When using a *carry and forward* based routing protocol, the end to end delivery delay is mainly contributed by the *carry and forward* procedure rather than wireless transmission. Therefore, the *end-to-end delivery* delay in a vehicular network can be further reduced by replicating multiple copies of data packets to other nodes in order to increase the chance of finding the destination node at small delivery delay.

5.1. Introduction

As mentioned in Section 2.1, vehicular network topology is very dynamic compared to traditional MANET because of the movement and speed of the vehicles. Thus, a vehicular network is always partitioned due to this reason, especially if the vehicle density is low. In this situation where a direct end-to-end path between source and destination can be considered as nonexistent, a regular ad hoc routing protocol with complete path discovery mechanism is not feasible since the routing path is usually disconnected due to the intermittent nature of network links. To overcome this problem, vehicles can be used as carriers to deliver messages using store-and-carry forwarding whenever forwarding option via wireless transmission is not available. VANET routing protocols [16–18,34] that employ this mechanism have been discussed extensively in Section 2.2.

Noting that the *carry and forward* mechanism can largely influence the end-to-end message delivery delay, majority of VANET routing protocols employ carry and forward method only to ensure intermittent connectivity in a vehicular network does not hinder packet delivery. Further, these protocols design their own unique solution to reduce end-to-end delay. For example, both the *Greedy Traffic-Aware Routing* (GyTAR) [17] and the *Vehicle-Assisted Data Deliver* (VADD) [16] protocols rely on traffic density in selecting forwarding path. VADD prefers a high traffic density path to a geographically shortest path especially, if the geographically shortest path contains partitioned networks, and packets have to be carried by vehicles instead of transmitted via wireless channel. At each intersection, GyTAR dynamically selects the next candidate junction by taking real-time vehicular traffic into account. The *Distributed Vehicular Broadcas* (DV-CAST) [34] and the the *Border Node based Routing* (BBR) [18] protocols use restricted flooding mechanism to deliver packets to the destination vehicles. DV-CAST suppress the flooding procedure in a dense network to avoid redundant broadcast and reduce overhead.

In this chapter, we design and implement a routing protocol with a carry and forward feature that is able to deliver data packets with small end-to-end delay without the use of real time traffic data to determine forwarding path, or flooding mechanism. The underlying idea of our protocol design is to forward multiple copies of data packets at road intersections to increase the chance of reaching the destination, and thus reduce the end-to-end delivery delay.

5.2. Proactive Multi-Copy Routing Protocol

5.2.1. Design Assumptions

In our *Proactive Multi-Copy (PMC)* design, we assume that each vehicle has the capability to obtain the road map data and its position information, which we consider as a valid assumption since nowadays most of the vehicles have a Global Positioning System (GPS) device.

In addition, we are assuming the source vehicle acquires the location of the destination vehicle via a location service, which is beyond the scope of our design and will be not discussed in this thesis. Once the destination vehicle's location is obtained, the information is carried in the packet so that intermediate nodes do not have to use the location service. However. due to the dynamic nature of a vehicular network, the destination node may have already left the area by the time packets arrive at the initial location. In this case, the packet carrier will obtain the new location of the destination node via a location service and forward the packets toward the new location. Further, we presume the use of a location service is limited only to acquiring the destination node location. Therefore, each vehicle in the network will depend on the beaconing system for its neighbor's information.

5.2.2. Proactive Multi-Copy (PMC) Protocol Design

The PMC protocol is based on the idea of a carry and forward protocol combining with a single hop beaconing system. In the PMC protocol, packet replication is made only at the intersection in which multiple road paths to the destination are available based on the candidates moving direction and destination vehicle positioning information. At the straight road section where there is no alternative paths, the protocol greedily forwards packets to the next intersection that leads towards the destination.

In this protocol, we assume that each node maintains neighborhood state information in an nb_table, which stores the id, the time of the beacon transmission, the current position, the speed, and the velocity of the neighboring vehicles at the time when the beacon is received. This information is acquired through single-hop beaconing system, where a vehicle would broadcast its information to its single-hop

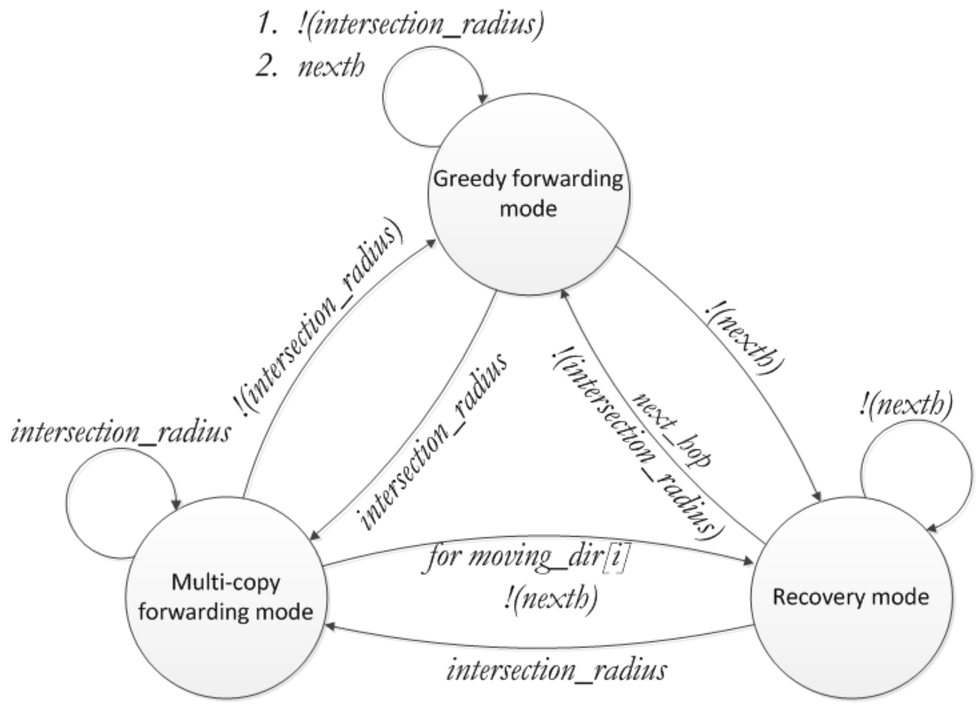


Figure 5.1. Transition modes in PMC

neighbors. The current protocol uses the same beacon packet format that is described in Section 4.3.1.

As shown in Figure 5.1, there are three packets modes in the PMC protocol; *greedy forwarding mode*, *multi-copy forwarding mode*, and *recovery mode*. When a vehicle receives a packet, the vehicle will determine whether it is located at an intersection or not (*intersection_ radius*). If the vehicle is currently at an intersection, the packet enters the *multi-copy forwarding mode*, or else, it enters the *greedy forwarding mode*. In the *greedy forwarding mode*, the packet carrier finds the best next hop (*nexth*) via the greedy algorithm with direction awareness. This algorithm will be explained later in Section 5.2.3. In either *multi-copy forwarding mode* or *greedy* *forwarding mode*, if the current vehicle is unable to find the closest node to the destination node other than itself, or also known as *local maxima*, the packets then enter the *recovery mode*. This situation is commonly happened in a sparse network, where a vehicle may not have any information on its neighbors at all or the neighbors already left its transmission range. In the *recovery mode*, packets are carried in the buffer and the carrying vehicle will try to retransmit once it receives beacons from its neighbor or when it arrives within other vehicle's transmission range.

5.2.3. Single-copy Forwarding

This section describes a *single-copy forwarding* (SCF) algorithm that is commonly used by VANET routing protocols. In this algorithm, the forwarding decision is made at each hop using position and direction information of neighboring vehicles. We ascertain that this algorithm is the basic forwarding algorithm used by the majority of VANET routing protocols.

Figure 5.2 demonstrates a scenario example where a *source vehicle* (v_s) is forwarding data packets to a *destination vehicle* (v_d). In the SCF algorithm, v_s finds a next hop from its neighboring vehicles (v_k) using *greedy forwarding algorithm*, which is outlined in Algorithm 5.1. As illustrated in Algorithm 5.1, v_s chooses a next hop, v_{nexth} with the smallest distance to v_d based on the information in the nb_table. In the example scenario of Figure 5.2, v_s determines v_1 as its next hop vehicle using greedy forwarding. Subsequently, v_1 chooses v_2 as its v_{nexth} and the data packets will then be forwarded through road segment of I_{12} ($r_{I_{12}}$) towards v_d. Since the traffic

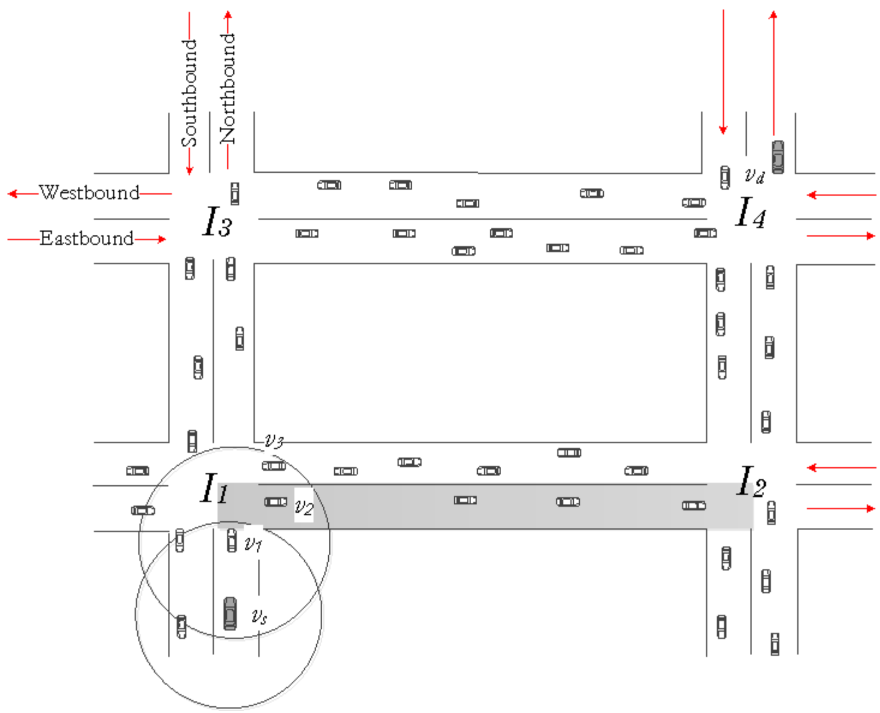


Figure 5.2. Scenario for single-copy routing in an intersection

density is low in that particular segment, the packets are then carried by intermediate vehicles until the latter are able to forward the packets wirelessly, in which high latency can occur during the carrying process.

Majority of forwarding algorithms in VANET have ascertained that greedy forwarding is not enough to find a suitable v_{nexth}. For example, in Figure 5.2, if we assume v_1 is the current forwarding vehicle, then v_2 or v_3 can be a suitable candidate for v_{nexth}. However, in Figure 5.2, if we consider the moving direction for the candidate vehicles, then v_3 cannot be taken into consideration since v_3 is moving away from v_d. It would be more effective if v_2 is selected as v_{nexth}.

To determine whether a candidate vehicle is moving towards or away from the

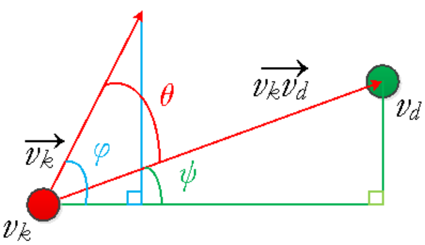


Figure 5.3. Visual representation of Equation (5.1)

destination vehicle, v_d, we utilize Equation (5.1) to calculate θ. In Equation (5.1), $\vec{v_k}$ is v_k vector that indicates an intermediate vehicle (v_k) speed value and moving direction. We define variable $\overrightarrow{v_k v_d}$ as distance vector from v_k to v_d. Figure 5.3 displays the visual representation of Equation (5.1). As illustrated in Figure 5.3, θ is the magnitude of the difference between angles ϕ and ψ where ϕ is the angle of the $\vec{v_k}$ and ψ is the angle of the $\vec{v_{kd}}$. If θ is smaller than or equal to $\frac{\pi}{2}$, than we consider the current vehicle is moving towards the destination vehicle.

$$\theta_{kd} = |\phi - \psi| \leq \frac{\pi}{2} \qquad (5.1)$$

where

$$\phi = \arctan\left(\frac{(\vec{v_k})_y}{(\vec{v_k})_x}\right) \text{ and } \psi = \arctan\left(\frac{(\overrightarrow{v_k v_d})_y}{(\overrightarrow{v_k v_d})_x}\right)$$

However, relying exclusively on θ in Eqn (5.1) is not enough to determine whether v_k is moving towards or away from v_d. The current forwarder also needs to discover the cardinal direction of a candidate vehicle, v_k, before calculating θ. Figure 5.4 displays a snap-shot for the example scenario in Figure 5.2. Each cardinal direction shown in Figure 5.4 is assigned with an angle of the polar coordinate system; *North*

$= 90°$, $South = 270°$, $East = 0°$, and $West = 180°$.

In Figure 5.4, both v_2 and v_3 are suitable next hop candidates for v_1. Without discovering the cardinal direction for v_2 and v_3, the calculation of θ from Eqn (5.1) will result in 0° for both v_2 and v_3 since both $(\vec{v_2})_y$ and $(\vec{v_3})_y$ have zero value.

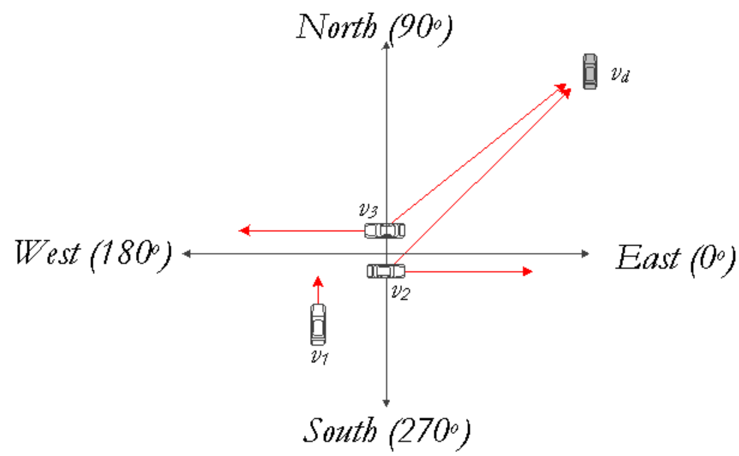


Figure 5.4. A snap-shot from the earlier example in Figure 5.2

By identifying the cardinal directions for v_2 and v_3, the calculation for θ_{2d} and θ_{3d} are as follows:

$$\theta_{2d} = |0° - \psi_{2d}| \quad \text{and} \quad \theta_{3d} = |180° - \psi_{3d}|$$

From the calculation above, it is evident that v_2 is the most suitable next hop vehicle for v_1. Algorithm 5.2 outlines the pseudo-code for discovering whether v_k is moving towards or away from v_d. The function in Algorithm 5.2 is implemented in Algorithm 5.1 at Line 5.

Algorithm 5.1 GREEDY FORWARDING

Require: v_k location; v_d location; v_c location

Ensure: v_{nexth}

$D_{min} \leftarrow dist\,(v_c, v_d)$

for each v_k in nb_table **do**

    $d_{min} \leftarrow dist\,(v_k, v_d)$

    if $d_{min} < D_{min}$ **then**

        if DIR(v_{kd}) is true **then**

            $D_{min} \leftarrow d_{min}$

            $v_{nexth} \leftarrow v_k$

        end if

    end if

end for

Algorithm 5.2 DIR(v_k)

Require: $\vec{v_k}$, $\vec{v_d}$

Ensure: v_k direction to v_d

Determine v_k cardinal direction

$\theta_{kd} \leftarrow \left|\arctan\left(\vec{v_k}, \vec{v_d}\right) - \arctan\left(\vec{v_k}\right)\right|$

if $\theta_{kd} \leq \frac{\pi}{2}$ **then**

    return 1

else

    return 0

end if

5.2.4. Multi-copy forwarding

In this section, we present the multi-copy forwarding algorithm, which is the main component of the PMC protocol. We combine the single-copy and multi-copy algorithms in the PMC protocol in order to reduce the delivery delay by proactively replicates data packets at intersections and forward the packets greedily at regular road segments. At any intersection, the PMC protocol replicates the packets through road segments where the road direction, $\overrightarrow{R}$ is moving towards v_d. We define four road directions in the PMC as $Northbound = 1$, $Southbound = 2$, $Eastbound = 3$, and $Westbound = 4$. Algorithm 5.3 outlines the main process of the PMC protocol and Algorithm 5.4 describes multi-copy forwarding algorithm which replicates multiple copies of a data packet when a vehicle arrives at an intersection.

In the PMC protocol, each vehicle has to maintain a *m_copy* table, which has data entries that include data packet id, destination vehicle id, and the location of an intersection at which the corresponding packet is replicated.

As outlined in Algorithm 5.3, upon receiving a data packet with id n ($DPkt_n$), an intermediate node, v_k examines its *nb_table* to determine if v_d is one of its current neighbors. If the condition is true, then v_k immediately forwards $DPkt_n$ to v_d. Otherwise, v_k determines whether it is located an intersection or not. If the condition is true, then v_k examines its *m_copy* table to determine if $DPkt_n$, which is bound for destination vehicle v_d, has already been replicated and forwarded beforehand in that particular intersection (I_c) in order to minimize redundant replication. If $DPkt_n$ has

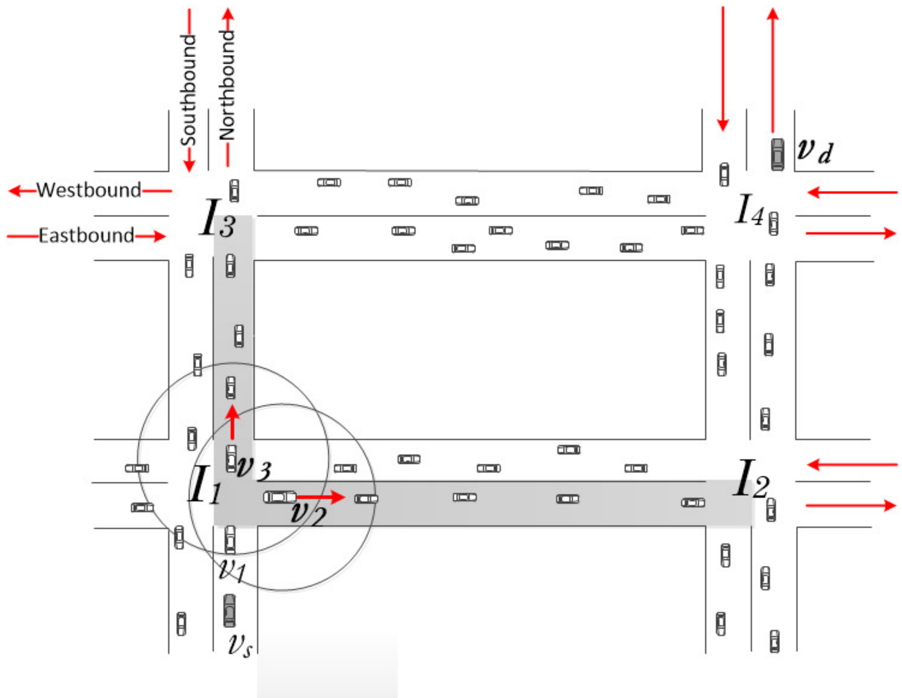


Figure 5.5. Example scenario for multi-copy forwarding at an intersection

never been replicated at I_c, then the protocol implements multi-copy forwarding on $DPkt_n$, as shown in Line 8 of Algorithm 5.3.

In Figure 5.5, after v_1 receives $DPKt_n$ from v_s at intersection I_1 and it has ascertained that $DPkt_n$ has never been replicated and forwarded at I_1 beforehand, v_1 implements multi-copy forwarding, as described in Algorithm 5.4, by examining $\overrightarrow{R}$ of each road segment at I_1 to find out which of the four directions are moving toward v_d. In the example scenario shown in Figure 5.5, road segment I_{13} with $\overrightarrow{R} = Northbound$ and I_{12} with $\overrightarrow{R} = Eastbound$ which are greyed out in the figure, are the best road segment candidates for packet replication. Henceforth, using greedy algorithm with direction awareness as shown in Algorithm 5.1 and 5.2, v_1 finds the next hop from its neighboring vehicles for the selected road segments and forwards the packets to the selected next hop vehicles, which in our example v_2 and v_3. After v_1 replicates $DPkt_n$ and forwards to v_2 and v_3, both v_2 and v_3 forward $DPkt_n$ to road segments $r_{I_{12}}$ and $r_{I_{13}}$, respectively via greedy forwarding with direction awareness. After forwarding packets to v_2 and v_3, v_1 stores $DPkt_n$, v_d and I_c information in the *m_copy* table. This process is then repeated once $DPkt_n$ reaches intersections I_2 and I_3.

Examining the *m_copy* table before replicating data packets may not be enough to minimize redundant replication since only the current forwarder, which in our example, v_1, stores the replication information in its *m_copy* table. To avoid redundant replication, our solution is the use confirmation message in the periodic beaconing packets. Additional fields (ACK, $DPkt_n$, v_d, I_c) are added to the packet header which

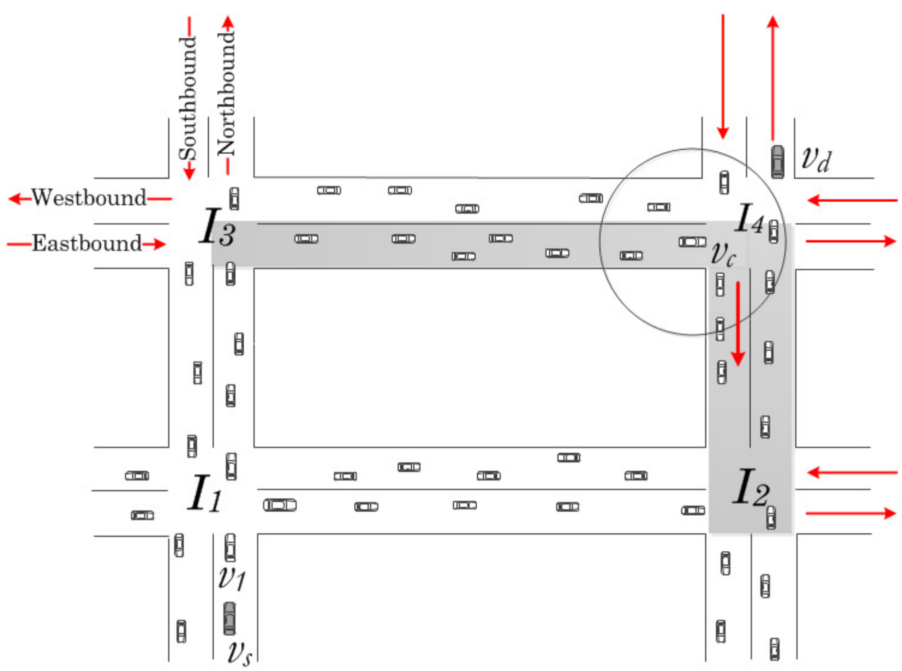


Figure 5.6. Example scenario for minimizing redundancy at an intersection

stores acknowledgment id, packet id, destination id and location of the current intersection. ACK field is used to inform the receiver of a beacon packet that the current beacon contains information on packet replication at the adjacent intersection. In Figure 5.6, we presume $DPkt_n$ in our previous example arrives at intersection I_4 via I_3; and the current forwarder (v_c) performs multi-copy forwarding at I_4. After forwarding the copies of $DPkt_n$, v_c sets the ACK field to 1 and attaches $DPkt_n$, v_d and I_c information to the most current beacon packet and broadcasts it to its neighbor along road segments $r_{I_{42}}$. The receivers of the beacon packet at road segment I_{42} save the information of the additional fields in their m_copy table. With this confirmation mechanism, other intermediate vehicles can minimize redundant copying at intersection I_4 from I_2.

Algorithm 5.3 main process of PMC

v_s = source vehicle

v_d = destination vehicle

v_c = current forwarding vehicle

v_k = intermediate vehicles/nodes

$DPkt_n$ = data packets with id $n = 1 \ldots m$

I_i = intersections available in the city map

I_c = current intersection

$r_{I_{i_a}}$ = road segments at intersection i where $a = 1 \ldots 4$

$\overrightarrow{R}$ = road direction vector

$nexth$ = next hop for forwarding data packets

Upon receiving $DPkt_n$:

if v_d is current neighbor **then**
    forward $DPkt_n$ to v_d
else
    if $radius\,(I_c) == true$ **then**
        if $[DPkt_n, v_d, I_c]$ exist in m_copy table **then**
            drop $DPkt_n$
        else
            MULTI_COPY FORWARDING
        end if
    else
        $v_{nexth} \leftarrow$ GREEDY FORWARDING
        if v_{nexth} is available **then**
            forward $DPkt_n$ to v_{nexth}
        else
            RECOVERY MODE
        end if
    end if
end if

Algorithm 5.4 MULTI_COPY FORWARDING

$mcopy \leftarrow 0$
for $a \leftarrow 1, 4$ **do**
    determine $\overrightarrow{R}$ for $r_{I_{ia}}$ to v_d is true
end for
for each $r_{I_{ia}}$ where $\overrightarrow{R}$ to v_d is true **do**
    find $nexth$
    if $nexth$ for $r_{I_{ia}}$ available **then**
        set $nexth\_status_{I_{ia}}$ = true
        $nexth_{I_{ia}} \leftarrow v_{nexth_{ia}}$
        $mcopy = mcopy + 1$
    else
        set $nexth\_status_{I_{ia}}$ = false
        $nexth_{I_{ia}} = NULL$
    end if
end for
if $mcopy \neq 0$ **then**
    for each $r_{I_{ia}}$ where $\overrightarrow{R}$ to v_d is true **do**
        copy $DPkt_n$ and forward to $v_{nexth_{ia}}$
        store $DPkt_n$, v_d, and I_i in m_copy table
    end for
else
    RECOVERY MODE
end if

5.3. Simulation Framework

In this section, we present our simulation framework in evaluating the performance the proposed PMC protocol and compare it to the single-copy forwarding algorithm via simulations conducted in NS-2 [54,55]. In our framework, we assume all vehicles in the network are equipped with wireless transceivers that allow the vehicles to transmit and receive packets via wireless channel. We assume all vehicles have a GPS device to enable the vehicles discover their location in the network. In addition, each vehicle has a high capacity buffer to store packets during recovery mode.

As we mentioned in Secton 5.2.2, the PMC protocol uses a single-hop beaconing system to acquire knowledge on local topology. However, we design the PMC protocol with a fixed beaconing system since the routing overhead is not the research focus in this chapter.

5.3.1. Mobility and Network Model

Both PMC and single-copy forwarding protocols are implemented in the network simulator NS-2 [55] for performance assessment. The simulation scenarios are configured in a 3 by 3 km urban grid model (Refer to Figure 5.7) with five different densities ranging from 50 vehicles to 175 vehicles. We use the VanetMobiSim *Intelligent Driver Model with Lane Changing* (IDM_LC) [63] to generate realistic vehicle mobility with maximum speed of 15 m/s. Using this model, each vehicle is able to adjust its speed based on the movement of the neighboring vehicles and change lane to overtake other vehicles in multi-lane roads. This model also supports smart intersection management, where vehicles slow down and stop at intersections, or they act accordingly at traffic lights. Table 5.1 summarizes the configuration parameters used in the simulation. The communication range is set at 250 meters and all vehicles are required to broadcast beacon packets every 0.5 second. Five pairs of source and destination vehicles are selected in random and each source transmits one data packet for every two seconds.

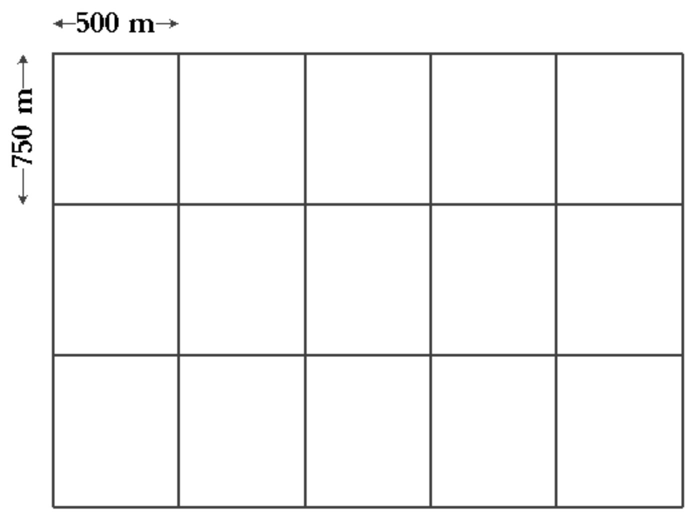


Figure 5.7. Manhattan grid topology used in the simulation

5.3.2. Performance Metrics

For each scenario. we execute our simulation with 100 iterations to ensure statistical validity for 95% confidence interval. The performance assessment is based on four metrics:

Table 5.1: Network Model Configuration

Area	3 km by 3 km
Number of vehicles	50, 75, 100, 125, 150, 175
Speed (meter/sec)	Between 5 to 15 m/s
Simulation time	1000 seconds
Number of connections	5
Traffic pattern	CBR Traffic
CBR Rate	5 packets/sec
Packet size	256 bytes
Beacon interval	0.5 second
Propagation Model	Two Ray Ground Model
MAC Layer	IEEE 802.11b
Transmission range	250 meters

1. ***Packet Delivery Ratio***: Measures the fraction of data packets that are successfully received by destination to those generated by traffic source.

2. ***Average End-to-End Delay***: Measures the average difference between the time a data packet is originated by an application and the time the same packet arrives at its destination

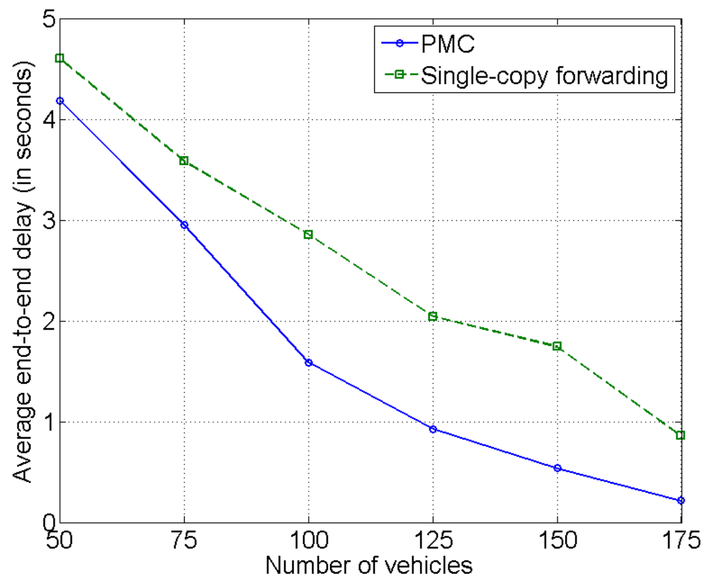
3. ***Routing Overhead Ratio***: Measures the fraction of total beacon packets emitted to total number packets transmitted in the network

4. ***Total Collision Ratio***: Measure the ratio of total number of collisions to total number of packets transmitted in the network

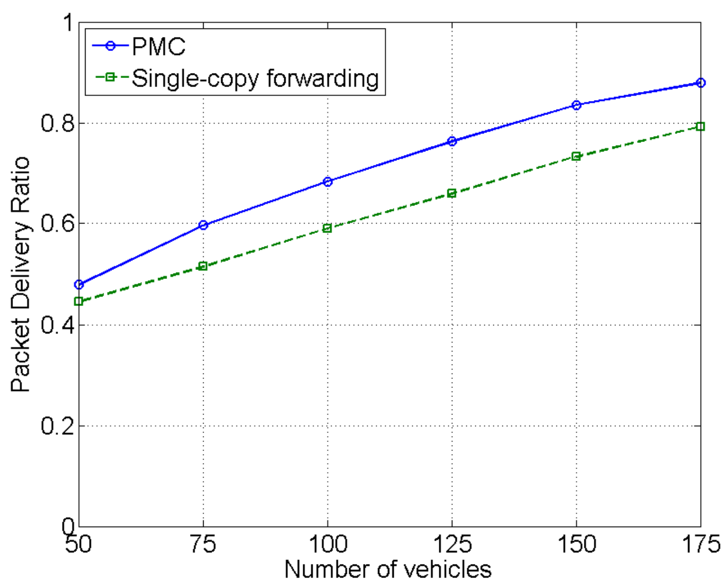
5.4. Simulation Results and Analysis

Figures 5.8 present the simulation results on performance comparison between the PMC protocol and the single-copy forwarding algorithm. Figure 5.8a displays the simulation results on the average end-to-end delay at different network densities. In the figure, PMC consistently outperforms the single-copy forwarding by the average of approximately 0.5 second in a disconnected network and 1 second in a well connected network. Figure 5.8a shows a low reduction of the end-to-end delay in a sparse network compares to a well-connected network because in a sparse network, data packets are carried by vehicles most of time. However, the reduction of the end-to-end delay shows that the PMC protocol is still able to perform efficiently in a sparse network since vehicles are most likely to stop or slow down at intersections. Figure 5.8b presents the delivery ratio comparison between PMC and the single-copy forwarding algorithm. The figure shows an increment of approximately 12% in delivery ratio when the PMC protocol is used in the scenario. From Figures 5.8a and 5.8b, we ascertain that using multi-copy forwarding in transmitting data packets can further reduce average end-to-end delay and increase packet delivery ratio.

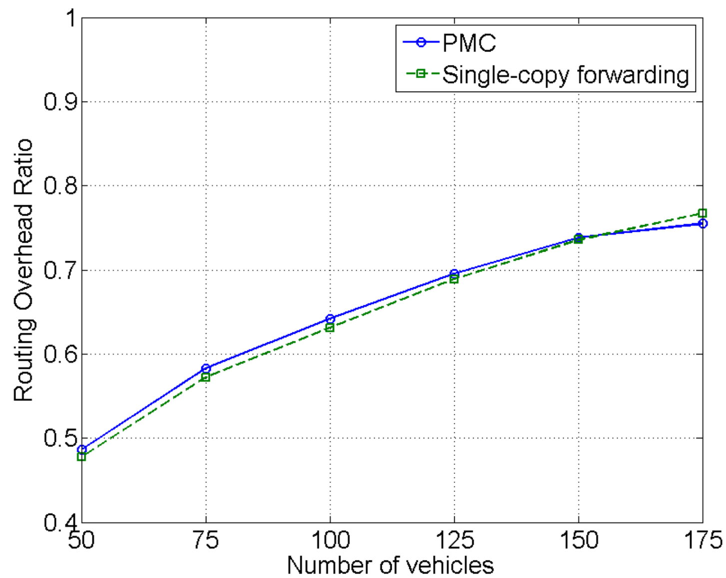
Figures 5.8c and 5.8d display the comparison overhead and collision ratios between PMC and single-copy forwarding in the simulation. In the figures, PMC and single-copy forwarding show similar overhead and collision ratios while achieving lower end-to-end delay. Even though the number of vehicles increases in a dense network, the PMC protocol is still able to maintain similar overhead ratios as single-copy forwarding since PMC uses the same beaconing system as single-copy forwarding with only small additional bytes of information added to the beacon packet.



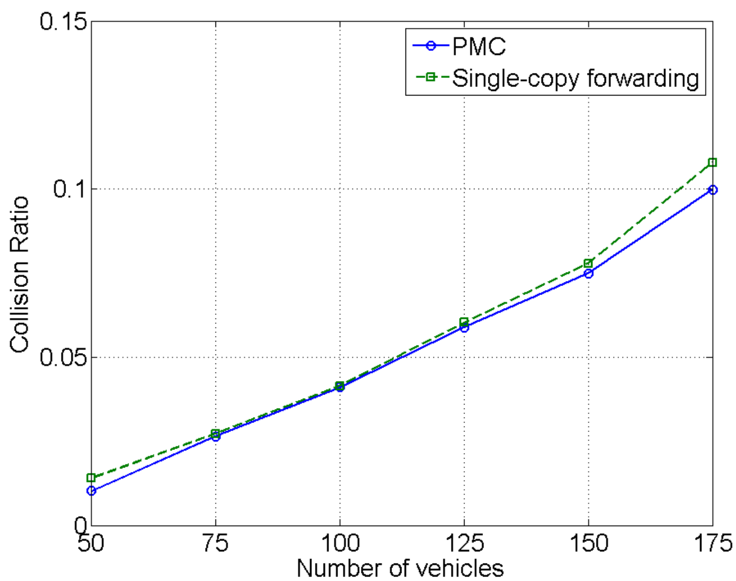
(a) Average end-to-end delay (in seconds)



(b) Packet Delivery Ratio



(c) Routing overhead ratio



(d) Collision ratio

Figure 5.8. Simulation results comparison between single-copy forwarding and PMC

5.5. Conclusion

In this chapter, we have designed and implemented a new proactive multi-copy (PMC) routing protocol that reduces end-to-end delay by proactively replicates data packets at intersections and forwards them to different intermediate nodes. By forwarding multiple copies of packets to different relays at different road segments, the protocol increases the chance of reaching the destination at low delivery delay. The forwarding mechanism is based on information that is commonly available via a GPS device. Simulation in an urban grid model has shown that the PMC protocol is able to reduce the average end-to-end delay and increase the delivery ratio compared to single-copy forwarding results. Despite having additional information added to beacon packets to minimize redundant replication, the results have shown that the proposed protocol is able to maintain similar overhead and collision ratios as the single-copy forwarding algorithm.

Chapter 6

Conclusions and Future Work

6.1. Conclusions

In this thesis, we investigate several issues related to the connectivity and performance of a carry and forward based routing protocol in a vehicular ad hoc network (VANET). We address the challenges posed by the unique properties of VANET, which are dynamic network topology and intermittent connectivity. Our research topics focus on investigating end-to-end delay for multiple disconnected clusters of vehicle on a unidirectional highway; developing an adaptive beaconing system using a fuzzy logic approach to reduce channel overhead and packet collisions; and implementing a routing protocol that proactively replicates data packets at intersections to increase the chance of reaching the destination with small end-to-end delay.

In VANET, the vehicles mobility and network topology are highly diverse especially in an urban area. Multi-hop forwarding through a large geographic area is usually expected for disseminating data to a far away vehicle. Due to VANET's dynamic nature, it is common for the forwarding path to go over some areas of the network where the number of vehicles is low and a next forwarding vehicle is hard to find. In this situation, an end-to-end connection over a large distance may not always exist. To support data dissemination in the presence of partitioned or disconnected networks, a carry and forward approach is used where a forwarding vehicles carries the data packet when a suitable next hop vehicle is not available, and forward the packet when a new vehicle moves into its vicinity. Although this approach can increase the chance of delivering the data packet to its destination, it can influence the packet's end-to-end delivery delay since the packet is moving with the speed of the carrying vehicle as opposed to the speed of light.

Thus, in Chapter 3, an analytical study is done on the data packet end-to-end delay between a source and a destination over multiple clusters of vehicles. In this chapter, the distribution model for end-to-end delay is derived which, to our best knowledge, has not been done in the literature. In this study, we are able to confirm that the carry and forward process often occurs in a low density network and the process has caused the packets to be delivered at high end-to-end delay. In addition, using the end-to-end delay model, we are able to estimate the probability of end-to-end delivery delay based on the average traffic flow rate. Nonetheless, the study is done without considering a number of real-world communication aspects such as channel fading, and contention issues at media access layer which can be considered as the research future works.

Transmitting packets on a single wireless communication channels often leads to channel contention problems especially in a dense network and all the vehicles are transmitting their periodic beacon packets and data packets at the same time. In Chapter 4, we concentrate on the effects of period local broadcast communication and design an adaptive approach for broadcasting beacon packets based on the logical reasoning of a VANET conditions. From our study in Chapter 4, we are able to show that by adapting the intervals for broadcasting the beacon packets based on the current network conditions, channel contention problems and packet collisions can minimized and at the same, we are able to reduce end-to-end delay and increase delivery ratio.

With the dynamic change in both topology and mobility, a vehicle can suffer intermittent connectivity or network disconnections in some parts of the network due to the low number of vehicles. In other areas where the number of vehicles is high, the data packets are normally forwarded using multi-hop forwarding via wireless channel. In Chapter 6, we design a routing protocol that can increase the chance of delivering the data packets at a small delay without the use of the real-time traffic information. Using a map of an area, the protocol reduces the delivery delay by replicating multiple copies of the same packet at an intersection. The replicated packets are then forwarded to the road segments that have directions towards the destination vehicle. The result shows that by replicating the same packets in a controlled manner, the protocol is able to reduce end-to-end delivery delay compare to the forwarding strategy that does not use replication.

The biggest challenge in a VANET is its mobility and dynamic topology. It is common for a VANET to experience extremely high network density and low network density. This thesis has addressed the main issue of a carry and forward based routing protocol which is end-to-end delivery delay. By studying the carry and forward process in both low and high density networks, we are able to design a suitable approach in broadcasting periodic beacon packets as well as forwarding data packets.

6.2. Future Research Directions

There are a number of promising directions for further research in VANETs. This section presents some of future works in the context of this thesis.

(i) The analytical framework for end-to-end delay presented in this thesis is derived for a unidirectional highway. The model can be extended to two directional highways and streets with intersections. In addition, in the model, packet transmissions are assumed to be forward transmission. Therefore, additional extensions for backward transmission for both one and two directional highways can be considered. In the proposed analytical model, vehicles' speed is based on a uniform distribution and assumed to remain unchanged once the vehicles enter the highway. It would be interesting to see the derivation of our model if we assume the vehicles' speed changes when moving on a highway.

(ii) Prototype verification for the adaptive beaconing system and the proactive multi-copy routing protocol can be considered extremely difficult and expensive. However, a protocol emulator using microcontrollers or field-programmable gate array (FPGA) chips can be used to imitate vehicles in a network and verify the performance of the protocols.

(iii) The adaptive beaconing system can be developed using a neural network model and compare its performance with fuzzy logic based beaconing system.

(iv) The proactive multi-copy routing protocol presented in this thesis is investigated using an urban grid model. The investigation can be extended to a realistic urban scenario since intersections are a common occurrence in realistic urban scenarios. The comparisons of the proactive multi-copy routing protocol can also include other VANET routing protocols available in the literature.

(v) The proactive multi-copy routing protocol can also be designed to be sensitive to the network condition. In a sparse network, instead of forwarding to selected relays at different road segments, the protocol should use a broadcast mechanism instead to further increase the chance reaching the destination. The protocol should be able to switch back to multiple unicast transmissions in a well-connected network.

Bibliography

[1] F. Li and Y. Wang, "Routing in vehicular ad hoc networks: A survey," *IEEE Vehicular Technology Magazine*, vol. 2, no. 2, pp. 12–22, 2007.

[2] H. Hartenstein and K. Laberteaux, *VANET: Vehicular Applications and Internetworking Technologies.* Wiley Chichester, 2010.

[3] Y. Toor, P. Muhlethaler, and A. Laouiti, "Vehicle ad hoc networks: applications and related technical issues," *IEEE Communications Surveys Tutorials*, vol. 10, no. 3, pp. 74–88, 2008.

[4] J. Jakubiak and Y. Koucheryavy, "State of the Art and Research Challenges for VANETs," in *5th IEEE Consumer Communications and Networking Conference*, 2008, pp. 912–916.

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

[6] R. Naja, "A Survey of Communications for Intelligent Transportation Systems," in *Wireless Vehicular Networks for Car Collision Avoidance*, R. Naja, Ed. Springer New York, 2013, pp. 3–35.

[7] M. Sichitiu and M. Kihl, "Inter-vehicle communication systems: a survey," *IEEE Communications Surveys Tutorials*, vol. 10, no. 2, pp. 88–105, 2008.

[8] S. Al-Sultan, M. M. Al-Doori, A. H. Al-Bayatti, and H. Zedan, "A comprehensive survey on vehicular Ad Hoc network," *Journal of Network and Computer Applications*, vol. 37, no. 0, pp. 380 – 392, 2014.

[9] P. Papadimitratos, A. La Fortelle, K. Evenssen, R. Brignolo, and S. Cosenza, "Vehicular communication systems: Enabling technologies, applications, and future outlook on intelligent transportation," *IEEE Communications Magazine*, vol. 47, no. 11, pp. 84–95, 2009.

[10] H. Hartenstein and K. Laberteaux, "A tutorial survey on vehicular ad hoc networks," *IEEE Communications Magazine*, vol. 46, no. 6, pp. 164–171, 2008.

[11] Z. Taysi and A. Yavuz, "Routing Protocols for GeoNet: A Survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 2, pp. 939–954, June 2012.

[12] R. Uzcategui and G. Acosta-Marum, "Wave: A tutorial," *IEEE Communications Magazine*, vol. 47, no. 5, pp. 126–133, 2009.

[13] A. S. C33, “Standard Specification for Telecommunications and Information Exchange Between Roadside and Vehicle Systems - 5 GHz Band Dedicated Short Range Communications (DSRC) Medium Access Control (MAC) and Physical Layer (PHY) Specifications,” 2003.

[14] Z. Mo, H. Zhu, K. Makki, and N. Pissinou, “MURU: A Multi-Hop Routing Protocol for Urban Vehicular Ad Hoc Networks,” in *2006 Third Annual International Conference on Mobile and Ubiquitous Systems: Networking Services*, 2006, pp. 1–8.

[15] H. Menouar, M. Lenardi, and F. Filali, “Movement Prediction-Based Routing (MOPR) Concept for Position-Based Routing in Vehicular Networks,” in *2007 IEEE 66th Vehicular Technology Conference, 2007. VTC-2007 Fall*, 2007, pp. 2101–2105.

[16] Z. Jing and C. Guohong, “VADD: Vehicle-Assisted Data Delivery in Vehicular Ad Hoc Networks,” *IEEE Transactions on Vehicular Technology*, vol. 57, no. 3, pp. 1910–1922, 2008.

[17] M. Jerbi, S. M. Senouci, T. Rasheed, and Y. Ghamri-Doudane, “Towards efficient geographic routing in urban vehicular networks,” *IEEE Transactions on Vehicular Technology*, vol. 58, no. 9, pp. 5048–5059, 2009.

[18] M. Zhang and R. Wolff, “Routing protocols for vehicular ad hoc networks in rural areas,” *IEEE Communications Magazine*, vol. 46, no. 11, pp. 126 –131,

November 2008.

[19] R. Schmidt, T. Leinmuller, E. Schoch, F. Kargl, and G. Schafer, "Exploration of adaptive beaconing for efficient intervehicle safety communication," *IEEE Network*, vol. 24, no. 1, pp. 14–19, 2010.

[20] N. Wisitpongphan, F. Bai, P. Mudalige, V. Sadekar, and O. Tonguz, "Routing in Sparse Vehicular Ad Hoc Wireless Networks," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 8, pp. 1538–1556, 2007.

[21] E. Schoch, F. Kargl, and M. Weber, "Communication patterns in VANETs - [topics in automotive networking]," *IEEE Communications Magazine*, vol. 46, no. 11, pp. 119–125, 2008.

[22] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *Second IEEE Workshop on Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA '99*, pp. 90–100.

[23] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing.* Kluwer Academic Publishers, 1996, pp. 153–181.

[24] V. Namboodiri, M. Agarwal, and L. Gao, "A Study on the Feasibility of Mobile Gateways for Vehicular Ad-hoc Networks," New York, NY, USA, pp. 66–75, 2004.

[25] G. Liu, B.-S. Lee, B.-C. Seet, C.-H. Foh, K.-J. Wong, and K.-K. Lee, *A Routing Strategy for Metropolis Vehicular Communications.* Springer Berlin / Heidelberg, 2004, vol. 3090/2004, pp. 134–143.

[26] R. A. Santos, A. Edwards, R. M. Edwards, and N. L. Seed, "Performance evaluation of routing protocols in vehicular ad-hoc networks," *International Journal of Ad Hoc and Ubiquitous Computing.*, vol. 1, no. 1/2, pp. 80–91, 2005.

[27] S. Y. Wang, C. C. Lin, Y. W. Hwang, K. C. Tao, and C. L. Chou, "A practical routing protocol for vehicle-formed mobile ad hoc networks on the roads," in *Intelligent Transportation Systems, 2005. Proceedings. 2005 IEEE*, pp. 161–166.

[28] B. Karp and H. T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in *Proceedings of the 6th annual international conference on Mobile computing and networking*, ser. MobiCom '00. New York, NY, USA: ACM, 2000, pp. 243–254.

[29] C. Lochert, H. Hartenstein, J. Tian, H. Fussler, D. Hermann, and M. Mauve, "A routing strategy for vehicular ad hoc networks in city environments," in *IEEE Intelligent Vehicles Symposium, 2003. Proceedings.*, 2003, pp. 156–161.

[30] H. Fussler, M. Mauve, H. Hartenstein, M. Kasemann, and D. Vollmer, "Location-based routing for vehicular ad-hoc networks," *ACM Mobile Computing and Communications Review (MC2R)*, vol. 7, no. 1, pp. 47–49, 2003.

[31] S. Panichpapiboon and W. Pattara-Atikom, "A review of information dissemination protocols for vehicular ad hoc networks," *IEEE Communications Surveys Tutorials*, vol. 14, no. 3, pp. 784–798, Third 2012.

[32] G. Korkmaz, E. Ekici, F. Özgüner, and U. Özgüner, "Urban multi-hop broadcast protocol for inter-vehicle communication systems," in *Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks*, ser. VANET '04. New York, NY, USA: ACM, 2004, pp. 76–85.

[33] A. Vahdat and D. Becker, "Epidemic Routing for Partially-Connected Ad Hoc Networks," Duke University, Tech. Rep., 2000.

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

[35] F. Schmidt-Eisenlohr, M. Torrent-Moreno, J. Mittag, and H. Hartenstein, "Simulation platform for inter-vehicle communications and analysis of periodic information exchange," in *Wireless on Demand Network Systems and Services, 2007. WONS '07. Fourth Annual Conference on*, 2007, pp. 50–58.

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

[37] C. Sommer, O. Tonguz, and F. Dressler, "Adaptive beaconing for delay-sensitive and congestion-aware traffic information systems," in *Vehicular Networking Conference (VNC), 2010 IEEE*, 2010, pp. 1–8.

[38] C. Thaina, K. Nakorn, and K. Rojviboonchai, "A study of adaptive beacon transmission on vehicular ad-hoc networks," in *Communication Technology (ICCT), 2011 IEEE 13th International Conference on*, 2011, pp. 597–602.

[39] A. Boukerche, C. Rezende, and R. Pazzi, "Improving Neighbor Localization in Vehicular Ad hoc Networks to Avoid Overhead from Periodic Messages," in *Global Telecommunications Conference, 2009. IEEE GLOBECOM 2009.*, 2009, pp. 1–6.

[40] W. Hao, R. M. Fujimoto, G. F. Riley, and M. Hunter, "Spatial propagation of information in vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 1, pp. 420–431, 2009.

[41] S. Durrani, X. Zhou, and A. Chandra, "Effect of Vehicle Mobility on Connectivity of Vehicular Ad Hoc Networks," in *2010 IEEE 72nd Vehicular Technology Conference Fall (VTC 2010-Fall)*, Sept 2010, pp. 1–5.

[42] A. Kesting, M. Treiber, and D. Helbing, "Connectivity Statistics of Store-and-Forward Intervehicle Communication," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 1, pp. 172–181, March 2010.

[43] Z. Zhang, G. Mao, and B. D. O. Anderson, "On the Information Propagation Process in Mobile Vehicular Ad Hoc Networks," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 5, pp. 2314–2325, Jun 2011.

[44] A. Agarwal, D. Starobinski, and T. D. C. Little, "Phase Transition of Message Propagation Speed in Delay-Tolerant Vehicular Networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 1, pp. 249–263, March 2012.

[45] E. Baccelli, P. Jacquet, B. Mans, and G. Rodolakis, "Highway Vehicular Delay Tolerant Networks: Information Propagation Speed Properties," *IEEE Transactions on Information Theory*, vol. 58, no. 3, pp. 1743–1756, March 2012.

[46] T. Khalaf and S. W. Kim, "Delay analysis in message ferrying system," in *IEEE International Conference on Electro/Information Technology, 2008. EIT 2008*, 2008, pp. 333–336.

[47] J. Fukuyama, "A delay time analysis for multi-hop V2V communications over a linear vanet," in *2009 IEEE Vehicular Networking Conference (VNC)*, October 2009, pp. 1–7.

[48] A. Abdrabou and W. Zhuang, "Probabilistic Delay Control and Road Side Unit Placement for Vehicular Ad Hoc Networks with Disrupted Connectivity," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 1, pp. 129–139, January 2011.

[49] H. Fussler, M. Mauve, H. Hartenstein, M. Käsemann, and D. Vollmer, "Mobicom poster: Location-based routing for vehicular ad-hoc networks," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 7, no. 1, pp. 47–49, 2003.

[50] S. M. Ross, *Introduction to Probability Models, Tenth Editionn.* Orlando, FL, USA: Academic Press, Inc., 2009.

[51] S. Chatterjee and B. Price, *Regression analysis by example, Fourth Edition.* Wiley-Interscience, 2006.

[52] R. Jain, *The art of computer systems performance analysis - techniques for experimental design, measurement, simulation, and modeling.*, ser. Wiley Professional Computing. Wiley, 1991.

[53] A. Leon-Garcia, *Probability and Random Processes for Electrical Engineering, Second Edition.* Reading, Massachusetts: Addison-Wesley, 1994.

[54] T. Issariyakul and E. Hossain, *Introduction to Network Simulator NS2*, 1st ed. Springer Publishing Company, Incorporated, 2008.

[55] "The Network Simulator NS-2," http://www.isi.edu/nsnam/ns/.

[56] D. Rossi, R. Fracchia, and M. Meo, "VANETs: Why Use Beaconing at All ?" in *IEEE International Conference on Communications, 2008. ICC '08*, 2008, pp. 2745–2751.

[57] M. Negnevitsky, *Artificial Intelligence: A Guide To Intelligent Systems*. ADDISON WESLEY Publishing Company Incorporated, 2005.

[58] T. J. Ross, *Fuzzy Logic with Engineering Applications*. John Wiley & Sons, Ltd, 2010.

[59] J. Mendel, "Fuzzy logic systems for engineering: a tutorial," *Proceedings of the IEEE*, vol. 83, no. 3, pp. 345–377, 1995.

[60] S. Bhattacharyya and P. Dutta, *Fuzzy Logic: Concepts, System Design, and Applications to Industrial Informatics*. IGI Global, 2012.

[61] B. S. Kerner, *The Physics of Traffic: Empirical Freeway Pattern Features, Engineering Applications, and Theory (Understanding Complex Systems)*, 2004, vol. 26.

[62] R. P. Roess, E. S. Prassas, and W. R. McShane, *Traffic Engineering (4th Edition)*. Prentice Hall, 2010.

[63] J. Härri, F. Filali, C. Bonnet, and M. Fiore, "VanetMobiSim: generating realistic mobility patterns for VANETs," in *Proceedings of the 3rd international workshop on Vehicular ad hoc networks*, ser. VANET '06. New York, NY, USA: ACM, 2006, pp. 96–97.

[64] M. M. Haklay and P. Weber, "OpenStreetMap: User-Generated Street Maps," *IEEE Pervasive Computing*, vol. 7, no. 4, pp. 12–18, Oct. 2008.

[65] H. Moustafa and Y. Zhang, *Vehicular Networks: Techniques, Standards, and Applications*, 1st ed. Boston, MA, USA: Auerbach Publications, 2009.

[66] Y. Wang and F. Li, *Vehicular Ad Hoc Networks.* Springer London, 2009, pp. 1–23.

[67] D. Reichardt, M. Miglietta, L. Moretti, P. Morsink, and W. Schulz, "CarTALK 2000: safe and comfortable driving based upon inter-vehicle-communication," in *IEEE Intelligent Vehicle Symposium, 2002*, vol. 2, pp. 545–5–50 vol.2.

[68] R. K. Schmidt, B. Kloiber, F. Schuttler, and T. Strang, "Degradation of communication range in VANETs caused by interference 2.0 - real-world experiment," pp. 176–188, 2011.

[69] R. Schmidt, T. Leinmuller, E. Schoch, F. Kargl, and G. Schafer, "Exploration of adaptive beaconing for efficient intervehicle safety communication," *IEEE Network*, vol. 24, no. 1, pp. 14–19, 2010.

[70] B. Seet, G. Liu, B. sung Lee, C. heng Foh, and K. kee Lee, "A-STAR: A Mobile Ad hoc Routing Strategy for Metropolis Vehicular Communications," in *NETWORKING 2004. Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications*, ser. Lecture Notes in Computer Science, vol. 3042. Springer Berlin/Heidelberg, 2004, pp. 989–999.

[71] E. Van de Velde and C. Blondia, "Adaptive REACT protocol for Emergency Applications in Vehicular Networks," in *32nd IEEE Conference on Local Computer Networks, 2007. LCN 2007*, 2007, pp. 613–619.

[72] V. Naumov and T. Gross, "Connectivity-Aware Routing (CAR) in Vehicular Ad-hoc Networks," in *26th IEEE International Conference on Computer Communications*, 2007, pp. 1919–1927.

[73] C. Sommer, O. Tonguz, and F. Dressler, "Traffic information systems: efficient message dissemination via adaptive beaconing," *IEEE Communications Magazine*, vol. 49, no. 5, pp. 173–179, 2011.

[74] H. T. Nguyen and E. A. Walker, *A First Course in Fuzzy Logic, Third Edition.* Chapman & Hall/CRC, 2005.

[75] C. Wu, S. Ohzahata, and T. Kato, "Fuzzy logic based multi-hop broadcast for high-density vehicular ad hoc networks," in *IEEE Vehicular Networking Conference (VNC), 2010*, 2010, pp. 17–24.

[76] C. Lin, F. Dong, and K. Hirota, "Fuzzy Road Situation Model Optimization Routing (FRSMOR) in Vehicular Ad-hoc Network (VANET)," in *Joint 6th International Conference on Soft Computing and Intelligent Systems (SCIS) and 13th International Symposium on Advanced Intelligent Systems (ISIS), 2012*, 2012, pp. 532–537.

[77] J. Heidemann, N. Bulusu, J. Elson, C. Intanagonwiwat, K. chan Lan, Y. Xu, W. Ye, D. Estrin, and R. Govindan, "Effects of detail in wireless network simulation," in *Proceedings of the SCS Multiconference on Distributed Simulation*, 2001, pp. 3–11.

[78] N. Jianwei, L. Chang, C. Canfeng, and M. Jian, "Adaptive copy and spread data dissemination in vehicular ad-hoc networks," in *Communications Technology and Applications, 2009. ICCTA '09. IEEE International Conference on*, 2009, pp. 934–939.

[79] Y.-W. Lin, Y.-S. Chen, and S.-L. Lee, "Routing Protocols in Vehicular Ad Hoc Networks: A Survey and Future Perspectives," *Journal of Information Science and Engineering*, vol. 26, no. 3, pp. 913–932, 2010.

[80] G. Karagiannis, O. Altintas, E. Ekici, G. Heijenk, B. Jarupan, K. Lin, and T. Weil, "Vehicular Networking: A Survey and Tutorial on Requirements, Architectures, Challenges, Standards and Solutions," *IEEE Communications Surveys Tutorials*, vol. 13, no. 4, pp. 584–616, Fourth 2011.

[81] S. M. Bilal, C. J. Bernardos, and C. Guerrero, "Position-based routing in vehicular networks: A survey," *Journal of Network and Computer Applications*, vol. 36, no. 2, pp. 685 – 697, 2013.

[82] X. Huang and Y. Fang, "Performance Study of Node-Disjoint Multipath Routing in Vehicular Ad Hoc Networks," *IEEE Transactions on Vehicular Technol-*

ogy, vol. 58, no. 4, pp. 1942–1950, May 2009.

[83] S. Dhurandher, S. Misra, M. Obaidat, M. Gupta, K. Diwakar, and P. Gupta, “Efficient angular routing protocol for inter-vehicular communication in vehicular ad hoc networks,” *IET Communications*, vol. 4, no. 7, pp. 826–836, April 2010.

[84] Y. Ding and L. Xiao, “SADV: Static-Node-Assisted Adaptive Data Dissemination in Vehicular Networks,” *IEEE Transactions on Vehicular Technology*, vol. 59, no. 5, pp. 2445–2455, Jun 2010.

[85] M. Gramaglia, P. Serrano, J. Hernandez, M. Calderon, and C. J. Bernardos, “New insights from the analysis of free flow vehicular traffic in highways,” in *2011 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, June 2011, pp. 1–9.

[86] S. Panichpapiboon and W. Pattara-Atikom, “Exploiting Wireless Communication in Vehicle Density Estimation,” *Vehicular Technology, IEEE Transactions on*, vol. 60, no. 6, pp. 2742–2751, July 2011.

[87] J. Jeong, S. Guo, Y. Gu, T. He, and D.-C. Du, “Trajectory-Based Data Forwarding for Light-Traffic Vehicular Ad Hoc Networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 5, pp. 743–757, May 2011.

[88] G. Yan and S. Olariu, "A Probabilistic Analysis of Link Duration in Vehicular Ad Hoc Networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1227–1236, Dec 2011.

[89] Q. Wang, J. Hu, and J. Zhang, "Performance evaluation of information propagation in vehicular ad hoc network," *IET Intelligent Transport Systems*, vol. 6, no. 2, pp. 187–196, June 2012.

[90] J.-J. Chang, Y.-H. Li, W. Liao, and I.-C. Chang, "Intersection-based routing for urban vehicular communications with traffic-light considerations," *IEEE Wireless Communications*, vol. 19, no. 1, pp. 82–88, February 2012.

[91] M. Khabazian, M. Mehmet-Ali, and S. Aissa, "Analysis of Continuous Communication Availability in Vehicular Ad Hoc Networks," *IEEE Systems Journal*, vol. 7, no. 1, pp. 137–150, March 2013.

[92] K. Rostamzadeh and S. Gopalakrishnan, "Analysis of Message Dissemination in Vehicular Networks," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 8, pp. 3974–3982, Oct 2013.

[93] C. Chen, X. Du, Q. Pei, and Y. Jin, "Connectivity Analysis for Free-Flow Traffic in VANETs: A Statistical Approach," *International Journal of Distributed Sensor Networks*, August 2013.

[94] Y. Cao, Z. Sun, H. Cruickshank, and F. Yao, “Approach-and-Roam (AaR): A Geographic Routing Scheme for Delay/Disruption Tolerant Networks,” *IEEE Transactions on Vehicular Technology*, vol. 63, no. 1, pp. 266–281, Jan 2014.

[95] Y.-A. Daraghmi, C.-W. Yi, and I. Stojmenovic, “Forwarding methods in data dissemination and routing protocols for vehicular Ad Hoc networks,” *IEEE Network*, vol. 27, no. 6, pp. 74–79, November 2013.

[96] P. Sermpezis, G. Koltsidas, and F.-N. Pavlidou, “Investigating a Junction-Based Multipath Source Routing Algorithm for VANETs,” *IEEE Communications Letters*, vol. 17, no. 3, pp. 600–603, March 2013.

[97] Y.-S. Chen, C.-S. Hsu, and Y.-T. Jiang, “A delay-bounded routing protocol for vehicular ad hoc networks with traffic lights,” *Wireless Communications and Mobile Computing*, 2013.

[98] S. Olariu and M. C. Weigle, *Vehicular Networks: From Theory to Practice*, 1st ed. Chapman & Hall/CRC, 2009.

[99] S. Misra, I. Woungang, and S. C. Misra, *Guide to Wireless Ad Hoc Networks*, 1st ed. Springer, 2009.

[100] W. Viriyasitavat, O. Tonguz, and F. Bai, “UV-CAST: an urban vehicular broadcast protocol,” *IEEE Communications Magazine*, vol. 49, no. 11, pp. 116–124, November 2011.

[101] B. Yu, C.-Z. Xu, and M. Guo, "Adaptive Forwarding Delay Control for VANET Data Aggregation," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 1, pp. 11–18, Jan 2012.

[102] M. Eiza and Q. Ni, "An Evolving Graph-Based Reliable Routing Scheme for VANETs," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 4, pp. 1493–1504, May 2013.

Appendix A

Tables of Confidence Intervals

A.1. Chapter 4 Simulation Results with 95% Confidence Interval

Adaptive Beaconing System

Table A.1: End-to-end delay

Number of vehicles	Mean	CI Lower Limit	CI Upper Limit
25	5.9066	5.7042	6.1089
50	3.1087	2.9553	3.2621
75	1.4723	1.4085	1.5361
100	1.1226	1.0165	1.2287
125	0.8539	0.7626	0.9452
150	0.7392	0.6606	0.8178
175	0.5245	0.4723	0.5768
200	0.1808	0.1670	0.1946

Table A.2: Packet Delivery Ratio

Number of vehicles	Mean	CI Lower Limit	CI Upper Limit
25	0.5429	0.5269	0.5588
50	0.8508	0.8404	0.8612
75	0.9541	0.9482	0.9599
100	0.9740	0.9704	0.9776
125	0.9920	0.9864	0.9975
150	0.9874	0.9845	0.9903
175	0.9901	0.9868	0.9933
200	0.9935	0.9961	0.9909

Table A.3: Routing Overhead Ratio

Number of vehicles	Mean	CI Lower Limit	CI Upper Limit
25	0.6043	0.6003	0.6084
50	0.5357	0.5340	0.5374
75	0.5836	0.5818	0.5854
100	0.6299	0.6290	0.6309
125	0.6723	0.6701	0.6745
150	0.7082	0.7077	0.7088
175	0.7367	0.7361	0.7374
200	0.7564	0.7558	0.7570

Table A.4: Collision Ratio

Number of vehicles	Mean	CI Lower Limit	CI Upper Limit
25	0.0205	0.0194	0.0215
50	0.0630	0.0609	0.0651
75	0.0724	0.0700	0.0748
100	0.0780	0.0758	0.0802
125	0.0760	0.0741	0.0779
150	0.0817	0.0799	0.0836
175	0.0896	0.0880	0.0911
200	0.1039	0.1018	0.1060

Adaptive Traffic Beacon Protocol

Table A.5: End-to-end delay

Number of vehicles	Mean	CI Lower Limit	CI Upper Limit
25	6.0080	5.5968	6.4192
50	4.0697	3.6829	4.4564
75	2.0077	1.8804	2.1349
100	1.4242	1.2994	1.5490
125	1.3020	1.1929	1.4112
150	1.1851	1.0750	1.2952
175	1.1137	0.9978	1.2296
200	0.6942	0.6325	0.7559

Table A.6: Packet Delivery Ratio

Number of vehicles	Mean	CI Lower Limit	CI Upper Limit
25	0.5501	0.5347	0.5655
50	0.7330	0.6941	0.7719
75	0.7948	0.7751	0.8145
100	0.8302	0.8136	0.8467
125	0.8368	0.8198	0.8537
150	0.8656	0.8495	0.8818
175	0.8846	0.8710	0.8982
200	0.8882	0.8749	0.9016

Table A.7: Routing Overhead Ratio

Number of vehicles	Mean	CI Lower Limit	CI Upper Limit
25	0.9362	0.9360	0.9364
50	0.9613	0.9612	0.9614
75	0.9220	0.9219	0.9221
100	0.9377	0.9376	0.9377
125	0.9486	0.9485	0.9486
150	0.9567	0.9567	0.9568
175	0.9621	0.9621	0.9622
200	0.9661	0.9660	0.9662

Table A.8: Collision Ratio

Number of vehicles	Mean	CI Lower Limit	CI Upper Limit
25	0.0296	0.0293	0.0298
50	0.1014	0.1008	0.1021
75	0.1015	0.1005	0.1025
100	0.1484	0.1474	0.1495
125	0.2202	0.2191	0.2213
150	0.2810	0.2798	0.2822
175	0.3853	0.3838	0.3867
200	0.4698	0.4715	0.4682

A.2. Chapter 5 Simulation Results with 95% Confidence Interval

Proactive Multi-copy (PMC) Routing Protocol

Table A.9: End-to-end delay

Number of vehicles	Mean	CI Lower Limit	CI Upper Limit
50	4.1875	3.9516	4.4234
75	2.9528	2.7423	3.1634
100	1.5888	1.5543	1.6232
125	0.9278	0.8808	0.9747
150	0.5360	0.5033	0.5687
175	0.2145	0.1998	0.2293

Table A.10: Packet Delivery Ratio

Number of vehicles	Mean	CI Lower Limit	CI Upper Limit
50	0.4787	0.4585	0.4990
75	0.6202	0.6030	0.6373
100	0.6904	0.6733	0.7076
125	0.7875	0.7797	0.7953
150	0.8391	0.8268	0.8513
175	0.8789	0.8714	0.8864

Table A.11: Routing Overhead Ratio

Number of vehicles	Mean	CI Lower Limit	CI Upper Limit
50	0.4869	0.4832	0.4907
75	0.5848	0.5800	0.5895
100	0.6469	0.6371	0.6566
125	0.6946	0.6835	0.7057
150	0.7222	0.7149	0.7294
175	0.7434	0.7317	0.7551

Table A.12: Collision Ratio

Number of vehicles	Mean	CI Lower Limit	CI Upper Limit
50	0.0102	0.0100	0.0103
75	0.0312	0.0310	0.0313
100	0.0382	0.0378	0.0386
125	0.0499	0.0496	0.0502
150	0.0754	0.0750	0.0758
175	0.0998	0.0990	0.1006

Single-copy Forwarding Algorithm

Table A.13: End-to-end delay

Number of vehicles	Mean	CI Lower Limit	CI Upper Limit
50	4.6053	4.2686	4.9421
75	3.5830	3.3668	3.7992
100	2.8567	2.7020	3.0114
125	2.0428	1.9822	2.1034
150	1.7426	1.6935	1.7916
175	0.8585	0.8222	0.8948

Table A.14: Packet Delivery Ratio

Number of vehicles	Mean	CI Lower Limit	CI Upper Limit
50	0.4449	0.4264	0.4634
75	0.5259	0.5095	0.5423
100	0.5734	0.5645	0.5823
125	0.6485	0.6312	0.6659
150	0.7583	0.7416	0.7750
175	0.7922	0.7788	0.8057

Table A.15: Routing Overhead Ratio

Number of vehicles	Mean	CI Lower Limit	CI Upper Limit
50	0.4776	0.4774	0.4877
75	0.5853	0.5852	0.5954
100	0.6388	0.6315	0.6461
125	0.6896	0.6829	0.6962
150	0.7385	0.7149	0.7294
175	0.7671	0.7592	0.7751

Table A.16: Collision Ratio

Number of vehicles	Mean	CI Lower Limit	CI Upper Limit
50	0.0140	0.0135	0.0144
75	0.0276	0.0268	0.0284
100	0.0403	0.0394	0.0412
125	0.0526	0.0515	0.0537
150	0.0732	0.0717	0.0747
175	0.1079	0.1059	0.1100

Appendix B

Source Codes Headers

```
* myfuzzy.h
 *
 *  Created on: 2012-12-12
 *      Author: root
 */

#ifndef MYFUZZY_H_
#define MYFUZZY_H_

#include <math.h>
#include <stdio.h>
#include <iostream>
#include <fstream>
#include <vector>
#include <algorithm>
#include <limits>

using namespace std;

#define NUMRULES        75
#define COMBINATION     75
#define INPUT           4
#define OUTPUT          3

struct delayMF
{
        double  dLow;
        double  dMedLow;
        double  dMedium;
        double  dMedHigh;
        double  dhigh;
};

//based on Matlab FIS file
struct delayMFInt
{
```

```
        static const int Low            = 1; //trapmf, [0 0 5 15]
        static const int dMediumLow     = 2; //trimf,  [5 15 25]
        static const int dMedium        = 3; //trimf,  [20 30 40]
        static const int dMediumHigh= 4; //trimf,  [35 15 55]
        static const int dHigh          = 5; //trapmf, [45 55 60 60]
};

struct numNeighborMF
{
        double  nSmall;
        double  nSmallMed;
        double  nMedium;
        double  nHighMed;
        double  nHigh;
};

struct numNeighborMFInt
{
        static const int nSmall         = 1; //trapmf,[0 0 10 20]
        static const int nSmallMed      = 2; //trimf, [10 25 40]
        static const int nMedium        = 3; //trimf, [30 45 60]
        static const int nHighMed       = 4; //trimf, [50 65 80]
        static const int nHigh          = 5; //trimf, [30 45 60]
};

struct speedMF
{
        double  spSlow;
        double  spMedium;
        double  spFast;
};

//based on Matlab FIS file
struct speedMFInt
{
        static const int spSlow         = 1; //trapmf, [0 0 5 13.89]
        static const int spMedium       = 2; //trimf, [5 10 15]
        static const int spFast         = 3; //trapmf, [13.89 25 20
           20]
};

struct intervalMF
{
        double  intrvVHigh;
        double  intrvHigh;
        double  intrvMedHigh;
        double  intrvMediumH;
        double  intrvNotAccept;
        double  intrvMediumL;
        double  intrvMedLow;
        double  intrvLow;
};

struct intrvMF
{
        double a_;
        double b_;
};

class MyFuzzy
{
```

```
private:
        static bool mySortFunction (const std::vector<double>& rowA,
            const std::vector<double>& rowB);

        double fuzzymin (double x, double y, double z);
        double downSlopeTri (double x, double m, double b);
        double upSlopeTri (double x, double m, double a);

        double MFTriangular (double input, double Range_a, double
           Range_m, double Range_b);

        double MFTrapezoid (double input, double Range_a, double
           Range_b, double Range_c, double Range_d);

        double MFTrapezoidLeft (double input, double Range_m, double
            Range_a);

        double MFTrapezoidRight (double input, double Range_m,
           double Range_b);

        double  downSlopeTriMu (double mu_x, double m, double b);

        double  upSlopeTriMu (double mu_x, double m, double a);

        intrvMF MFTriangularMu (double mu_x, double Range_a, double
           Range_m, double Range_b);

        intrvMF MFTrapezoidLeftMu(double mu_x, double a, double b);
        intrvMF MFTrapezoidRightMu(double mu_x, double a, double b);
        double  centroidTriangular(double x1, double x2, double x3);
        double  centroidCenterValue(int rule);

public:
        delayMFInt              dMFInt;
        numNeighborMFInt nMFInt;
        speedMFInt              sMFInt;
        double                  combMFRules[COMBINATION][INPUT];
                                //fuzzy rules matrix results based
                                   on inputs
        double                  MFResult[COMBINATION][OUTPUT];
                                //[fisrules][result][a][b][m]
        double                  OutRules[COMBINATION];

        double                  totalDelay;
        double                  totalSpeed;
        int                             totalNumNb;

        double                  avgDelay;
        double                  avgSpeed;
        int                             avgNumNb;
        int                             numCalculate;

        MyFuzzy(double d, double s, double n, int num);
        ~MyFuzzy() {};

        delayMF                 initializeMFdelay(delayMF dMF_);
        numNeighborMF   initializeMFNeighbor(numNeighborMF nMF_);
        speedMF                 initializeMFSpeed(speedMF sMF_);
        intervalMF              initializeMFInterval(intervalMF iMF_
           );
```

```
        delayMF                evaluateMFdelay (double delayIn,
            delayMF dMF_);

        numNeighborMF   evaluateMFNeighbor(double neighborIn,
            numNeighborMF nMF_);

        speedMF                evaluateMFSpeed (double speedIn,
            speedMF sMF_);

        void                   evaluateMFResults();
                               //evaluate matrix result using
                                   interval MF

        void                   evaluateMFMatrix();
                               //evaluate the output of fuzzy rules
                                    matrix result

        void                   insertMFResults(delayMF dMF_,
            numNeighborMF nMF_, speedMF sMF_);
                               //create fuzzy rules matrix result
                                   based on user inputs

        void                   printMFResults();
                               //printout fuzzy rules matrix result
        void                   InsertFuzzyRulesOutput();
                               //insert output MF into fuzzy rules
                                   matrix
        void                   copyRulesOutput();
                               //copy the result and the fuzzy
                                   rules matrix into a new array
        void                   sortMFRules();
                               //sorted fuzzy rules matrix in
                                   MFResult based on column 0
        double                 defuzzification();
                               // centroid defuzzification process

        double          defuzzTriangular(double x1, int rule);
                               // defuzzification process via
                                   centroid

        double          receiveInput(double delay_, double speed_,
            int numNeighbor);
                               // receive inputs from main code
};


#endif /* MYFUZZY_H_ */
```

```
/*
 * mcr_btable.h
 *
 *  Created on: 2013-06-10
 *      Author: root
 */

#ifndef MCR_BTABLE_H_
#define MCR_BTABLE_H_

#include<packet.h>

#define CURRENT_TIME            Scheduler::instance().clock()

struct mbcast_table
{
        int                             pid_;
        // data packet id being multi-copied
        nsaddr_t                dest_addr_;
        // destination address for the packet
        double                  inter_x_;
        // intersection (pos x) where packet is being multi-copied
        double                  inter_y_;
        // intersection (pos y) where packet is being multi-copied
        double                  time_copied;
        // time when the packet is being copied
        nsaddr_t                bcast_node;
        // node id that broadcast ANC
        mbcast_table*   fnext_;
        // linked list traversal
};

/*
   ======================================================================

 * Broadcast List
 *
    =====================================================================

 */

class mcr_btable
{
        friend class MCR;
        friend class mcr_pktlist;
public:
        mbcast_table* bchead_;
        mbcast_table* bctail_;
        mcr_btable();
        ~mcr_btable();

        bool    btab_empty();
        void    btab_insert(int id, nsaddr_t rxn, double int_x,
           double int_y, double time, nsaddr_t bnode);
        bool    btab_search(int id, nsaddr_t rxn);
        void    btab_delete(int id, nsaddr_t rxn);
};


#endif /* MCR_BTABLE_H_ */
```

```
/*
 * mcr_pktlist.h
 *
 *  Created on: 2013-06-07
 *      Author: root
 */

#ifndef MCR_PKTLIST_H_
#define MCR_PKTLIST_H_

#include <packet.h>
#include <mobilenode.h>

#define BUFFER_SIZE                     10000
#define CURRENT_TIME            Scheduler::instance().clock()

/*
   ===========================================================================
 * Packet List
 *
    ===========================================================================
 */

class mPacket
{
public:
        Packet          *pkt_;
        mPacket         *mnext_;
        int                     id_;
        nsaddr_t        dst_;

        mPacket(int pid_, nsaddr_t rx, Packet *p, mPacket *mlink_ =
           0)
        {
                pkt_    = p;
                mnext_  = mlink_;
                id_             = pid_;
                dst_    = rx;
        }
};

/*
   ===========================================================================
 * class to manage packet list
 *
    ===========================================================================
 */

class mcr_pktlist
{
        friend class MCR;
        friend class mcr_nblist;
        friend class mcr_btable;

public:
        mcr_pktlist();
        ~mcr_pktlist();
```

```
        mPacket          *mp_head_;
        mPacket          *mp_tail_;


        mPacket* mp_head()                 { return mp_head_; }
        mPacket* mp_tail()                 { return mp_tail_; }

        bool             list_empty() const;
                                 // return true if packet buffer is
                                    empty
        void             insertLast(Packet *p);
                                 // save packet at the last slot of
                                    the buffer
        void             deletePkt_id(int id);
                                 // delete packet based on packet id
        bool             is_pkt_inBuffer(int pid);
        Packet*          search_packet(int id);
        void             deleteHead();
        int                      countPacket();

};


#endif /* MCR_PKTLIST_H_ */
```

```
/*
 * mcr_nblist.h
 *
 *  Created on: 2013-06-03
 *      Author: root
 */

#ifndef MCR_NBLIST_H_
#define MCR_NBLIST_H_

#include <mobilenode.h>
#include <packet.h>
#include <vector>
#include <fstream>
#include <iostream>
#include <stdio.h>
#include <god.h>

using namespace std;
#define CURRENT_TIME    Scheduler::instance().clock()

/*
   ===========================================================================

 * Neighbor List - c++ linked list
 *
    ==========================================================================

 */
/*
class mnb_info
{
public:
        nsaddr_t                addr_;          // neighbor's
           address
        double                  posx_;          // neighbor's
           location - position x
        double                  posy_;          // neighbor's
           location - position y
        double                  posz_;          // neighbor's
           location - position z
        double                  speed_;         // neighbor's speed
        double                  velox_;         // neighbor's
           velocity - position x
        double                  veloy_;         // neighbor's
           velocity - position y
        double                  veloz_;         // neighbor's
           velocity - position z
        double                  angle_;         // angle to
           destination node
        int                                     card_dir_;      // cardinal
           direction at intersection
        double                  time_;          // timestamp - when
           the info is saved
        int                                     notupdates_;// how many time
            neighbor info is not update
        // < 3 beacon cycle - >= 3 beacon cycle - info deleted
        mnb_info*               mnb_next;
```

```
        mnb_info(nsaddr_t add, double x, double y, double z, double
            sp_, double vx, double vy, double vz, double ang, int dir
            , double t, int up)
        {
                addr_           = add;
                posx_           = x;
                posy_           = y;
                posz_           = z;
                speed_          = sp_;
                velox_          = vx;
                veloy_          = vy;
                veloz_          = vz;
                angle_          = ang;
                card_dir_       = dir;
                time_           = t;
                notupdates_     = up;
                mnb_next        = NULL;
        }
};

class mcr_nblist
{
        friend class MCR;
        friend class mcr_pktlist;
        friend class mnb_info;
private:
        mnb_info                *mnb_first_, *mnb_last_;
        int     ind_cardinalDirection(double vx, double vy);

public:
        mcr_nblist();
        ~mcr_nblist();
        mcr_routing     mroute_;

        mnb_info* mnb_first()   { return mnb_first_; };
        mnb_info* mnb_last()    { return mnb_last_; };

        void insert_nb(nsaddr_t nid, double nx, double ny, double nz
            , double nspeed, double nvx, double nvy, double nvz);
        // save neighbor information at the beginning of the list
        nsaddr_t search_nb_greedy(nsaddr_t ma, nsaddr_t prevn,
            nsaddr_t rxn);
        // search the closest neighbor via greedy forwarding
        void delete_nbid(nsaddr_t nid);
        // delete neighbor information based on its address
        void delete_allnb();
        // delete all neighbor's information in the table
        int nb_number();
        // calculate number of neighbor in the table
        void where_mynb();
        // determine where my neighbor is going
        // based on cardinal directions:
        // 1 - North(Up), 2 - South(Down), 3 - East(Right), 4 - West
            (Left)
        bool mynb_inList(nsaddr_t nid);
        // return true if the neighbor information already exists in
             the list

};

#endif /* MCR_NBLIST_H_ */
```

```
/*
 * mcr_routing.h
 *
 *  Created on: Aug 14, 2013
 *      Author: root
 */

#ifndef MCR_ROUTING_H_
#define MCR_ROUTING_H_

#include <mobilenode.h>
#include <node.h>
#include <math.h>

class mcr_routing
{
public:
        void mcr_location(nsaddr_t id, double *x, double *y, double
           *z);

        void mcr_velocity(nsaddr_t id, double *vx, double *vy,
           double *vz);

        double mcr_distance(double x1, double y1, double z1, double
           x2, double y2, double z2);

        double mcr_node_angle(double xv1, double yv1, double xv2,
           double yv2);

        double mcr_velo_angle(double xv, double yv);

        double mcr_intersection_angle(double x1, double y1, double
           x2, double y2);

};



#endif /* MCR_ROUTING_H_ */
```

Appendix C

Mobility and Network Simulation Tools

C.1. Simulators for Vehicle Mobility Pattern

VanetMobiSim [63] extends the CANU Mobility Simulation Environment (CanuMobiSim which is a flexible framework for user mobility modeling) and its mobility patterns have been validated against TSIS-CORSIM, a well-known and validated traffic generator. The simulator emphasizes on vehicular mobility and features realistic automotive motion models at both macroscopic and microscopic levels. At the macroscopic level, VanetMobiSim can import maps from the US Census Bureau TIGER database, or randomly generates them using Voronoi tessellation [63]. At the microscopic level, VanetMobiSim able to implement mobility models which able to provide car-to-car and car-to-infrastructure communications. VanetMobiSim's functionalities are decomposed into macro- and micro-mobility features of a vehicular environment to produce realistic urban mobility traces. The macro-mobility part is composed of motion constraints and a traffic generator, while the micro-mobility part controls cars acceleration and deceleration in order to keep a safe inter-distance and avoid accidents and overlapping. The output from VanetMobiSim resulted in a mobility trace file, which can be of any selected format compatible to NS-2, QualNet, GlomoSim or OPNET file for its further use in network simulation.

C.2. Network Simulation Tools

One of the main objectives for any VANETs communication system is to evaluate its benefits and limitations, specifically in terms of its performance. Network simulation is generally used to model computer network configurations before they are implemented in the real world. Using simulation, performance of different network setups can be compared, which makes it possible to recognize and resolve performance problems without having to perform potentially expensive field tests. Network simulations are commonly used in research especially to evaluate the behavior of newly developed network protocols [77].

The ns-2 network simulator is a discrete event-driven simulator, which focuses almost entirely on dynamic nature of communication networks [54]. Development of ns-2 started in 1989, which was then shorthand for Version 2 of The Network Simulator. During that time, it was a fork of the REAL network simulator developed by

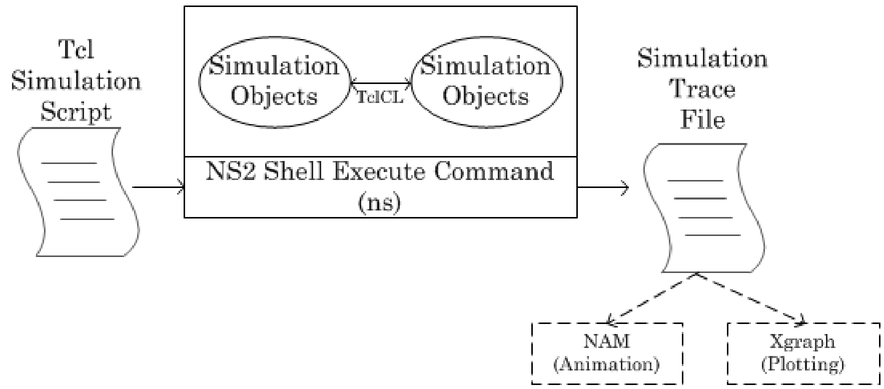


Figure C.1. NS2 Basic Architecture

Cornell University and University of California, which was based on earlier simulators[37]. While there is no IDE or graphical execution environment available for ns-2, the simulator can record detailed packet traces that can be written to disk and, later, visualized using the included NAM (short for Network Animator) tool.

Fig. C.1 shows the basic architecture of NS2, in which provides its users with an executable command ns that takes the name of a Tcl simulation script file as its input argument with a simulation trace file as its output. The simulation core of most ns-2 modules is formed by a wide array of C++ classes. Object Tcl (OTcl), an object oriented dialect of the more popular Tcl language is used to set up, run, and control simulations; as well as for large parts of the module library, and to interface the C++ objects with Tcl simulation script [54].

By using OTcl code to declare module structure, module behavior, and simulation control that can be seamlessly interwoven with the C++ core modules, NS2 becomes an extremely flexible simulator for networking research,. Further flexibility is afforded by the fact that no rigid constraints on event types or module coupling thar are enforced by the simulation kernel. Any ns-2 object in the simulation can schedule an arbitrary object derived from event to be delivered to any other ns-2 object, or an arbitrary OTcl statement to be executed. Therefore, a number of conventions, as illustrated in have proven helpful for structuring simulations [54]:

i. Nodes resemble hosts in the simulation. They contain at least one classifier, termed the node entry point, which will handle packets (i.e., events of type Packet) sent to that node.

ii. Classifiers handle packets in a node, passing each to one or more higher-layer classifiers in the node or delivering them to outbound links. An agent is a special form of classifier that constitutes the end of a packet handling chain, creating new packets or consuming the packets sent to it.

iii. Links resemble channels between nodes in the simulation. They contain at least one connector, termed the link entry point, which will handle packets sent to that link.

iv. Connectors handle packets in a link, passing each packet either to the connector target or to a special drop target.

Even with these agreed-upon conventions, however, the degree of flexibility offered by NS-2 means that great care needs to be exercised if simulations are to be re-used in another context or if efforts from different research groups are to remain

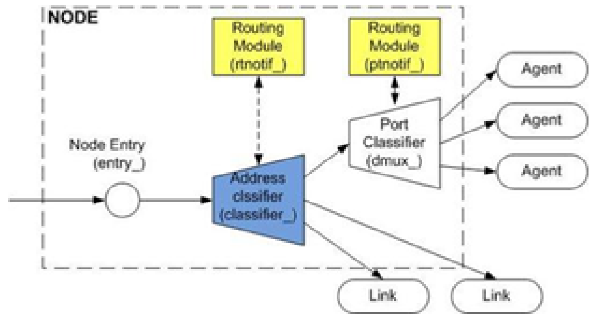


Figure C.2. Common convention of modeling in NS-2

compatible. Moreover, debugging ns-2 simulations requires detailed knowledge of the OTcl components, as their statements are interpreted at run time and, thus, cannot easily be inspected with common debugging tools.

Appendix D

List of Publications

A. Hassan, M. H. Ahmed, and M. A. Rahman, "Estimation of End-to-End Delay for Vehicular Networks under Unidirectional Highway Scenario," submitted to *2014 IEEE International Conference on Communications (ICC 2014)*

A. Hassan, M. H. Ahmed, and M. A. Rahman, "Adaptive Beaconing System based on Fuzzy Logic Approach for Vehicular Network," *2013 IEEE 24rd International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*, 2013.

A. Hassan, M. H. Ahmed, and M. A. Rahman, "An application of vehicular ad hoc wireless network for hybrid electric vehicle," *2011 IEEE International Electric Machines Drives Conference (IEMDC),* pp. 1486–1491, 2011.

A. Hassan, M. H. Ahmed, and M. A. Rahman, "Performance evaluation for multicast transmissions in VANET," *2011 24th Canadian Conference on Electrical and Computer Engineering (CCECE)*, pp. 001105–001108, 2011

A. Hassan, M. H. Ahmed, and M. A. Rahman, "IEEE 802.11p Performance Evaluation in a City Environment," in *IEEE 20th Annual Newfoundland Electrical and Computer Engineering Conference (NECEC 2011),* 2011.
Available: http://necec.engr.mun.ca/ocs2011/viewabstract.php?id=48

A. Hassan, M. H. Ahmed, and M. A. Rahman, "Evaluation of MANET routing protocols for multiple receivers in VANET," in *IEEE 19th Annual Newfoundland Electrical and Computer Engineering Conference (NECEC 2010),* 2010. Available: http://necec.engr.mun.ca/ocs2010/viewabstract.php?id=21