Improving the Reliability and Efficiency of Data Transmission in Vehicular Ad-Hoc Network

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Xiaoxiao Jiang

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
Doctor of Philosophy

David H.C. Du

June, 2015

© Xiaoxiao Jiang 2015
ALL RIGHTS RESERVED

Acknowledgements

First and Foremost, I would like to express my special appreciation to my advisor Professor David Du. He always inspires me and motivates me through my PhD. He is the greatest advisor and shows me how to do research. He has encouraged me to work on interesting problems and keep on finding the best solutions. He also provides insightful discussions and suggestions not only about my research, but also about my career. I am very grateful for his enthusiasm, patience, inspiration and enormous knowledge. Without his guidance my PhD would not have been achievable.

I would like to thank my PhD committee members, Professor Anand Tripathi, Professor Zhi-Li Zhang and Professor Zongxuan Sun. They provided very valuable feedback for my dissertation. Thanks for their time and brilliant comments.

I would like to thank my collaborators and friends over my PhD: Xiang Cao, Clark Jau, Ziqi Fan, Weiping He, Cheng Jin, Kewal Panchputre, Sarah Sharafkandi, Alireza Haghdoost. Thanks for all their supports and helps.

Last but not least, I would like to express my deep and sincere gratitude to my mother Weining Wang and father Li Jiang for all their love. Thanks to their great helps and supports over the years. They raised me with a love of science and always encourage my academic interests.

Abstract

With the increment of population, vehicle transportation problems like traffic congestion and accidents are becoming more severe in our society. To enhance the transportation environment, Intelligent Transportation Systems (ITS) have been deployed to bring smart connectivity to transportation. As a core component of ITS, Vehicular Ad-Hoc Network (VANET) has received considerable attention in information sharing and data delivery services. It is able to offer direct communications between vehicles and between vehicles and roadside units (RSUs). Connected vehicles can send and receive hazard information on the current traffic situation and therefore alert drivers the potential dangerous conditions like icy road or impending collisions. A large number of applications have been proposed based on the advent of DSRC devices which is the defacto communication devices for VANET for both safety and non-safety purposes. In order to support a variety of applications, we aim at improving the reliability and efficiency of data transmission in VANET.

A new two-tier BUS-VANET architecture is proposed which fully integrates traffic infrastructures with public transportation and private vehicles. In this new architecture, the communications of vehicles, not only benefit from the existence of buses, but also consider the effects of using RSUs and Traffic Control Center (TCC). RSUs are used to ensure service coverage while TCC is helpful for locating the destination vehicle quickly. We also investigate the benefits that can be obtained by taking advantage of traffic infrastructures. Comparing to traditional VANET, better performance can be achieved in BUS-VANET with less delivery delay and higher delivery rate.

To overcome the high packet collision probability under high traffic density as the main weakness of IEEE 802.11p protocol, time division multiple access (TDMA) based MAC protocols have been proposed in VANET. However, considering the real two-way traffic, packet collisions still occur due to the contention or multiple vehicles using the same slot while approaching each other called encounter collisions. We proposed two TDMA based MAC protocols: MAT-MAC and PTMAC. MAT-MAC is designed for two-way traffic. It aims to reduce the number of both encounter and contention collisions while maintain high slot utilization even under unbalanced traffic scenario.

PTMAC is a novel predication based MAC protocol. Most of the encounter collisions can be predicted and potentially eliminated before they really happen. It is not only suitable for two-way traffic but also for four-way intersections.

Varieties of applications have been developed taking advantage of Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communications. Unlike wired communication, wireless communication is relatively unreliable, which significantly impacts the service quality. To further improve the transmission performance, it is necessary to investigate and deeply understand the performances of different types of applications over DSRC transmission. A real environment test-bed is developed using DSRC devices to investigate the performance of multi-hop multimedia streaming transmission. We also evaluate the performance of MapReduce applications over a Vehicular Cloud. Impacts of different parameters and appropriate parameter settings are discussed based on VANET transmission features.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

With the increment of population, vehicle transportation problems like traffic congestion and accidents are becoming more severe in our society. Based on 2012 traffic statistic report from US National Highway Traffic Safety Administration (NHTSA) [5], 277 billion dollars a year have been cost on traffic crashes. Over 30,000 people were killed and over 2 million people got injured annually because of the traffic crashes, which is an unacceptable high loss of lives. Enormous amount of time and gasoline are also wasted on the traffic congestion. People in America waste about 4.2 billion hours in traffic congestion every year. As announced by the U.S. Department of Transportation (USDOT) [6], vehicles are also the major sources of carbon dioxide, nitrous oxide and methane. These fumes are very harmful to human health. Additionally, vehicles that are stationary or traveling at reduced speeds due to congestion will emit more fumes. Therefore, it is important to reduce traffic congestion and gas emissions.

Generally, traffic accidents are caused by drivers' inability to assess the traffic and road condition quickly. Drivers may make wrong decisions based on the incomplete and inaccurate information. On the other hand, vehicles are able to avoid traffic congestion if they can receive the traffic information ahead and chose alternate routes. To enhance the transportation safety and address the traffic congestion problem, Intelligent Transportation Systems (ITS) have been deployed to bring smart connectivity to transportation through communication technologies [7]. The USDOT released the Intelligent Transportation Systems Strategic Research Plan 2010-2014 to feature a connected transportation environment among vehicles, infrastructures and passengers' mobile devices.

The focuses of this research plan include: significantly reducing the highway crashes, dynamically adapting the traffic signals based on information provided by vehicles to avoid unnecessary stops, accurately accessing multi-modal transportation system performance, and getting precise travel time information of all options as well as the potential impact. The USDOT recently released new plan 2015-2019 for ITS research and priorities for the second half of the decade. The two key ITS program priorities are realizing connected vehicle implementation and developing automation related technologies.

As a core component of ITS, Vehicular Ad-Hoc Network (VANET) has received considerable attention in information sharing and data delivery services. It is able to offer direct communications among vehicles and between vehicles and roadside units (RSUs). It is aiming to handle the safety, mobility and environment issues. As indicated by the USDOT, 82% of crash scenarios can be avoided. This will prevent thousands of automobile crashes every year. Connected vehicles can send and receive hazard information on the current traffic situation and therefore alert drivers the potential dangerous conditions like icy road or impending collisions. VANET is built over two basic components: Road Side Unit (RSU) and On-Board Unit (OBU). While OBUs are typically installed on vehicles, RSUs are stationary units that are deployed at fixed locations (e.g., road intersections). Both Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communications are supported by OBUs and RSUs to share different kinds of information in VANET. Besides RSUs, Traffic Control Center (TCC) is regarded as another type of traffic infrastructure. It is a trusted agency that collects and maintains current information of vehicles without exposing their locations to others. Through wired or wireless connections with RSUs, TCC is able to collect and process traffic information and disseminate the operation decisions. In this way, it can assist in controlling the traffic [8]. NHTSA announced that it will begin to enable vehicle-to-vehicle (V2V) communication technology for light vehicles and already issued an Advance Notice of Proposed Rulemaking to begin implementation of V2V communications technology in 2014.

Dedicated Short Range Communications (DSRC) is a short to medium range wireless communications protocol that is specifically designed for both Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communications. Its communication range is adjustable from 300m to 1000m. DSRC supports outdoor high vehicle speed, low latency wireless communications and can tolerant extreme weather conditions. In North

America, the US Federal Communications Commission (FCC) has already allocated 75 MHz in the 5.9 GHz frequency for DSRC to be used by Intelligent Transportation Systems (ITS). In Europe, to support road traffic safety applications, 30 MHz has been set for vehicular communications at 5.8755.905 GHz. Meanwhile, a 20-MHz band at 5.8555.875 GHz is assigned for non-safety related applications. In Japan, the allocated frequency is 5.8 GHz.

As shown in Figure 1.1, the DSRC spectrum is divided into seven channels and each with 10MHz bandwidth. One channel is called Control Channel (CCH) that is mainly used for disseminating control information and safety related messages. Multiple channels are defined as Service Channels (SCHs) that used for transmitting non-critical information for both safety and non-safety purposes [9]. With such number of channels, large family of vehicular safety and non-safety applications can be supported. The FCC have designated channel 172 for V2V safety communications for accident avoidance and safety of life applications. Channel 184 is also assigned for high-power and long-distance communications. Each device should alternate between CCH and SCHs. The supported data transmission rate can be varied from 3 Mbps to 27 Mbps. The USDOT is piloting a deployment concept to encourage embedding and retrofitting vehicles with DSRC interfaces to support a variety of applications for both safety and non-safety purposes. Currently, it is constructing a large scale pilot to demonstrate the effectiveness of DSRC with a mix of light, heavy and transit vehicles, with the goal of accelerating the introduction and commercialization of DSRC [10]. Several vendors such as Toyota and GM are also working on this and planning to release new vehicles with DSRC devices.

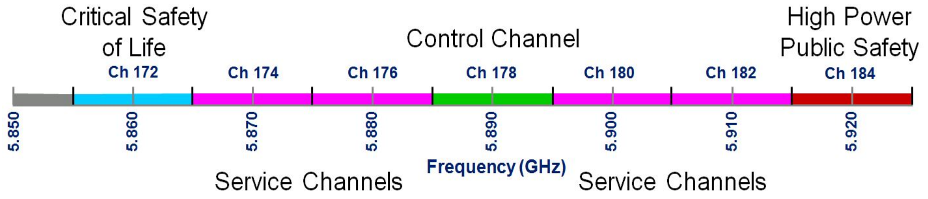


Figure 1.1: US DSRC Channel Allocation [2]

To support V2V and V2I communications, several standards for different layers are utilized in DSRC protocol stack. For PHY and MAC layers, DSRC uses IEEE 802.11p Wireless Access for Vehicular Environments (WAVE) [9] which is a modified version of IEEE 802.11 standard. IEEE 1609 group also defines 1609.4 for channel switching. For network and transport layers, DSRC utilizes 1609.3 for network services and 1609.2 for security services. DSRC also supports internet protocols including IPv6, UDP and TCP. For single-hop messages, the WAVE Short Message Protocol (WSMP) is generally used for saving bandwidth. IPv6 is mostly used for multi-hop packets for its routing capability. For application layer, a set of message formats are defined by the SAE J2736 message set dictionary standard to support different types of VANET applications. 1609.1 specifies the managements required for the operation of the applications. Besides the event driven messages, the basic safety messages (BSMs) and WAVE-Basic Service Advertisements (WSAs) have been developed [11,12]. The BSM contains critical vehicle status information like position and speed. To support most of the applications and make sure the potential dangers can be detected on time, every vehicle is required to broadcast and exchange BSMs periodically, that is, at least every 100ms [13]. WSAs are also needed to be periodically broadcasted by RSUs or vehicles to support non-safety services.

In order to satisfy the research as well as testing purposes for VANET, the USDOT has established a real-world, operational test-bed at Michigan (known as the Southeast Michigan Test-Bed) in 2007 that offers vehicles, infrastructures and equipment [14]. Approximately 55 RSUs were deployed and 27 vehicles equipped with DSRC were employed. It has gone through numerous enhancements, including geographical expansion and technical updates for more general uses. It currently cover 45 miles comprising 75 linear miles of roadways [15]. A Traffic Management Center (TMC) is also deployed in the Southeast Michigan Test-Bed for collecting, storing and processing traffic related data. It is responsible for making decisions of transportation operations like signal control and traffic flow management. In addition to the location in Michigan, test-bed capabilities have been expanded to affiliated test-beds in other states like Virginia, Florida, California, New York, and Arizona. These sites focus on different testing capabilities and they share information, tools and resources with each other.

To improve the transportation environment, a large number of applications have been proposed based on the advent of DSRC devices for both safety and non-safety purposes. Non-safety related applications like traffic congestion detection, advertisement broadcasting, music downloading and many others are developed for improving the driving experience, comfort and traffic efficiency. Meanwhile, the safety applications are designed to save the lives or protect the properties, which include traffic accident altering, road condition monitoring, collision avoidance and so on. They are helpful for warning the driver about the potential hazard and avoiding traffic crashes. Most of the safety related applications are challenging for protocol designing since they are real-time based and have strict low latency and high reliability requirements. Basically, the driver reaction time is about 750ms [16] and most of the alerting message should be received no later than 100ms ahead to give the driver enough time to react.

As a special form of Mobile Ad-Hoc network (MANET), VANET has some unique characteristics. First of all, because of the high mobility feature of vehicle, the VANET topology dynamically and frequently changes over time. Network disconnection may also happen frequently due to vehicle's mobility. This makes the deployments of RSU become necessary for improving the connectivity and service coverage. Additionally, the moving pattern of a vehicle is restricted by roadmap and is potentially predictable through predefined trajectory. The Global Positioning Systems (GPSs) mounted on vehicles are helpful for providing the vehicle current and future trajectories in some level. However, the privacy issue should be considered since drivers may not want to share their trajectories with others. Additionally, the behaviors and decisions of drivers are somehow unpredictable, for example, which route a driver will choose. Furthermore, the time constrain of delivering the message is critical in order to detect dangers on time. Besides the communication capability, computation and storage resources are also provided on nowadays vehicles and RSUs for data processing and storage [17, 18]. Unlike other MANET, power in VANET is not a critical challenge. All these factors should be considered when developing applications and protocols for VANET.

Although VANET brings considerable benefits for improving the transportation environment, there are some research issues that need to be solved. Most of the real-time VANET applications have strict latency constrain that makes efficient data transmission becomes a must. In VANET, packet delivery is usually conducted by *forwarding* through multiple vehicles by multi-hop. If the vehicle holding the packet cannot deliver the data to others due to its limited communication range, it has to *carry* the data until *forwarding* is possible. The most time consuming part of the transmission is the *carrying* phase, since the vehicle speed is much lower than the wireless transmission speed.

Therefore, reducing the probability of packet *carrying* is an efficient way to decrease transmission delay. However, the high vehicle mobility and frequent topology changes in VANET make it a challenge to keep connectivity between vehicles. This potentially increases the packet *carrying* probability.

In addition to the latency problem, VANET is also facing the reliability issue. Although IEEE 802.11p has been approved as the standard medium access control (MAC) protocol, its contention-based nature potentially incur high collision probability and packet losses under high traffic density situation. When more than one vehicle within a communication range is attempting to access the channel concurrently, the transmission collision may happen and none of the packets can be successfully received. For broadcasting, hidden terminal problem cannot be handled well since no RTS/CTS is used because of the broadcast storm. Therefore, to meet the requirements of both safety and non-safety related applications, it is important to decrease the packet collision probability for improving the reliability of data transmission. Besides, for further improving the transmission performance, it is necessary to investigate and deeply understand the performances of different types of applications in VANET over DSRC transmission.

In order to solve the above problems, this dissertation develops new architecture, protocols and innovation solutions that aim at investigating three main research issues. The research issues to be addressed include:

- How to efficiently transmit the data among vehicles and between vehicles and traffic infrastructures (RSUs) with lower latency?
- How to improve the data transmission reliability with fewer packet collisions and higher packet delivery rate under different traffic scenarios?
- What are the network performances of different types of VANET applications? And how to improve their performances?

In the following, Chapter 2 gives an overview of the background knowledge and related works in VANET. Detailed information about applications, routing and MAC protocols in VANET are provided. Chapter 3 explains a new proposed vehicular network taking advantage of public transportation. This aims at reducing the packet delivery delay. Chapter 4 describes two proposed MAC protocols called MAT-MAC and PTMAC for improving the transmission reliability by decreasing the number of packet collisions for both two-way and four-way traffic scenarios. Performance measurements and improvements of real-time multimedia streaming and MapReduce applications over DSRC transmission are investigated in Chapter 5. Conclusion of the dissertation is given in Chapter 6.

Chapter 2

Background

In this chapter, an overview of the basic background knowledge in VANET is provided. Firstly, varieties of applications that have been developed and used in VANET are introduced. We discuss the requirements and benefits of these applications. Then, different types of routing protocols include topology-based and position-based protocols are introduced. Besides, the CSMA/CA based 802.11p and TDMA based MAC protocols are also explained.

2.1 Applications in VANET

A large number of applications have been developed to address real-world traffic problems by providing wide range of information through V2V and V2I communications. VANET applications are basically classified into two types: safety related and non-safety related applications. Safety related applications aim at preventing traffic accidents to save lives and properties. This type of applications is sensitive to latency and has requirement of gathering information of surrounding vehicles and road conditions. Traffic information can be collected by vehicles' or road side sensors and then disseminated to other vehicles who are potentially affected. Non-safety related applications focus on improving the driving comfort, providing entertainment services and enhancing the traffic efficiency.

For safety related applications, the Vehicle Safety Communications Consortium (VSCC) suggests eight high potential applications [19]: traffic signal violation warning, curve speed warning, emergency electronic brake light, pre-crash sensing, cooperative forward collision warning, left turn assistant, lane-change warning, and stop sign movement assistant. There are two types of safety messages that are considered as helpful: event-driven messages and periodic messages [20]. The Event-driven messages are sent when a dangerous condition is detected. For example, if a traffic accident or an urgent brake happens ahead, the detected vehicle or RSU needs to broadcast the warning messages to alert others about the hazardous situation. Such messages have to be quickly disseminated in order to obtain benefits. Meanwhile, the periodic messages are broadcasted by every vehicle periodically. It usually contains the vehicle status information like speed, position and moving direction. Since every vehicle has the knowledge of its neighbor vehicles, unsafe situation can be avoided. Such messages are required to be broadcasted frequently enough in order to provide the most updated information. The VSCC suggests that the periodic messages should be broadcasted at a frequency of at least 10 messages per second.

The non-safety related applications are further categorized into two types according to their purposes. One type of non-safety related applications aims at improving the traffic efficiency and driving experience. The typical example applications that are helpful for reducing the traffic congestions are intelligent navigation and signal control. Base on the information of real-time traffic condition and other vehicles' routing choices, the intelligent navigation application is able to recommend routes with less traffic. Meanwhile, through collecting the vehicle density on each road, the traffic signals can be dynamically adjusted to smooth the traffic flow. To enhance the driving experience, information like weather condition, nearby gas station locations and available parking lots can also be provided. Another type of non-safety related applications has the objective of providing entertainments to drivers and passengers. Services like music downloading, real-time sport streaming, video or audio conference can be supported. Basically, the non-safety related applications have lower priority compare to the safety related applications and have relative loose requirements for transmission latency and reliability.

2.2 Routing Protocols in VANET

To provide efficient and reliable communications, routing in VANET has been studied and many protocols have been proposed. Different types of communications are supported in VANET: unicast communication, multicast/geocast communication and broadcast communication. The unicast is to provide communication from a source node to a destination node through multi-hop. The multicast/geocast is to perform data transmission from a node to a group of destination nodes. These destination nodes usually locate at particular geographic positions based on requirement. The broadcast is used to disseminate information to all the neighbor vehicles.

2.2.1 Unicast Communication

The main goal of designing the unicast routing protocol in VANET is to build a path with shorter delivery delay from a source to a destination. The delay of forwarding between each hop is very small while long latency can be potentially caused by the *carrying* delay. The concept of the packet *carrying*-and-*forwarding* is widely used in MANET to deal with network partitioning and merging. A packet can be forwarded if there are other vehicles nearby. Otherwise, this packet has to be carried until the carrier reaches other vehicles in its communication range. Basically, there are two types of routing protocols that have been designed in VANET: topology-based and position-based protocols. Topology-based protocols like Ad-Hoc On-demand Distance Vector (AODV) [21] and Dynamic Source Routing (DSR) [22] are usually used in an Ad-Hoc Network and they forward packets based on link information that stored in the routing table. To reduce the overhead, they usually do not store the route unless needed. Since VANET has the characteristics of self-organization and self-management, most mobile Ad-Hoc routing protocols are applicable. On the other hand, position-based protocols are aware of vehicles' or RSUs' position information during the routing process. Differs from traditional MANET, VANET has its own features. Considering the high mobility feature of vehicle, position-based protocols have been proved as more suitable for VANET [23]. Each node (Vehicle and RSU) is required to decide its own and neighbors' locations through the assistance of street map or on-board GPS.

One of the well-known position based routing protocols is Greedy Perimeter Stateless

Routing (GPSR). Assuming every vehicle knows all its neighbors' locations, GPSR lets a vehicle selects the one who is the closest to the destination as the next hop. However, such greedy forwarding is not suitable for VANET since it did not consider the vehicle high mobility and may miss some candidates to forward the packet. Therefore, the link connectivity cannot be guaranteed and packet carrying will be incurred. Zhao and Cao proposed a data forwarding scheme and constructed a link delay model called vehicle-assisted data delivery model (VADD [24]). They investigated the data forwarding delay using a traffic stochastic model to achieve lower delivery delay. Based on the traffic density, VADD decides which portion of the street that data will be forwarded and which portion of street that data will be carried. VADD aims at avoiding the packet carrying and transmitting packets through wireless channels as much as possible. If a packet has to be carried by a vehicle through a road, the vehicle with higher speed is chosen firstly. Geographical Opportunistic Routing (GeOpps [25]) is another routing algorithm that exploits the availability of information from the navigation system in order to opportunistically route a data packet to a certain geographic allocation. GeOpps selects vehicles that are likely to carry the information closer to the final destination of the packet.

Unlike traditional MANET, the trajectory of a moving vehicle in VANET is potentially predictable base on its current location and speed. Therefore, [26], [27] and [28] take advantage of vehicle trajectory for designing routing algorithms to shorten the delivery latency. TSF is a trajectory based routing algorithm for selecting an appropriate RSU as the target point in VANET, which is proposed by Jeong et al. in [26]. TSF selects an optimized target point (RSU) for reducing the delay and utilizes vehicle trajectory information to compute the Expected Delivery Delay (EDD). This EDD will be shared with other neighbors and the vehicle with the shortest EDD will be chosen as the next hop. Jeong et al. also proposed the trajectory based statistical forwarding (TBD) for finding the vehicle as the next hop to minimize the delivery delay from a vehicle to a RSU [27]. While [26] paid attention to the transmission from one RSU to another RSU, [27] focused on vehicle to RSU transmission. Xu et al. designed a shared trajectory based data forwarding scheme for V2V transmission in [28] which used the predicted encounter graph. Vehicles have to share their trajectories with others for the encounter time prediction. In this way, vehicles can select the next hop with less latency based on the estimated encounter points. However, these trajectory based routing algorithms are potentially hard to be realized in the real world since people may not want to share their trajectories with others considering the privacy issue.

2.2.2 Multicast/Geocast Communication

The objective of geocast routing is to send packets from a source node to all other nodes within a geographical region according to their interests. One example application that needs such geocasting is crash warning. When a crash happens, the warning message should be broadcasted to the group of vehicles who will be potentially affected. Most geocast routing protocols are developed for restricting the message overhead and network congestion. One key factor that should be considered for designing the geocast protocol is the target vehicle groups. Bachir and Benslimane have proposed a geocast protocol called Inter-Vehicle Geocast (IVG) in [29]. The alarm messages are sent to vehicles in the risk areas. IVG determines such areas according to the driving directions, speeds and the positions of vehicles.

Another key factor of designing the geocast routing protocol is how to reduce the number of rebroadcasts and network overhead. Briesemeister et al. proposed a geocast scheme in [30]. When a vehicle receives a packet, instead of rebroadcasting it immediately, this vehicle holds the packet for a while. The holding time depends on the distance between the holding vehicle and the sending vehicle. When the holding time runs out, if this vehicle did not receive the same packet from others, it will rebroadcast the message. They also applied the maximal-hop-number to limit the range of the flooding.

The designing of geocast routing also needs to consider the vehicle high mobility. Another classical geocast routing protocol has been proposed in [31] to reduce the packet losses and network overhead. The basic idea is to add cache to the routing layer to store current unroutable packets. If the caching vehicle gets notification about any newly discovered neighbors or changes in neighbors' positions, the cached message is likely to be transmitted. The authors also proposed a modified distance aware neighborhood strategy that takes neighbor changes into account. The closest vehicle to the destination inside a range that smaller than the communication range is selected. In this way, the delivery delay can be shortened.

2.2.3 Broadcast Communication

Broadcast is widely used in VANET for disseminating basic vehicle status information and event-driven messages through both single-hop and multi-hop. The simplest method to implement broadcast is flooding that every vehicle just rebroadcasts the received messages to all its neighbors except the one who sent it the messages. The performance of such scheme will significantly degrade and a large number of packet collisions will happen when the vehicle density becomes high. Therefore, some protocols have been proposed for achieving efficient broadcast with fewer collisions and lower network overhead. Selective forwarding is usually used to avoid network traffic congestion by reducing the number of vehicles for rebroadcasting. Durresi et al. designed a broadcast protocol called BROADCOMM in [32] for handling the emergency on highway environment. They divided the highway into several virtual cells and separated vehicles into two levels. The first level contains all the vehicles in a cell while the second level is composed of a few vehicles that locate close to the geographical center of a cell and they are called cell reflectors. They perform as cluster heads and will be responsible for forwarding the packets received from its cell members or vehicles from the neighbor cells. This protocol outperforms flooding but only work for simple highway scenario.

Another protocol named Urban Multi-hop Broadcast (UMB) [33] is designed for handling the packet collisions and hidden terminal problem for broadcasting in urban area. The basic idea of UMB is choosing the furthest node in the communication range and broadcast direction to finish the packet forwarding without using the network topology information. When there is an intersection in the packet disseminate path, new broadcasts to other directions will be initiated to inform vehicles at all directions. Such new broadcasts will be completed by repeaters which are suggested to be installed at each intersection.

2.3 MAC Protocols in VANET

2.3.1 802.11p MAC Protocol

As one part in the Wireless Access in Vehicular Environment (WAVE) protocol stack developed by IEEE, 802.11p has already been approved as the standard MAC protocol in VANET. It employs contention-based Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) with Enhanced Distributed Channel Access (EDCA). For CSMA/CA in 802.11p MAC protocol, if the channel is sensed as idle for an arbitration inter frame space (AIFS), the vehicle starts the transmission directly. Otherwise, this vehicle needs to randomly pick up a back-off value from the interval [0, CW] and then starts a countdown procedure. CW stands for the Contention Window which is initially set as CW_{min}. The back-off value will be decreased when the channel is free and the transmission will begin when the back-off value reaches 0. The AIFS is calculated based on the Short Interframe Space (SIFS) and AIFS-number (AIFSN) as Equation 2.1.

$$AIFS(AC_i) = AIFSN(AC_i) \times Slot + SIFS \tag{2.1}$$

Table 2.1: Default Parameter Settings for Different Applications in 802.11p [1]

AC	CW_{min}	CW_{max}	$AIFSN$
AC0	15	1023	9
AC1	15	1023	6
AC2	7	15	3
AC3	3	7	2

If multiple vehicles within two-hop range are trying to access the channel simultaneously, a collision will happen and none of the packets can be successfully received. In this case, vehicles have to re-compete for the channel to resend the packets. A sender of unicasting needs to wait for an acknowledgement (ACK) from the recipient. If the ACK is not received with in a period of time, the packet will be retransmitted. An exponential back-off scheme which extends the CW size will be applied for the retransmission until CW_{max} is reached. In this way, the probability of retransmission contention collision can be reduced. After a successful transmission or the maximum number of transmission attempts is reached, the value of CW will be set back to CW_{min}. Based on different latency requirements and critical levels, EDCA classifies packets into four Access Classes (ACs) with different priorities. Different ACs have different AIFSN and CW values to ensure packets with higher priority can access the channel earlier. The default EDCA parameters of 802.11p are shown in Table 2.1 and the overall architecture of DSRC MAC

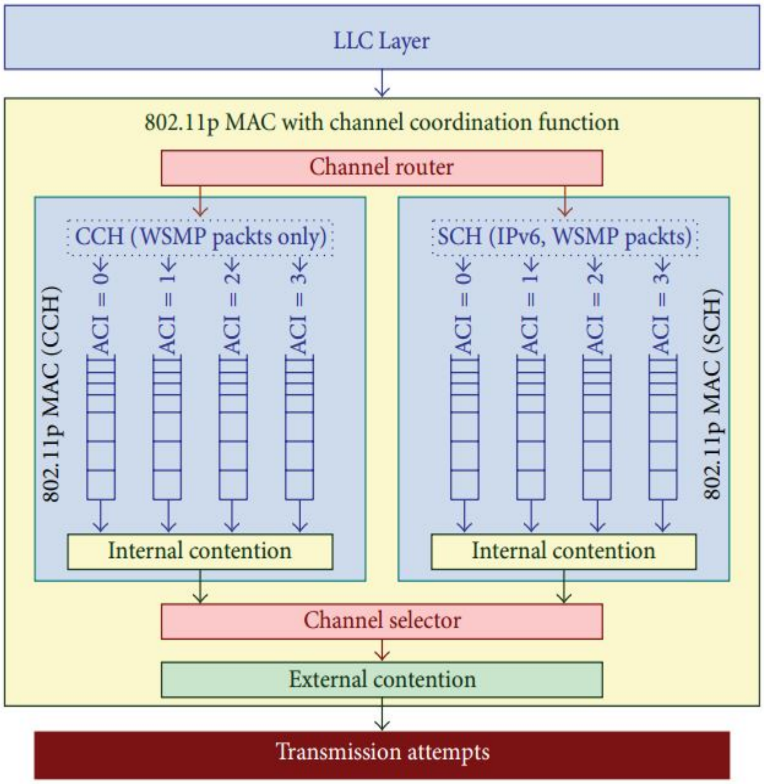


Figure 2.1: Overall Architecture of DSRC MAC [3]

is illustrated in Figure 2.1. Each AC queue is implemented on a per-channel basis.

However, as a contention-based scheme, 802.11p has a serious issue of potential high packet collision probability under high traffic density scenario. It has the drawback of potential unbounded channel access delay [34]. If a vehicle has multiple packets, it has to contend for multiple times. Besides, although a small CW size set by 802.11p allows high priority packets to be transmitted with less delay, it may introduce more transmission collisions within the same class. No exponential back-off scheme can be used for broadcasting, which also causes the high probability of a packet collision [35]. Furthermore, 802.11p is very vulnerable to hidden terminal problem. Considering the broadcast storm, it cannot use RTS/CTS mechanism for packet broadcasting [35]. In this case, packet collisions cannot even be detected. Therefore, 802.11p is potentially not suitable for real-time traffic applications.

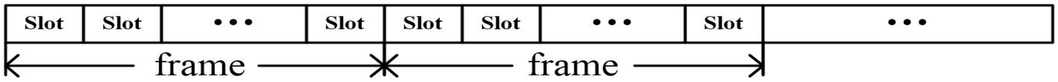


Figure 2.2: Time Frame and Slot Structure in TDMA

2.3.2 TDMA based MAC Protocol

To overcome the shortcomings of IEEE 802.11p, time division multiple access (TDMA) based MAC protocols have been proposed to facilitate efficient transmission in VANET [9, 36]. Basically, there are two types of TDMA based MAC protocols: distributed TDMA and centralized TDMA. Each node manages its time slot by itself in distributed TDMA schemes while all the time slots are allocated by a central node such as a RSU or a cluster head in a centralized TDMA [13, 17, 34]. Considering the high mobility nature of vehicles and the potential lacking of RSUs in highway scenario, distributed TDMA based MAC protocols provide more flexibility on slot management. Some previous studies provided the performance comparison between 802.11p and TDMA based MAC protocols [4, 37, 38]. They have shown that TDMA performs more reliable and robust when comparing to IEEE 802.11p.

The most basic distributed TDMA based MAC protocol has been proposed in [39] using time slotted structure. The time is partitioned into repeated frames and each frame is composed of fixed number of slots as shown in Figure 2.2. Each vehicle selects a specific available time slot to transmit data. If successful, it keeps on using the same slot at subsequent frames until a collision occurs or the slot is no longer needed. Every vehicle is required to broadcast a Frame Information (FI) at every frame which contains the slot information about all its one-hop neighbors. In this way, vehicles can get their neighbors' slot information within two-hop range. A newly joining vehicle first listens to the channel for a frame and then selects an available time slot to transmit data at next frame. The newly joining vehicles are those who have not reserved their slots and intend to transmit packets. Therefore, the probability of transmission collisions is reduced and each vehicle is guaranteed to access the channel at least once in each frame if a reservation is successfully made. There is no need for each individual packet to compete for the channel.

Chapter 3

Public Transportation based Vehicular Network Architecture

In this chapter, we address the issue of how to efficiently transmit data among vehicles and between vehicles and traffic infrastructures (RSUs) with lower latency. Most of the real-time VANET applications have strict delay constrain. Therefore, we aim at designing a new VANET architecture called BUS-VANET that fully integrates buses and traffic infrastructures for providing better data delivery service. RSUs can be used to compensate for the shortage of buses and ensure the service coverage. Traffic Control Center (TCC) is helpful for locating the destination vehicle quickly. Comparing with traditional VANET architecture, BUS-VANET has better performance with higher packet delivery rate and shorter delivery delay.

3.1 Introduction

In order to collect current traffic condition and deliver traffic control information to vehicles, Intelligent Transportations System (ITS) has to effectively use VANET and traffic infrastructures consisting of Road Side Units (RSUs) and Traffic Control Center (TCC). Road Side Units (RSUs) are used for collecting traffic statistical data, temporarily buffering data, identifying vehicle current locations and shortening the communication delay [40]. Traffic Control Center (TCC) is a trusted agency that collects and maintains current information of vehicles without exposing their locations to others.

Based on the collected traffic and vehicle information, TCC can assist in controlling the traffic [41]. This type of traffic infrastructure has been built to support many safety and non-safety related applications like accident alert, traffic congestion detection and music downloading [42].

Some VANET architectures have been proposed based on the predictable routes and schedules of buses in order to improve the transmission performance. However, none of them take advantage of the existing traffic infrastructures. The service coverage by buses cannot be guaranteed and there is no efficient destination location identification method that has been developed in traditional VANET. Therefore, we aim at designing a new VANET architecture that fully integrates buses and traffic infrastructures for providing better data delivery service. RSUs can be used to compensate for the shortage of buses to ensure the service coverage while TCC can be helpful for locating the destination location quickly.

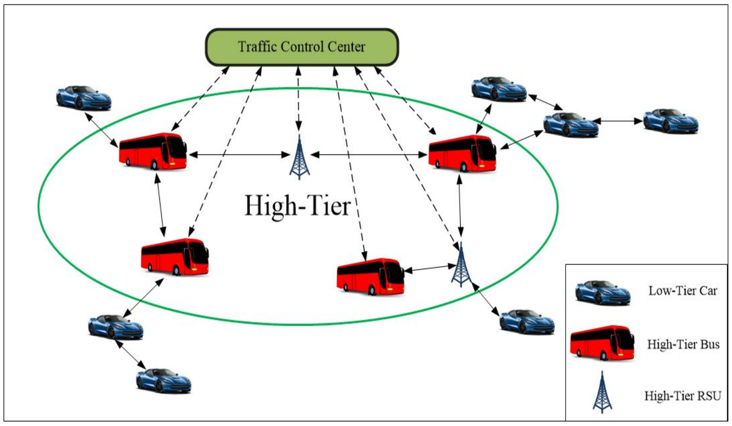


Figure 3.1: The Proposed BUS-VANET Architecture

Nowadays, Wi-Fi service becomes more and more common for passengers to access the Internet on public transportation. Several states like California, Texas and Nevada are providing free Wi-Fi service on city rails or buses [43]. A system called WiRover has been running on some buses in Wisconsin since April 2010. It provides Wi-Fi hotspots, through which passengers can connect to the Internet [44]. Thus, we can assume that buses and RSUs are able to carry the interface with longer communication range like Wi-Fi or WiMAX. Based on this assumption, we propose a new two-tier BUS-VANET architecture integrated with traffic infrastructures. Buses and RSUs are high-tier nodes that constitute the mobile backbone for data delivery, while the low-tier is composed of common vehicles. All the high-tier nodes form a connected topology, directly or indirectly connect to the TCC because of their Internet accessing capability. RSUs and buses have two types of interfaces: DSRC and Wi-Fi. DSRC is the communication protocol used in VANET and its communication range is from 300m to 1000m. Wi-Fi has longer range around 5km. The low-tier nodes are just equipped with DSRC. The route and schedule of every bus and the location of each RSU are shared with all the vehicles. The basic architecture of our BUS-VANET is shown in Figure 3.1.

Comparing with traditional VANET, BUS-VANET provides more stable and efficient data delivery service. Longer communication range of buses and RSUs can decrease the probability of *carrying* and reduce the delivery delay. The destination localization in our proposed architecture is completed quickly by the corporation of buses, RSUs and TCC. Each vehicle will register with a nearby bus or a RSU. The information collected by a bus or RSU will be reported to TCC and the destination vehicle can be found quickly. Besides, since a private vehicle does not share its trajectory with other vehicles, its privacy is protected. Our main contributions can be summarized as follows:

- We propose a new BUS-VANET architecture which fully integrates traffic infrastructures with buses and vehicles. We investigate how to take advantages of RSUs and TCC that already been provided by ITS to improve the VANET performance.
- Based on the proposed BUS-VANET, we use the registration technology to improve the transmission performance and provide a new method of selecting registration node to reduce the number of switches from common vehicles to high-tier nodes. We also proposed a new scheme for identifying the destination location more efficiently.
- Through measuring our BUS-VANET architecture and comparing it with other two VANET architectures in [45] and [46], we show that BUS-VANET has better performance with higher packet delivery rate and shorter delivery delay.

In the following, Section 3.2 provides related work. Section 3.3 shows our proposed BUS-VANET architecture and explains the data delivery in our BUS-VANET. Section

3.4 discusses two important research issues in BUS-VANET and the potential solutions. We show the simulation results in Section 3.5 and compare the proposed BUS-VANET with other VANET architectures. Finally, the conclusion of this works is given in Section 3.6.

3.2 Related Work

Most of the routing protocols in VANET are designed to build a path with shorter delivery delay from a source to a destination vehicle. The delay of forwarding between each hop is very small while most of the latencies can be potentially caused by the *carrying* delay. Vahdat and Becker proposed an epidemic routing algorithm with a transmission model in a partially connected MANET in [47] in order to maximize the delivery rate and minimize the delivery latency. They introduced the idea of the *carry*-and-*forward* which is widely used in dealing with frequent network partitioning and merging. Packets can be forwarded if there are other nodes nearby. Otherwise, packets have to be carried until the carrier reaches other nodes in its communication range. However, this protocol is not specifically designed for VANET and ignored the fact that the trajectory of a moving vehicle is predictable base on its current location, speed and traffic condition. To modify the model and let it become suitable for VANET, Zhao and Cao proposed a data forwarding scheme and constructs a link delay model called vehicle-assisted data delivery model (VADD [24]). They investigated the data forwarding delay using a traffic stochastic model to achieve lower delivery delay. Based on the traffic density, they decide which portion of the street that the data will be forwarded and which portion of street that the data will be carried.

[26] and [28] take advantage of the vehicle trajectory for designing routing algorithms with shorter latency. TSF is a trajectory based routing algorithm for selecting an appropriate RSU as the target point in VANET, which is proposed by Jeong et al. in [26]. TSF selects an optimized target point (RSU) for reducing the delay and utilizes vehicle trajectory information to compute the Expected Delivery Delay (EDD). This EDD will be shared with other neighbors and the vehicle with the shortest EDD will be chosen as the next hop. Xu et al. designed a shared trajectory based data forwarding scheme for V2V transmission [28] which used the predicted encounter graph to minimize

the delivery delay. Vehicles have to share their trajectory with others for the encounter time prediction. In this way, vehicles can select the next hop with less latency based on the estimated encounter points. However, these trajectory based routing algorithms are hard to be realized in the real world since people may not want to share their own trajectories considering the privacy issue.

The concept of using public transportation for data delivery has been considered in some previous works. Wong et al. proposed an architecture of BUSNet in [45]. Through several experiments, they showed that common vehicles (we simply call them vehicles in this dissertation) with different speeds and unpredictable paths may degrade the performance of VANET. They justified through simulation that vehicles are constrained by the road conditions and can be partitioned into disjointed parts. Therefore, they attempted to take advantage of public transportation with predictable routes for improving the inter-vehicle communications. The basic idea of BUSNet is to build the data transmission path by buses for two vehicles that are geographically far away. If these two vehicles are far from each other, the package delivery will be finished by buses instead of common vehicles.

In [46], Luo et al. introduced a two-tier VANET, in which buses constitute the backbone for data delivery. One tier is the vehicles while another composed of the buses. Similar to [45], packet delivery is completed by buses. They also provided a registration method based on the estimated connection time. Base on the current locations, running directions and speeds of the vehicle and bus, they estimated when their distance will become larger than the communication range. However, their method of computing the connection time constrains that vehicles cannot change their speeds and directions. Besides, they did not take advantage of traffic infrastructures which can be used to improve the VANET performance.

Several routing algorithms have been proposed taking advantage of buses. Kitani et al. used buses to efficiently collect and propagate traffic information in [48]. Buses are regarded as message ferries which move along predetermined routes. They collect traffic information from nearby vehicles and disseminate such information to other vehicles in other areas. Meanwhile, regular cars send packets including current area traffic information to buses and request information of other areas from buses. Li et al. used the Expected Min-Max Delay (EMMD) to select bus as relay node in [49]. The data delivery is completed by buses and their routing decisions are based on the minimum maximum delay. Knowing the routes and schedules of buses, they estimated the maximum delay of each potential path and the one with the smallest maximum delay will be selected for data delivery.

Lai proposed a Footmark Leaving scheme in [50] to locate the destination vehicle. It lets each vehicle maintain a table of recent passed vehicles in each road segment. A destination finding request is broadcasted until it reaches a vehicle that keeps the information about the destination vehicle. Thus, the packets can be transmitted along the road segment recorded in the table. Then, buses are responsible for delivering the packets to an appropriate road segment based on the trajectories of the bus and the destination vehicle.

All of the previous studies ignored the important impact of the existing traffic infrastructures. Although the schedules and routes of buses are predictable, the number of buses can vary from one hour to the next. Therefore, the VANET performance cannot be guaranteed without using some number of stationary RSUs and TCC. On the other hand, the Footmark Leaving scheme proposed in [50] is not very efficient for destination identification since it uses a form of broadcasting.

3.3 Proposed BUS-VANET Architecture

Several significant improvements are introduced in our new two-tier BUS-VANET architecture since we fully integrate BUS-VANET with traffic infrastructures. In this section, we overview the proposed BUS-VANET architecture and describe how the network operates for data delivery.

3.3.1 Assumptions

Firstly, three assumptions are made in our BUS-VANET:

1) All the vehicles, buses and RSUs are equipped with DSRC devices for communicating with each other and GPS-based navigation system with a digital road map. Current information about traffic statistics is also available.

2) Buses and RSUs are additionally equipped with a Wi-Fi communication capability. Therefore, they truly form a backbone of VANET.

3) The route and schedule of every bus and the location of each RSU are shared with all the other vehicles.

3.3.2 Operation of Proposed BUS-VANET

In our proposed BUS-VANET, data delivery will be carried out by the corporation of buses, RSUs and TCC. The mobile buses and fixed location RSUs are dynamically forming a connected topology. For Infrastructure to Vehicle Communication (I2V), TCC first identifies the destination vehicle's location (i.e., which bus or RSU that the destination vehicle is the most recently registered with), and then forward the packet to this RSU or bus which can further transmit the packet to the specified destination vehicle. For Vehicle to Infrastructure Communication (V2I), vehicles can send the packet to a nearby bus or RSU (i.e., the high-tier node that they are currently registered with) which can forward the packet to the TCC. For Vehicle to Vehicle (V2V) communication, the packet is first transmitted to a nearby bus or RSU. Then this high-tier node checks whether it already knew the destination information or not. If yes, the packet can be directly sent to the destination without the help from TCC. Otherwise, it has to check with TCC to find out the target bus or RSU that the destination vehicle is currently registered with. After getting feedback from TCC, they can start the transmission to the target bus or RSU and then to the destination vehicle.

To obtain data delivery service, each vehicle needs to register with a nearby high-tier node directly or through multi-hop communication. High-tier nodes (buses and RSUs) periodically broadcast beacon messages containing their locations and speeds. Vehicles that received such messages need to propagate these beacons and chooses one high-tier node they can hear to finish its registration. All the beacon messages have a life time that indicates whether it is still active. If a beacon exceeds its life time, it will be ignored by the received vehicles and no longer be broadcasted. When a vehicle loses connection to its current registered high-tier node, it needs to switch to another high-tier node for getting further service. Every bus and RSU holds and keeps on updating its own registration table which contains the information of the vehicles that are currently registered with them. There are several reasons that why we need this registration. Firstly, registration helps to locate the destination vehicle quickly. Besides, without registration, a vehicle needs to construct a delivery path each time when it wants to send a packet. This not only wastes resource, but also increases the delay. Although the registration process is important, there are some issues remained to be solved. For instance, if there are multiple buses or RSUs nearby, which one should be selected as registration node and what metric should be based on to make such a decision? If there is no bus or RSU nearby, what a vehicle should do? We will discuss these issues in Section 3.4.

Broadcasting is widely used in most of the previous studies for identifying a destination vehicle. A source vehicle that has no knowledge about where is the destination vehicle has to broadcast a request to find where it is. Others received such request need to do further broadcasting until someone knows the location of the destination vehicle and send back a feedback to the source vehicle. This not only causes heavy packet traffic, but also increases the time of searching for the destination. Although broadcasting is inefficient for destination identifying, there are fewer better ways we can use without the help of traffic infrastructures. By integrating TCC into our BUS-VANET, we propose a more efficient scheme for identifying the destination vehicle location quickly. We add a location table in TCC which records the location information of all the vehicles. All high-tier nodes are required to report their registration tables to TCC periodically. Since all the high-tier nodes connect to TCC directly or via Wi-Fi, checking the location table will be much faster than broadcasting among vehicles and the delivery delay can be reduced. We will also discuss more details related to the destination location identification in Section 3.4.

3.4 Research Issues in BUS-VANET

Although the proposed two-tier BUS-VANET architecture brings many benefits, there are several issues need to be solved in order to improve the BUS-VANET performance. Two main issues and the possible solutions are discussed in this section.

3.4.1 Selection of Registration Node

In our proposed BUS-VANET, each vehicle needs to register with a nearby high-tier node for getting data delivery service. How to determine which bus or RSU should be selected for registration is an important issue if a vehicle received several beacons from different high-tier nodes. When a vehicle receives an active beacon from a bus or RSU, this bus or RSU will be regarded as a candidate registration high-tier node and will be put into a candidate set. If a vehicle loses connection with its currently registered bus or RSU, it needs to switch its registration to another high-tier node. Since switching from one bus or RSU to another will cause path re-computation and rebuilding, we aim at reducing the number of such switches. The bus or RSU with the longest registration time will be selected as the registration node from the candidate set. The registration time here means how long a vehicle can keep the registration with a bus or RSU before it has to switch to another high-tier node.

The meaning of connection is difference in our proposed BUS-VANET comparing with previously proposed VANET architectures. In previously proposed VANETs, the connection between two nodes means that they are in the communication range R of each other and they will be regarded as disconnected when their distance is larger than R. However, in our BUS-VANET, a vehicle can connect to a bus or RSU via multi-hop communications, which makes the previously proposed judgment of disconnection inapplicable. Thus, we use the estimated package delivery delay based on the distance and traffic density between two nodes as the metric to judge disconnection. Since a vehicle knows its own speed and trajectory, it can predict its location at any time. Note that buses and RSUs do not need to know the trajectory of a vehicle, only each vehicle knows its own. Two nodes are regarded as disconnected when their estimated package delivery delay is larger than a given threshold T.

Since packet delivery delay is composed of two parts: *Forwarding* and *Carrying*, following notations are defined for estimating the delivery delay:

1) L_{ij}: the length of the road r_{ij} from intersection i to j
2) L_f and L_c: the distance of *forwarding* and *carrying*
3) d_{ij}: the delivery delay from intersection i to j
4) v_{ij}: the average speed for vehicles running on road r_{ij}

If the vehicle communication range R is smaller than the length of the road L_{ij}, the total delivery delay from one vehicle to a bus or RSU is the sum of delays for all the road segments along a delivery path. For one road segment, the delivery delay can be expressed as:

$$\begin{cases} d_{ij} = \alpha \times \dfrac{L_f}{R} + \dfrac{L_c}{v_{ij}} \\ L_f = P[F] \times L_{ij} \\ L_c = L_{ij} - L_f - R \end{cases} \tag{3.1}$$

where α is a constant that is used to adjust the forwarding delay and P[F] stands for the probability of packet forwarding. Assuming the inter-vehicle distance obeys the exponential distribution, we can get P[F] = (1-e$^{-R\times\rho_{ij}}$). ρ_{ij} is the traffic density of the road segment from Intersection i to j. For a delivery path from Intersection m to Intersection n, the total delay D is:

$$D = \sum_{\substack{m \le x \le n-1 \\ m+1 < y < n}} d_{xy} \tag{3.2}$$

On the other hand, we consider the case that the vehicle communication range R is larger than the length of the road. In this case, the delivery delay is not computed based on every road segment. We use the distance L(AB) between vehicle A and the high-tier node B instead of using the length of the road. The distance between them does not mean the Euclidean distance. It is the sum of the lengthes of all the road segments on a delivery path. An example is given in Figure 3.2. The red and blue dash lines represent different delivery paths from vehicle A to Bus B. For different delivery paths, the distance L(AB) can be compute as:
L1(AB) = L(AC) + L(CD) + L(DE) + L(EB)
L2(AB) = L(AF) + L(FG) + L(GH) + L(HE) + L(EB)

Then we can compute the delivery delay from A to B for each delivery path as:

$$\begin{cases} D(AB) = \alpha \times \dfrac{L_f}{R} + \dfrac{L_c}{v(AB)} \\ L_f = P[F] \times L(AB) \\ L_c = L(AB) - L_f - R \end{cases} \tag{3.3}$$

Similarly, we know that P[F] = (1-e$^{-R\times\rho(AB)}$) and $\rho(AB)$ is the traffic density on the delivery path between A and B. Since there are several possible delivery paths that the packet can be transmitted, we pick up the one with the smallest delay. If there are n

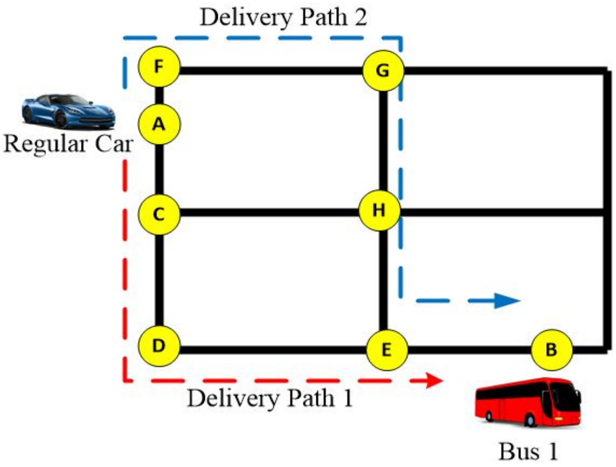


Figure 3.2: Distance between Vehicle and Bus/RSU

possible paths:

$$EstimatedDeliveryDelay = min\{D_1, D_2...D_n\} \tag{3.4}$$

In order to find out when a vehicle will lose connection with a candidate bus or RSU, we sample at each intersection and check whether they are still connected with each other. Besides, a bus or RSU that is regarded as disconnected at one intersection may return to be connected for the rest of the intersections. It is inefficient to eliminate a bus or RSU if the disconnection duration is very short. Therefore, we set a threshold N_t as the maximum tolerant number of disconnected intersections. If a high-tier node has been judged as disconnected for more than N_t (N_t is set to a small number like 2) intersections, this high-tier node will be discarded from the registration candidate set.

We show an example in Figure 3.3, where a vehicle has four candidates for registration: Bus 1, Bus 2, RSU 1 and RSU 2. The blue line is the trajectory of the regular vehicle while the red lines stand for the trajectories of buses. When the vehicle arrives at intersection A, buses will reach their point 1. When the vehicle arrives at intersection B, buses will reach point 2. At intersection C, the delivery delay between this vehicle and Bus 2 becomes larger than the threshold T and the connection will not recover later on. That is, exceeding the maximum tolerant number of disconnected intersections N_t. So Bus 2 will be discarded from the registration candidate set. Similarly, RSU 1 and RSU 2 will also be eliminated from the candidate set and the vehicle will finally choose

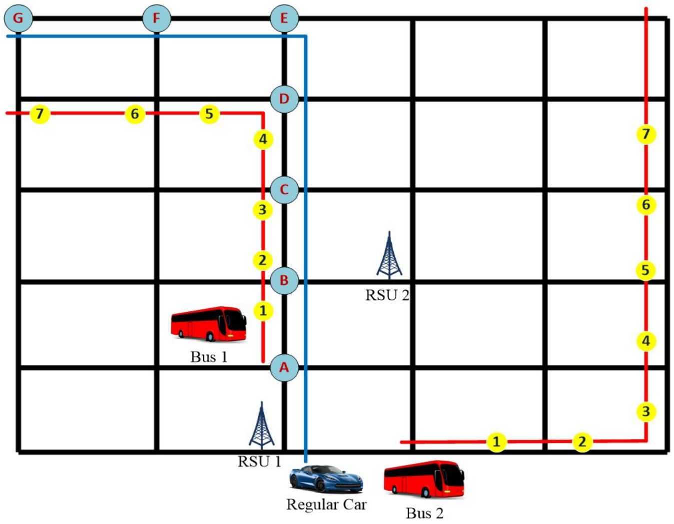


Figure 3.3: Example of Registration Node Selection

Bus 1 for registration.

Since the speed and location of a vehicle are affected by the real road condition, the prediction of a vehicle's location may become inaccurate with the elapsing time. The traffic density also changes over time, which influences the accuracy of the estimated delivery delay. Thus, we set a maximum number of prediction intersections as N_i. We only do the prediction and assume the vehicle's trajectory and traffic density are correct within these N_i intersections. If more than one high-tier nodes are left after checking N_i intersections, the one with the smallest average delivery delay will be selected for registration. Similarly, if a vehicle finds that no bus or RSU is left in the candidate set after checking an intersection, it will back to the previous intersection and select the one with the smallest average delivery delay as registration node. We define the registration node as Reg, the Longest Registration Time Algorithm is shown as Algorithm 1.

Furthermore, we considered the situation that there is no high-tier node nearby. That is, a vehicle receives no beacon messages from a bus or RSU. Base on its own trajectory, the routes of buses and the locations of RSUs, a vehicle is able to predict whether it can get service in a short duration. If yes, it just holds the packets and waits for the bus or RSU. Otherwise, it shall transmit the packets through other low-tier vehicles since they are likely registered with a bus or RSU. Later, when this vehicle receives a beacon, it can finish the registration. For packets receiving, if a vehicle lost connection with its current registration node, the current high-tier node needs to keep this vehicle's information in its registration table for a period of time. In this way, the high-tier nodes have a rough idea about where is the vehicle even if they are disconnected.

3.4.2 Destination Vehicle Location Identification

By integrating TCC and RSUs with buses and vehicles, we design a new scheme for identifying the destination vehicle quickly. In this subsection, we will provide more details about this TCC identification scheme including how to find the correct location of a destination and how to decrease the workload of TCC. As we mentioned, each bus or RSU keeps a registration table recording which vehicles are currently registered with them. These registration tables will be reported to TCC periodically and TCC maintains a location table to store these collected information. An example of the location table in TCC is shown as Table 3.1, which records the information about each registered vehicle.

Table 3.1: Example Location Table

Bus/RSU ID	Vehicle ID
Bus 1	Vehicle 1
	Vehicle 2
	Vehicle 3
Bus 2	Vehicle 4
	Vehicle 5
RSU 1	Vehicle 6
RSU 2	Vehicle 7

Although reporting to TCC for every change lets TCC keep the most up-to-date information, the overhead will be extremely high since many changes will happen in a high mobile vehicular network environment. Thus, we let buses and RSUs report their registration tables to TCC periodically. However, the information stored in the location table in TCC may not be the latest and a vehicle may have switched to other RSU or bus. To deliver the packet to the correct destination, high-tier nodes need to keep track of vehicles when they switched to a new one. The registration table that is hold at each bus and RSU is represented as Table 3.2. The registration table not only records the current registered vehicles, but also keeps the information about where the previously registered vehicles switched to. The Null in Table 3.2 means that the vehicle still belongs to the current high-tier node. The newly registered RSU or bus by a vehicle needs to inform the previously registered bus or RSU to track this vehicle for a short duration. Thus, when a packet is sent to the old bus or RSU, it can be forwarded to the destination vehicle following the destination tracking path. Such vehicle tracking information will be deleted from the registration table after the RSU or bus reports to TCC.

Table 3.2: Example Registration Table

Vehicle ID	Bus/RSU ID that the vehicle has switch to
Vehicle 1	Null
Vehicle 2	Null
Vehicle 3	Bus 2

To further reduce the workload of TCC, each bus and RSU can also keep a routing table to record the destination vehicles and their registered buses or RSUs as shown in Table 3.3. For packets that are sent to a repeated destination in a limited time, a source bus or RSU can record the destination location after asking TCC and directly transmit the packet for the next time. Similarly, when a destination bus or RSU receives a packet, they learns that the source vehicle is registered with which high-tier node and puts such information into its routing table for future communications. Therefore, through past destination vehicle identification and backward learning, buses and RSUs can check the routing table to know the location of a destination vehicle. This avoids duplicated destination finding requests to TCC. Since vehicles are always moving, RSUs and buses also have to clear and update their routing table each period of time in order to remove the stale information.

Table 3.3: Example Routing Table for Learned Destinations

Vehicle ID	Registered Bus or RSU
Vehicle 3	Bus 2
Vehicle 4	Bus 2
Vehicle 6	RSU 1

3.5 Simulation and Performance Evaluation

We constructed a simulator by integrating SUMO and NS3 to simulate our proposed BUS-VANET performance in a real world environment. SUMO is a traffic simulator that generates road topology and traffic mobility patterns. NS3 is a network simulator that simulates the network performance like packet transmission. As shown in Figure 3.4 (a), we use part of downtown Minneapolis for our simulation and Figure 3.4 (b) provides the translated map that we used in our simulator. Base on the real bus schedule, six bus lines are created and the bus regeneration time is 5 minutes. In our simulation, detectors are settled on each road segment, which can give us the density information about the number of vehicles running through them. Table 3.4 shows the detailed parameter settings of our simulation. Notice here that although we do not focus on the physical aspects, the wireless link quality can be degraded considering the moving feature of buses.

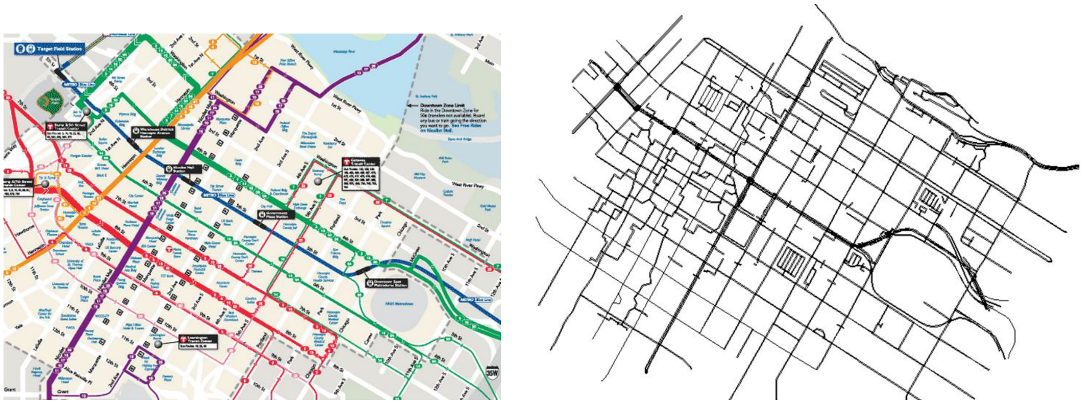
Table 3.4: Parameter Settings for BUS-VANET Simulation

Simulation Map Range	about 2×1.8km
Number of Generated Vehicles	100 vehicles per 200s
Max Vehicle Velocity	15m/s
Max Bus Velocity	10m/s
Number of Bus Lines	6
Bus Regeneration Time	5min
Communication Range (low-tier)	300m
Communication Range (high-tier)	300m and 1500m
Simulation Time	1200s

3.5.1 Comparison of Registration Schemes

We first compare four registration schemes: random selection, shortest distance selection, our proposed longest registration time selection and a scheme used in [46]. The metric that we focus on is the number of registration switches. In [46], the authors estimated the connection time between a bus and a vehicle using their current speeds and positions without considering the direction changes.

As shown in Figure 3.5 (a), the average number of switches for the longest registration time scheme is the least while the shortest distance scheme performs the worst

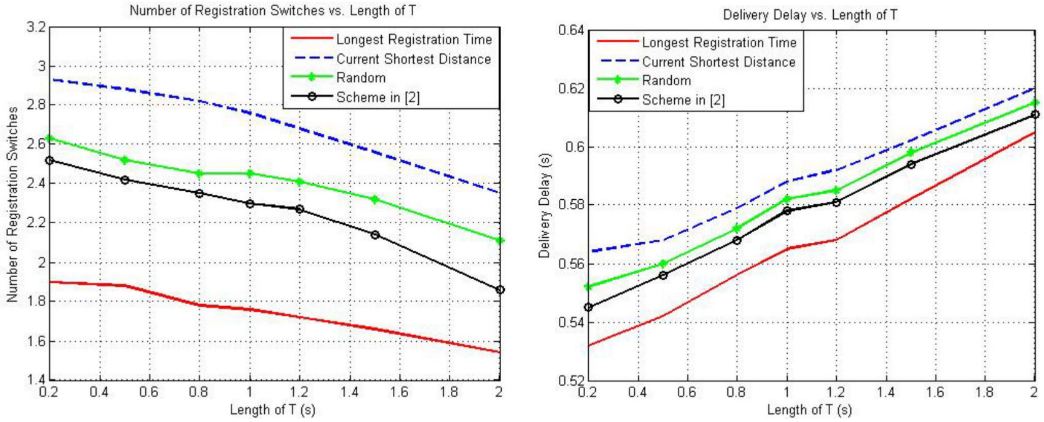


(a) Map of Downtown Minneapolis (b) Translated Map in Simulation

Figure 3.4: Maps for BUS-VANET Simulation

since it does not consider the direction and speed difference between a vehicle and a bus or RSU. They may lose connection right away since the vehicle and the bus are running at opposite directions regardless how close they are currently. On the other hand, the random selection scheme chooses the registration bus or RSU randomly. Sometimes, it may select the one with longer registration time while other times selects a worse one. For the scheme used in [46], it performs a little bit better than the random scheme. Since it cannot handle the situation that vehicles change their directions and it only considers one-hop connection, the improvement is not obvious. When the maximum tolerant delay T is set as 1s, our registration scheme can reduce the number of switches by 24% and 28% comparing to the scheme in [46] and the random selection respectively.

As we mentioned above, the threshold T is used to judge disconnection. It will also affect the network performance. We varied the length of T from 0.2 to 2s and the results are shown in Figure 3.5. When a small length of T, vehicles have to change their registration nodes more frequently. But the delivery delay becomes smaller in this case. On the other hand, if we set a larger T, the delivery delay will increase, but the number of switches will be reduced.



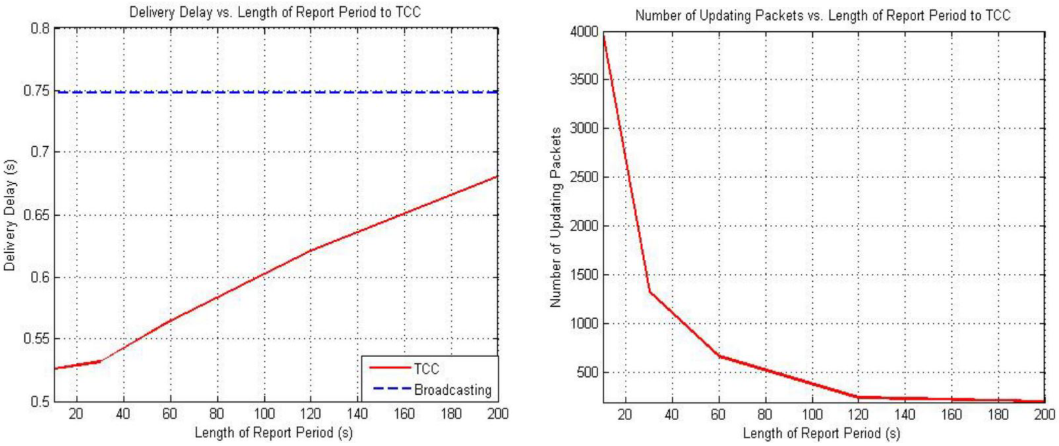
(a) Number of Registration Switches

(b) Delivery Delay

Figure 3.5: Comparison of Registration Schemes

3.5.2 Comparison of Destination Identification Schemes

We also compare two destination identification schemes: broadcasting and our TCC identification scheme. We run these two schemes separately in our BUS-VANET architecture. For TCC identification scheme, the length of time between each report from buses and RSUs to TCC influences the delivery delay. As shown in Figure 3.6, shorter the report period (interval between each report) we set, more accurate the information that TCC records and less delivery delay can be achieved. On the other side, a longer report period causes more time to find the location of a destination, but fewer control packets are generated. We vary the report period from 10 to 200s for observing its impact. Here we notice that the delay is almost the same when the report period is varied from 10 to 30s. This is because the location of a vehicle does not change too much within 30s. However, when we increase the report period to 200s, the difference becomes visible. Considering the balance between delivery delay and TCC workloads, we select 60s as the report period. About 24% shorter delay can be achieved for TCC identification scheme comparing with broadcasting when the report time is set as 60s. Even when the report time is 200s, our TCC identification scheme still performs better than that of broadcasting with 9% shorter delay.



(a) Delivery Delay

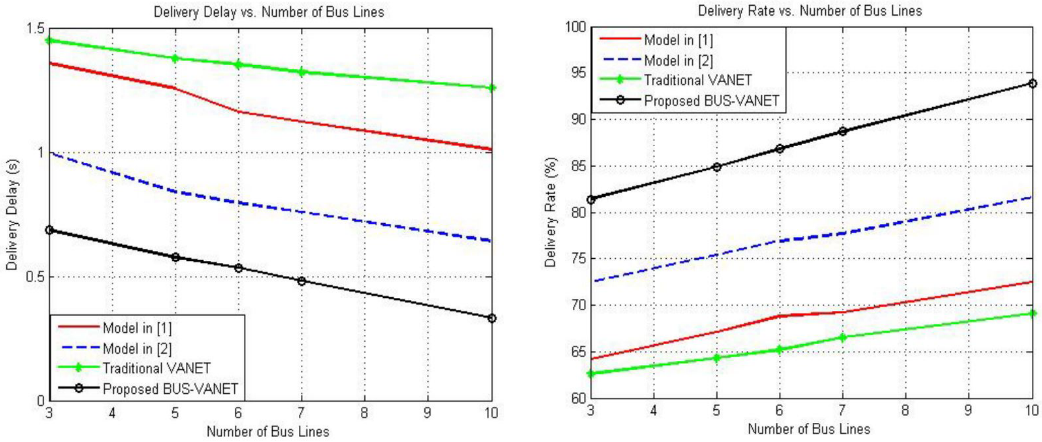
(b) Number of Updating Packets

Figure 3.6: Comparison of Destination Identification Schemes

3.5.3 Comparison of Proposed BUS-VANET and Existing VANETs

Then we compare the network performance of four VANET architectures: traditional VANET, the architecture proposed in [45], the architecture proposed in [46], and BUS-VANET. For these four architectures, 10 RSUs are uniformly distributed on the simulated map. We vary the number of bus lines from 3 to 10 in order to investigate how it impacts the performance. In the traditional VANET, vehicles and buses are treated equally. RSUs have the same communication range as vehicles. For the architecture in [45], buses are responsible for most of the packets transmissions but have the same communication range as common vehicles. No registration is used in this architecture. The difference between the architecture in [45] and [46] is that buses have longer communication range and a registration method is used in [46].

As shown in Figure 3.7, our BUS-VANET has the smallest delivery delay and the highest delivery rate. For a realistic bus system (6 bus lines) setting, comparing to the traditional VANET, architectures proposed in [45] and [46], the delivery delay in our BUS-VANET reduces by 60%, 46% and 33% respectively. Meanwhile, the delivery rate increases by 25%, 21% and 12% respectively. With the increment of number of bus lines, better performance can be achieved since larger number of buses can provide better transmission coverage and service. The traditional VANET performs the worst



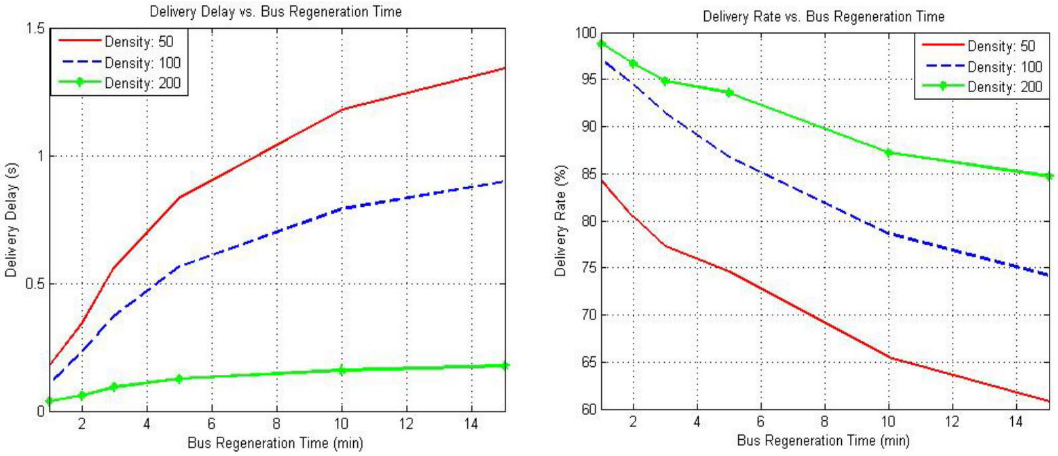
(a) Delivery Delay

(b) Packet Delivery Rate

Figure 3.7: Impact of Number of Bus Lines

since it treats every node equally. The performance of the traditional VANET changes slightly when we varying the number of bus lines. This is due to the fact of changing the number of bus lines equals to changing the number of vehicles and the difference of these numbers is not considerable. The architecture proposed in [45] works a little bit better than the traditional one since it only utilizes the predictable bus routes but does not give buses longer communication range. The architecture in [45] performs better than [46] since buses have better ability on handling the data transmission through longer communication range and a registration method is used.

We also explore the influence of bus frequency in our BUS-VANET. The results are shown in Figure 3.8. As we increase the generation time between two consecutive buses of the same bus line from 1 to 15 minutes, the average delivery delay increases by 88% while the average delivery rate drops by 24% when we generate 100 vehicles for a duration of 200 seconds. Lower frequency of buses causes fewer chances for vehicles to finish their registration. Higher frequency of bus regeneration improves the performance of packets transmission. If we fix the bus frequency as 5 minutes and increase the number of vehicles generated for a duration of 200 seconds from 50 to 200, the delivery delay reduces by 85% while the delivery rate increases by 20%. The reason is that higher traffic density increases the probability of building connection among vehicles and decreases the probability of packet carrying.



(a) Delivery Delay

(b) Packet Delivery Rate

Figure 3.8: Impact of Bus Frequency

In our BUS-VANET, fewer RSUs can be required with the help of buses. While ensuring the service coverage and maintaining the quality of data delivery, avoiding installing more RSUs can save certain costs. We compare two VANET architectures here: one is our two-tier BUS-VANET and the other is a traditional VANET without buses. For BUS-VANET, we set the number of bus lines as six and the bus regeneration time is 5 minutes. The number of RSUs is fixed as 10 for the traditional VANET and we vary the number of RSUs in BUS-VANET from 0 to 10. As shown in Figure 3.9, the red dash line stands for the delivery delay of a traditional VANET while the blue line expresses our BUS-VANET. These two lines meet each other when we reduce the number of RSUs to around 5 or 6. Thus, we can infer that our BUS-VANET with 5 or 6 RSUs performs similar to a traditional VANET with 10 RSUs.

3.6 Conclusion

In this dissertation, we propose a new two-tier BUS-VANET that is fully integrated with traffic infrastructures for improving the performance of VANET. We take advantage of RSUs and TCC that already required and constructed by ITS and investigate the benefits we can obtain from this realistic environment. By integrating RSUs and TCC with buses, the coverage of the high-tier nodes can be ensured and the probability

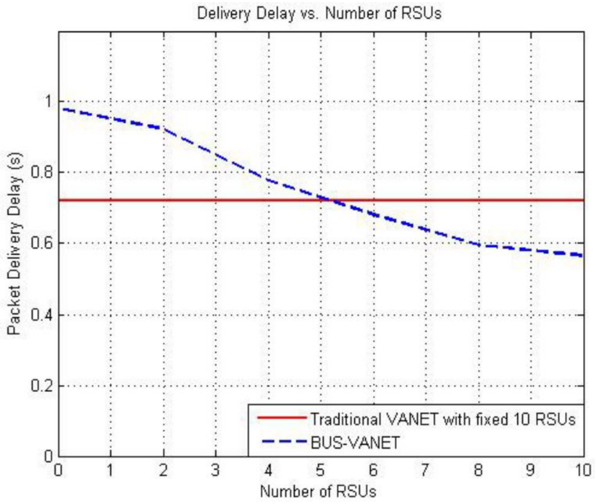


Figure 3.9: Impact of Number of RSUs

of packet carrying is reduced. TCC is helpful for quickly identifying the location of the destination vehicle. Comparing to traditional VANET, better performance can be achieved in BUS-VANET with less delivery delay and higher delivery rate.

Algorithm 1 Longest Registration Time Algorithm

```
$S$ = candidate registration set; $D$ = discard set;
$N_i$ = maximum number of prediction intersections;
$N_t$ = maximum number of disconnected intersections can be tolerated;
$N_d(x)$ = number of disconnected intersections for high-tier node $x$;
$T$ = maximum delivery delay can be tolerated;
while number of checked intersections $\leq N_i$ and more than one entry left in $S$ do
    set the next intersection for checking;
    for every bus or RSU $x$ in $S$ do
        compute the estimated delivery delay $t$;
        compute and record the average delivery delay for all the intersections have been checked;
        if $t > T$ then
            $N_d(x) = N_d(x) + 1$;
            if $N_d(x) > N_t$ then
                put this node $x$ into $D$;
                remove this node $x$ from $S$;
    end for
    if $S$ has more than one bus or RSU then
        Clear $D$;
end while
if $S$ has more than one bus or RSU then
    $Reg$ = the one with the smallest average delay in $S$
else if no bus or RSU left in $S$ then
    $Reg$ = the one with the smallest average delay in $D$
    Clear $D$;
else
    $Reg$ = the only one bus or RSU in $S$
return $Reg$;
```

Chapter 4

TDMA based MAC Protocol for Reducing Packet Collisions

In this chapter, new MAC protocols are introduced in order to improve the data transmission reliability in VANET. Smaller number of packet collisions and higher packet delivery rate can be achieved under different traffic scenarios. Two TDMA based MAC protocols are proposed: MAT-MAC and PTMAC. MAT-MAC is designed for two-way traffic. It aims to reduce the number of both encounter and contention collisions while maintain high slot utilization even with unbalanced traffic densities. PTMAC is a novel protocol based on the important observation that most of the encounter collisions can be predicted and potentially avoided. It is not only suitable for two-way traffic scenario but also for four-way intersections in an urban area.

4.1 MAT-MAC: A Migration-based Adaptive TDMA MAC for Reducing Packet Collisions in VANET

4.1.1 Introduction

Different from other Ad-Hoc networks, VANET has the unique characteristics of high node mobility, dynamic topology changes and strict delay constrains. These issues must be considered in developing MAC protocols for VANET to support both safety and non-safety related applications. Although the carrier sense multiple access/collision avoidance (CSMA/CA) based IEEE 802.11p [51] has been approved as the standard Medium Access Control (MAC) protocol, it suffers high collision probability under high traffic density situations [35, 37]. Because of its contention-based nature, 802.11p also has the drawback of potential unbounded channel access delay.

To overcome the shortcomings of IEEE 802.11p, time division multiple access (TDMA) based MAC protocols have been proposed to facilitate efficient transmission in VANET. Basically, there are two types of TDMA based MAC protocols: distributed TDMA and centralized TDMA. Each node manages its time slot by itself in a distributed TDMA while all the time slots are allocated by a central node such as a RSU or a cluster head in a centralized TDMA [52–54]. Considering the high mobility nature of vehicles and the potential lacking of RSUs in highway scenario, we focus on distributed TDMA based MAC protocols which provide a more flexible way for slot management. The most basic distributed TDMA based MAC protocol has been proposed in [39] using time slotted structure. The time is partitioned into repeated frames and each frame is composed of fixed number of slots. Each vehicle selects a specific available time slot to transmit data. If successful, it keeps on using the same slot at subsequent frames until a collision occurs or the slot is no longer needed. Since every vehicle is required to broadcast the slot information about all its one-hop neighbors, a vehicle is able to know which slots have already been occupied in two-hop communication range. Therefore, the probability of transmission collisions is reduced and each node is guaranteed to access the channel at least once in each frame if a reservation is successfully made. There is no need for each individual packet to compete for the channel.

However, such TDMA based MAC protocol still cannot avoid packet collisions. Considering the real two-way traffic environment, two types of packet collisions may happen: contention collision and encounter collision. A contention collision usually happens between newly joining vehicles who are trying to reserve the same available slot within two-hop range. The newly joining vehicles are defined as those who have not reserve a slot and intend to transmit packets. An encounter collision happens between vehicles that are occupying the same time slot. They are originally out of two-hop range but will encounter each other in the near future. Encounter collisions are caused by the unique mobile characteristic of vehicles and they happen more frequently between vehicles from opposite directions and driving towards each other.

In order to reduce the number of encounter collisions, slot partition based MAC protocols have been proposed by separating slots in each frame into disjointed sets for different directions. However, the major problem of such partition is that the slot utilization becomes low when the traffic density is high in one direction while low in another. Vehicles running at the direction with denser traffic may also suffer more contention collisions. Although VeMAC proposed in [55] used a kind of adaptive scheme, its random slot borrowing scheme incurs encounter collisions which may make the partition scheme becomes meaningless. Besides, a vehicle may already experience several contention collisions before it is allowed to contend for a slot from the other direction. Therefore, we design a new Migration-based Adaptive TDMA MAC protocol (MAT-MAC) to reduce the number of both encounter collisions and contention collisions while maintain high slot utilization even under the scenario with unbalanced traffic densities. Our main contributions can be summarized as follows:

- Proposing a new Migration-based Adaptive TDMA MAC protocol (MAT-MAC) for two-way traffic to reduce the number of both encounter and contention collisions while maintain high slot utilization even under scenario with unbalanced traffic densities.
- High contention collision probability is avoided through dynamic slot adaptation based on the real-time traffic condition. We also design a method to estimate the number of competing vehicles which is used for computing the contention collision probability.
- Through performance evaluation and comparison, we show that MAT-MAC has better performance with smaller number of collisions and higher slot utilization for two-way traffic regardless of the unbalanced traffic densities.

In the following, Section 4.1.2 overviews the related work. Section 4.1.3 explains our proposed MAT-MAC protocol. How to estimate the contention collision probability is discussed in Section 4.1.4. In Section 4.1.5, we evaluate the performance of our MAT-MAC protocol and compare it with other TDMA based MAC protocols and Section 4.1.6 gives a conclusion of our work.

4.1.2 Related Work

Comparisons between 802.11p and TDMA based MAC protocols in VANET have been studied in some previous works [37,38,56]. These works demonstrated that TDMA performs more reliable and robust when comparing with IEEE 802.11p. A basic TDMA based protocol named ADHOC MAC has been proposed by Borgonovo et al. in [39] for VANET. It was designed for Ad-Hoc Networks to provide efficient and reliable data delivery service. It grouped a set of time slots into a frame and defined a concept of Frame Information (FI) which contains the time slot status. Each vehicle is responsible for broadcasting its FI to inform others the occupied slots by its one-hop neighbors and itself. In this way, every vehicle can get all its neighbors' slot information within two-hop range. A new joining vehicle needs to listen to the channel for a frame and then selects an available time slot to transmit data at next frame. Once a vehicle gets a slot successfully, it keeps on using the same slot at subsequent frames until a collision happens or it does not need the slot anymore. However, this slot reservation scheme cannot handle the encounter collisions.

Some improved MAC protocols have been developed [57–59] based on such slotted structure under one-way traffic. An adaptive distributed MAC protocol named A-ADHOC is proposed by Liu et al. in [57]. Since the fixed-size frames may waste slots and introduce unnecessary delay under a sparse traffic condition, A-ADHOC dynamically adjusts the length of a frame based on the real-time traffic density. They showed that A-ADHOC can enhance the performance with less transmission delay. Yu and Biswas proposed a self-configuring protocol called VeSOMAC in [58]. In VeSOMAC, every vehicle uses a bit map to record their neighbors' slot information which is also shared with others. Unlike other schemes that select a time slot randomly, the authors paid more attention to the ordering of time slots. They ordered time slots in the same sequence as the vehicles appear on the road to reduce the packets forwarding delay. In [59], Bharati et al. developed a cooperative protocol named CAH-MAC. Their scheme allows the neighbors who detected a transmission failure from a vehicle to retransmit the packet using an unreserved slot. But they ignored the fact that new joining vehicles may also contend for the same unreserved slot. Thus, more contention collisions will be introduced. However, these proposed protocols only considered one-way traffic and did not focus on reducing the transmission collisions.

A few MAC protocols have been proposed using slot partition method for solving the encounter collision issue under two-way traffic scenario. Zhou et al. proposed an Even-Odd MAC protocol in [60]. They regulated that vehicles running to the right can only contend for even slots while vehicles heading left can only reserve the odd slots. In this way, encounter collisions caused by vehicles from the opposite directions can be entirely avoided. However, the slot utilization of this scheme is low under unbalance traffic densities scenario. High contention collision probability will also be incurred for vehicles running at the direction with dense traffic. Omar et al. proposed another TDMA based MAC protocol called VeMAC in [55] that also uses partition method. Unlike the Even-Odd MAC in [60], slot partition in VeMAC is not strict, that is, if a vehicle cannot successfully reserve a slot within τ frames, it is allowed to contend for any available slots. However, since the borrowed slots are just randomly selected, the encounter collisions from vehicles at opposite directions may be introduced. Besides, a vehicle already suffered high contention collision probability before it can contend for a slot from the other direction. Moreover, the authors did not discuss what value τ should be set.

4.1.3 Proposed MAT-MAC Protocol

To avoid the encounter collisions from vehicles running at opposite directions, Even-Odd scheme is used that vehicles moving right are only allowed to reserve even slots while vehicles heads left can only contend for odd slots. If the traffic densities are similar for the two directions, vehicles need to follow such even and odd scheme strictly. However, if the traffic densities are unbalanced, that is, low in one direction while high in the other direction, vehicles heading the direction with denser traffic will be allowed to borrow some extra slots from the other direction. We make some assumptions that exactly follow the basic ADHOC MAC [39] protocol.

1. Every vehicle broadcasts a FI message at every frame which includes its own and each one-hop neighbor's occupying slot information.
2. Every vehicle keeps the slot information about its one-hop and two-hop neighbors.
3. Each newly joining vehicle that has not obtained a slot and wants to get a slot needs to listen to the channel for one frame. Then, they can randomly choose an available slot assigned for its direction at the next frame for transmission.

Slot Migration Scheme

An important issue caused by slot borrowing is the potential high probability of encounter collision. If a vehicle randomly borrows a slot from the opposite direction, its transmission may collide with others from the opposite direction. If the number of borrowed slots is large, the partition scheme becomes meaningless and it cannot efficiently avoid encounter collisions anymore. Therefore, we design a Slot Migration scheme which allows vehicles to borrow extra slots without suffering encounter collisions with others heading opposite direction.

The basic idea is to put the occupied slots in the front of each frame and lend out slots from the tail of each frame. In this way, the borrowed slots and occupied slots become disjointed and encounter collisions from vehicles heading different directions can be prevented. A newly joining vehicle first randomly contends for an available slot for its direction. After it reserves a slot successfully, it needs to check if there are any "hole" before its current reserved slot. Here a "hole" means the non-adjacent even or odd slots. A vehicle needs to check the occupied slots of all its neighbors within two-hop range at the same direction. If a "hole" is found, slot migration will be processed to fill this "hole". Before a vehicle migrates to another slot, it needs to announce where it will switch to using its original slot. A vehicle cannot access a slot that has been announced by another vehicle to reserve. In this way, the contention collisions among migrating vehicles and newly joining vehicles can be avoided. Such announcement also indicates that the current slot will become available and open for contention from next frame. Therefore the Slot Migration will not cause slot wasting.

Notice here that vehicles may have different neighbors within two-hop range so their slot arrangements in a frame may not be the same. For example, in Figure 4.1, vehicle heading left are occupying odd slots while vehicles running to the right are reserving even slots. Vehicle B, C, D, E, F, G, X and Y are all the neighbors of vehicle A within two-hop. However, vehicle C, D and E are not in the two-hop range of X and Y. From vehicle A's point of view, there is no "hole" ahead of its current slot (slot 3) so no migration is needed. On the other hand, from the view of vehicle X, there is a "hole" of slot 1 so it will migrate to slot 1. Since vehicle X will announce its migration and Y's slot locates behind X's slot, Y is able to learn that slot 1 will be filled by X. Therefore, vehicle Y will migrate to slot 5 to fill the "hole" between Slot 3 and 9. Figure 4.2 shows the slot arrangements and migration processes for vehicle A, X and Y respectively.

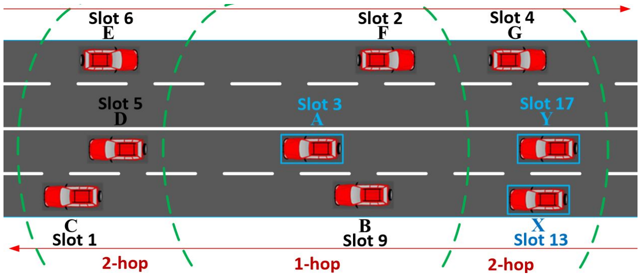
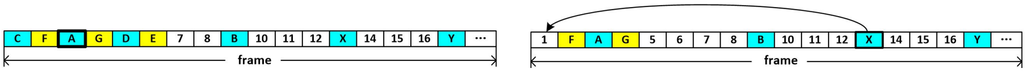
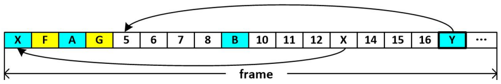


Figure 4.1: Traffic Scenario for Slot Migration



(a) Migration From A's View

(b) Migration From X's View



(c) Migration From Y's View

Figure 4.2: Example of Slot Migration

Slot Borrowing and Returning

The slot borrowing decision is made base on the real-time traffic condition and the contention collision probabilities on both directions. On the one hand, if the contention collision probability is too high in one direction, vehicles in this direction should begin to consider borrowing slots. On the other hand, they need to make sure that vehicles in the opposite direction where the slots will be borrowed from can afford lending out these slots temporarily. Meanwhile, if the difference of contention collision probabilities between the two directions is not too large, that is, the traffic densities on the two directions are similar to each other, the slot borrowing is unnecessary and may greatly impact the direction that lend out the slots. Therefore, we set three borrowing conditions for each vehicle to decide whether it should borrow slots from the other direction or not:

1. The contention collision probability of the same direction is larger than the Borrowing Threshold T_B

2. The contention collision probability of the opposite direction is not larger than the Borrowing Threshold T_B
3. The difference of contention collision probability between the two directions is larger than the Difference Threshold ΔT

Once a vehicle finds that all the borrowing conditions have been satisfied, extra slots originally assigned for the other direction at each frame will be marked as borrowed. A one bit flag is added for each slot to indicate if this slot has been borrowed or not. Such borrowing information will be included in the FI messages and broadcasted at each frame until being returned. Notice here that a slot borrowing decision can only be made by a vehicle that already reserved a slot, since it is able to inform others about the borrowing information. Newly joining vehicles heading to the same direction heard such borrowing information are able to learn that extra slots are open for contention. Vehicles in the opposite direction received such borrowing information are forbidden to contend for these lend out slots. Meanwhile, the one-hop neighbors who received the borrowing decision will check its contention collision probability. If they find that all the borrowing conditions are satisfied, they will also mark the borrowed slots.

In order to increase the slot utilization and balance the contention collision probabilities, the number of borrowed slot is chosen to make the contention collision probabilities on the two directions become similar to each other. Although VANET is a kind of distributed system that every vehicle may have different contention collision probabilities, the borrowing scheme cannot work if everyone borrows different numbers of slots. Vehicles with different numbers of borrowed slots within two-hop range are hard to cooperate with vehicles running on the opposite direction. Therefore, the vehicle that first makes the borrowing decision needs to decide the number of slots for borrowing and other borrowing neighbors received the borrowing information only follow this number of borrowed slots.

For slot returning, we consider two cases. One is that vehicles with borrowed slots can handle the slot competitions well without borrowing extra slots. That is, the traffic density in this direction decreases. The other possible case is that vehicles who lent out slots cannot afford such lending anymore, which means the traffic density in this direction increases. Thus, two returning thresholds T_{RL} and T_{RH} and two returning conditions are set for each slot borrowed vehicle to decide whether it should return the slots or not:

1. The contention collision probability on the same direction is smaller than the Low Returning Threshold T_{RL}

2. The contention collision probability on the same direction is smaller than the High Returning Threshold T_{RH} and the one on the opposite direction is not smaller than the T_{RH}. Meanwhile, their contention collision probability difference is larger than the difference threshold ΔT

The returning condition 2 is similar to the borrowing condition. Once a vehicle with borrowed slot finds that either the condition 1 or condition 2 is satisfied, it will return its current borrowed slots by un-marking those borrowing flags. Generally, T_{RL} is set as a fixed small percentage (like 2%) and T_{RH} is initially set as the original borrowing threshold T_B. Both borrowing T_B and returning threshold T_{RH} are dynamically adjusted. Similar to the borrowing processing, all the one-hop neighbors received the returning information should check their own contention collision probability. If one of the returning conditions is satisfied, the returning information will be prorogated and the borrowed slots will be returned.

Dynamic Threshold Setting

Thresholds are used for making the borrowing and returning decisions. While Difference Threshold ΔT and Low Returning Threshold T_{RL} are fixed value, Borrowing Threshold T_B and High Returning Threshold T_{RH} are dynamically adjusted base on the real-time traffic conditions and previous borrowing decisions. This is for avoiding unnecessary slot adjustments and slot thrashing problem. If the contention collision probabilities of both directions become larger than the current T_B but the difference between them is still smaller than the ΔT, the slot borrowing is not allowed and T_B will be adjusted to $T_B + \alpha$. Here α is a constant number stands for the adjusting percentage. On the other side, if the contention collision probability for one of the directions reduces to smaller than $T_B - 2\alpha$, the borrowing threshold will be adjusted to $T_B - \alpha$.

When the slot borrowing happen, T_{RH} is initially set as current T_B and then the borrowing threshold is changed to $T_B + \alpha$. Such adjustment of borrowing threshold is used for vehicles that already borrowed some slots but may need to borrow more with heavier traffic density. During slot borrowing, if the contention collision probabilities of both directions become larger than the current returning threshold T_{RH} but the difference between them is still smaller than the ΔT, T_{RH} will be changed to $T_{RH} + \alpha$. And if the contention collision probability on one of the directions becomes smaller than the $T_{RH} - \alpha$, the returning threshold will be adjusted to $T_{RH} - \alpha$.

4.1.4 Contention Collision Probability Estimation

In order to make the slot borrowing and returning decisions, vehicles need to estimate the contention collision probability P_C for both directions. P_C is computed based on the number of competing vehicles M and the number of empty slots N. The number of empty slots N can be easily obtained from observing the slot arrangement at each frame. Some papers have analyzed the contention collision probability and suggested appropriate number of empty slots for different numbers of competing vehicles [57]. However, a practical issue that has not been mentioned is how to estimate the number of competing vehicles. In this dissertation, we estimate the number of competing vehicles M based on slot observation and packet collision detection. Slot observation provides the number of competing vehicles through possibility point of view, while collision detection is helpful for improving the estimation accuracy. Assuming there are totally $M(i)$ vehicles competing at frame i, the number of successful reserved vehicle $M_S(i)$ can be learned through observing and comparing the slot arrangements at frame i and $i+1$. A vehicle also has the knowledge of the number of empty slots $N(i)$ at frame i. The probability $P_S(i)$ that a competing vehicle can successfully reserve a slot at frame i is computed as Equation 4.1:

$$P_S(i) = (1 - \frac{1}{N(i)})^{(M(i)-1)} \tag{4.1}$$

Having $M(i) \times P_S(i) = M_S(i)$, we can solve the only unknown factor M by Lambert Function as Equation 4.2:

$$M(i) = \frac{W[M_S(i) \times (1 - \frac{1}{N(i)}) \times ln(1 - \frac{1}{N(i)})]}{ln(1 - \frac{1}{N(i)})} \tag{4.2}$$

Besides estimating the number of competing vehicles through the possibility of successful competing, we improve the estimation accuracy by adding collision detection information. If a vehicle finds that no collision happen at frame i, the number of competing vehicles will be exactly the number of successful competing vehicles $M_S(i)$. On the other hand, if a vehicle finds that there are $C(i)$ collisions happened at frame i, it learns that there are at least $2C(i)$ vehicles that failed the contention. Therefore, the final estimation of the number of competing vehicles $M'(i)$ can be expressed as Equation 4.3:

$$M'(i) = \begin{cases} M_S(i) & \text{if C(i)} = 0 \\ MAX[M(i), M_S(i) + 2C(i)] & \text{if C(i)} \neq 0 \end{cases} \tag{4.3}$$

4.1.5 Simulation and Performance Evaluation

We evaluate the performance of our proposed MAT-MAC protocol and compare it with ADHOC MAC [39], Even-Odd MAC [60] and VeMAC [55]. MATLAB and SUMO are used to construct a simulation environment. SUMO is a traffic simulator that generates a real world mobility model includes road map and vehicles' moving patterns. MATLAB is used for building a VANET communication environment with the MAC protocols implementation. As shown in Figure 4.3, we use part of highway I-94 MN for our simulation. There are several entrances and exits on the highway. Vehicles can enter the highway from any one of the entrances or the beginning of the road while leave through any one of the exits or the end of the road. Different Traffic Balanced Rates are tested in the simulation. The Traffic Balance Rate (TBR) is defined as the ratio of the number of vehicles in the direction with sparser traffic to the number of vehicles in the direction with denser traffic. Therefore, the traffic densities are exactly the same for the two directions when TBR is equal to 1 and a small TBR means a scenario with severe unbalanced traffic densities. The detailed simulation parameters are summarized in Table 4.1.

Although VeMAC only evaluated the performance of VeMAC-0 which has no tolerance for the contention collision and VeMAC-inf that has infinite tolerant for the contention collision, we also test the performances of VeMAC-1 and VeMAC-5. VeMAC-1 and VeMAC-5 allow vehicles to contend for all the available slots after 1 and 5 times contention failures respectively. Note here that the slot arrangement of VeMAC-0 is similar to the ADHOC MAC while the VeMAC-inf equals to the strict Even-Odd MAC.

Figure 4.3: Road Map for Simulation [4]

Table 4.1: Parameter Settings for MAT-MAC Simulation

Highway Road Length	5000m
Vehicle Max Speed	30m/s
Number of Slots in Each Frame	80
Per Frame Length	0.1s
DSRC Communication Range	300m
Total Number of Generated Vehicles	600
Simulation Time	400s

Number of Collisions

We first evaluate the total number of packet collisions, including both contention collisions and encounter collisions. As shown in Figure 4.4, every bar is separated into two parts by a black line. The part below the line stands for the number of contention collisions while the part above the line is the number of encounter collisions. With 0.8 and 1 TBR, all the protocols except ADHOC MAC work great with a small number of collisions. Since no slot partition is used, ADHOC MAC suffers large number of encounter collisions. On the other hand, the number of contention collisions of ADHOC MAC is not greatly impacted by different TBRs. Since the traffic densities are kind of balanced for 0.8 and 1 TBRs, VeMAC-5 and our MAT-MAC work similar to Even-Odd and no slot borrowing is involved. VeMAC-1 introduces a few extra encounter collisions since vehicles borrow slots from the opposite direction even if the traffic densities are balanced.

However, when we decrease the TBR to 0.4 and 0.2, the performance of Even-Odd MAC sharply degrades with large number of contention collisions since no slot adaptation scheme is applied. For VeMAC-5, a vehicle has to wait for five times of contention collisions before it can borrow slots. Therefore, the number of contention collisions is not efficiently decreased. Both VeMAC-5 and VeMAC-1 introduce additional encounter collisions from vehicles running to the opposite directions because vehicles randomly borrow slots from the other direction, especially for VeMAC-1. Since VeMAC does not adapt the slot based on the real-time traffic condition, the timing of slot borrowing may be unsuitable. For our MAT-MAC, extra slots are borrowed before a vehicle begins the slot contention based on the contention collision probability. The migration scheme is also helpful for avoiding the encounter collisions from vehicles driving to the opposite directions. Therefore, our MAT-MAC protocol performs the best with the least number of packet collisions for different TBRs. With 0.2 TBR, comparing with ADHOC MAC, Even-Odd, VeMAC-1 and VeMAC-5, MAT-MAC has 78.1%, 78.6%, 42.1% and 56.7% fewer collisions respectively.

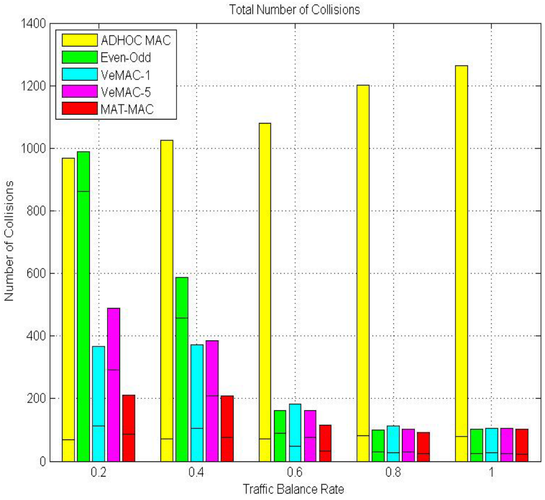


Figure 4.4: Total Number of Packet Collisions

Slot Utilization

Slot utilization is another important factor for evaluating TDMA based MAC protocols. It is defined as the ratio of number of vehicles that have occupied slots within two-hop range to the total number of slots in a frame. Since different vehicles have different slot arrangements, we measure the average slot utilization from all the vehicles at every time unit and focus on the maximum slot utilization during the simulation. As shown in Figure 4.5, MAT-MAC has better performance since it dynamically adapts the numbers of slots of the two directions based on the real-time traffic condition. Without slot adaptation, the slot utilization of Even-Odd is substantially impacted by small TBRs. Even if there are still a lot of empty slots left for the direction with sparse traffic, the direction with dense traffic suffers high contention collision probability or even faces the predicament that no empty slot left for this direction. Since VeMAC-1 and VeMAC-5 cannot efficiently reduce the number of contention collision and introduce additional encounter collisions, their performances are not significantly improved. Although ADHOC MAC is not greatly impacted by different TBRs, the large number of encounter collisions potentially influences its performance. With 0.2 TBR, MAT-MAC has 8.7%, 5.4% and 7.6% higher maximum slot utilization comparing to Even-Odd, VeMAC-1 and VeMAC-5 respectively.

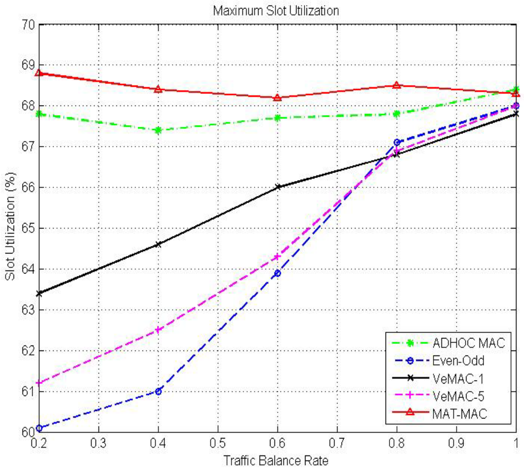


Figure 4.5: Maximum Average Slot Utilization

Table 4.2 shows the number of unnecessary failed contentions that happen when all the slots assigned for the direction with dense traffic have been reserved but there are still some empty slots left for the direction with sparse traffic. These contention failures are unnecessary since they can be fully prevented if the number of available slots for each direction is well adjusted. Such contention failure does not happen with 0.6, 0.8 and 1 TBRs since the differences of the traffic densities on the two directions are not large enough. Although VeMAC-1 and VeMAC-5 can somehow reduce the number of contention failures, they cannot fully avoid such unnecessary failures. This is because that the slot borrowing is not allowed before contention collisions really happen even current contention collision probability is already very high for one direction. For both ADHOC MAC and MAT-MAC, no such contention failure happens no matter what TBR has been set. While ADHOC MAC does not employ disjoined slot sets for vehicles in different directions, our MAT-MAC is able to adjust the number of available slots for each direction to prevent the high contention collision probability. Therefore, the slots for the direction with dense traffic will not be used up unless both directions have really high traffic densities.

Table 4.2: Number of Unnecessary Contention Failures

MAC Protocol	TBR=0.2	TBR=0.4	TBR=0.6, 0.8, 1
ADHOC MAC	0	0	0
Even-Odd	236	66	0
VeMAC-1	16	8	0
VeMAC-5	64	38	0
MAT-MAC	0	0	0

Contention Collision Probability

Besides, we investigate the effect of different TBRs on the probability of contention collision which is the ratio of the number of contention collisions to the total number of contentions. Figure 4.6 displays the contention collision probabilities of the five MAC protocols. When TBR is set as 0.8 or 1, all the four protocols using slot partition perform better than AHDOC MAC around 1% since the number of competing vehicles is decreased for each direction. However, the contention collision probabilities of Even-Odd and VeMAC-5 severely increase for 0.2 and 0.4 TBRs. Both VeMAC-1 and MAT-MAC are helpful for preventing high contention collision probability. But since MAT-MAC can adapt the number of slots for each direction before the happening of high contention collision probability, it decreases the contention collision probability more efficiently than the VeMAC-1. Comparing to Even-Odd, VeMAC-1 and VeMAC-5, MAT-MAC has 44%, 1.6% and 17% smaller contention collision probabilities respectively when TBR is equal to 0.2. Although ADHOC MAC has the smallest contention collision probabilities under unbalanced traffic scenarios, the large number of encounter collisions significantly degrades its performance for two-way traffic.

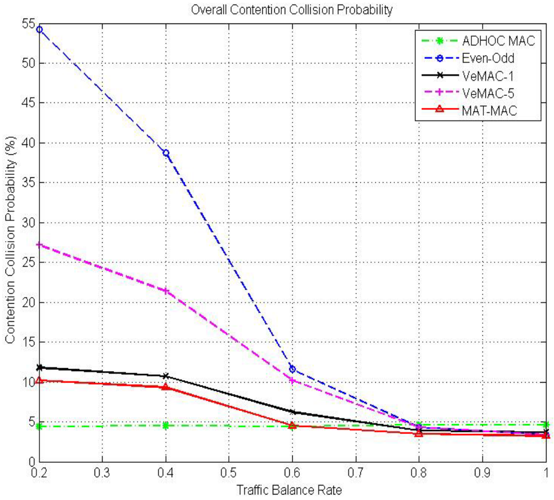


Figure 4.6: Overall Contention Collision Probability

4.1.6 Conclusion

In this dissertation, we propose a new Migration-based Adaptive TDMA MAC protocol (MAT-MAC) for decreasing the number of both encounter collisions and contention collisions while maintaining high slot utilization even under the scenario with unbalanced traffic densities. The number of slots assigned for each direction can be dynamically adjusted based on current traffic condition and the estimated contention collision probability. Our simulation results show the effectiveness of the proposed protocol on reducing the total number of collisions and keeping high slot utilization.

4.2 PTMAC: A Prediction-based TDMA MAC Protocol for Reducing Packet Collisions in VANET

4.2.1 Introduction

Although MAT-MAC shows it efficiency on reducing the number of encounter and contention collisions while maintaining high slot utilization, it cannot eliminate the encounter collisions among vehicles along the same direction. Vehicles currently behind may catch up with and pass over other vehicles. Furthermore, none of the previously proposed MAC protocols work well at four-way intersections which are much more complicated than the two-way traffic. Encounter collisions caused by vehicles from different directions cannot be avoided anymore. Therefore, instead of using slot partition method, we design a new Prediction-based TDMA MAC protocol (PTMAC) to reduce the probability of encounter collision. To the best of our knowledge, PTMAC is the first protocol that is designed for both two-way traffic and four-way intersections. Our main contributions can be summarized as follows:

- Designing a new Prediction-based TDMA MAC protocol (PTMAC) for decreasing the probability of encounter collisions while maintaining high slot utilization and without introducing additional overheads. Most of the encounter collisions can be predicted and potentially eliminated before they really happen. The prediction is based on the vehicle information that is already provided to support safety related application.
- Our newly designed PTMAC protocol is demonstrated to be suitable for both two-way traffic scenario and four-way intersections in an urban area. Unbalanced traffic densities will not degrade the performance of PTMAC.
- Through measuring and comparing our PTMAC protocol with the basic ADHOC MAC protocol in [8] and Even-Odd TDMA MAC in [11], we show that PTMAC has better performance with smaller number of collisions and higher delivery rate for both two-way and four-way intersection scenarios regardless of the traffic loads on different road segments.

In the following, we introduce our proposed PTMAC protocol under two-way scenarios in Section 4.2.2 and extent it for four-way intersections in Section 4.2.3. Detailed performance analysis of ADHOC MAC, Even-Odd MAC and our proposed PTMAC is provided in Section 4.2.4. In Section 4.2.5, we evaluate the performance of our PTMAC protocol and compare it with other two TDMA-based MAC protocols proposed in previous studies and Section 4.2.6 gives a conclusion of our work.

4.2.2 PTMAC Protocol for Two-way Traffic

We have made an important observation that most of the encounter collisions can be predicted and potentially avoided based on vehicles' moving patterns and the traffic condition. Therefore, instead of using slot partition method, we propose a novel MAC protocol that takes advantage of prediction to remove potential collisions. Our PTMAC protocol is described under two-way traffic scenario in this section and it will be extended to four-way intersections in the next section. For both two-way and four-way scenarios, there are three steps need to be processed in PTMAC protocol: Potential Collision Detection, Potential Collision Prediction and Potential Collision Elimination. A potential collision needs to be detected first. Then we can predict if this potential collision will really happen in the future based on the real-time traffic condition and moving patterns of vehicles. Finally, we reschedule the slots to eliminate this potential collision. Detailed descriptions of these three steps will be provided in the following of this section. Notice that the collisions we mention here means encounter collisions, so as the following of this work, unless we point out that it is a contention collision. Recall that in a TDMA based protocol, each vehicle will first contend for an empty slot in a frame. It will continuously use this slot if it successfully transmitted first time. A contention collision happens if multiple vehicles within two-hop communication range contend for the same slot. An encounter collision is caused by two vehicles approaching each other while using the same slot in a frame.

Assumptions

Firstly, some assumptions are made based on the basic TDMA MAC protocol that has been proposed in VANET:

1. Every vehicle broadcasts a message at every frame which includes its own and each one-hop neighbor's location, speed, moving direction and occupying slot information. Most of the information are required by safety related applications.
2. Every vehicle keeps the information about its one-hop and two-hop neighbors which are shared by its one-hop neighbors.
3. Each newly joining vehicle that has not obtained a slot and wants to get a slot needs to listen to the channel for one frame. Then, they can randomly choose an available

slot at the next frame for transmission.
4. Each vehicle is equipped with a GPS device that provides the information about its own location, moving direction and speed. Road information like road length is also available. Such information can also be obtained from RSU broadcasting.

The assumptions about the slot information follow the ADHOC MAC proposed in [39]. ADHOC MAC does not need additional information other than the frame information and works well for one-way traffic. PTMAC needs more vehicle information for predicting the potential encounter collisions. Fortunately, information like vehicle speeds, locations and moving directions are generally required by most of the applications for safety purposes. Thus, PTMAC can directly take advantage of this information without introducing additional overheads.

To support most of the safety related applications, USDOT considers two types of safety messages as helpful for disseminated: event-driven messages and periodic messages [20]. The Event-driven messages are sent when a dangerous condition is detected. Meanwhile, the periodic messages are broadcasted by every vehicle periodically. It usually contains the vehicle status information like speed, position and moving direction. Since each vehicle is able to aware of its neighbor vehicles, unsafe situation can be avoided. This type of packets is required to be broadcasted frequently enough in order to provide the most updated information. The Vehicle Safety Communications consortium (VSCC) suggests that the periodic messages should be broadcasted at a frequency of at least 10 messages per second. Example applications identified by VSCC include traffic signal violation warning, curve speed warning, emergency electronic brake lights, pre-crash warning, cooperative forward collision warning, left turn assistant, lane-change warning, and stop sign movement assistant [19]. The packet size basically ranges from 200 to 500 bytes [61] and SAE [62] defined over 70 data elements and how many bits each element required. About 30 elements are commonly used for safety related applications. Therefore, there is typically no additional overhead introduced by PTMAC since the vehicle information has already been provided for ensuring the traffic safety.

Potential Collision Detection

We start from the first step of our PTMAC protocol: how to detect a potential encounter collision. In order to detect the potential collisions that may happen between vehicles currently out of each other's two-hop range, each vehicle needs to learn the information of its neighbors at least three-hop away. The most naive solution is to require every vehicle to broadcast the information of its two-hop neighbors in addition to its one-hop neighbors. The major drawback of this approach is that significant overheads will be introduced with longer packet length. Therefore, to avoid such additional overheads, we use "intermediate vehicles" to detect the potential collisions between vehicles currently out of two-hop range. Since each vehicle is able to obtain the information of its two-hop neighbors from its one-hop neighbors, the intermediate vehicles are able to get knowledge of its two-hop neighbors ahead and two-hop neighbors behind. In this way, these intermediate vehicles can detect potential collisions between vehicles three or four-hops away that are reserving the same slot. This is the most essential observation for our proposed protocol. Notice that if the traffic density is very low, intermediate vehicle may not exist between two vehicles with a potential collision. In this case, the PTMAC protocol still works and performs similar to ADHOC MAC. Vehicles get encounter collision will re-contend for an available slot to transmit packet. Meanwhile, the packet collision will not become a problem in such sparse traffic condition.

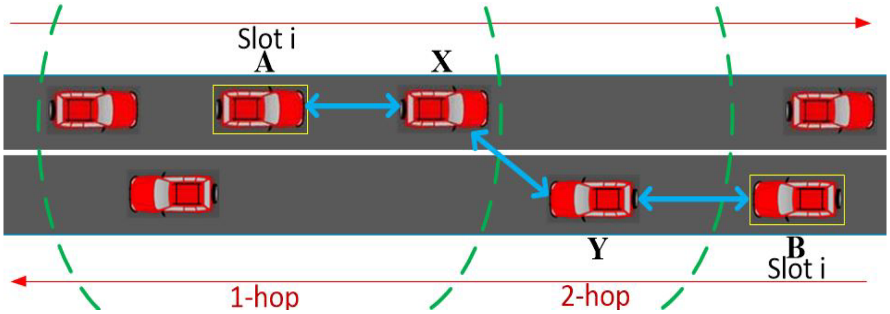


Figure 4.7: Potential Collision Detection for Vehicles from the Same Direction

The process of potential collision detection can be described as: Based on the message containing the frame information received from other vehicles (one-hop neighbors), every vehicle needs to check if any two of its one-hop or two-hop neighbors are occupying the same time slot. Every vehicle learns the information of its two-hop neighbors from its one-hop neighbors. So a potential collision can be detected among two vehicles at most four-hop away. Actually, since each vehicle tries to avoid reserving the same slot with other vehicles within two-hop range, a potential collision can only be detected between two vehicles that are three-hop or four-hop away. However, since two vehicles with four-hop potential collision are still far away from each other and will be safe for a period of time, we only need to concern the potential collisions detection for vehicles that are thee-hop away. For example, Figure 4.7 and 4.8 display the potential collisions that are detected among vehicles at the same direction and opposite directions respectively. In both cases, vehicles A and B are currently three-hop away. They are occupying the same slots i but they cannot find this potential collision by themselves. Instead, the intermediate vehicles X and Y have the slot information about both A and B, so they are able to detect this potential collision between A and B.

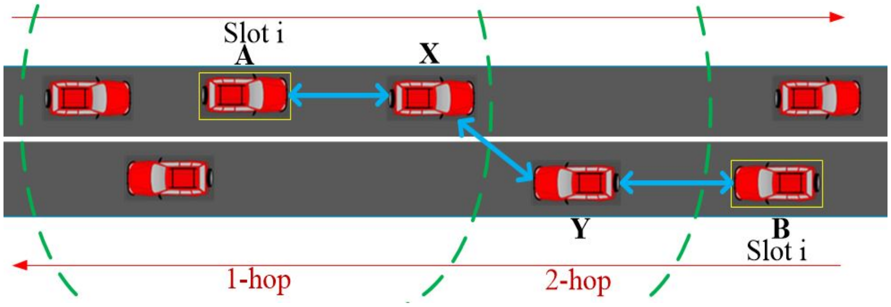


Figure 4.8: Potential Collision Detection for Vehicles from the Opposite Directions

Potential Collision Prediction

When an intermediate vehicle I detects a potential collision between two other vehicles, let's say vehicle A and B, it needs to predict whether A and B will "encounter" each other and the potential collision is really going to happen. The "encounter" means that two vehicles come into two-hop communication range of each other. This prediction can be done based on the current locations, the speeds and moving directions of these two vehicles. A potential collision that is predicted to happen is considered as "active". We classify the potential encounter collisions into two types: potential collisions among vehicles running at the same direction and among vehicles driving at the opposite directions. Different methods will be used for these two types of potential collisions to predict if they are active or not.

Firstly, we explain the same direction potential collision prediction. For vehicles running along the same direction, they are likely to catch up with each other if the one behind has much faster speed. The distance between two vehicles may shorten to or less than two-hop communication range (2R) in a short duration of time from now due to their speed difference. Assuming vehicle A locates behind B and they are occupying the same slot, this potential collision is regarded as active if the distance between them can be reduced to 2R in a short duration of times, where R is the communication range of a vehicle. This is shown in Equation 4.4.

$$\begin{cases} (V_a - V_b) \times T \geq D - 2R \qquad (if V_b < V_a) \\ T = min\{K, \dfrac{L_b}{V_b}\} \end{cases} \tag{4.4}$$

V_a and V_b are the speeds of vehicle A and B respectively. L_b is the length of the road that B has not finished and D is the current distance between A and B. T stands for a short duration time which is used to check if vehicle A and B can run into two-hop range of each other within this short duration T or not. It is unnecessary for a potentially collided vehicle to change its slot too early. If two potential collided (in terms of transmission) vehicles will not encounter each other within time T, the potential collision can be removed later. K represents a short period of time that enables a potentially collided vehicle to change its slot with high success probability. It is still possible that a potentially collided vehicle can safely switch its slot to a new slot, but get a potential collision with another three-hop neighbor. If we set a larger K, the potentially collided vehicle will have multiple chances to switch its slot and higher probability of removing the collision can be achieved. On the other hand, if we set K to be too large, the original slot for the potentially collided vehicle will be open for competition and other vehicles may take over this slot. In this case, a new potential collision may appear right away and the whole process needs to be done again. Therefore, a smaller K can save resource and slot utilization. T equals to either K or the time before B leaves the road, depends on whichever is smaller. Since T is a really short period of time (for example, less than 1s), we regarded V_a and V_b as constant within T and their variability is less important. If A is faster than B and Equation 4.4 is satisfied, this potential collision is considered as active and has to be eliminated. Otherwise, if V_a is not larger than V_b or Equation 4.4 is not satisfied, this potential collision is currently harmless.

For vehicles running at opposite directions, potential collisions may be detected between vehicles that are running toward each other or farther away from each other. An example is shown in Figure 4.9. Vehicles A and B are reserving the same slot and driving toward each other. Thus, the potential collision detected by intermediate vehicle I_1 and I_2 will definitely happen in the future. On the other hand, if intermediate vehicle I_1' and I_2' detect a potential collision between vehicle A and B', this collision can be ignored since A and B' are running farther away from each other. We set two conditions for intermediate vehicles to check if the potential collided vehicles are approaching or running farther away from each other:

1. The intermediate vehicle finds that one of the potential collided vehicles which running at the same direction locates behind it.
2. Meanwhile, the other potentially collided vehicle which running at the opposite direction locates ahead of it.

If both conditions are satisfied, the intermediate vehicle knows that two potential collides vehicles are approaching each other. Otherwise, this potential collision can be ignored. Assuming the DSRC communication range is 300m and vehicle speeds are around 30m/s (67Mph) on highway, for a three-hop away potential collision, the time to shorten the distance between two potential collided vehicles to two-hop range is about 5s. Therefore, there are plenty of times for a potentially collided vehicle to change its slot since every vehicle needs to broadcast its information at least every 100ms which is required by most of the safety related applications. Similar to the same direction collision cases, only if the distance between A and B can be reduced to 2R in a short duration of time T, the potential collision is considered as active. This is shown in Equation 4.5.

$$(V_a + V_b) \times T \geq D - 2R \tag{4.5}$$

Where T equals to the K in Equation 4.4. When an intermediate vehicle finds that two potential collided vehicles are approaching each other and Equation 4.5 is satisfied, it regards this potential collision as active.

Potential Collision Elimination

If an active potential collision is found, we need to prevent this collision to happen in the near future. One of the potential collided vehicles needs to give up its current

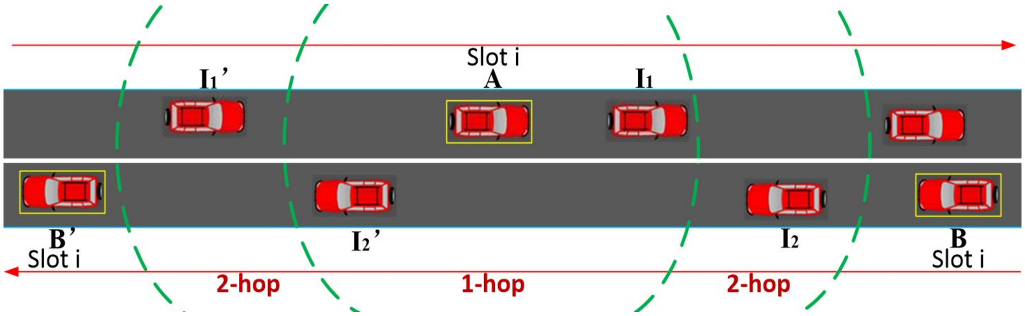
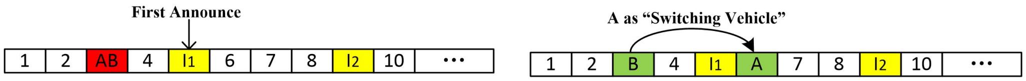


Figure 4.9: Potential Collision Prediction for Vehicles from the Opposite Directions

reserved slot and switch to another available slot. Since there may be multiple intermediate vehicles detect the same potential collision, we need to select one of them to handle this potential collision. Then the selected intermediate vehicle has the responsibility to decide which one is the "switching vehicle" to release its current reserved time slot. Finally, the switching vehicle needs to switch to another empty slot after receiving a switching notification from the responsible intermediate vehicle. Recall that we only focus on the potential collisions detected between vehicles three-hop away.

The basic rule is selecting the potentially collided vehicle which is a one-hop neighbor of the responsible intermediate vehicle as the switching vehicle. There may be more than one intermediate vehicle that can detect the same potential collision. When an intermediate vehicle finds an active potential collision, it first listens to the channel until its own reserved slot comes. If it has not received any notification from others about this active potential collision, it becomes the responsible intermediate vehicle to broadcast a notification about this potential collision. Meanwhile, the potentially collided vehicle within one-hop of this responsible intermediate vehicle is selected as the switching vehicle. In this way, the responsible intermediate vehicle can directly inform the switching vehicle without further forwarding. Notice here that the intermediate vehicles detect the same potential collision must be in the communication range of each other. Thus they are able to receive the notification about the potential collision from each other.

A one bit flag will be added into the broadcasted Frame Information (FI) of the responsible intermediate to indicate that a slot has an active potential collision. Assuming



(a) Before Potential Collisions Elimination (b) After Potential Collision Elimination

Figure 4.10: Example of Potential Collision Elimination

slot i is currently occupied by the switching vehicle A, the responsible intermediate vehicle will broadcast its FI with an active flag on slot i. Therefore, when vehicle A receives the FI from the intermediate vehicle, it finds its slot will conflict with another vehicle and can conclude that it has to change its slot. Other intermediate vehicles found the same active potential collision can just broadcast their FIs without adding an active flag on the same slot. After the switching vehicle changes to a new slot, it will update its FI and transmit using its new slot. Vehicles that received such updated FI from A will update their frame information. We can also allow a switching vehicle to use its original slot one more time to preannounce which slot it will switch to. In this way, other vehicles received such messages can avoid selecting the same slot. The contention collisions among multiple switching vehicles from different potential collisions and newly joining vehicles can be prevented.

We take Figure 4.9 as an example and the original slot arrangement is shown as Figure 4.10 (a). Vehicle A and B are occupying the same slot 3. This potential collision will be detected by both the intermediate vehicle I_1 and I_2. Since I_1 has not received any notification about the detected potential collision, it will become the responsible intermediate vehicle who needs to broadcast a notification about this potential collision. As I_1's one-hop neighbor, vehicle A will be selected as the "switching vehicle". Meanwhile, since the slot of vehicle I_2 is behind the slot of I_1, I_2 is able to hear the switch notification from I_1 and does not need to broadcast a duplicated notification. As shown in Figure 4.10 (b), when vehicle A receives the notification, it will randomly switch to another available slot. After A switches to a new slot, all its neighbors will update their Frame Information about A. Therefore, the potential collision between vehicle A and B can be eliminated before it really happens.

4.2.3 PTMAC Protocol for Four-way Traffic

More Assumptions and Traffic Model

After explaining our PTMAC under two-way traffic scenario, we extend it into a four-way intersection scenario. Vehicles can drive at four possible directions: North, South, West and East. We consider a Traffic Light Model in which the intersection has traffic light for controlling the traffic from all four directions. Vehicles can go straight, turn right, turn left or make a U-turn at a four-way intersection. More assumptions besides what we mentioned in Section 3 are made for four-way traffic:

1. Each vehicle periodically broadcasts their turning direction at the coming intersection before passing the intersection. This information can come from the turning left/right signal or from GPS device base on a predetermined route.
2. The location of an intersection and the phases of its traffic lights are provided by RSU broadcasting.

The PTMAC protocol still processes with the three steps for four-way intersection scenario. The steps of Potential Collision Detection and Potential Collision Elimination are similar to what we described in the two-way traffic scenario. We will focus on explaining the most different part: Potential Collision Prediction under four-way scenario. We further separate the Potential Collision Prediction into two parts: road segment prediction and intersection prediction. The road segment prediction concerns the collisions between vehicles running on the same road segment, either heading the same direction or the opposite directions. The intersection prediction pays attention to the potential collisions among vehicles driving on different road segments while approaching to or leaving the intersection. Since the road segment prediction is the same as what we has explained in two-way traffic scenario, we concentrate on describing the intersection prediction which is used to check whether two vehicles are currently out of two-hop range and reserving the same slot will encounter each other or not.

Potential Collision Prediction at An Intersection

We take an example of four-way intersection scenario as shown in Figure 4.11 to explain our PTMAC protocol. Vehicles A and B are occupying the same slot i and are currently three-hop away from each other. After the intermediate vehicle X and Y detected this potential collision, they need to predict whether this collision is active or not. We consider the potential collision prediction in three possible cases based on the current locations of two potentially collided vehicles A and B. The first case is that both A and B have passed the intersection. The second case is that one of them has passed the intersection while the other one has not. The third case is that none of them have passed the intersection.

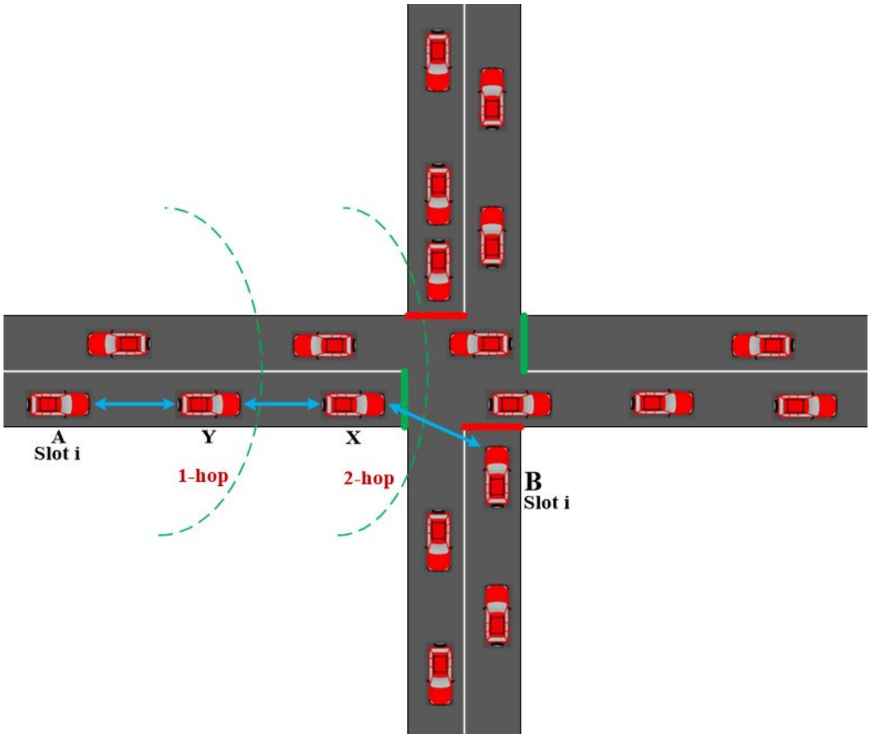


Figure 4.11: Example of Potential Collision Prediction for Four-way Traffic

In the first case, both A and B already passed the intersection and are driving away from the intersection. If A and B have turned to different directions, they are running farther away from each other and no collision will happen. If A and B have turned to the same direction, the prediction problem becomes the road segment prediction. The second possible case is that A has already passed the intersection, but B has not. B will turn to the same direction or different direction with A's. If B did not encounter A before it passes the intersection and turns to the same direction as A, the problem will become road segment prediction again. On the other hand, if B did not encounter

A before it passes the intersection and turns to a different direction, their distance will become larger and they have no more chance to encounter each other. Thus, we only need to check whether the potential collision is active before B passes the intersection.

We also consider the third case that neither A nor B has passed the intersection. The distance between these two vehicles will be shortened before one of them passes the intersection. Assuming A passes the intersection first, if A turns to the opposite direction of B's current driving direction of the same road segment, then the problem becomes road segment prediction. If A turns to other directions, then the problem becomes similar to the second case. Thus, for this case, we only need to check whether the potential collision is active or not before one of A and B passes the intersection. We summarize all the cases in Table 4.3.

Table 4.3: Intersection Collision Prediction with Different Positions of Vehicle A and B

A passed	B passed	Action of Intermediate Vehicle
√	√	N/A
√	×	Check if potential collision is active before B passes intersection
×	×	Check if potential collision is active before one of A and B passes the intersection

In order to check whether two vehicles can encounter each other before passing the intersection, we first need to compute when they will pass the intersection. Based on a travel time estimation model proposed in [63], we develop an improved model to estimate the travel time that a vehicle needs before it passes the intersection. We provide more accurate estimation with the help of VANET communication and real-time information. The total travel time T_t from vehicle's current position to the intersection is separated into two components: signal delay time T_s and cruise time T_c. Vehicles' behaviors are complicated when they approaching to the intersection and they are greatly influenced by the traffic lights, especially when they need to wait for a red signal. We use the simplified Webster formula which is the widely used model to estimate the signal delay. It is computed based on the traffic light phases, traffic volume (vehicles/sec) and degree of saturation. The degree of saturation is the traffic volume to capacity ratio. The capacity is the maximum rate at which vehicles can pass through a point in a period of time. Both the capacity and the traffic volume can either be estimated by an individual vehicle or from RSU broadcasting. Knowing the current locations and speeds of all its two-hop neighbors, an intermediate vehicle can compute the number of vehicles passing a point in a period of time and then get the traffic volume. Actually, this information is also widely used in many traffic control applications like intelligent traffic light [64]. Each vehicle also receives the red and green light phases information at an intersection from RSU broadcasting. Therefore the signal delay can be calculated by the intermediate vehicle.

On the other hand, we modify the model in [63] by computing the cruise time based on real-time traffic information from vehicles periodically broadcasting information instead of using loop detector. The cruise time will be the time cost from a vehicle's current location to the end of the queue at an intersection. Since every vehicle has its two-hop neighbor's location and speed information, one simple way to estimate the queue length is counting the number of vehicles whose speeds are 0. Therefore, an intermediate vehicle is also able to compute the cruise time of two potentially collided vehicles. Actually, the way we compute cruise time is more accurate than that in [63], since they considered the cruise time is from a vehicle's current location to a fix point (loop detector) regardless of the current queue length.

We set T_a and T_b as the total travel times needed for A and B to pass the intersection respectively. L_a and L_b stand for the current distance from A and B to the intersection respectively. V_a and V_b are the speeds of vehicles A and B. We regard V_a and V_b as constant in a short period of time. T_{ac} and T_{bc} are the cruise times of vehicles A and B while T_{aq} and T_{bq} are the waiting times of vehicles A and B respectively. If they do not need to wait for the signal, the cruise time will be exact the total travel time. For the second case that A already passed the intersection, if $V_a < V_b$, then we directly check if Equation 4.6 or 4.7 is satisfied or not. K is the short duration of time that we set for switching vehicle to change slot. If $K < T_b$, Equation 4.6 is used. If $K \geq T_b$, then Equation 4.7 will be checked. In Equation 4.6, the first sub-equation checks two vehicles that are running at different directions and different road segments. The second sub-equation is used for two vehicles running at the same direction but different road segments. Similarly, in Equation 4.7, Y and Y' are used for two vehicles that are running at different directions and the same direction on different road segments respectively. The shortest distance between A and B may appear at two points: when

B reaches the waiting queue and when B is passing the intersection. Therefore, we check both points. X in Equation 4.7 estimates the distance between A and B when B reaches the waiting queue. Y or Y' stands for the distance between A and B when B is passing the intersection. If Equation 4.6 or 4.7 can be satisfied, the potential collision is considered as active.

$$\begin{cases} \sqrt{(L_a + V_a \times T)^2 + (L_b - V_b \times T)^2} \leq 2R \\ (L_a + L_b) + (V_a - V_b) \times T \leq 2R \\ T = min\{K, T_{bc}\} \end{cases} \tag{4.6}$$

$$\begin{cases} X = L_a + V_a \times T_b \\ Y = \sqrt{(L_a + V_a \times T_{bc})^2 + (L_b - V_b \times T_{bc})^2} \\ Y' = (L_a + L_b) + (V_a - V_b) \times T_{bc} \\ min\{X, Y(Y')\} \leq 2R \end{cases} \tag{4.7}$$

For the third case, assuming $T_b > T_a$, that is, vehicle A will pass the intersection first, Equation 4.8 or 4.9 will be used to check if the distance between A and B can reduce to 2R or smaller before A passes through the intersection. If $K < T_a$, then Equation 4.8 is used while Equation 4.9 will be checked when $K \geq T_a$. In Equation 4.8, the first sub-equation checks two vehicles that are running at different directions and different road segments. The second sub-equation is used for two vehicles that are running at the same direction but different road segments. Equation 4.9 checks the distance between A and B when A is passing the intersection. If Equation 4.8 or 4.9 is satisfied, this potential collision is active. Otherwise, we need to check whether the distance between A and B can be reduced to 2R or smaller before B passes the intersection using Equation 4.6 or 4.7. This becomes the same situation as the second case.

$$\begin{cases} \sqrt{(L_a - V_a \times T_1)^2 + (L_b - V_b \times T_2)^2\}} \leq 2R \\ (L_a - V_a \times T_1) + (L_b - V_b \times T_2) \leq 2R \\ T_1 = min\{K, T_{ac}\} \\ T_2 = min\{K, T_{bc}\} \end{cases} \tag{4.8}$$

$$\begin{cases} L_b - V_b \times T \leq 2R \\ T = min\{T_a, T_{bc}\} \end{cases} \tag{4.9}$$

4.2.4 Performance Analysis

In this section, we provide a detailed performance analysis of ADHOC MAC, Even-Odd MAC and our proposed PTMAC protocol.

Probability of Contention Collisions

As a reminder, there are two types of collisions: contention collision and encounter collision. We first investigate the contention collisions probability of the three MAC protocols. N denotes the totally number of slots in each frame. N_E and N_R stand for the number of empty slots and reserved slots respectively. M is defined as the number of new joining vehicles within two-hop communication range. They currently do not occupy slots but are trying to compete for slots. Here we only consider the case that M is larger than 1. Otherwise, no contention collision will happen. For the basic ADHOC MAC protocol, if N_E is greater than 1, the probability that a vehicle among these M competitors can reserve a slot successfully is computed as Equation 4.10. If the N_E is less or equal to 1, the contention collision will definitely happen and no one can make a reservation successfully.

$$P_S = (1 - \frac{1}{N_E})^{M-1} \tag{4.10}$$

If the number of current empty slots is equal to or larger than the number of competitors, that is, $N_E \geq M$, then the probability that all these M vehicles can successfully gain a slot is computed as Equation 4.11. Since PTMAC does not use slot partition method, the way of computing the contention collision probability for PTMAC is similar to ADHOC MAC.

$$P_{ALLS} = \frac{\prod_{i=0}^{M-1}(N_E - i)}{{N_E}^M} \tag{4.11}$$

For Even-Odd MAC protocol, the total number of available slots is halved for each direction. Assuming the traffic densities are totally balanced for both directions, there will be $\frac{N_E}{2}$ empty slots left for each direction. The contention collision can be analyzed in two cases. In the first case, if there are M number of competing vehicles from the same direction and $\frac{N_E}{2}$ is greater than 1, the probability that one of them can reserve a slot successfully is computed as Equation 4.12.

$$P_S = (1 - \frac{2}{N_E})^{M-1} \tag{4.12}$$

Then if $\frac{N_E}{2} \geq M$, the probability that all these M vehicles gain a slot successfully is computed as Equation 4.13.

$$P_{ALLS} = \frac{\prod_{i=0}^{M-1}(\frac{N_E}{2} - i)}{(\frac{N_E}{2})^M} \tag{4.13}$$

In another case, if there are totally M competing vehicles within two-hop communication range, half of them are running to the left while half of them are driving to the right, the probability that one of them can reserve a slot successfully is computed as Equation 4.14. And if $N_E \geq M$, the probability of all these M vehicles gaining slots successfully in this case will be computed as Equation 4.15.

$$P'_S = (1 - \frac{2}{N_E})^{\frac{M}{2}-1} \tag{4.14}$$

$$P'_{ALLS} = (\frac{\prod_{i=0}^{\frac{M}{2}-1}(\frac{N_E}{2} - i)}{(\frac{N_E}{2})^{\frac{M}{2}}})^2 \tag{4.15}$$

Therefore, we can see that when using Even-Odd MAC, more contention collisions are introduced in the first case while smaller contention collision probability is achieved in the second case. However, since more contentions are happen between newly joining vehicles and they are heading the same direction, the first case happens more frequently. The second case is more suitable for the collisions that happen between new joining vehicles and re-competing vehicles or among re-competing vehicles.

For all the three MAC protocols, the contention collisions are not only caused by the newly joining vehicles, but also from the vehicles who have suffered encounter collisions and have to re-compete for new slots. Therefore, reducing the number of encounter collisions is also helpful for decreasing the number of contention collisions.

Probability of Encounter Collisions

An encounter collision is caused by two vehicles that are currently reserving the same slot and out of two-hop range, but will encounter each other in the near future. Assuming there are two newly joining vehicles A and B and they are trying to reserve their slots. If we know that they will encounter in the future (for example, driving at opposite directions and approaching each other), the probability that A and B will select the same slot (an encounter collision will happen) is computed as Equation 4.16.

$$P_{EC} = \frac{N_E(A \cap B)}{N_E(A) \times N_E(B)} \tag{4.16}$$

Notice here that vehicle A and B have different neighbors and slots allocations. $N_E(A)$ and $N_E(B)$ stand for the numbers of empty slots from the view of A and B respectively. $N_E(A \cap B)$ expresses the number of empty slots from both A and B's views. Notice here that for Even-Odd MAC protocol, there is no encounter collision happens between vehicles running at opposite directions. But it cannot avoid the encounter collisions from the same direction. Actually, since the number of available slots is halved in Even-Odd protocol, the probability of the encounter collision from vehicles at the same direction is increased.

Probability of Removing Potential Collisions

In our proposed PTMAC protocol, it is likely that a detected potential collision cannot be successfully removed under heavy traffic density. One possible situation is that there is no other empty slot for the switching vehicle to switch to. Another possible situation is that the switching vehicle switches its slot to a new slot, but this new slot incurs a new potential collision with another vehicle. So under a dense traffic density, this vehicle may have to keep on changing its slot until it finds a slot without potential conflict with others or the collision really happens. Assuming a vehicle A is detected having a potential collision with vehicle B, the probability that this potential collision can be removed at the next frame is expressed as Equation 4.17. Here $N_E(A)$ is the number of empty slots and $A(E)$ expresses the empty slots from A's view. $A(3)$ stands for the three-hop neighbors of A and they will encounter A within T (a short duration). We call those neighbors as three-hop encounter neighbors. So $N(A(E) \cap A(3))$ expresses the number of slots which are empty from A's point of view and meanwhile are occupied by A's three-hop encounter neighbors.

$$P_{RM1} = 1 - \frac{N(A(E) \cap A(3))}{N_E(A)} \qquad (N_E(A) > 0) \tag{4.17}$$

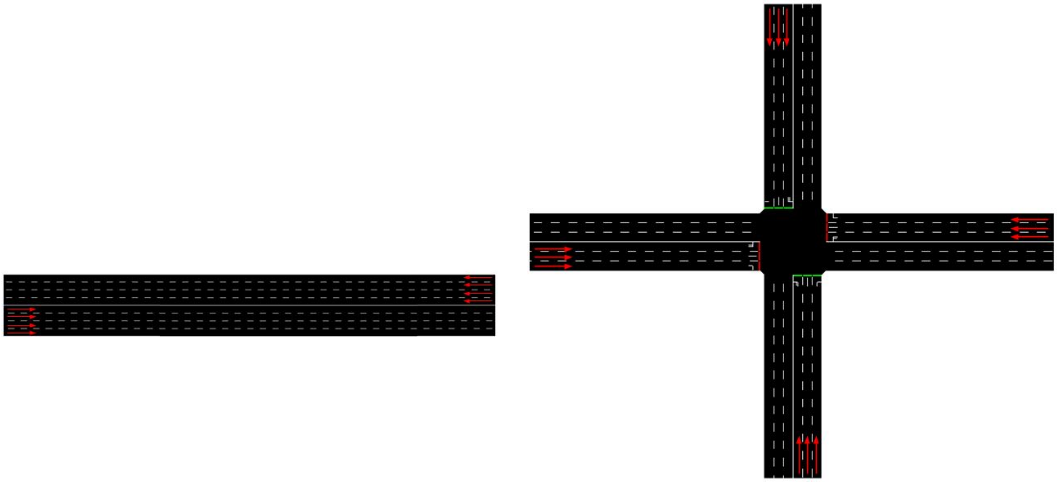
As long as the potential collision has not really happened, vehicle A still has chance to switch to elsewhere. If A has N_{SW} number of chances to change its slot, the probability that vehicle A can eventually remove the potential collision is computed as Equation 4.18. Therefore, we can see that with higher traffic density, a vehicle may need to switch it slot multiple times in order to avoid the encounter collision. This is also the reason that T should not be too small. Otherwise, A only has one or two chances to switch it slot, which may cause failed collision elimination especially under heavy traffic density scenario.

$$P_{RM} = 1 - \prod_{i=0}^{N_{SW}} \frac{N(A(E) \cap A(3)) - i}{N_E(A) - i} \qquad (N_E(A) > i) \tag{4.18}$$

4.2.5 Simulation and Performance Evaluation

In this section, we evaluate the performance of our proposed PTMAC protocol. We use MATLAB and SUMO to construct a simulation environment where both two-way and four-way traffic scenarios are considered. SUMO is a traffic simulator to generate real world mobility models including road map, traffic light information and vehicle's moving pattern. Vehicles' speeds are adjusted based on the traffic condition and traffic light information when they are approaching an intersection. A mobility trace file which contains the position of each vehicle at any time is generated by SUMO and input to MATLAB. MATLAB is used for building a VANET communication environment and implementing the MAC protocols. We compare our PTMAC with a general TDMA based protocol named ADHOC MAC proposed in [39] and an Even-Odd TDMA protocol developed in [60].

The first simulation scenario is a highway with two-way traffic. Vehicles are running at different speeds within different maximum speeds. A vehicle can catch up with and pass over other vehicles if its speed is faster. We measure the performances using different traffic densities. In total 200, 400 and 600 vehicles are generated for each direction during 600s simulation time. We also investigate the impact of unbalanced traffic densities for different directions on these three MAC protocols. The second simulation scenario is an intersection with four-way traffic. There are three lanes for each direction. The right lane is for vehicles turning right, the middle lane is for vehicles go straight while the left lane is used for vehicles turning left or making a U-turn. The number of vehicles that has been generated for each direction is varied from 150 to 200 vehicles within 600s simulation time. Figure 4.12 illustrates the simulation scenarios in SUMO. For both scenarios, based on the report from SAE [62], the packet size is assumed as 400 Byte and the data rate is 6 Mbps.



(a) Two-Way Highway (b) Four-Way Intersection

Figure 4.12: Simulation Maps in SUMO

To ensure the driving safety, the 3-second rule is generally used which suggests that a vehicle should stay 3 seconds behind the vehicle in front of it. For highway scenario, we consider two-way traffic and each direction with 4 lanes. Assuming the average vehicle speed is 30 meters per second (67mph), the distance between the two vehicles should be 90m. Assuming the average vehicle length is 4m, one vehicle will have maximum 48 neighbor vehicles in its communication range (6 vehicles for each lane). For urban area, the average vehicle speed is assumed as 20 meters per second (45mph) and the distance between the two vehicles should be 60m. If there are 3 lanes for each direction, maximum 54 vehicles will be in the communication range (9 vehicles for each lane).

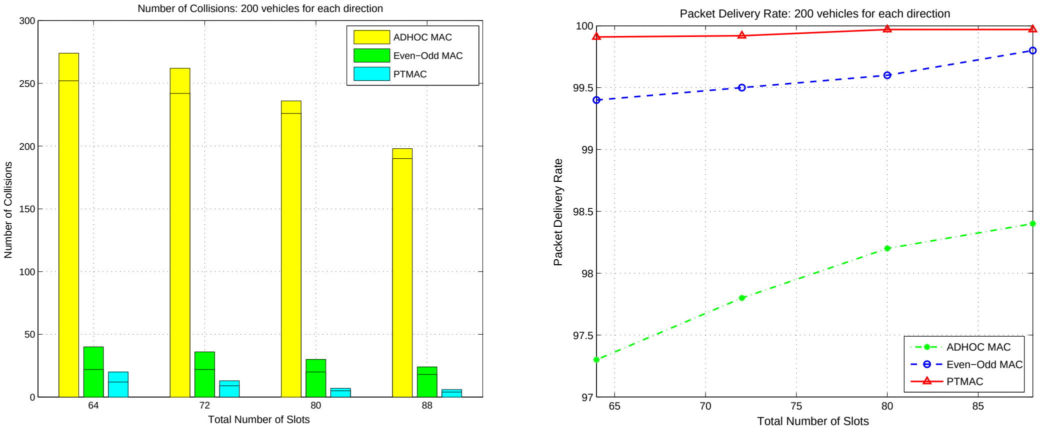
Considering the aggressive driving and stops at the intersection, we vary the number of slots in a frame from 64 to 88 and investigate its impact on the performance. The detailed simulation parameters are summarized in Table 4.4. To focus on the packet collisions, the simulation runs using an ideal physical channel, that is, the packet will be successfully transmitted within the communication range if there is no packet collision.

Table 4.4: Parameter Settings for PTMAC Simulation

Parameter	Highway	Intersection
Highway Road Length	2000m	N/A
Urban Road Length before Traffic Light	N/A	1500m
Vehicle Max Speed	25-32m/s	20-25m/s
Green Light Phase	N/A	42s
Number of Lanes (each direction)	4	3
Traffic Density (number of vehicles per direction)	200, 400, 600	150, 175, 200
Communication Range	300m	300m
Per Frame Length	0.1s	0.1s
Data Rage	6Mbps	6Mbps
Packet Size	400 Bytes	400 Bytes
Simulation Time	600s	600s

Two-way Simulation Results

We first evaluate the performance of these three MAC protocols under two-way traffic with balanced traffic densities. We focus on two metrics: packet delivery rate and total number of collisions. We measure the performances under different traffic densities and different numbers of slots within a frame. Figure 4.13 shows the results of number of packet collisions and packet delivery rate with 200 vehicles generated for each direction. In Figure 4.13 (a), every bar is separated into two parts by a black line. The part below the line stands for the number of encounter collisions while the part above the line is the number of contention collisions. From the results we can see that with the increment of the number of slots, all of the three MAC protocols get better performance since more available slots decreases the collision probability. With 64 slots per frame, our PTMAC works better than ADHOC MAC and Even-Odd with 92.7% and 50% fewer collisions respectively. This is because that PTMAC not only eliminates the collisions among vehicles from opposite directions, but also avoids the collisions from the same direction. The delivery rate of PTMAC also improves by 2.6% and 0.5% comparing to ADHOC MAC and Even-Odd respectively. Besides having the least number of encounter collisions, PTMAC also has fewer contention collisions. The number of contention collisions is affected by the number of encounter collisions since the collided vehicles have to re-compete for slots. Therefore, reducing the number of encounter collisions is also helpful for decreasing the number of contention collisions. Since the traffic density is pretty low in this case, the problem of packet collision is not severe.

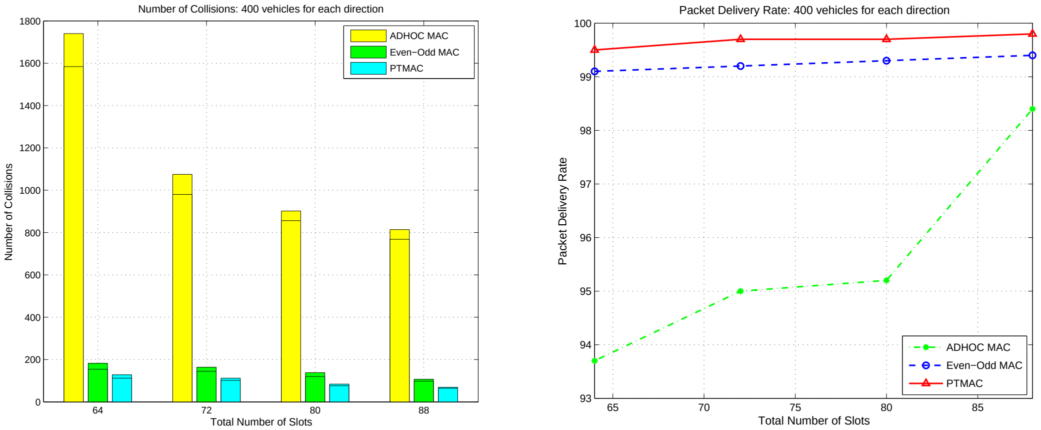


(a) Number of Collisions

(b) Packet Delivery Rate

Figure 4.13: Performance with 200 Vehicles for Each Direction on Highway

Then we increase the traffic density by generating 400 vehicles for each direction. Results about number of collisions and packet delivery rate are shown in Figure 4.14. With 64 slots per frame, PTMAC has 90.6% and 29.7% fewer collisions than ADHOC MAC and Even-Odd respectively. The packet delivery rate of PTMAC improves about 5.8% and 0.4% comparing to the ADHOC and Even-Odd respectively. We continue increasing the traffic density to 600 vehicles for each direction and Figure 4.15 shows the results. Unlike the results we got from previous traffic situations, the number of contention collisions of Even-Odd protocol is sharply increased. Basically, Even-Odd scheme has no great impact on the number of contention collisions among vehicles driving at different directions since both the number of available slots and the number of



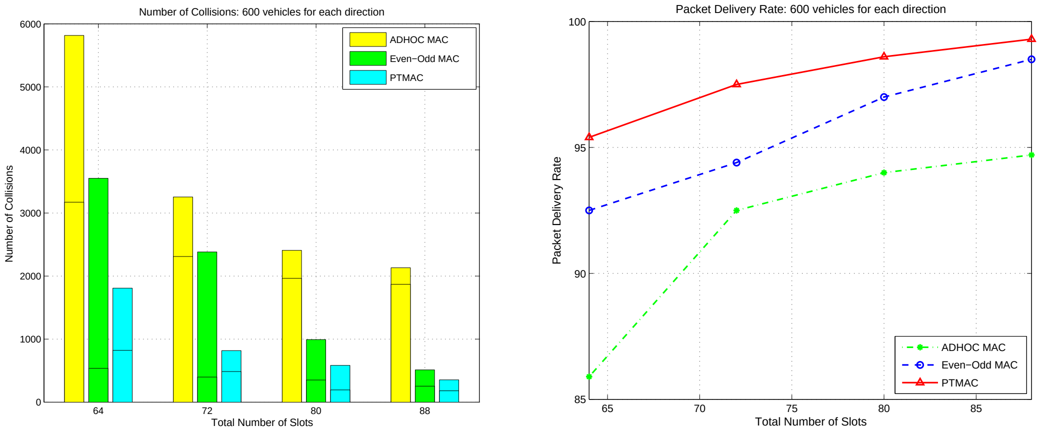
(a) Number of Collisions

(b) Packet Delivery Rate

Figure 4.14: Performance with 400 Vehicles for Each Direction on Highway

competing vehicles have been halved. However, if a contention happens among vehicles at the same direction (among newly joining vehicles and re-competing vehicles), higher probability of contention collision may occur since only half of the slots are available. Higher traffic density means that more newly joining vehicles are generated and more encounter collisions happen among vehicles along the same direction. Thus, more contention collisions are introduced for Even-Odd scheme. On the other hand, our proposed PTMAC has a little bit more number of encounter collisions than Even-Odd protocol with 64 and 72 numbers of slots in a frame. This is because the total number of available slots is not enough that even if a potential collision has been identified, there is no other available slots to change to. Besides, it is likely that a switching vehicle switches to another slot but collides with another vehicle. The denser the traffic is, the more likely a vehicle cannot successfully eliminate the potential collision. However, even with 64 slots per frame under this dense traffic, the proposed PTMAC still has better overall performance with 9.2% and 3.4% higher delivery rate than ADHOC MAC and Even-Odd respectively. When we increase the number of slots in a frame to 80 and 88 (i.e. enough number of slots is provided for vehicles to switch to when eliminating the potential collisions), PTMAC has fewer encounter and contention collisions comparing with the Even-Odd scheme.

Additionally, we study the influence of unbalanced traffic densities on these three



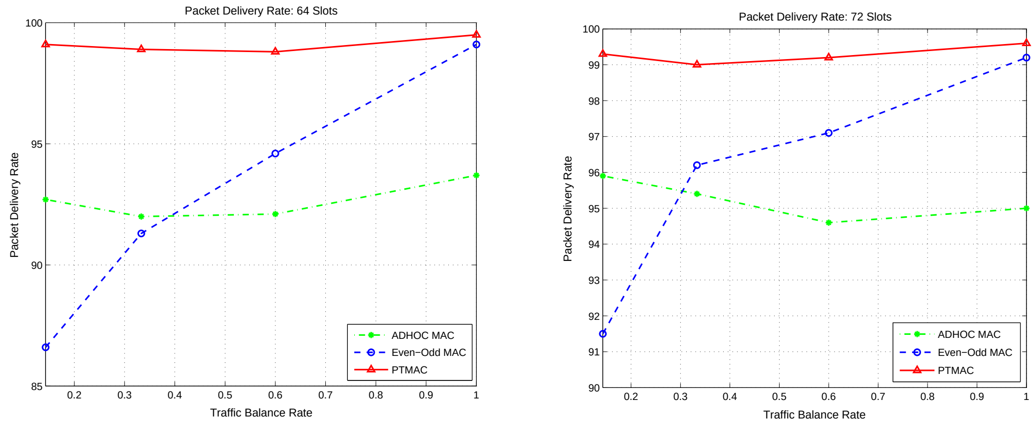
(a) Number of Collisions

(b) Packet Delivery Rate

Figure 4.15: Performance with 600 Vehicles for Each Direction on Highway

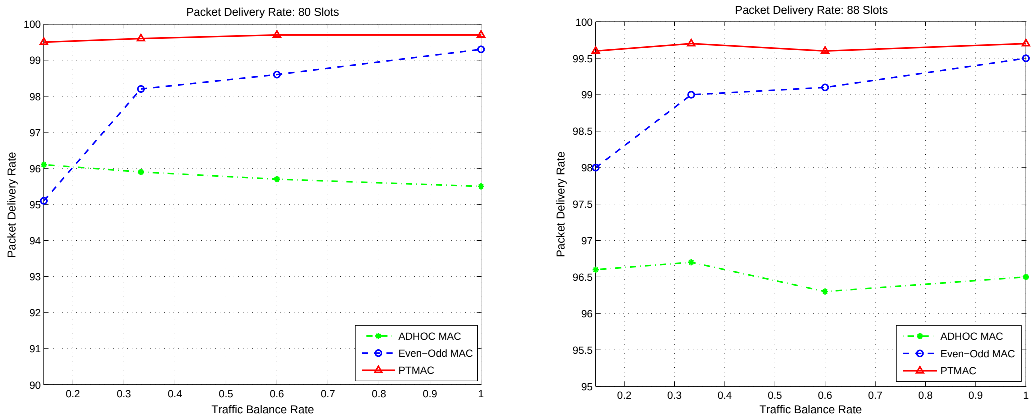
MAC protocols. Here we use the parameter called Traffic Balance Rate (TBR). It is computed as the ratio of the number of vehicles in the direction with sparser traffic to the number of vehicles in the direction with denser traffic. We fix the total number of vehicles that are generated through the simulation as 800 and measure the packet delivery rates of the three MAC protocols using different TBRs. Figure 4.16 (a), (b), (c) and (d) represent the packet delivery rates with 64, 72, 80 and 88 number of slots in a frame respectively. As we can see from the results, the performances of ADHOC MAC and PTMAC are not greatly impacted and degraded by the unbalanced traffic densities since these two protocols do not use slot partition. A vehicle is able to contend for any available slots without restriction based on its moving direction. On the other hand, although Even-Odd protocol reduces the number of encounter collisions, it shows its sensitivity to a small TBR with low packet delivery rate, especially for smaller number of slots in a frame, like 64 and 72. With 64 slots in a frame, the performance of Even-Odd is worse than the ADHOC MAC when TBR is set as 1/7 or 1/3. Thus, vehicles in the direction with heavier density will suffer high probability of contention collision, even there are many empty slots left in another direction. With 64 slots and 1/7 TBR, PTMAC has higher delivery rate of 6.4% and 12.5% comparing to ADHOC MAC and Even-Odd respectively.

In addition, for Even-Odd scheme, if a vehicle finds that all the slots assigned for



(a) 64 Slots Per Frame

(b) 72 Slots Per Frame



(c) 80 Slots Per Frame

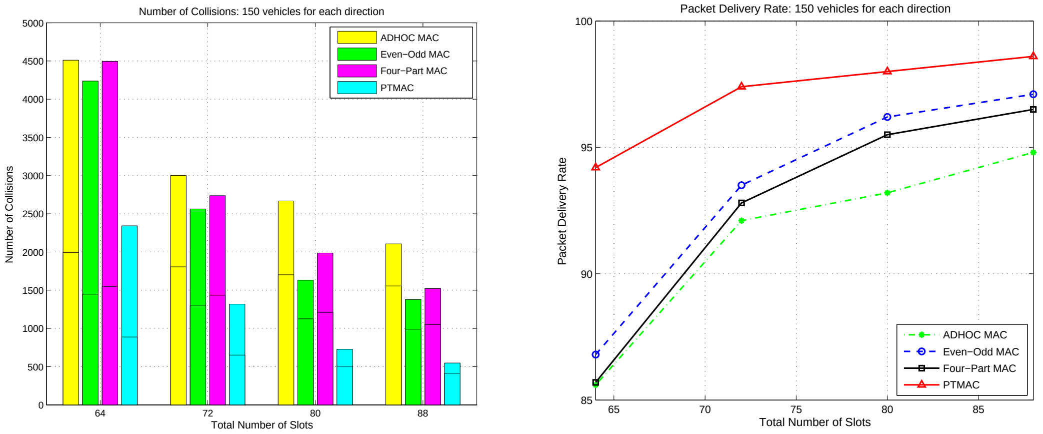
(d) 88 Slots Per Frame

Figure 4.16: Packet Delivery Rate with 800 Vehicles

its direction have been occupied, it will not have a chance to access the channel even if there are still empty slots left for the other direction. It can only begin the slot contention after someone in its direction release the slots. In this case, the slots of the sparse traffic density side will be wasted and some failed contentions will incur. These failed contentions are considered as unnecessary since they can be fully prevented if the number of available slots can be well adapted. Both PTMAC and ADHOC MAC do not suffer such unnecessary failures since vehicles are freely select any available slots for channel contention. Therefore, we can conclude that Even-Odd MAC protocol is very sensitive to the traffic density and the traffic balance rate. Its performance is easily to be degraded by higher or unbalanced traffic densities. On the other hand, PTMAC is not impacted a lot by the traffic unbalance rate but its ability of reducing the number of encounter collisions can be restricted by a high traffic density and a small number of slots in a frame.

Four-way Simulation Results

We also evaluate the performances of the MAC protocols under four-way intersection scenario. Similar to the two-way scenario, we measure their performances using different numbers of slots in a frame and traffic densities. For the Even-Odd protocol, we regulate that vehicles moving to the East and North can only use even slots while vehicles driving to the West and South can only reserve odd slots. Besides, we measure another MAC protocol called Four-Part MAC. In Four-Part MAC, all the slots in each frame is evenly partitioned into four disjointed parts: one part for each direction. Therefore, there will be no interference between vehicles running to the different directions.

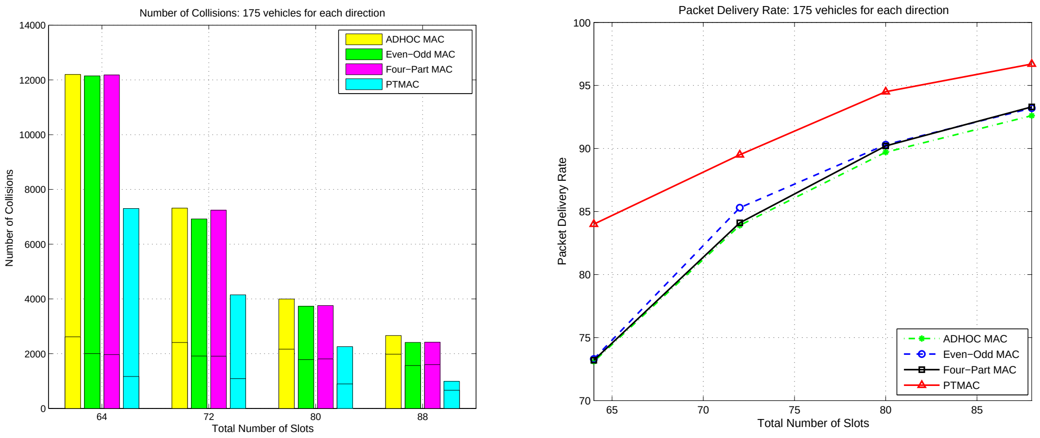


(a) Number of Collisions

(b) Packet Delivery Rate

Figure 4.17: Performance with 150 Vehicles for Each Direction at Intersection

Figure 4.17, 4.18 and 4.19 represent the results under different traffic densities. From the simulation results we can see that PTMAC still works the best with the least number of collisions and highest delivery rate. Both Even-Odd and Four-Part do not have obvious improvement for this four-way intersection scenario. They even perform worse with heavier traffic density. More encounter collisions happen among vehicles at the same direction in this four-way intersection since a vehicle ahead may need to stop and wait for the red signal so it is easy to be caught up by other vehicles behind. Such collisions cannot be handled by Even-Odd and Four-Part. Besides the contention collisions from vehicles at the same direction, Even-Odd cannot avoid the contention collisions that happen near the intersection between vehicles using the same set of slots (like vehicles heading North and East that both use the even slots). For the Four-Part MAC, although no contention collision will happen between vehicles originally driving at different directions, vehicles may change their directions at the intersection. Furthermore, the quartered number of available slots not only increases the probability of contention collisions but also incurs more encounter collisions between vehicles running at the same direction.

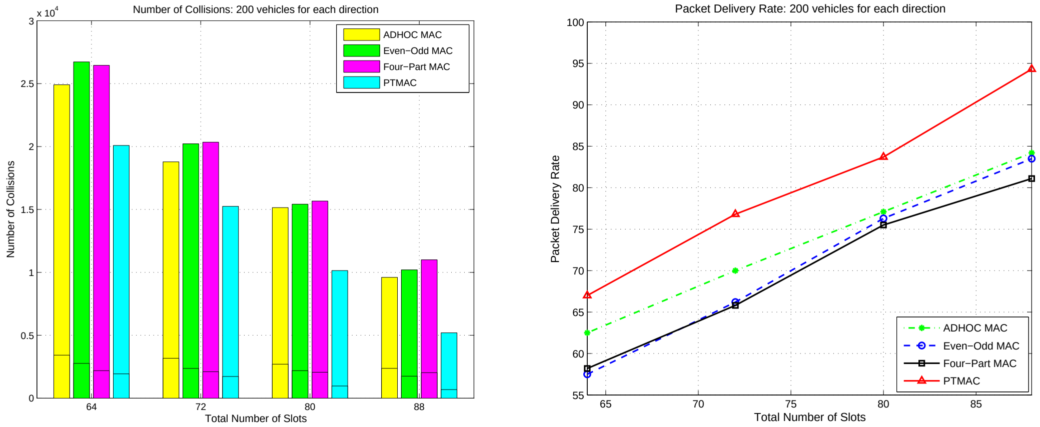


(a) Number of Collisions

(b) Packet Delivery Rate

Figure 4.18: Performance with 175 Vehicles for Each Direction at Intersection

Contrasting with ADHOC MAC, Even-Odd and Four-Part protocols, PTMAC performs better with 48.1%, 44.7% and 47.9% fewer collisions respectively when we set 64 slots in a frame and 150 vehicles for each direction. In the same environment, the packet delivery rate of PTMAC improves about 8.6%, 7.4% and 8.5% comparing to ADHOC, Even-Odd and Four-Part protocols respectively. When we increase the traffic density to 175 vehicles for each direction, PTMAC has 10.9%, 10.7% and 10.8% higher delivery rate when comparing to ADHOC, Even-Odd and Four-Part. For heavier traffic



(a) Number of Collisions

(b) Packet Delivery Rate

Figure 4.19: Performance with 200 Vehicles for Each Direction at Intersection

density with 200 vehicles for each direction, the efficiency of PTMAC is weakened with smaller number of slots in each frame since the number of slots is not enough. But it still has 5.5%, 10.5% and 8.8% higher delivery rate than that of ADHOC, Even-Odd and Four-Part respectively for 64 slots per frame.

4.2.6 Conclusion

In this dissertation, we propose a new Prediction-based TDMA MAC protocol (PTMAC) for decreasing the number of packet collisions, especially for encounter collisions. Potential collisions among vehicles who are currently out of their two-hop communication range can be detected by intermediate vehicles, predicted and then eliminated before they really occur. Our simulations show the effectiveness of the proposed protocol. Since no slot partition is used, unbalanced traffic densities will not degrade the performance of PTMAC. Unlike a few existing MAC protocols that only work for one-way or two-way traffic scenarios, PTMAC is also suitable for handling four-way traffic.

Chapter 5

Performance Measurement and Improvement of VANET Applications

In this chapter, we investigate the network performances of different types of VANET applications. Varieties of applications have been developed that taking advantage of V2V and V2I communications for both safety and non-safety purposes. Unlike wired communication, wireless communication is relatively unreliable. Because of the contention nature of 802.11p protocol, packet collisions and losses are common phenomena and significantly impact the service quality in VANET. Therefore, for further improving the transmission performance, it is necessary to understand the performances of different types of applications over DSRC transmission. Both real-time multimedia and MapReduce applications are measured and analyzed over DSRC transmission in this chapter.

5.1 Multihop Transmission and Retransmission Measurement of Real-Time Video Streaming over DSRC Devices

5.1.1 Introduction

In addition to the traditional data services, many real-time video streaming applications are increasingly being developed over VANET. While the real-time video is very useful for entertainments and social activities [65], it also helps to enhance the driving safety by providing more precise and clear traffic information [66]. Generally, there are two types of real-time video streaming applications that can be implemented in VANET: interactive based video streaming and non-interactive based video streaming. The interactive applications such as videophone support two-way communications and do not require very high video quality but have very tight delay constraint. The non-interactive applications like live sports and road condition monitoring need higher video quality but can tolerant longer startup delay and support pauses during the video playing.

However, due to the lossy vehicular wireless links and the feature of contention based MAC protocol (CSMA/CA) employed in VANET [51], packet loss is a very common phenomenon. Packet collisions caused by concurrent transmissions are inevitable. This greatly impacts the video quality and makes it hard to satisfy user's requirement. Moreover, since no RTS/CTS mechanism can be used for broadcasting in VANET, the lost packets cannot even be detected. Fortunately, unlike other data packets, video streaming are usually composed of a number of continuous packets with consistent sequence numbers. Therefore we can detect the lost streaming packets by checking the sequence numbers and increase the successful packet delivery rate through retransmissions. Despite of the powerful ability of reducing packet losses, aggressive retransmissions may lead to additional inter-arrival delay, reduce bandwidth efficiency, block normal transmissions and even worsen the video quality. Therefore, we need to control the number of retransmissions and investigate the optimal retransmission strategies under different transmission environments. Startup caching is also necessary to ensure smooth video playing and relief the downside of retransmission.

Although it is desirable to deliver the packets directly through single hop communication, such condition is not always available in the real environment. Multihop transmissions not only increase the communication range, but also provide more stable communications. Thus, to support the video applications and further research on streaming transmission in VANET, we need to deeply understand their performance under multihop conditions and even with interferences. Interferences from other vehicles and environment are also inevitable in the real world. The communication range of DSRC is 300m to 1000m that all vehicles in this range will affect each other by concurrently sending packets. Therefore, we focus on measuring the performance of multihop video streaming transmission under different scenarios with and without interferences. We also investigate the influences of different retransmission and startup caching strategies on video quality improvement.

The main contributions can be summarized as follows:

- We develop a real environment test-bed with IWCU DSRC devices to investigate the performance of multihop video streaming and the impact of interferences from others. Practical assessments have been done under two scenarios: multihop video transmission with and without interference.
- To overcome the high packet loss rate which is a main shortage of video streaming over VANET, we improve the application by adding packet retransmission and startup caching. We also analyze how to optimize the retransmission scheme for minimizing the number of collided and late packets based on the transmission rates of the interference node.
- Performance of different retransmission and startup caching strategies have been evaluated over real environment based experiments.

In the following, the related works are introduced in Section 5.1.2. Our developed test-bed and testing scenarios are described in Section 5.1.3. In Section 5.1.4, we analyze the packet collision probability in each scenario and point out the shortage of current video transmission over VANET through experiment results. Different retransmission and startup caching strategies are discussed in Section 5.1.5. We also analyze how to optimize the retransmission scheme in this Section. Section 5.1.6 shows the experiment results and the performance of video streaming. The summary of this work is given in Section 5.1.7.

5.1.2 Related Work

Some research efforts have been done on studying the performance of video streaming transmission over VANET. Xie et al. studied the streaming performance under different traffic conditions and data forwarding schemes in [67]. They compared the sender based forwarding (SBF) and the receiver based forwarding schemes (RBF) for bi-direction environment. While SBF lets the sender select an appropriate forwarder, RBF allows the receiver to decide whether it is worth to forward the packets. From the simulation results, they showed that RBF is more suitable for video transmission in VANET with higher PSNR. Authors in [4] evaluated and compared single hop transmission with multihop transmission using routing protocols DSDV and AODV. Although single hop transmission provides lower delay and packet loss rate, its overall throughput and connectivity are much worse than the multihop transmission. However, DSDV and AODV have been originally used in Ad-hoc network, and they are not specially designed and very feasible for VANET.

To improve the video streaming quality, Asefi et al. proposed a MAC retransmission adaptation scheme in [68] considering the multihop transmission from a RSU to a destination vehicle. They concerned on minimizing the video startup delay and frequency of playback freezes probabilities based on the channel statistics. Different with their work, we focus more on reducing the number of collided and late packets given different transmission rates of the interference, which greatly influences the video quality.

All of the above reviewed works are simulation based and overlook many details in the real environment. Only a few papers tried to measure the video transmission over real devices. Vinel et al. have developed an inter-vehicle video test-bed in [69]. They tested one to one transmission case and concentrated on measuring the impact of distance between a sender and a receiver on video quality. But their testing scenario was too simple comparing to the real environment since they did not consider the interferences from others and also the multihop transmission. They also ignored the collided and useless packets which highly impact the video quality. Another paper [70] contributed on testing the real video transmission over vehicles but they did not employ

802.11p but 802.11b which is not specially designed for VANET.

5.1.3 Test-Bed Description

DSRC Devices

The developed test-bed is composed of four IWCU DSRC devices and two laptops. Four IWCU DSRC devices are performed as sender, receiver, relay node and interference node respectively. The IWCU (ITRI WAVE/DSRC Communication Unit) is an integrated wireless communication system that is designed for employing Intelligent Transportations System (ITS) applications and already been selected as research qualified product from the USDOT. It is compliant with IEEE 802.11p and IEEE 1609 which are known as the WAVE/DSRC standards especially designed for VANET. For fitting well with vehicular applications, an omni-directional antenna operating at 5.9GHz frequency range is attached to the IWCU unit. GPS system is also attached to the device for providing location information and synchronizing time among DSRC devices. MIPS processor based Linux kernel is embedded into the IWCU unit. With such capabilities, applications based on both Roadside-to-Vehicle Communications (RVC) and Inter-Vehicle Communications (IVC) can be supported by this DSRC device. Figure 5.1 shows the IWCU DSRC device and its characteristics are reported in Table 5.1.

Table 5.1: Characteristics of IWCU DSRC Devices

Processor	Qualcomm Atheros AR7130 300MHz
Antenna	5.9GHz, 7dBi
Operating System	Linux Kernel 2.6.32
Bandwidth	10/20 MHz
Standards	IEEE 802.11p, IEEE 1609
Power	6-25 dBm
Bitrates (10MHz)	3-27 Mbps

Testing Scenario

As shown in Figure 5.2 and 5.3, we focus on two testing scenarios: Two-hop Transmission with and without interference node. The sender sends the streaming packets to the relay node and the relay node forwards the packets to the receiver. Two laptops are

Figure 5.1: IWCU DSRC Device

connected to the sender and receiver for generating streaming packets and reconstructing the video respectively. Since we are testing the multihop transmission, the sender and receiver are set out of communication range of each other while the relay node and interference node can reach all of the devices. When the sender is sending the streaming packets, the interference node is also broadcasting other packets. For the sending laptop, the video files are imported into a media player called VLC which transfers the video into streaming packets and sends them to the DSRC device as sender through Ethernet. As long as the DSRC device performs as receiver receives a streaming packet, it forwards the packet to the receiving laptop which will play out the video by the VLC player. Based on our observation, there is no or very few packet losses happen over Ethernet transmission, thus we concentrate on the wireless transmission.

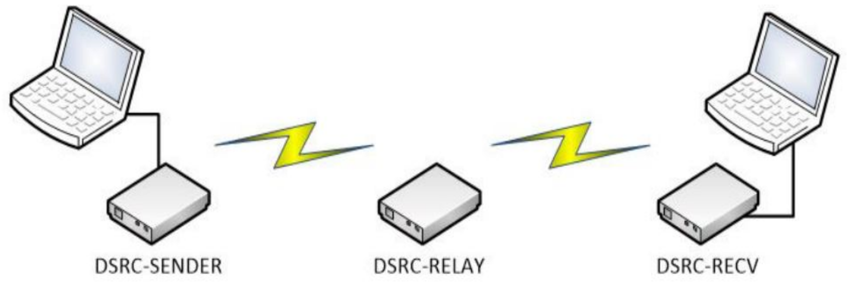


Figure 5.2: Scenario 1: Two-hop Transmission without Interference

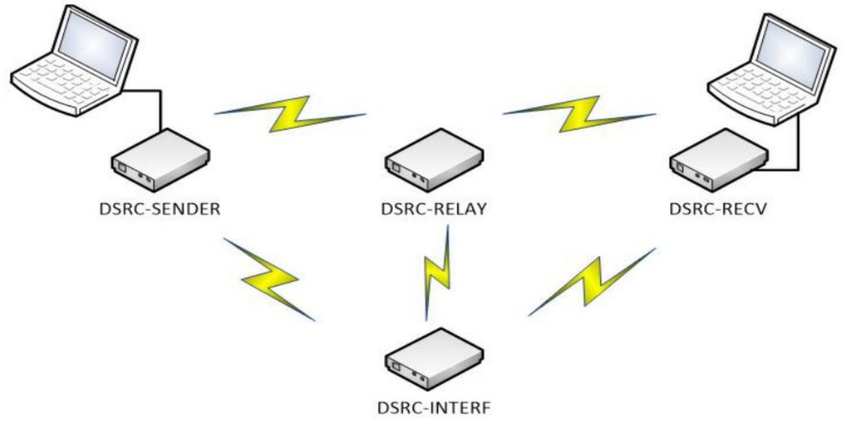


Figure 5.3: Scenario 2: Two-hop Transmission with Interference

Video Streaming Transmission

In our experiments, video contents are encoded by MPEG-4 and streamed using UDP based RTP protocol. Real-Time Protocol (RTP) is a standardized packet format that is especially designed for delivering real-time video stream. Although RTP does not guarantee successful real-time packets delivering, it provides sequence number and time stamp. These can be used for further retransmission and startup caching. Therefore, by checking the sequence number of each received streaming packet, the receiver can recognize which one has been lost, which enables the measurement of packets lost rate and in order packets inter-arrival delay.

All the DSRC devices in our experiments transmit packets under broadcasting mode, which means that no RTS/CTS mechanism is used. Two video files from the same content but with different qualities have been tested in our experiments. Their detailed characteristics are shown in Table 5.2. The frame size we choose is 720 × 480 which is fit for the video playing on vehicle. File 1 is typically used for testing interactive real-time video applications like videophone while file 2 is more feasible for non-interactive videos such as live sports. We use fixed packet size 1370B for the streaming transmission.

Table 5.2: Characteristics of Testing Video Files

File Name	File 1	File 2
Bitrates	384 Kbps	800 Kbps
File Size	4.24 MB	7.17 MB
Frame Rate	15 fps	20 fps

5.1.4 Measurement of Multihop Streaming Transmission

We first measure the 2-hop transmission scenario that is composed of the sender, the relay node and the receiver. Even without interference from other nodes, some packet losses still happen. It is likely that the relay node and the sender are trying to deliver concurrently, which means that relay node will miss some streaming packets from the sender. Assuming the sender wants to send a packet at frame i while the relay node would like to deliver later at frame j, the probability $P_{SR}(i)$ that the sender and the relay node deliver data concurrently can be computed as Equation 5.1. In this case, j has to be equal or larger than i.

$$P_{SR}(i) = P_S(i) \times \sum_{j=i}^{i+x-1} \frac{P_R(j) \times (x - j + i)}{x^2} \tag{5.1}$$

Here x is the contention window size. A node will randomly select a number less or equal to x in order to contend for the channel. $P_S(i)$ and $P_R(j)$ stand for the possibilities that the sender and relay node want to send packets at frame i and j respectively. As long as the sender has not started the transmission and the relay node finds the channel is idle, it is likely that they contend for the same time frame for delivering the packets, which leads to collision.

Then, we introduce an interference node and let it broadcast packets periodically. In this situation, more packet losses are observed and there are three causes for the packet losses. Firstly, similar to scenario 1, there are packets losses caused by the concurrent transmission of the sender and the relay node. Second, if the interference node and the sender access the channel at the same time, collisions will happen that packets from both sides will be lost. Third, collision will also happen if the interference node and the relay node are transmitting packet simultaneously. Therefore, assuming P_{SR}, P_{SI} and P_{RI} are the probabilities of collision between sender and relay node, sender and interference node, relay node and interference respectively, the probability of total packets transmission collision P_C can be represented as Equation 5.2.

$$P_C = 1 - ((1 - P_{SR})(1 - P_{SI})(1 - P_{RI})) \tag{5.2}$$

Typically, there are three metrics that greatly affect the received video quality: Packet inter-arrival delay, delay jitter and packet loss rate. Since the live video streaming, especially the interactive streaming is time sensitive, packets that arrive later than the requested deadline are regarded as useless. The delay jitter that stands for the variation of end to end delay also impacts the video reconstruction since the video frames should be displayed at a constant rate. Since the jitter that is defined in RFC3550 only focuses on streaming packets inter-arrival delay and has not considered the packet sequence number, we call such jitter as non-sequence jitter. The packet loss rate stands for how many streaming packets have been lost during the transmission. It also has destructive impact on the video quality.

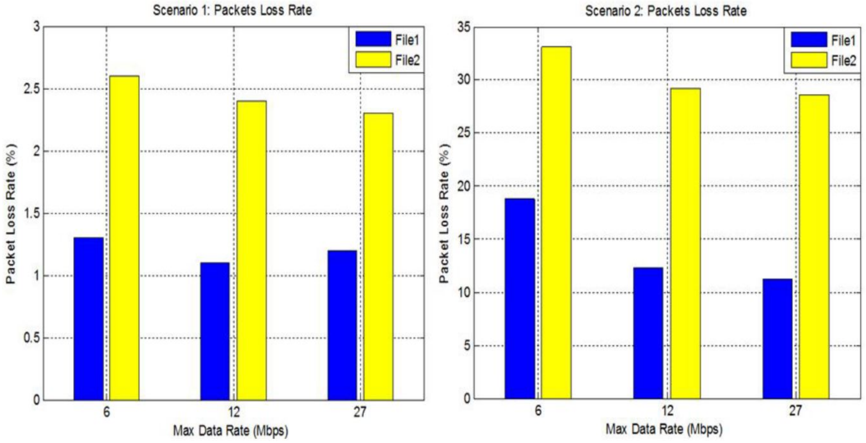


Figure 5.4: Packet Loss Rate without Retransmission

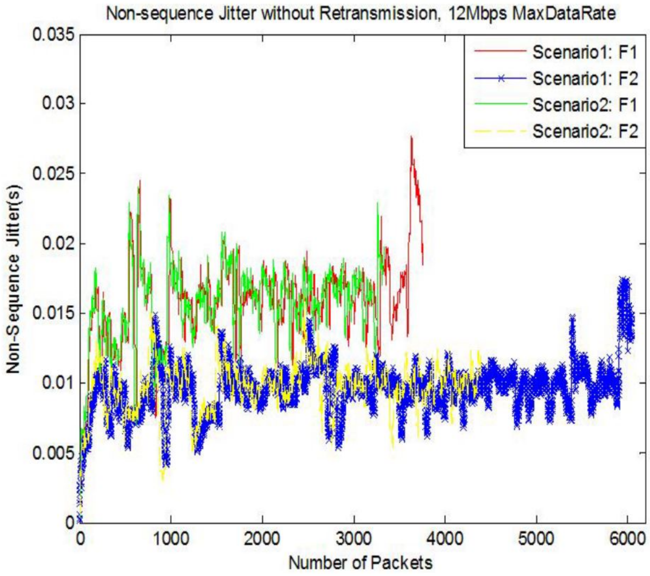


Figure 5.5: Non-Sequence Jitter without Retransmission

In our experiments, we use channel 178 with 10MHz bandwidth for DSRC to transmit the packets. Figure 5.4 and 5.5 represent the packet loss rate and non-sequence delay jitter respectively. For scenario 1, the packet loss rate is around 1.1% for file 1 and 2.4% for file 2 under 12Mbps maximum data rate. The delay jitter varies between 0.2ms and 27ms for file 1 while between 0.2ms and 17ms for file 2. Both packet loss rate and delay jitter are within the tolerant value. Therefore, the video quality is acceptable for scenario 1. However, for scenario 2 which has an interference node, the packet loss rate increases significantly. When we use 12Mbps as the maximum data rate, about 12.3% and 29.2% of packets have suffered collisions for file 1 and file 2 separately. The video quality is greatly impacted at the receiving side when such number of packets have been lost. On the other side, the non-sequence jitter has not been greatly affected by the interference node. Therefore, the main problem for video streaming transmission is the packet loss rate which needs to be improved to ensure the video quality.

5.1.5 Strategy of Retransmission and Startup Caching

Measurement of Retransmission and Startup Caching Schemes

Considering that packet collisions are inevitable and it is common to have interferences from other vehicles on the road while transmitting packets, we use retransmission to reduce the packet loss rate. The sequence number attached in RTP helps the receiver to detect the lost streaming packets and request for retransmission. If the receiver receives streaming packets with an inconsistent sequence number, it knows some packets have been lost and a request will be sent back for retransmission. The lost packet detections are carried out on the relay node and the receiver. Relay node will request for the lost packets from the sender while the receiver will ask for retransmission from the relay node. Therefore, buffers with limited size are kept at the relay node and the sender to hold the delivered packets for a while in case they need to be resent. Although retransmission is powerful on reducing the packet loss rate, it may lead to additional transmission delay and may block the normal packet transmission. Moreover, if the number of retransmissions is too large, heavier delivery traffic will be caused and more collisions may be incurred. Besides, since streaming data is time sensitive, some packets may become useless if they arrive too late. To avoid such situations, we have to avoid unlimited retransmissions and give up some retransmissions based on the latest receiving sequence number and the transmission rate of the interference node.

In order to reduce the sequenced inter-arrival delay which is caused by retransmission, startup caching is inserted into the receiver DSRC device. Before video being played out, the streaming packets will be held in a startup caching within a period of time which is defined as maximum startup delay. In this way, the downside of retransmission can be reduced by avoiding some useless packets that arrive late. However, this strategy is not feasible for interactive real-time videos and pauses are not allowed during the video playing for this type of applications. On the other side, the non-interactive real-time applications are more suitable for startup caching. It also supports pauses during the video playing. If the receiver finds that the startup caching is already empty, a pause will be applied for re-caching the packets. The video will be played out again from the freeze point after waiting for a period of time which is the same as the maximum startup delay.

Noticing here that the non-sequence jitter which has been defined in RFC3550 cannot reflect the influence of retransmission and it does not care about the order of streaming packets. Since the video is reconstructed base on the packets with consistent sequence numbers, the non-sequence jitter is meaningless if the received packets are out-of-order because of retransmission. Thereby we define a new method to measure the impact of retransmission and check whether it causes more inter-arrival packets delay in the order of sequence number. Packets will be considered as useless when they arrive later than the inter-arrival deadline. Assuming the required video reconstruction rate is at least R bps and the packet size is S bits, the packets inter-arrival deadline is computed as Equation 5.3.

$$InterarrivalDeadline = \frac{S}{R} \tag{5.3}$$

Another metric that we focus on is the number of pauses during the video playing for the non-interactive real time video. Pauses allow the streaming to be re-cached and enable fluent playing out. However, too many pauses will also affect user's watching experience. On the other hand, although longer startup delay is helpful for reducing the number of pauses, it may cause dissatisfaction for users who are impatient for waiting. Therefore we have to balance the length of startup delay and the number of pauses. The total waiting time includes two parts: startup delay and times of pausing. Assuming T_D is the startup delay and N is the number of pauses during the video playing, the total waiting time is shown in Equation 5.4.

$$TotalWaitingTime = (N+1) \times \ T_D \tag{5.4}$$

Optimal Retransmission Scheme Analysis

To avoid aggressive retransmissions that may worsen the video quality, we would like to find out how to optimize the retransmission scheme based on the transmission rate of the interference node. For a retransmission, the possibility of losing packet will not be affected by the contention window back-off mechanism since that is not applicable for broadcasting in DSRC MAC protocol. We only concern on the lost and useless packets in this dissertation while ignore the error packets. Denoting the total number of failed packet as N_F without retransmission and N_{FR} with retransmission, the number of retransmissions should be limited to ensure N_F is less than N_{FR}. We can also optimize the retransmission times by finding the minimum value of N_{FR}. Assuming at most T times of retries are allowed and there are N_P streaming packets to be transmitted (retransmission packets are not included), the number of failed packets is computed as Equation 5.5 and 5.6.

$$N_F = N_P \times P_C + N_P \times P_{useless} \tag{5.5}$$

$$N_{FR} = N_P \times (P_{CR})^{min(T,D)} + N_P \times P'_{useless} \tag{5.6}$$

Where P_C is the probability of packet collision without retransmission and P_{CR} stands for the probability of packet collision with retransmission. We also define a new parameter D as "maximum retransmission sequence number difference". The retransmission is only allowed if the difference between the sequence numbers of the missed packet and the received packet with largest sequence number is less than this "maximum retransmission sequence number difference". For example, if we set D as 3, the sequence number of the missed packet is 5 and the largest sequence number of the received packets is 10, then this missed packet will be ignored and no retransmission is allowed.

As we discussed in the last section, the packet losses can be separated into three parts. We will show the probability of collision P_{SI} between the sender and the interference node while the other two parts can be inferred in a similar way. Then the final P_{CR} can be computed based on Equation 5.2. Although the streaming packets are generated based on a fixed interval, sender's transmission rate is variable with time because of different numbers of retransmissions. Given the video rate as R_V Kbps, we can estimate the overall transmission rate R_S of sender with retransmissions as Equation 5.7. The maximum length of retransmission queue must be less or equal to D, which will affect the transmission rate. In the worst case, D packets are waiting for retransmission. In this case, to satisfy the video reconstruction requirement, the transmission rate of the sender should be increased to $(R_V \times (D + 1))$ Kbps. If the available bitrate is less than this required transmission rate, useless packets will be generated.

$$R_S = R_V \times (1 + \sum_{a=1}^{min(T,D)} (P_{CR})^a) \tag{5.7}$$

We set the transmission rate of the interference node as R_I Kbps with packet size as S_I bits and disabled the retransmission. Assuming the packets coming from the sender follow Poisson process and the interference node generates packets at a fixed interval, Equation 5.8 is used to compute the probability of packet collision between the sender and the interference node if the sender wants to send a packet at frame i. Here S_S and S_I are the packet sizes of the sender and the interference node respectively. f is the frame size defined in DSRC MAC protocol. The retransmission helps to reduce the overall packet collision rate from P_C to $(P_{CR})^{min(T,D)}$ by providing multiple transmission chances. On the other hand, the probability of concurrently transmission P_{CR} itself becomes higher since the retransmission increases the data rate to R_S.

$$\begin{cases} P_{SI}(i) = P_S \times P_I \times \sum_{j=i}^{i+x-1} \frac{(x-j+i)}{x^2} \\ P_S = \frac{f \times R_S}{S_S} \times e^{-\frac{f \times R_S}{S_S}} \\ P_I = min(\frac{f \times R_I}{S_I}, 1) \end{cases} \tag{5.8}$$

The number of useless packets is also impacted by the number of retransmissions. The probability of streaming packets arriving late is computed based on the current maximum available bitrate R_B and the transmission rate. If the transmission rate R_S of the sender is larger than R_B because of the retransmission, the packet will not arrive on time. The transmission rate R_S is closely related to the number of retransmission packets which have been inserted between two normal sequence packets. The number of retransmission packets N is composed of the collided packets need to be retransmitted as well as other packets that already been retransmitted but failed again and wait for another retry. We compute the probability that N is larger than a value M for packet n based on its previous D packets. The probability that packets become useless is computed as Equation 5.9 and 5.10. Therefore, in order to optimize the number of failed packets, we need to minimize the sum of $(P_{CR})^{min(T,D)}$ and $P'_{useless}$.

$$P'_{useless} = P\{R_S > R_B\} = P\{R_V(1+N) > R_B\} = P\{N > \frac{R_B}{R_V} - 1\} \tag{5.9}$$

$$\begin{aligned} P\{N > M\} = 1 - \sum_{a=1}^{M}(P_{CR}\binom{D}{a}(\sum_{b=1}^{min(T,D)}(P_{CR})^b)^a) \\ - ((1-PCR)(\sum_{c=0}^{min(T,D)}(P_{CR})^C))^D \end{aligned} \tag{5.10}$$

5.1.6 Experiment Results

Retransmission without Startup Caching

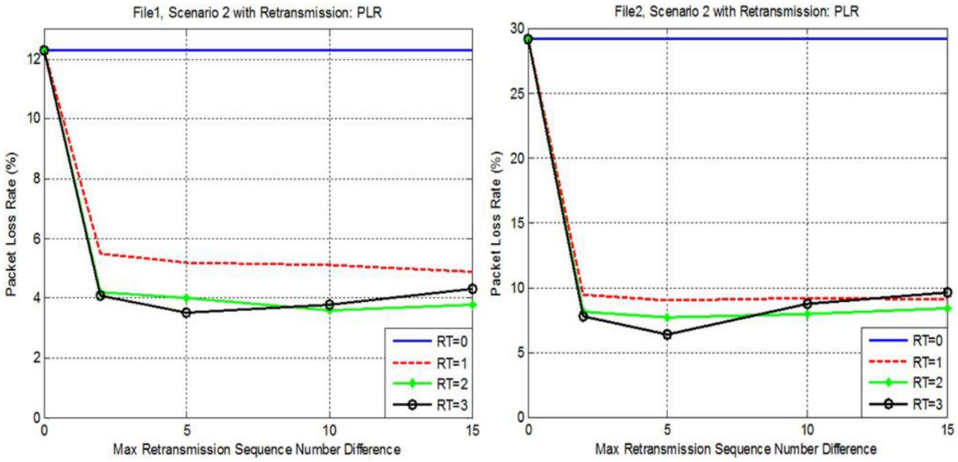


Figure 5.6: Packet Loss Rate with Retransmission

We first investigate the streaming performance for the scenario with retransmission but without startup delay and pause. Although this scheme is more feasible for interactive real-time applications (file 1), we measure both file 1 and file 2 for comparison purpose. Figure 5.6 and 5.7 display the packet loss rate and useless packet rate of both files in scenario 2 respectively. Comparing to the case without retransmission, the packets loss rates of the scenario with retransmission reduce to 3.5% for file 1 and 6.4%

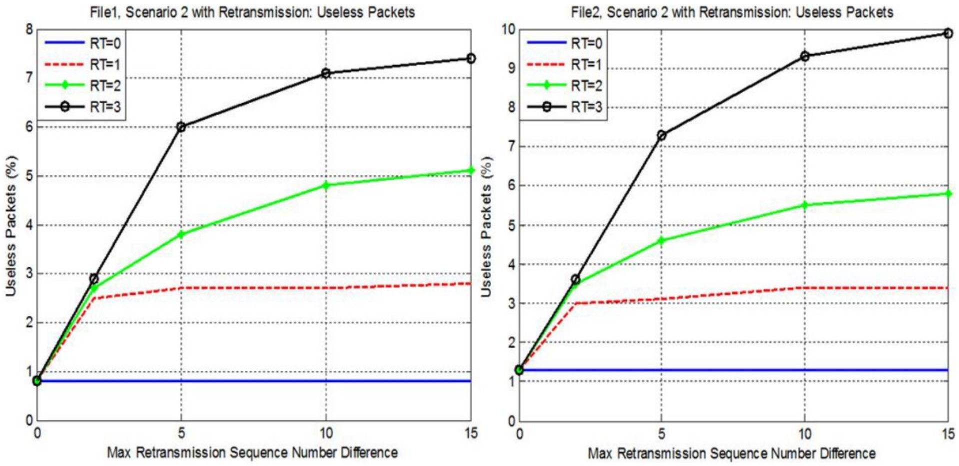


Figure 5.7: Useless Packet Rate with Retransmission

for file 2 under 12Mbps maximum data rate. However, we can observe from the results that if the maximum retransmission sequence number difference is set to be too large, the performance will not be improved anymore for two or three times of retransmission. The packet loss rate stops reducing but begins increasing. Since the retransmitted packets are likely to be lost again, they will be retransmitted until being successfully delivered or reaching the maximum retransmission times. In these cases, with elapsing of time, the packets that need to be retransmitted will be queued up and the maximum queue length must be less than the maximum sequence number difference that we set. As a result, retransmissions potentially block the normal streaming transmission and incur heavy traffic as well as more packet collisions. Notice here that this maximum retransmission sequence number difference does not affect the performance too much if we only allow one time retransmission since the failed retransmitted packets is not allowed to be further retransmitted. The retransmission queue will not be larger than 1.

Although the performance of packet loss rate is improved, the number of useless packets increases with the increment of number of retransmissions. From Figure 5.7, we can see that the useless packet rate significantly increases with larger sequence number difference when we set maximum three times of retransmission. File 2 is more sensitive to the number of retransmissions than file 1. This is because that file 2 has higher video requested bitrates and it generates more packets in a time unit comparing to file 1.

On the other side, the performance is not big impacted by maximum sequence number difference if only one time of retransmission is set.

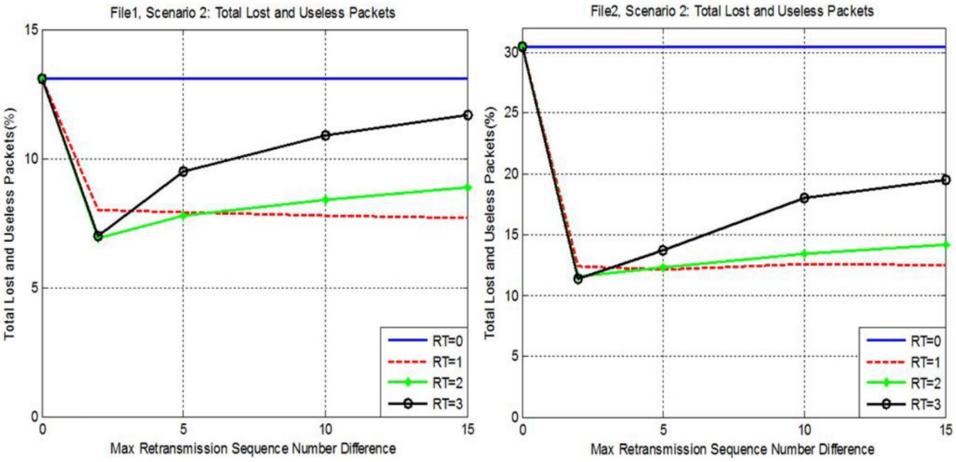


Figure 5.8: Total Lost and Useless Packet Rate with Retransmission

Figure 5.8 shows the rate of lost and useless packets. The performance is improved by at most 5.4% for file 1 and 17.8% for file 2. However, we also notice that the packet loss rate increases with the increment of maximum sequence difference if we allow two or three times retransmission. When we set the maximum sequence number difference as 15 with at most three times retransmission, its performance is even close to the one without retransmission. Therefore, controlling the number of retransmissions is necessary for efficiently improving the streaming transmission performance.

We also investigate the effect of the interference node through different transmission rates: 320Kbps, 1000Kbps, 2000Kbps, 4000Kbps and 8000Kbps. As shown in Figure 5.9, the failed packet rates are separated into two types: The part below the line of a bar stands for the collided packet rate while the part above the line is the rate of useless packets that arrived late. So an entire bar represents the packet failure rate. The maximum data rate is fixed at 12Mbps but the actual available resources for data transmission will be smaller than this since parts of resources are used for serving control packets and other purposes like AIFS in MAC protocol. From the results, we can see that basically, larger the transmission rate that the interference node has, larger the packet failure rate we will get. The benefits of retransmission for reducing the packets collision rate is not so obvious for smaller interference transmission rates like 320Kbps

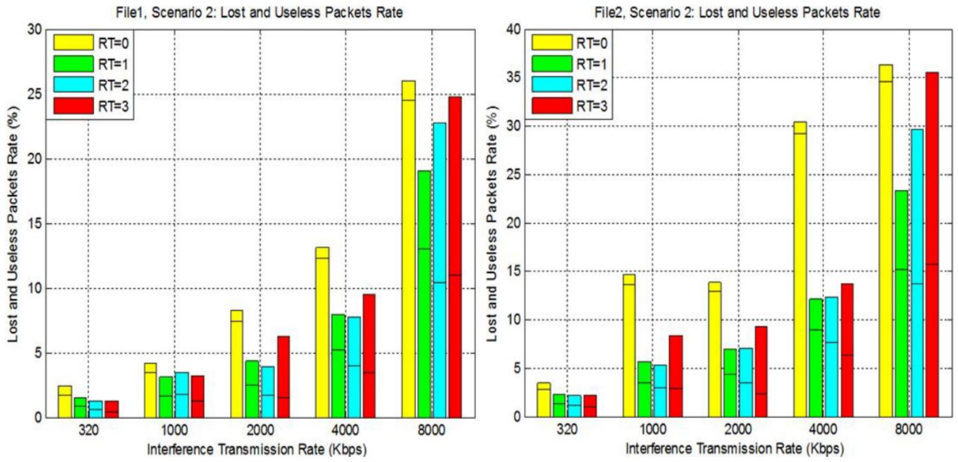


Figure 5.9: Effect of Interference Transmission Rate

since the effect of interference is not so great as well. In most cases, retransmissions can reduce the number of collisions except the case when the channel is under heavy load. For example, if the interference node is transmitting at 8000Kbps, three times of retransmission will worsen the performance by incurring more collisions and increasing the useless packet rate dramatically.

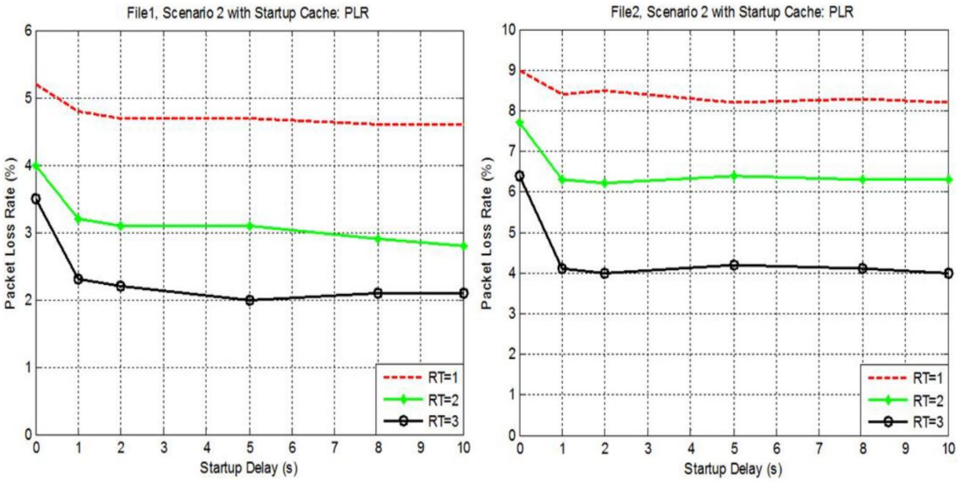


Figure 5.10: Packet Loss Rate with Retransmission and Startup Caching

Retransmission with Startup Caching

Then, we evaluate the streaming performance for the scenario with both retransmission and startup caching. We vary the maximum number of retransmissions and the length of startup delay to observe the changes of packet loss rate as well as the number of pauses. The value of maximum retransmission sequence number difference is fixed as 5. From Figure 5.10, we can see that the packet loss rate dramatically drops down with the helps of retransmission and startup caching. Since the pauses are applied when the startup caching is empty, the packet loss rate does not change too much with different startup delays. More retransmissions allowed, fewer packet losses will happen. Since the retransmission rarely block the normal streaming transmission in this case, the number of useless packets also reduces.

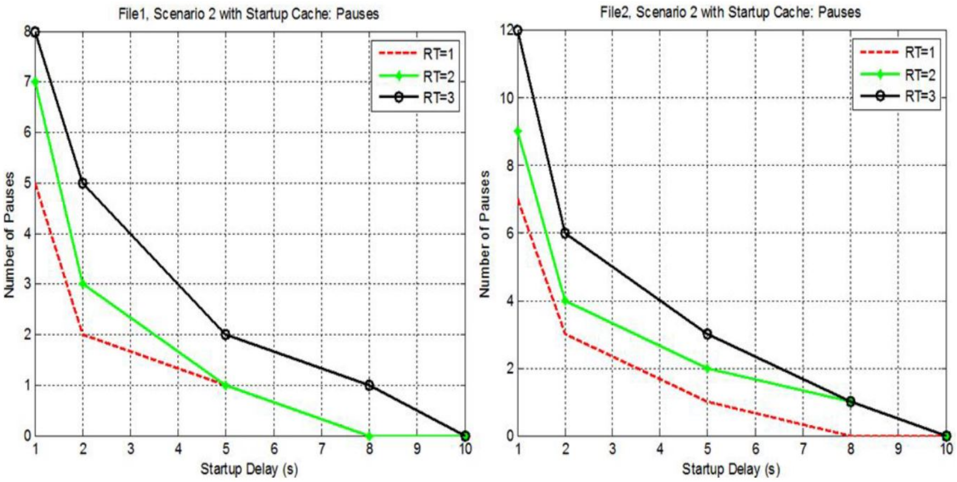


Figure 5.11: Number of Pauses with Retransmission and Startup Caching

We also measure the number of pauses during the video playing which is closely related to the user's watching experience. Both long startup delay and frequent pauses may cause dissatisfaction. Therefore, balancing these two factors is necessary. The results are shown in Figure 5.11. Longer the startup delay we set, smaller number of pauses will occur later. This means that more packets are accumulated during the startup phase. It is harder for the startup caching to become empty during the transmission. When we increase the startup delay to 10s, both file 1 and file 2 do not need pauses anymore. Besides, we can observe that more pauses will be introduced if we increase the number of retransmissions. Although a large number of retransmissions will result in a small packet loss rate, we do not want too many pauses during the video playing which will reduce user's watching experience. So under the condition of acceptable packets loss rate, we prefer smaller number of retransmissions.

The results of the total waiting time are shown in Figure 5.12 which combine the time of startup delay and pauses. We can observe that as we increase the startup delay, the total waiting time first increases and then drops down and then increases again for file 1 with one time or two times retransmission. The reason is that no more pause is needed when the startup delay is larger or equal to 8s. In this case, the total waiting time is equal to the startup delay. The same explanation is applied for file 2. Although the total waiting time are closely related to user's watching experiences, it cannot fully reflect the satisfaction degree. For example, too many pauses may annoy the user even if the total waiting time is less than other cases.

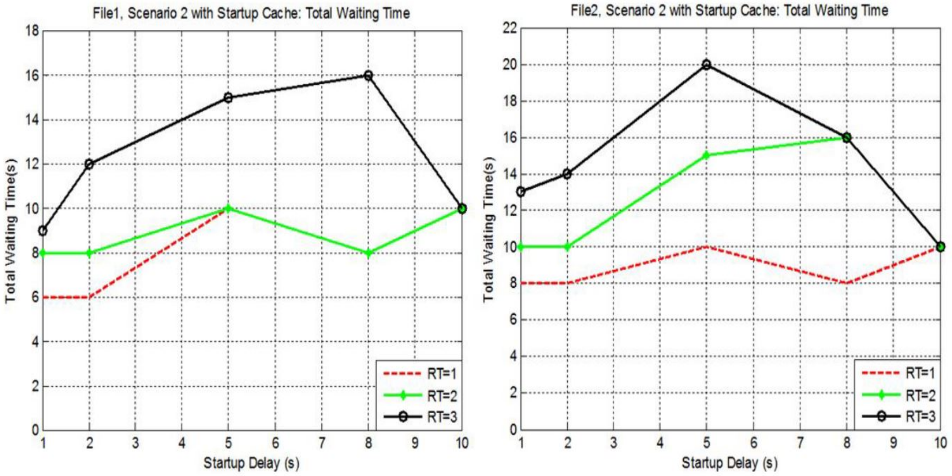


Figure 5.12: Total Waiting Time with Retransmission and Startup Caching

5.1.7 Conclusion

In this dissertation, we develop a DSRC device based test-bed to investigate the performance of multihop real-time video streaming transmission. The quality of the video at the receiving side is acceptable for the 2-hop transmission scenario without interference. However, if we add another DSRC device as the interference node, the streaming packet loss rate becomes unacceptable. To overcome such high packet loss rate, we introduce retransmission and startup caching. For the interactive applications which only support retransmission but do not allow long startup delay and pauses, the packet lost rate reduces but the packet inter-arrival delay becomes longer which causes more useless packets. We also noticed that large number of retransmissions may block the normal streaming transmission and cause more useless packets.

5.2 VeDoop: Performance Study of MapReduce Application over Vehicular Cloud

5.2.1 Introduction

Nowadays, vehicles are becoming more and more powerful. A modern vehicle is typically equipped with powerful on-board computing, storage and communication capabilities. A vast number of vehicles on the road and parking lots can be recognized as abundant and under-utilized computational resources to provide varieties of services. On the other hand, a massive volume of traffic related data is also transmitted, collected and processed in VANET. Many VANET applications like in-vehicle multimedia entertainment and vehicular social networking have been developed that demand complex computation and large storage resources. Although modern vehicles have powerful onboard capabilities, it is still difficult for an individual vehicle to efficiently finish some complicated tasks and many applications have strict time restriction [17, 71]. To fully take advantage of the computation resources of vehicles and support more advanced applications, the concept of Vehicular Cloud (VC) has been proposed [72]. While vehicles parking at the same parking lot can easily build a cluster, a set of vehicles with similar mobility running on the road can also form a small cluster. Clusters of vehicles are able to do cluster computing and provide more powerful computing and storage capabilities. The ultimate goal of VC is to take advantage of vehicles' resources and provide collaborations among nearby vehicles to support advanced applications that single vehicle cannot accomplish alone [73].

Basically, there are two types of vehicular cloud models: static vehicular cloud and dynamic vehicular cloud [17]. In a static cloud, vehicles are regarded as stationary nodes providing cloud services. Such model is suitable for parking lots or traffic congestion scenarios that takes advantage of tens or even hundreds of vehicles. This is usually considered as a relative stable scenario that vehicles infrequently join in and leave the cluster. A dynamic cloud is formed on the road by a cluster of moving vehicles. In such cloud model, the main difference between vehicles and standard nodes in a conventional cloud is the high mobility. Vehicles freely join in or leave the vehicle cloud time by time and their moving patterns are impacted by many factors like traffic density and traffic light. Very frequent topology changes and communication disconnections happen in this case. Such high mobility may threaten the availability of the cloud resource and potentially results in collaboration failure. The number of vehicles that can be used for forming the cluster is also limited. Therefore, comparing to static vehicular cloud, dynamic vehicular cloud has more challenges and is more difficult to be realized.

Over the past few years, a tremendous increasing of data is generated in our society. To handle the Big Data, MapReduce is widely used in clusters for efficient data processing and computing. MapReduce applications are generally studied and implemented over Data Center Network (DCN) that based on a cluster of wired connected servers. It has outstanding ability on processing data efficiently by splitting large work into smaller tasks and executing in parallel. Hadoop is one of the most popular implementation of MapReduce and has been successfully used by many companies. Since vehicles are able to form a cluster and do cluster computing, it is necessary to investigate the performance of MapReduce applications that operate over VANET environment. Most of the previous works about vehicular cloud were focusing on the high level architecture [17, 71–74] and rarely go deep into the performance details. Different with wired communication, data transmission in wireless environment is unreliable and the transmission interferences between the nodes have to be considered. The large number of packet collisions also causes longer total time for finishing a MapReduce task. Thus, to improve the performance of MapReduce applications over Vehicular Cloud with DSRC as transmission protocol, it is important to handle the high packet collision probability and the long transmission time.

In this dissertation, we concentrate on measuring and improving the performance of MapReduce applications over static Vehicular Cloud. Our main contributions can be summarized as follows:

- To adapt MapReduce to Vehicular Cloud environment, we measure and compare the performances of MapReduce applications over wireless VANET connection and wired Ethernet connection.
- We investigate how to improve the MapReduce application performance over Vehicular Cloud through appropriate parameter settings from both MapReduce and DSRC transmission perspectives. Impacts of different parameters are investigated.
- To decrease the interferences between vehicles, we also discuss the cooperation strategy among vehicles within a vehicular cloud.

In the following, Section 5.2.2 overviews the related works about vehicular cloud and MapReduce. Section 5.2.3 describes the experiment environment and simulation settings for measuring and comparing the performances of MapReduce applications under wired and wireless connections. How to improve the performance of MapReduce applications over Vehicular Cloud and how the parameters affect the performance are investigated in section 5.2.4. In section 5.2.5, we discuss the cooperation strategy among vehicles within a vehicular cloud. The conclusion is given in Section 5.2.6.

5.2.2 Related Work

A thoroughly survey on vehicular cloud computing is provided in [17]. The concept of Vehicular Cloud (VC) was firstly proposed and defined by Abuelela and Olariu in [72] that leverages the under-utilized computational resources of vehicles. To illustrate the power of VC, the authors presented some ideas about how the VANET can be used for providing services. For computing as a service, a data cloud can be formed by hundreds of vehicles in a parking lot. Taking advantage of hundreds of vehicles in a mall parking lot, a data center is able to be built. Besides, vehicles involves in the traffic jam can put their resources together to support some applications like traffic light rescheduling. For network as a service, vehicles with internet connectivity can act as access points to the internet. For storage as a service, the available storage provided by vehicles can be used in many applications in the cloud.

Besides, Lee et al. provided an introduction of Vehicular Cloud Networking (VCN) architecture and its design principles in [73]. The vehicular cloud is formed by interconnected resources that are available in both vehicles and RSUs. VCN operation details about cloud resource discovery, cloud formation, task assignment, content sharing, cloud maintenance and cloud release are also explained. Besides the communications and routing services, a VCN also offers security privacy, monitoring, and visualization services. Yu et al. proposed a hierarchical architecture for VC in [71]. They create a flexible cloud environment for moving vehicles by integrating traffic infrastructures in order to fully utilize the resources in an entire network. The cloud architecture is separated into three layers: Vehicular Cloud, Roadside Cloud and Central Cloud. In this way, the computation and storage resources are all merged into the cloud. The authors also mentioned some promising applications of VC like real-time navigation with computation resource sharing, cooperative download/upload with bandwidth sharing and video surveillance with storage resource sharing. All these previous works focused on VC architecture design, operation and general application scenarios.

Enormous of works have been conducted to study and improve the MapReduce performance over conventional Data Center Network. Jiang et al. in [75] studied the performance of MapReduce and identify five factors that may impact the performance, including I/O mode, indexing, data parsing, grouping scheme and block-level scheduling. To improve the MapReduce performance, various job schedulers have been proposed [76, 77]. In [78], Ahmad et al. suggested a multi-tenant scheduler called ShuffleWatcher. Based on the network load, it shapes the shuffle traffic by delaying or elongating a job's shuffle. Map and Reduce tasks placements are also shuffle-aware and consider the locality. Hammound et al. designed a reducer scheduler in [79]. They selected the reducers based on the localities of mappers and also consider partition skew to reduce the traffic during the shuffle phase.

Different with wired environment, performance of wireless communication is greatly influenced by packet collisions. Thus, it is necessary to find out the suitable parameter settings and configurations for both MapReduce and DSRC to achieve better performance.

5.2.3 Performance Evaluation of MapReduce Application

To study the performances of MapReduce under different environments, we first build a conventional cluster environment and evaluate the performance of MapReduce through test-bed experiments. Following the data processing and traffic generating patterns that collected from the experiments, we further measure the performance of MapReduce under VANET wireless environment through simulation by simulating the MapReduce process and DSRC transmission.

MapReduce Experiment over Conventional Cloud

We first study the performance and collect the traffic pattern of MapReduce application under conventional cloud environment with wired connection. Totally four lab machines are used for the experiment. While one node is assigned as Master, the other three nodes perform as Workers. All nodes locate at the same rack and are connected through 1Gbps switch. TeraSort is used as the testing MapReduce application. It is a well-known benchmarking application to test both HDFS and MapReduce layers of Hadoop. TeraGen is first used for generating the input data and then TeraSort executes the sorting job. We experimented with different sizes of input data from 5 GB to 25 GB. The job is split into 40 map tasks for 5 GB, 80 map tasks for 10 GB, 120 map tasks for 15 GB, 160 map tasks for 20 GB and 200 map tasks for 25 GB. The detailed parameter settings are shown as Table 5.3. In the experiment, two workers work as pure mappers while the other work processes both map and reduce tasks. Therefore, in the shuffle phase, the two pure mappers need to send the intermediate mapping results to the one that operates reduce task.

Table 5.3: Parameters for MapReduce

Model	Intel(R) Xeon(R) CPU E5-2407 0 @ 2.20GHz
Hadoop Version	2.2.0
Number of Cores	4
Number of Nodes	4
Number of Masters	1
Number of Workers	3

The data processing in MapReduce essentially has three phases: Map, Shuffle and

Reduce. When an application job is submitted, the resource manager at the Master node will first lunch an Application Manager on one of the workers. Then the Application Manager will further request resources for starting more containers on the works to run tasks. The input data is split into several parts so multiple map tasks can be processed concurrently. After mappers finished the map tasks and generated mapping results, the reducer needs to pull data from them to complete the reduce task and outputs the reducing results. The phase of pulling data from the mappers is called shuffling. When does the reducer start the shuffle phase is decided by a parameter called Shuffle Start Time (SST, defined as mapreduce.job.reduce.slowstart.completedmaps in Hadoop). It represents the percentage of maps in a job that have finished before the reducer can start the shuffling [80]. For example, a value of 0.05 means that the reducer will start the shuffle phase when 5% of the maps are completed. In this way, the execution of map tasks can be overlapped with the shuffling of intermediate results, which reduces the completion time of a MapReduce job.

MapReduce over Vehicular Cloud

As one part in the Wireless Access in Vehicular Environment (WAVE) protocol stack developed by IEEE, 802.11p [1] has already been approved as the standard MAC protocol in VANET. It employs contention-based Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) with Enhanced Distributed Channel Access (EDCA). For CSMA/CA in 802.11p MAC protocol, if the channel is sensed as idle for an arbitration inter frame space (AIFS), the vehicle starts the transmission directly. The AIFS is computed based on the Short Interframe Space (SIFS) and AIFS-number (AIFSN), that is, $AIFS = AIFSN \times Slot + SIFS$. If the channel is currently busy, a vehicle needs to randomly pick up a back-off value from the interval [0, CW] and then start a countdown procedure. CW stands for the Contention Window which is initially set as CW_{min}. The back-off value will be decreased when the channel is free and transmission will begin when the back-off value reaches 0.

If multiple vehicles within the communication range are trying to access the channel simultaneously, a collision will happen and none of the packets can be successfully received. In this case, vehicles have to re-compete for the channel to resend the packets. A sender of unicasting needs to wait for an acknowledgement (ACK) from the recipient.

If the ACK is not received with in a period of time, the packet will be retransmitted. An exponential back-off scheme which extends the CW size will be applied to the retransmission until CW_{max} is reached. In this way, the probability of retransmission contention collision can be reduced. Based on the different application requirements and critical levels, EDCA classifies packets into four Access Classes (ACs) with different priorities. Different ACs have different AIFSN and CW values to ensure packets with higher priority can access the channel earlier. AC3 owns the highest priority. The default EDCA parameters of 802.11p are shown in Table 5.4.

Table 5.4: Default Parameter Settings for EDCA in 802.11p [1]

AC	CW_{min}	CW_{max}	$AIFSN$
AC0	15	1023	9
AC1	15	1023	6
AC2	7	15	3
AC3	3	7	2

In order to measure the performance of MapReduce over VANET, all the data transmissions between nodes are carried out through wireless connection instead of Ethernet connection. The MapReduce data processing and traffic generating patterns follow the data collected from the experiment. We simulate the MapReduce process based on the gathered parameters include number of map and reduce tasks, the time spend on each phase, the time cost for each task and the number of data for transmission. Meanwhile, the DSRC MAC protocol is simulated using MATLAB. We first assign the Access Classes (ACs) as 2. So the AIFSN is 3 while CW_{min} and CW_{max} are 7 and 15 respectively. The data rate is simulated as 27Mbps which is the maximum data rate that can be support by DSRC. Detailed parameter settings for DSRC transmissions are shown in Table 5.5.

Generally, the network performance of the wireless network is worse than wired network. Assuming the computational capabilities are the same for conventional cloud environment and vehicular environment, the most different part between wired and wireless environments is the shuffle phase which involves most of the data transmission. Packet collisions and losses are usually happen under wireless environment. If multiple mappers are trying to send the map results to the reducer concurrently, a packet collision will happen. Since the reducer needs all the data from all the mappers to finish the computation of the reduce work, a mapper has to re-compete for the channel and retransmit the lost packet until the reducer receives it. Therefore, when more collisions happen, longer overall time is expected for completing the MapReduce job. Besides the collision problem, the data rate supported by DSRC is from 3 to 27 Mbps, which is much slower than the Ethernet connection. This will also cause longer shuffling time. Since we focus more on the shuffle phase, we assume the map tasks are carried out locally in the simulation.

Table 5.5: Parameter Settings for DSRC Transmission

$Slot$	10 us
$SIFS$	30 us
$AccessClass(AC)$	2
$AIFSN$	3
CW_{min}	7
CW_{max}	15
$DataRate$	27 Mbps

Performance Evaluation and Comparison

The metrics that we mainly focus on are the shuffle time and the packet collision rate during the shuffle phase. The shuffle time indicates the total time needed to finish the shuffle phase and this will directly impact the job completion time. The packet collision rate represents the ratio of number of packet collisions to the total number of transmissions. As shown in Figure 5.13, the MapReduce shuffle time under DSRC transmission is about 10 times to the Ethernet connection. The packet collision rate of shuffling using DSRC transmission is shown in Table 5.6. The average packet collision rate over all the measured data sizes is 9.3%. Therefore, we can see that the performance of MapReduce greatly degrades because of wireless transmission.

Table 5.6: Packet Collision Rate for Vehicular Cloud

Data Size (GB)	5	10	15	20	25
Packet Collision Rate	9.5%	9.1%	9.3%	9.4%	9.4%

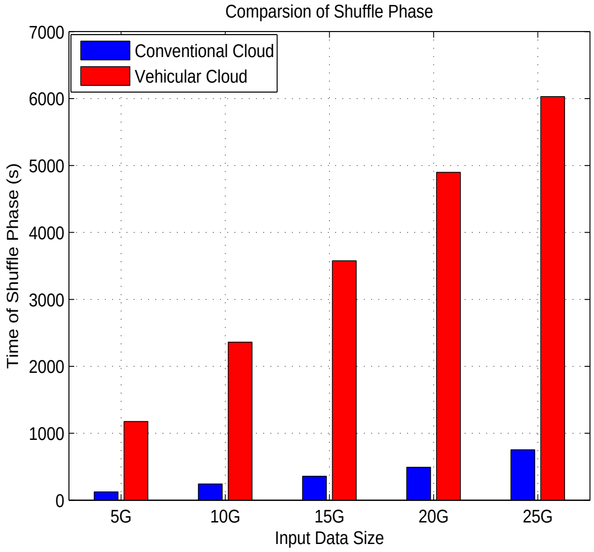


Figure 5.13: Comparison between Conventional and Vehicular Cloud

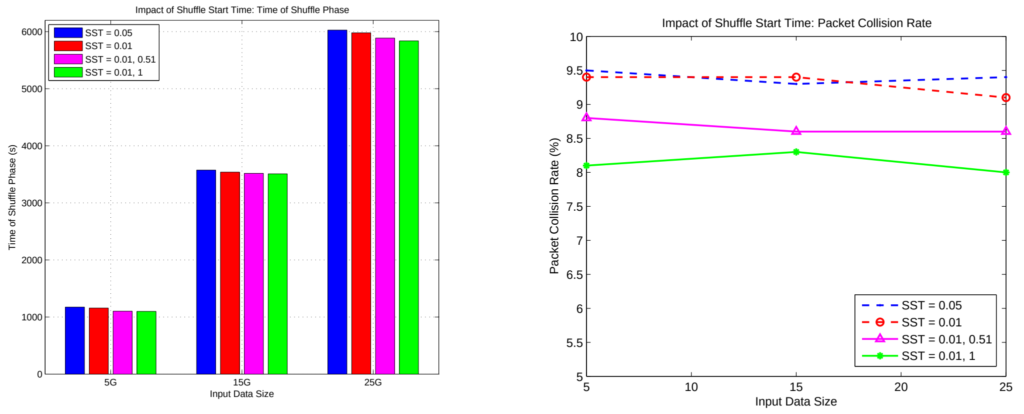
5.2.4 Parameter Settings of MapReduce Application

In order to improve the performance of MapReduce application over VANET, it is critical to shorten the shuffle phase. The time of shuffle phase can be reduced through reducing the shuffle traffic or the number of packet collisions. To achieve less shuffling traffic, some new task schedulers has been proposed for conventional cloud. In this section, we explain how to improve the performance of MapReduce by finding appropriate settings for both MapReduce and DSRC. We investigate the effects of the parameters include total number of nodes, shuffle start time for MapReduce and Access Category (AC) for DSRC. We observe that smaller number of packet collisions can be achieved using suitable parameter settings.

Impact of Shuffle Start Time

We first investigate the impact of Shuffle Start Time (SST) on the MapReduce performance. The AC is set as 2 and four-node cluster simulation is used. Totally four groups of settings are tested for SST and each group is measured for three data sizes as 5, 15 and 25GB. We first compare the settings of SST as 0.05 and 0.01. Then, since there are two workers that need to transmit the intermediate results to the third worker, we are interested to see the performance when the shuffle phase on each worker start at different time. The third group is to set the SST for the first worker as 0.01 and the second worker as 0.51. We also set the fourth group that the SSTs are 0.01 and 1 for the two workers separately. In this case, the first worker will start the shuffle phase as long as 1% maps have been done while the other work can only begin the shuffling when all the maps have been finished.

Figure 5.14 represents the time taken for shuffle phase and the packet collision rate. As we can see from the results, the smaller SST does not impact the shuffle time and packet collision rate obviously. This because that the smaller SST only allows the shuffle phase to start earlier. When we put different SSTs on different workers, shuffle time becomes shorter and packet collision rate reduces. When one worker is transmitting, the other one is still waiting for the start of shuffling and queuing the packets. Thus there is no interference before the second worker starts its shuffle phase. Greater the shuffle start time difference between the two works, fewer packet collisions will happen. Comparing with 0.01 SST, the settings of the third and forth group have 82s and 132s less shuffle time respectively with 25G input data. Therefore, the strategy of setting different SSTs on different workers is suitable for an individual job.



(a) Shuffle Time

(b) Packet Collision Rate

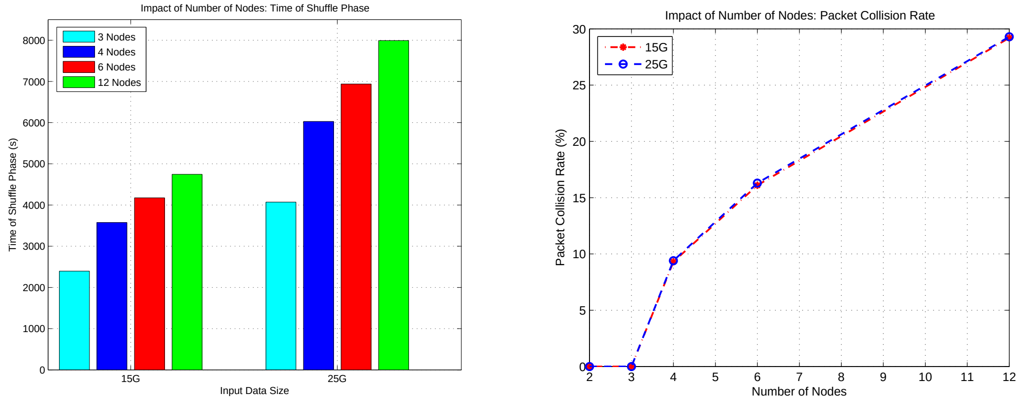
Figure 5.14: Impact of Shuffle Start Time

Impact of Number of Nodes

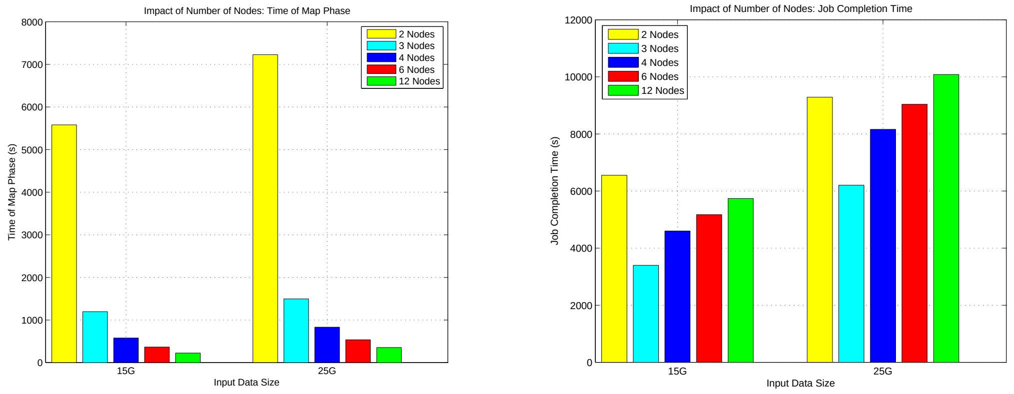
Although using a larger number of nodes for processing data in parallel is helpful for finishing the map tasks earlier, workers potentially suffer high packet collision rate since more interferences are introduced. Such higher packet collision rate will increase the time of shuffle phase. In order to investigate how the cluster size influences the performance, we simulate the MapReduce and DSRC transmission over 2, 3, 6 and 12 nodes following the task processing pattern of the four-node experiments. For all the cases, one node is used as master node while the others are regarded as workers. All the works will run as mappers and one of the workers will also perform as reducer. Similar to the four-node experiment, all the workers except the one running the reduce task need to transmit intermediate map results to the reducer. We use AC2 for the simulation and SST is fixed as 0.05.

Figure 5.15 shows the results of the shuffle time, the packet collision rate, the time of map phase and the time of job completion time. We can see that more nodes are used for MapReduce, more packet collisions happen. The two-node cluster only has one worker therefore no shuffle transmission is needed. For three-node cluster case, no packet collision happens because there is only one worker that transmits map results to the reducer in the shuffling phase. When a large number of nodes are attempting to contend for the channel, they are more likely to access the channel at the same time and interrupt each other. Since more collisions and retransmissions happen, the shuffle time also greatly increases. With 25GB input data, comparing to six-node and twelve-node cluster, the four-node cluster has 841s and 1895s less time for shuffling and 6.9% and 19.9% less percentage of collisions respectively.

On the other hand, we observe that the time of map phase is reduced when using larger cluster size since more nodes working in parallel to finished the map tasks. When there is only one worker, the map phase cost as high as 5580s and the entire job completion time is 6547s for 15G data. When we increase the number of workers to two, the time of map phase reduces significantly to 1196s and the job completion time shortens to 3394s for 15G data. However, when we keep on increasing the number of nodes, the total job completion time becomes longer. This is because that the map phase is partially or mostly covered by the shuffle phase and its reduction is not as much as the increment of shuffling time. Therefore, smaller size of cluster is recommended for



(a) Shuffle Time (b) Packet Collision Rate



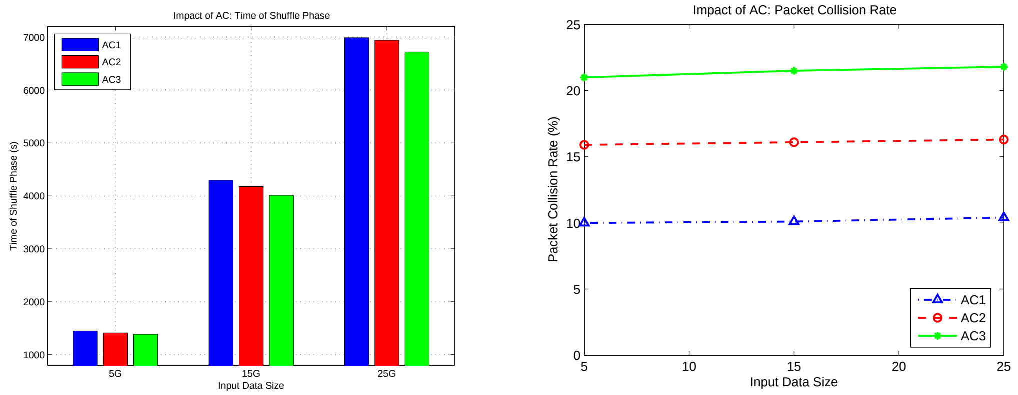
(c) Map Time (d) Job Completion Time

Figure 5.15: Impact of Number of Nodes

MapReduce in VANET.

Impact of ACs in DSRC

We also investigate the influence of different ACs for DSRC transmission. The performances of AC1, AC2 and AC3 are measured. Six-node cluster is used for the simulation and SST is fixed as 0.05. The values of AIFSN and CW are set based on Table 5.4 for each AC. Figure 5.16 demonstrates the shuffle time and the packet collision rate. As shown in the results, when higher AC is set, less shuffle time can be achieved. This because that a packet with higher AC has smaller AIFSN and shorter CW size. If the channel is idle, packet with AC3 can wait for shorter time before the transmission begins. If the channel is busy, a smaller CW helps packets with AC3 get a small backoff value. Therefore, packets with higher AC are able to access the channel earlier with less waiting time. With 25GB input data, AC3 has less shuffle time of 270s and 221s comparing to AC1 and AC2.



(a) Shuffle Time

(b) Packet Collision Rate

Figure 5.16: Impact of AC

On the other hand, a packet with higher AC suffers more packet collisions because of its smaller CW size. For example, AC3 has CW_{min} as 3 and AC2 has CW_{min} as 7. This means that packets with AC2 have larger CW range to choose from. If two packets begin channel contention at the same time, they may select the same backoff value with probabilities of 33.3% and 14.3% for AC3 and AC2. With 25GB input data, AC3 has higher packet collision rate of 5.5% and 11.4% when comparing with AC1 and AC2 respectively. However, the effect of shorter waiting time is greater than the packet collision rate. Therefore, for packets from the same application, higher AC is preferred to achieve shorter shuffle time even the packet collision rate may become higher.

5.2.5 Cooperation within Vehicular Cloud

As we shown in the above section, a small number of vehicles work together performs better than a large number of vehicles considering the interference problem. Assuming a parking lot is 54 meters by 96 meters and each parking space is 4 meters by 5 meters, the total number of available parking spaces is approximately 120. This example parking lot is shown in Figure 5.17. If DSRC communication range is 300m, a vehicle will be able to transmit data to all the other vehicles in the parking lot. All the vehicles parking in this parking lot can form a vehicular cloud. However, if 120 vehicles are trying to shuffle at the same, the impact of interferences will become huge and the time of shuffle phase will significantly increase. Therefore, we divide the whole vehicular cloud into several groups and each group is composed of a small number of vehicles as workers. Assuming the group size is 3, there will be 40 groups of vehicles in the example parking lot. Each job will only be completed by one group of vehicles. One vehicle will serve as master node in the VC and it assigns jobs to each group of vehicles. The master node can be selected as the vehicle that will stay at the parking lot for the longest time or a RSU can be deployed and it works as a Master. The members of each group can by dynamically changed based on the job and resource requirements.

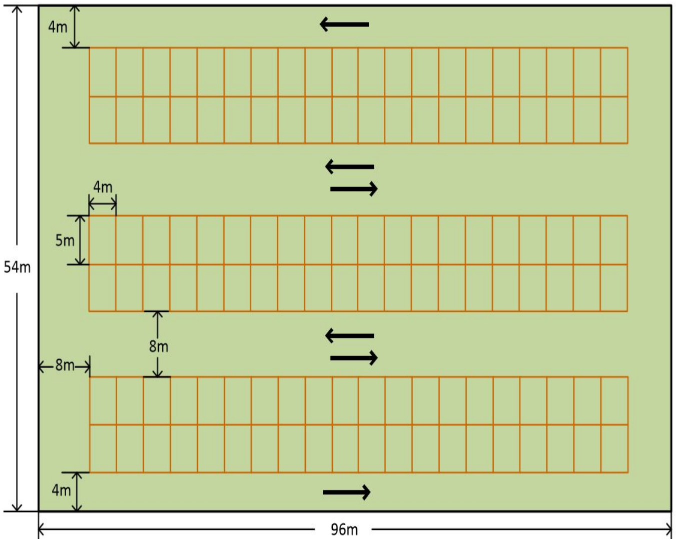


Figure 5.17: Example Parking Lot

Although tuning down the power to shorten the DSRC communication range is helpful for decreasing the interferences, the flexibility and resource availability are potentially reduced. Vehicles as workers may leave or join in the cloud and they may park at different space. If using a short communication range, a vehicle may have less chance to cooperate with others. Besides, for different jobs, it is likely that the data is kept at different vehicles. Thus longer communication range allows vehicles to work with different vehicles and avoids some data transfers. However, one problem that is introduced by the longer communication range is the interference between each group. To avoid such interference, the master node needs to manage the shuffle phase of each group. Since vehicles in parking lot are considered as static nodes, centralized TDMA is a good choice to manage all the groups. Each time unit can be separated into several time slots. Having work statuses of all the nodes, the master is able to allocate time slots to each group and a group can do the shuffling at its assigned slots. The number of slots assigned for each group can be based on the size of shuffling data, job priority and other parameters.

5.2.6 Conclusion

In this dissertation, we compare the performances of MapReduce application over conventional cloud with Ethernet connection and Vehicular Cloud with wireless DSRC connection. Impacts of different parameters include number of nodes, shuffle start time for MapReduce as well as AC for DSRC are discussed. Smaller number of nodes, separating the shuffling start time on different workers and higher AC are helpful for reducing the shuffling time. We also discuss the cooperation within the vehicular cloud in order to reduce interferences.

Chapter 6

Conclusion

In this dissertation, we focus on improving the reliability and efficiency of data transmission in VANET. We concentrate on addressing three research issues: 1) How to efficiently transmit data among vehicles and between vehicles and traffic infrastructures (RSUs) with lower latency? 2) How to improve the data transmission reliability with fewer packet collisions and higher packet delivery rate? 3) What are the network performances of different kinds of VANET applications? And how to improve their performances? New architecture, protocols and innovation solutions are developed for investigating these research issues.

To address the first issue, we developed a new VANET architecture called BUS-VANET that takes advantage of both buses and traffic infrastructures. In this architecture, the coverage of the high-tier nodes can be ensured and the probability of packets carrying is reduced. TCC is helpful for quickly identifying the location of the destination vehicle. Comparing to the traditional VANET, better performance can be achieved in BUS-VANET with less delivery delay and higher delivery rate. Additionally, to improve the transmission reliability and reduce the number of packet collisions, two TDMA based MAC protocols are proposed: MAT-MAC and PTMAC. MAT-MAC aims at reducing the number of collisions while maintaining high slot utilization even under unbalanced traffic scenario. High contention collision probability can be avoided through dynamic slot migration and adaptation based on the real-time traffic condition. PTMAC is a novel protocol based on the important observation that most of the encounter collisions can be predicted and potentially avoided. It is not only suitable for two-way traffic but also for four-way intersections in an urban area. As a result, the proposed protocols improve the transmission performance with fewer packet collisions under different scenarios. Besides, to further support and improve the performances of varieties of applications through V2V and V2I communications for both safety and non-safety purposes, we studied the performances of real-time multimedia and MapReduce applications over DSRC transmission. The real-time video is useful for entertainments, social activities, and enhancing the driving safety. MapReduce applications are widely used in clusters for processing and computing data efficiently. Impacts of different strategies and parameter settings are also discussed. Upon using appropriate retransmission and startup caching, the performance of multimedia transmission can be improved. Smaller number of nodes, separating the shuffling start time on different workers, higher AC and corporations within the vehicular cloud are helpful for reducing the shuffling time of the MapReduce applications.

References

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

[2] Qi Chen, D. Jiang, and L. Delgrossi. Ieee 1609.4 dsrc multi-channel operations and its implications on vehicle safety communications. In *IEEE Vehicular Networking Conference (VNC 2009)*, pages 1–8, 2009.

[3] Yasser L. Morgan. Managing dsrc and wave standards operations in a v2v scenario. *International Journal of Vehicular Technology*, 2010.

[4] H. Venkataraman, A. d'Ussel, T. Corre, C. H. Muntean, and G. M. Muntean. Performance analysis of real-time multimedia transmission in 802.11p based multihop hybrid vehicular networks. In *Proceedings of the 6th International Wireless Communications and Mobile Computing Conference (IWCMC 2010)*, pages 1151–1155, 2010.

[5] National Highway Traffic Safety Administration (NHTSA). Traffic safety facts. Technical Report DOT HS 812 032, 2012.

[6] US DOT. Transforming transportation through connectivity: Its strategic research plan, 2010-2014 (progress update, 2012). Technical Report FHWA-JPO-12-019, 2012.

[7] RITA. Guide to Federal Intelligent Transportation System (ITS) Research. http://www.its.dot.gov/factsheets.

[8] Ing-Chau Chang, Yuan-Fen Wang, and Cheng-Fu Chou. Efficient vanet unicast routing using historical and real-time traffic information. In *IEEE 17th International Conference on Parallel and Distributed Systems (ICPADS 2011)*, pages 458–464, 2011.

[9] Ieee standard for wireless access in vehicular environments (wave)–networking services corrigendum 1: Miscellaneous corrections. *IEEE Std 1609.3-2010/Cor 1-2012 (Corrigendum to IEEE Std 1609.3-2010)*, pages 1–19, July 2012.

[10] ITSA. The Connected Vehicle Next Generation ITS. http://www.itsa.org/industryforums/connectedvehicle.

[11] C. Campolo, A. Vinel, A. Molinaro, and Y. Koucheryavy. Modeling broadcasting in ieee 802.11p/wave vehicular networks. *IEEE Communications Letters*, 15:199–201, 2011.

[12] C. Campolo, A. Molinaro, and A. Vinel. Understanding the performance of short-lived control broadcast packets in 802.11p/wave vehicular networks. In *IEEE Vehicular Networking Conference (VNC 2011)*, pages 102–108, 2011.

[13] National Highway Traffic Safety Administration (NHTSA). Vehicle safety communications project: Task 3 final report: Identify intelligent vehicle safety applications enabled by dsrc. Technical Report DOT HS 809 859, 2005.

[14] USDOT. Southeast Michigan Test Bed Resource Guide. http://www.its.dot.gov/testbed/pdf/se-mi-resource-guide-9-3-1.pdf.

[15] USDOT. Southeast Michigan Test Bed. http://www.its.dot.gov/testbed/testbed_semichigan.htm.

[16] ND Lernern. Brake perception-reaction times of older and younger drivers. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, pages 206–210, 1993.

[17] Md Whaiduzzaman, Mehdi Sookhak, Abdullah Gani, and Rajkumar Buyya. A survey on vehicular cloud computing. *J. Netw. Comput. Appl.*, 40:325–344, 2014.

[18] Yi Qian and N. Moayeri. Design of secure and application-oriented vanets. In *IEEE Vehicular Technology Conference (VTC Spring 2008)*, pages 2794–2799, 2008.

[19] National Highway Traffic Safety Administration (NHTSA). Vehicle safety communications project final report. Technical Report DOT HS 809 859, 2005.

[20] Michael McGurrin. Vehicle information exchange needs for mobility applications version 3.0. Technical Report FHWA-JPO-13-065, April 2013.

[21] C.E. Perkins and E.M. Royer. Ad-hoc on-demand distance vector routing. In *IEEE Workshop on Mobile Computing Systems and Applications (WMCSA 1999)*, pages 90–100, 1999.

[22] David B. Johnson, David A. Maltz, and Josh Broch. Ad hoc networking. chapter DSR: The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks, pages 139–172. 2001.

[23] Boon chong Seet, Genping Liu, Bu sung Lee, Chuan heng Foh, and Keok kee Lee. A-star: A mobile ad hoc routing strategy for metropolis vehicular communications. In *International Conference on Information Networking (ICOIN)*, pages 989–999, 2004.

[24] Jing Zhao and Guohong Cao. Vadd: Vehicle-assisted data delivery in vehicular ad hoc networks. In *IEEE International Conference on Computer Communications (INFOCOM 2006)*, pages 1–12, 2006.

[25] I. Leontiadis and C. Mascolo. Geopps: Geographical opportunistic routing for vehicular networks. In *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2007)*, pages 1–6, June 2007.

[26] Jaehoon Jeong, Shuo Guo, Yu Gu, Tian He, and D.H.C. Du. Tsf: Trajectory-based statistical forwarding for infrastructure-to-vehicle data delivery in vehicular networks. In *IEEE International Conference on Distributed Computing Systems (ICDCS 2010)*, pages 557–566, 2010.

[27] Jaehoon Jeong, Shuo Guo, Yu Gu, Tian He, and D.H.C. Du. Tbd: Trajectory-based data forwarding for light-traffic vehicular networks. In *IEEE International Conference on Distributed Computing Systems (ICDCS 2009)*, pages 231–238, 2009.

[28] Fulong Xu, Shuo Guo, Jaehoon Jeong, Yu Gu, Qing Cao, Ming Liu, and Tian He. Utilizing shared vehicle trajectories for data forwarding in vehicular networks. In *IEEE International Conference on Computer Communications (INFOCOM 2011)*, pages 441–445, 2011.

[29] A. Bachir and A. Benslimane. A multicast protocol in ad hoc networks inter-vehicle geocast. In *The 57th IEEE Semiannual Vehicular Technology Conference VTC 2003-Spring*, volume 4, pages 2456–2460, 2003.

[30] L. Briesemeister, L. Schafers, and G. Hommel. Disseminating messages among highly mobile hosts based on inter-vehicle communication. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 522–527, 2000.

[31] C. Maihofer and R. Eberhardt. Geocast in vehicular environments: Caching and transmission range control for improved efficiency. In *2004 IEEE Intelligent Vehicles Symposium*, pages 951–956, 2004.

[32] M. Durresi, A. Durresi, and L. Barolli. Emergency broadcast protocol for inter-vehicle communications. In *11th International Conference on Parallel and Distributed Systems*, volume 2, pages 402–406, July 2005.

[33] Gökhan Korkmaz, Eylem Ekici, Füsun Özgüner, and Ümit Özgüner. Urban multi-hop broadcast protocol for inter-vehicle communication systems. In *Proceedings of the 1st ACM International Workshop on Vehicular Ad Hoc Networks*, VANET '04, pages 76–85, 2004.

[34] H. Menouar, F. Filali, and M. Lenardi. A survey and qualitative analysis of mac protocols for vehicular ad hoc networks. *IEEE Wireless Communications*, 13:30–35, 2006.

[35] T.V. Nguyen, F. Baccelli, Kai Zhu, S. Subramanian, and Xinzhou Wu. A performance analysis of csma based broadcast protocol in vanets. In *IEEE International*

Conference on Computer Communications (INFOCOM 2013), pages 2805–2813, 2013.

[36] Ieee standard for wireless access in vehicular environments (wave)–multi-channel operation. *IEEE Std 1609.4-2010 (Revision of IEEE Std 1609.4-2006)*, pages 1–89, Feb. 2011.

[37] Katrin Bilstrup, Elisabeth Uhlemann, Erik G. Strom, and Urban Bilstrup. On the ability of the 802.11p mac method and stdma to support real-time vehicle-to-vehicle communication. *EURASIP Journal on Wireless Communications and Networking*, 2009:1–13, 2009.

[38] K. Sjoberg, E. Uhlemann, and E.G. Strom. Delay and interference comparison of csma and self-organizing tdma when used in vanets. *Wireless Communications and Mobile Computing Conference (IWCMC)*, pages 1488–1493, 2011.

[39] F. Borgonovo, A. Capone, M. Cesana, and L. Fratta. Adhoc mac: New mac architecture for ad hoc networks providing efficient and reliable point-to-point and broadcast services. *Wireless Networks*, 10:359–366, 2004.

[40] G.G.Md. Nawaz Ali, Edward Chan, and Wenzhong Li. Supporting real-time multiple data items query in multi-rsu vehicular ad hoc networks (vanets). *Journal of Systems and Software*, pages 2127–2142, Aug. 2013.

[41] A Soua and H. Afifi. Adaptive data collection protocol using reinforcement learning for vanets. In *International Wireless Communications and Mobile Computing Conference (IWCMC)*, pages 1040–1045, July 2013.

[42] IA Soomro, H. Hasbullah, and J.L. bin Ab Manan. User requirements model for vehicular ad hoc network applications. In *International Symposium in Information Technology (ITSim)*, pages 800–804, June 2010.

[43] Public Transportation with Free WiFi. http://www.openwifispots.com/.

[44] Joshua Hare, Lance Hartung, and Suman Banerjee. Beyond deployments and testbeds: experiences with public usage on vehicular wifi hotspots. In *International Conference on Mobile Systems, Applications, and Services (MobiSys)*, pages 393–406, 2012.

[45] Kai-Juan Wong, Bu sung Lee, Boon chong Seet, Genping Liu, and Lijuan Zhu. Busnet: Model and usage of regular traffic patterns in mobile ad hoc networks for inter-vehicular communications. In *Patterns in Mobile Ad Hoc Networks for Inter Vehicular Communications, ICT 2003*.

[46] Jie Luo, Xinxing Gu, Tong Zhao, and Wei Yan. Mi-vanet: A new mobile infrastructure based vanet architecture for urban environment. In *IEEE Vehicular Technology Conference Fall (VTC 2010-Fall)*, pages 1–5, Sept. 2010.

[47] Amin Vahdat and David Becker. Epidemic routing for partially-connected ad hoc networks. Technical report, 2000.

[48] T. Kitani, T. Shinkawa, N. Shibata, K. Yasumoto, M. Ito, and T. Higashino. Efficient vanet-based traffic information sharing using buses on regular routes. In *IEEE Vehicular Technology Conference (VTC-Spring 2008)*, pages 3031–3036, 2008.

[49] Chen Li, Li Zhijun, and Jiang Shouxu. An efficient routing protocol in a bus-based vehicular ad hoc network. In *International Conference on Pervasive Computing Signal Processing and Applications (PCSPA)*, pages 70–73, 2010.

[50] Wei Kuang Lai, Kai-Ting Yang, and Meng-Chong Li. Bus assisted connectionless routing protocol for metropolitan vanet. In *International Conference on Genetic and Evolutionary Computing (ICGEC)*, pages 57–60, Sept. 2011.

[51] 802.11-2012. Ieee standard for information technologytelecommunications and information exchange between systems local and metropolitan area networksspecific requirements part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications. 2012.

[52] A. Ahizoune, A. Hafid, and R. Ben Ali. A contention-free broadcast protocol for periodic safety messages in vehicular ad-hoc networks. In *IEEE 35th Conference on Local Computer Networks (LCN)*, pages 48–55, 2010.

[53] S. Sharafkandi and H.C.D. David. A new mac layer protocol for safety communication in dense vehicular networks. In *IEEE 35th Conference on Local Computer Networks (LCN)*, pages 637–644, 2010.

[54] M.S. Almalag, S. Olariu, and M.C. Weigle. Tdma cluster-based mac for vanets (tc-mac). In *2012 IEEE International Symposium on World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 1–6, June 2012.

[55] H.A. Omar, Weihua Zhuang, and Li Li. Vemac: A tdma-based mac protocol for reliable broadcast in vanets. *IEEE Transactions on Mobile Computing*, 12(9):1724–1736, Sept. 2013.

[56] A. Alonso and C.F. Mecklenbrauker. Stabilization time comparison of csma and self-organizing tdma for different channel loads in vanets. In *12th International Conference on ITS Telecommunications (ITST)*, pages 300–305, Nov. 2012.

[57] Fei Xie, K.A. Hua, Wenjing Wang, and Y.H. Ho. A-adhoc: An adaptive real-time distributed mac protocol for vehicular ad hoc networks. In *Communications and Networking in China 2009*, pages 1–6, 2009.

[58] Fan Yu and S. Biswas. Self-configuring tdma protocols for enhancing vehicle safety with dsrc based vehicle-to-vehicle communications. volume 25, pages 1526–1537, 2007.

[59] S. Bharati and Weihua Zhuang. Cah-mac: Cooperative adhoc mac for vehicular networks. *IEEE Journal on Selected Areas in Communications*, 31(9):470–479, Sept. 2013.

[60] Xuejun Zhuo, Sha Hua, Limin Miao, and Yiqi Dai. Direction matters: A decentralized direction-based tdma scheduling strategy for vanet. In *IEEE International Conference on Communication Technology (ICCT 2011)*, pages 566–571, 2011.

[61] C. L. Robinson, L. Caminiti, D. Caveney, and K. Laberteaux. Efficient coordination and transmission of data for cooperative vehicular safety applications. In *Proceedings of the 3rd International Workshop on Vehicular Ad Hoc Networks*, VANET '06, pages 10–19, 2006.

[62] SAE. Dedicated short range message set (dsrc) dictionary. Technical Report Tech. Rep. Standard J2735, 2006.

[63] X. Xie, R. Cheu, and D. Lee. Calibration-free arterial link speed estimation model using loop data. *Journal of Transportation Engineering*, 127:507–514, 2001.

[64] Mircea Augustin ROSCA and Aura RUSCA. Methods for green times allocation in under-saturated signalized intersection. *UPB Scientific Bulletin, Series D*, 74, Oct. 2012.

[65] Ridong Fei, Kun Yang, and Xueqi Cheng. A cooperative social and vehicular network and its dynamic bandwidth allocation algorithms. In *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS 2011)*, pages 63–67, 2011.

[66] Zhenyu Yang, Ming Li, and Wenjing Lou. Codeplay: Live multimedia streaming in vanets using symbol-level network coding. In *18th IEEE International Conference on Network Protocols (ICNP)*, pages 223–232, 2010.

[67] Fei Xie, K.A. Hua, Wenjing Wang, and Y.H. Ho. Performance study of live video streaming over highway vehicular ad hoc networks. In *IEEE 66th Vehicular Technology Conference,VTC-2007 Fall*, pages 2121–2125, 2007.

[68] M. Asefi, Jon W. Mark, and Xuemin Shen. A mobility-aware and quality-driven retransmission limit adaptation scheme for video streaming over vanets. *IEEE Transactions on Wireless Communications*, 11(5):1817–1827, May 2012.

[69] A. Vinel, E. Belyaev, O. Lamotte, M. Gabbouj, Y. Koucheryavy, and K. Egiazarian. Video transmission over ieee 802.11p: Real-world measurements. In *2013 IEEE International Conference on Communications Workshops (ICC)*, pages 505–509, 2013.

[70] P. Bucciol, E. Masala, N. Kawaguchi, K. Takeda, and J.C. De Martin. Performance evaluation of h. 264 video streaming over inter-vehicular 802.11 ad hoc networks. In *Proc. IEEE PIMRC 2005*, volume 3, pages 1936–1940, 2005.

[71] Rong Yu, Yan Zhang, S. Gjessing, Wenlong Xia, and Kun Yang. Toward cloud-based vehicular networks with efficient resource management. *IEEE Network*, 27:48–55, 2013.

[72] Mohamed Eltoweissy, Stephan Olariu, and Mohamed Younis. Towards autonomous vehicular clouds. *Ad-Hoc Networks*, 49:1–16, 2010.

[73] Euisin Lee, Eun-Kyu Lee, M. Gerla, and S.Y. Oh. Vehicular cloud networking: Architecture and design principles. *IEEE Communications Magazine*, 52:148–155, 2014.

[74] Mahmoud Abuelela and Stephan Olariu. Taking vanet to the clouds. In *Proceedings of the 8th International Conference on Advances in Mobile Computing and Multimedia (MoMM 2010)*, pages 6–13, 2010.

[75] Dawei Jiang, Beng Chin Ooi, Lei Shi, and Sai Wu. The performance of mapreduce: An in-depth study. *Proc. VLDB Endow.*, 3(1-2):472–483, 2010.

[76] Matei Zaharia, Dhruba Borthakur, Joydeep Sen Sarma, Khaled Elmeleegy, Scott Shenker, and Ion Stoica. Delay scheduling: A simple technique for achieving locality and fairness in cluster scheduling. In *Proceedings of the 5th European Conference on Computer Systems*, EuroSys '10, pages 265–278, 2010.

[77] K. Kc and K. Anyanwu. Scheduling hadoop jobs to meet deadlines. In *IEEE Second International Conference on Cloud Computing Technology and Science (CloudCom)*, pages 388–392, 2010.

[78] Faraz Ahmad, Srimat T. Chakradhar, Anand Raghunathan, and T. N. Vijaykumar. Shufflewatcher: Shuffle-aware scheduling in multi-tenant mapreduce clusters. In *2014 USENIX Annual Technical Conference (USENIX ATC 14)*, pages 1–13, 2014.

[79] Mohammad Hammoud, M. Suhail Rehman, and Majd F. Sakr. Center-of-gravity reduce task scheduling to lower mapreduce network traffic. In *Proceedings of the 2012 IEEE Fifth International Conference on Cloud Computing*, CLOUD '12, pages 49–58, 2012.

[80] Tom White. *Hadoop: The Definitive Guide*. O'Reilly Media, Inc., 2012.