2011

Simulating a Universal Geocast Scheme for Vehicular Ad Hoc Networks

Benjamin L. Bovee
University of Massachusetts Amherst

SIMULATING A UNIVERSAL GEOCAST SCHEME FOR VEHICULAR AD HOC NETWORKS

A Thesis Presented

by

BENJAMIN L. BOVEE

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL AND COMPUTER ENGINEERING

May 2011

Electrical and Computer Engineering

SIMULATING A UNIVERSAL GEOCAST SCHEME FOR VEHICULAR AD HOC NETWORKS

A Thesis Presented

by

Benjamin L. Bovee

Approved as to style and content by:

Russel G. Tessier, Chair

Daiheng Ni, Member

Hossein Pishro-Nik, Member

Christopher V. Hollot, Department Head
Electrical and Computer Engineering

ACKNOWLEDGMENTS

I'd like to thank Prof. Tessier for not only his help and advice on this project but for also allowing me the opportunity to work on it. I'd like to thank Mohammad Nekoui for his supervision and patience with me throughout everything. I would also like to thank Prof. Pishro-Nik and Prof. Ni for their help and input with the project and the New England UTC for funding the research. I would also like to thank all the members of Prof. Tessier's Reconfigurable Computing Group who helped me with questions, problems, and reviews throughout the entire research and simulation process. A big thanks to Benjamin St. Peter who helped me overcome numerous software obstacles and lapses in rational thinking. Of course I owe a debt of gratitude to my family and friends for supporting me, especially Marnie Levine. I'd also like to formally thank God that this thesis is complete.

ABSTRACT

SIMULATING A UNIVERSAL GEOCAST SCHEME FOR VEHICULAR AD HOC NETWORKS

MAY 2011

BENJAMIN L. BOVEE, B.S.E.E., UNITED STATES COAST GUARD ACADEMY

M.S.E.C.E., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor Russell G. Tessier

Recently a number of communications schemes have been proposed for Vehicular Ad hoc Networks (VANETs). One of these, the Universal Geocast Scheme (UGS) proposed by Hossein Pishro-Nik and Mohammad Nekoui, provides for a diverse variety of VANET-specific characteristics such as time-varying topology, protocol variation based on road congestion, and support for non line-of-sight communication. In this research, the UGS protocol is extended to consider inter-vehicle multi-hop connections in intersections with surrounding obstructions along with single-hop communications in an open road scenario. Since UGS is a probabilistic, repetition-based scheme, it supports the capacity-delay tradeoffs crucial for periodic safety message exchange. The approach is shown to support both vehicle-to-vehicle and vehicle-to-infrastructure communication. This research accurately evaluates this scheme using network (NS-2) and mobility (SUMO) simulators, verifying two crucial elements of successful VANETs, received packet ratio and message delay. A contemporary wireless radio propagation model is used to augment accuracy. Results show a 6% improvement in received packet ratio in intersection simulations combined with a decrease in average packet delay versus a previous, well-known inter-vehicle communication protocol.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Motivation

There are more than 40,000 people killed in traffic accidents each year in the United States, and in 2006 there were 42,642 reported fatalities from highways alone [1]. Also in the United States there were 6 million traffic crashes in 2006, which resulted in injuries to just under 2.6 million people [1]. That adds up to a traffic crash every 5 seconds, someone sustaining a traffic-related injury every 12 seconds, and a traffic related fatality every 12 minutes. These accidents also contribute to the congestion problem, which, in 2005, resulted in 4.2 billion hours of travel delay, 2.9 billion gallons of wasted fuel, and a net urban congestion cost of nearly $80 billion [1]. Many of these accidents and the congestion that they cause are avoidable. In 2005, of the 43,000-recorded fatalities, 21,000 were caused by roadway departures and intersection related incidents [3]. If vehicles were able to communicate with one another, departing vehicles, those leaving highways, and vehicles about to cross through intersections could let the other vehicles in those areas know about their presence. If a vehicle is doing any of these actions in an unsafe manner, information updates could be sent to the other applicable vehicles to make the drivers aware of the danger, thus helping to reduce the number of accidents. Also, older drivers are quickly becoming a significant fraction of the driving public. These drivers are challenged by changes in their visual acuity and a reduction in their ability to respond quickly to changes in road conditions.

Information updates could especially help older drivers understand their driving environment and to assist them in avoiding potential road hazards.

1.2 Intelligent Transportation Systems

In order to advance transportation science, technology, and analysis, and to improve the coordination of transportation research the U.S. Department of Transportation (USDOT) created the Research and Innovative Technology Administration (RITA) in 2005 [6]. Two of the main functions of RITA are to: 1) Coordinate the USDOT research and education programs and 2) Bring advanced technologies into the transportation system. The main office of RITA, which focuses on these two functions, is the Intelligent Transportation System Joint Program Office (ITS JPO) [2]. The focus of the ITS program is intelligent vehicles, intelligent infrastructure and the creation of an intelligent transportation system through integration with and between these two components. The overall advancement of ITS is done through investments in it's major initiatives to improve safety, mobility, and productivity. The allocation of 75 MHz in the 5.9 GHz band for Dedicated Short Range Communications (DSRC) was proposed by the FCC in order to help ITS achieve these goals [1]. ITS then created the Vehicle Infrastructure Integration (VII) initiative to utilize this communications band. VII proposed to use DSRC to establish vehicle-vehicle (V2V) and vehicle-infrastructure components (V2I) communications to deliver timely information necessary for collision and congestion avoidance. In the past two years, the VII initiative has been replaced by the IntelliDrive (SM) initiative, which has subsequently been replaced by the Connected Vehicle program, which is a multimodal initiative, that aims to enable safe, interoperable networked wireless communications among vehicles, the infrastructure, and passengers' personal communications devices [5]. The Connected Vehicle research program envisions that each future vehicle will be equipped with an On-Board Equipment (OBE), which includes a DSRC transceiver, a Global Positioning System (GPS) receiver, and a computer. Also equipped with similar devices, Roadside Equipment (RSE) will be deployed at selected roadside locations. Therefore, vehicles will be able to communicate with each other and with the roadside by means of DSRC. There is also continuing research being done to include smart phone technologies, which may be used to supplement the OBE [2].

The 75 MHz frequency band allocated to DSRC is divided into seven 10 MHz wide channels. These seven designated channels are divided up as follows: one is assigned to V2V public safety communication (ch. 172), one is assigned to intersection public safety (ch. 184), four channels are assigned to public safety and private applications (ch. 174, ch. 176, ch. 180, ch. 182), and one channel is the control channel (ch. 178) used mainly for broadcast traffic. The ITS architecture utilizing all of these channels, thus showing the importance of DSRC communications, can be seen in Fig.1 below.

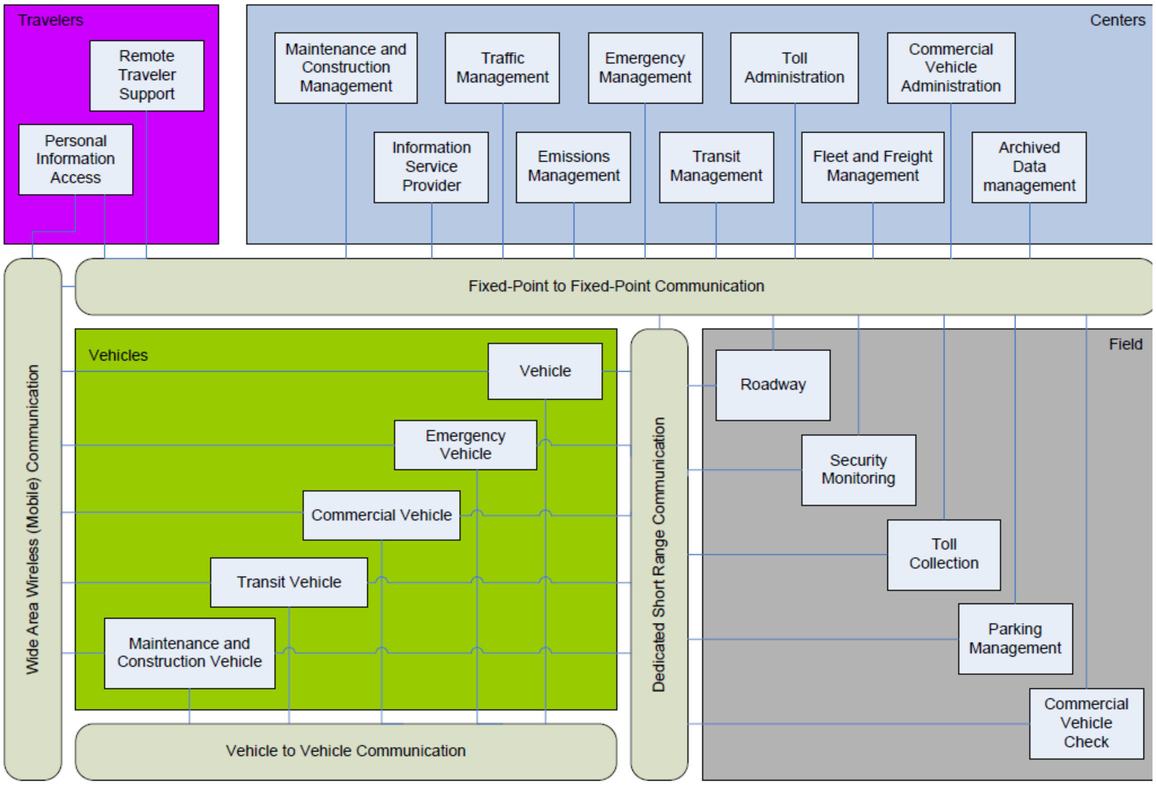
Travelers
Remote Traveler Support
Personal Information Access
Centers
Maintenance and Construction Management
Traffic Management
Emergency Management
Toll Administration
Commercial Vehicle Administration
Information Service Provider
Emissions Management
Transit Management
Fleet and Freight Management
Archived Data management
Fixed-Point to Fixed-Point Communication
Wide Area Wireless (Mobile) Communication
Vehicles
Vehicle
Emergency Vehicle
Commercial Vehicle
Transit Vehicle
Maintenance and Construction Vehicle
Vehicle to Vehicle Communication
Dedicated Short Range Communication
Field
Roadway
Security Monitoring
Toll Collection
Parking Management
Commercial Vehicle Check

Fig. 1.1: Intelligent Transportation Systems Architecture [7]

The end product of the communication architecture is the Connected Vehicle applications for safety and mobility. Connected Vehicle safety applications would allow vehicles to have 360-degree awareness to inform a vehicle operator of hazards and situations they can't see. These safety applications have the potential to reduce crashes through advisories and warnings. For instance, vehicle operators may be advised of a school zone; sharp ramp curve; or slippery patch of roadway ahead [2]. Drivers could also be advised of the presence of Connected Vehicle -equipped bicycles and pedestrians around them, which would enhance the safety of pedestrians and bicyclists as well as motorists. Warnings could be provided in more imminent crash situations, such as during merging operations that put vehicles on a collision path, or when a vehicle ahead stops suddenly. The mobility applications are intended to provide a connected, data-rich travel environment based on information transmitted anonymously from thousands of vehicles that are using the transportation system at a particular time. This information could help transportation managers monitor and manage transportation system performance. Adjusting traffic signals or transit operations, or dispatching maintenance crews or emergency services could do this. This information could also help transportation agencies and fleet operators to manage crews and use resources as efficiently as possible [2]. Providing travelers with real-time information about traffic congestion and other travel conditions helps them make more informed decisions that can reduce the environmental impact of their trip. Informed travelers may decide to avoid congestion by taking alternate routes or public transit, or by rescheduling their trip – all of which can make their trip more fuel-efficient and eco-friendly. The ability for vehicles to "talk to" the infrastructure could provide information to the vehicle operator so that he or she can drive through a traffic signal network at optimum speeds to reduce stopping. Many transportation management activities that enhance mobility, by reducing vehicle idling due to traffic congestion, also potentially reduce emissions [2].

1.3 Communication Requirements of ITS Connected Vehicle

As previously stated, one of the goals of the Connected Vehicle application is to give the user a 360-degree awareness of their surroundings. An example figure of this objective can be seen below.

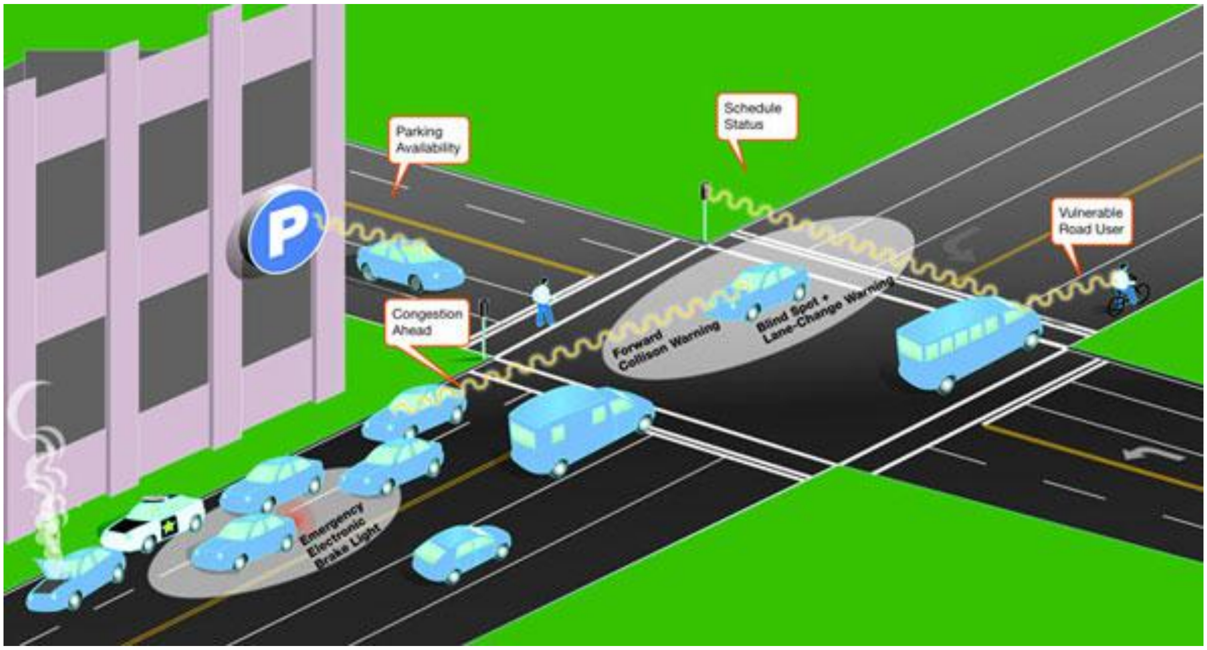

Fig. 1.2: A 360° situational awareness provided by Connected Vehicle [1]

In order to accomplish this goal, each vehicle would require a map of the relative position of all neighboring vehicles. This issue is at the heart of the safety applications provided by Connected Vehicle. By knowing the distance to all the vehicles in the immediate area, the safety system can inform the driver of any potentially hazardous situations. Most vehicles in the near future are expected to maintain the digital road maps that are already in many of today's current vehicles that provide directions with the help of GPS location data. Using these maps, along with relative distances of other vehicles, the safety system could help the driver in higher risk situations such as changing lanes on a highway, merging traffic when highway lanes decrease, traversing intersections/roundabouts, and many others. By knowing other vehicles speed and direction, the safety systems could predict future positions and calculate when vehicles are either in or about to enter hazardous situations. For instance, a senior design project at Umass investigated a scenario in which a vehicle that is about to run a red light at an intersection alerts all of the other vehicles in the intersection of the impending danger. By knowing the vehicles speed and distance to the intersection, the system can predict if the vehicle will run the red light in such a manner that it can send a warning to all other drivers prior to the actual event [9].

There are two types of messages that need to be sent by the communications scheme: periodic and event-driven. To give the 360-degree view to the user, the majority of messages that would need to be sent by the vehicles would be periodic status updates. These messages would let other vehicles in the area know other vehicle's information such as current location, speed, and rate of acceleration. The other, less frequent type of sent messages would be event-driven safety messages. These messages are disseminated throughout the network in case of emergency. Because of the high priority of these messages, they can be sent on a channel dedicated to ensuring the safety of life [10]. Also, since they are sent much less frequently, they do not raise that much of a capacity concern.

It has been shown in previous research that periodic update messages can be as small as 51 bytes per packet [3]. Location information, located on Earth's surface with 1 cm resolution, can be delivered with $\log2(2\pi6.4 \times 106\ \mathrm{m}/10-2\ \mathrm{m}) + \log2(\pi6.4 \times 106\ \mathrm{m}/10-2\ \mathrm{m}) = 62.81 \leq 63$ bits where $6.4\times106\mathrm{m}$ is the Earth's radius [3]. Relative location information within 100m (in a 200m×200m square centered at the reference point) in a Cartesian system with 1 cm resolution can be delivered with $2\ \log2(200\mathrm{m}/10-2\ \mathrm{m}) = 28.6 \leq 29$ bits. Assuming each vehicle transmits its position in absolute form, its velocity, and the relative positions and velocities of vehicles immediately in front, behind, left, and right, $63 + 29 + 4(29 + 29) = 324$ bits or 41 bytes need to be transmitted. Adding 2 bytes for the ID of each vehicle, 51 bytes in total are necessary. Additional bytes are allocated for other uses, such as detection of an obstacle and its position information, emergency car and its position information, emergency braking, etc, along with 80 bytes for standard network protocol headers [11]. It is assumed that the periodic messages will be about 200 bytes altogether [10].

Having established the size requirement of vehicle network safety messages, the next step is to explore the frequency and range of transmissions. At 100km/hr (62 mph), a vehicle moves 6m in 216ms. A 6m distance is the approximate accuracy of GPS and most off the shelf GPS devices have an update rate of less than 5 Hz [10]. Therefore, an update frequency of approximately 5 messages/second or a new message every 200ms guarantees the accurate and up-to-date status of vehicles. Broadcast ranges should lie between 50 and 300 meters. When a vehicle transmits its safety message it does so to inform surrounding vehicles of its state of motion. Oncoming vehicles that are close need to be told immediately. To make data transport economical, oncoming vehicles that are far away should be told when they are closer [11]. The distinction between near and far can be made precise by thinking of the message as having a critical range. A vehicle should receive the message before it reaches the critical range. For example, if a vehicle is stopped, it would like its message to reach oncoming vehicles at freeway speeds before they hit 250 meters to give them ample time to take evasive action. Hence, we assume that a stopped vehicle message would be presented to the data transport service with a specified range of 300 meters. In general, the critical range number would depend on the content of the message and the message range would be a value greater than the critical range. The 50-meter lower bound is derived from the vehicle density in a jammed lane. This value is about 217 vehicles/lane/mile. It corresponds to about 5 meters between cars. A car itself is about 5 meters in length. This adds up to 10 meters. To cover the width of a multi-lane highway with its merge ramps, etc., one can assume the minimum communication range will be 50 meters [11].

The final requirement, which is related to the frequency of safety messages, is the maximum allowable delay. Since the useful lifetime of a packet has been established as 200ms, the packet will basically be useless after this amount of time. It is also important to note the criticality of the packet delay. Not only will vehicles be traveling at speeds up to and beyond 80 mph (128 km/hr or 7.1 meters every 200ms) but there are also the unavoidable delays due to human reaction time. This time can be anywhere from 500ms to 1.2s from the moment an event occurs until an actual decision is made, depending on how unexpected the event is [3]. In the future this delay may become avoidable through the use of fully autonomous safety systems, but that is outside the scope of this research.

1.4 Introduction to a Universal Geocast Scheme Proposed for VANETs

This scheme accounts for a diverse variety of VANET-specific characteristics such as the gradual introduction of technology, highly dynamic topology, road-constrained vehicle movement and the presence of obstacles [12]. Most packet collisions in VANETs occur due to hidden nodes. In unicast communications, a two-way handshaking is performed prior to the actual transmission in order to alleviate the hidden node problem. However, this procedure congests the network with a lot of overhead in the case of broadcast messaging, which is the dominant mode of communication in VANETs. This is especially true for periodic safety messages that can be just as small as the RTS/CTS and ACK messages sent to set-up and confirm reception. The proposed universal geocast scheme incorporates a geometrical framework and is based on retransmissions rather than the two-way handshaking. This makes it appropriate for urban as well as rural area deployments. Moreover, by making the scheme probabilistic, capacity-delay tradeoffs crucial for safety message exchange are addressed. The scheme is able to take advantage of any infrastructure in place, such as roadside transceivers for forwarding vehicle messages, although the network can still operate in a purely ad hoc manner. Very simple simulation results, done previously [12], confirm that this scheme can dramatically improve the probability of the reception between nodes in two scenarios.

1.5 Problem Statement

Because of the size, frequency, and expected number of receivers of periodic safety messages, traditional wireless protocols such as IEEE 802.11 need to be drastically revised in order to work with the Connected Vehicle system. The 'Universal Geocast Scheme for VANETs' presented by Mohammad Nekoui and Hossein Pishro-Nik [12] hopes to accomplish this by making changes to the various parts of the proposed IEEE 802.11p protocol. Although their scheme had been tested in theory, using probability models for average reception probability, it needed to be proven with a better, more defined simulation architecture that could encompass more of the behaviors of a mobile wireless communications network. The simulations done in this research provide better comparisons to other leading proposals for the new inter-vehicle communication protocol and has allowed improvements to be made by evaluating situations in which UGS formerly performed poorly.

This research has addressed this problem by using a network simulator along with a mobility model to test the protocol under the most realistic conditions possible. This included integrating a new propagation model from recent research into the propagation loss of DSRC signals in an urban environment. During the testing of the protocol specific changes were made to improve it. An example of a place where improvement was made was in the determination of the size of the interval from which the backoff will be selected, based on message retransmission number and current vehicle density. The backoff is the randomly chosen amount of time that a node will defer its transmission for upon finding the channel busy. It has already been seen [12] that a change to this backoff interval has shown vast improvements in the number of vehicles that are able to receive another vehicle's transmission. The two measures of success for the communication scheme are reception ratio and delay. The *delay* is the amount of time required for a transmitted message to reach the intended receivers. The *reception ratio* is the number of other vehicles in the transmitting vehicle's geocast range that receive the packet, within the packet's proposed 200 ms lifetime.

1.6 Thesis Organization

This thesis document is organized as follows. Chapter 2 provides an overview of VANETs and the communications protocols they are expected to use. The chapter also gives an introduction and short description of the IEEE 802.11p protocol proposed for VANETs along with a more thorough explanation of the proposed Universal Geocast Scheme for VANETs. Chapter 3 discusses previous work on the other VANET communications schemes. It also discusses recent work that has been done to more accurately simulate radio propagation models, and also mobility and traffic models.

Chapter 4 discusses the network simulator chosen for this project, NS-2. It discusses the current capabilities of NS-2 and briefly describes the evolution of mobile networking in NS-2. There is also a discussion of the typical output from NS-2 simulations and how this output can be analyzed to determine received ratio and delay. Chapter 5 then goes on to discuss the changes that were made to NS-2 and the IEEE 802.11 framework, along with the implementation of the new propagation model. Chapter 6 discusses the specific methods of extracting results from NS-2 data output. There is a discussion of the results of the simulations and a comparison with the results of similar schemes and their simulations. Finally, Chapter 7 concludes the thesis by reviewing the work that has been completed in this thesis, and describes what future work may be done.

CHAPTER 2

BACKGROUND

Before an explanation of the communication protocol described in this thesis, it is best to also describe some of the other, popular, VANET communication protocols. First, a brief description of vehicular ad hoc networks and their use in the intelligent transportation system will be given. This chapter presents a brief discussion of vehicular ad hoc networks, their infrastructure, and why they are best suited for vehicular communications. There is also a discussion of possible choices for communication schemes in ad hoc networks.

2.1 Vehicular Ad hoc Networks

A wireless ad hoc network is a decentralized wireless network. The network is ad hoc because it does not rely on a pre-existing infrastructure, such as routers in wired networks or access points in infrastructure wireless networks. Instead, each node participates in routing by forwarding data for other nodes, and so the determination of which nodes forward data is made dynamically based on network connectivity [13]. The decentralized nature of wireless ad hoc networks makes them suitable for a variety of applications where central nodes can’t be relied on, and may improve the scalability of wireless ad hoc networks compared to wireless managed networks. However, theoretical and practical limits to the overall capacity of such networks have been identified. The presence of a dynamic and adaptive routing protocol will enable ad hoc networks to be formed quickly [13].

A mobile ad hoc network (MANET) consists of mobile platforms, often referred to as nodes and consisting of a router with multiple hosts and wireless communications devices, which are free to move about arbitrarily. The nodes may be located in any number of vehicles including airplanes, ships, tanks, cars, and even on people or very small devices. A MANET is an autonomous system of mobile nodes. The system may operate in isolation, or may have gateways to and interface with a fixed network. MANET nodes are equipped with wireless transmitters and receivers using antennas, which may be omni-directional (broadcast), highly directional (point-to-point), steerable, or some combination thereof. At a given point in time, depending on the nodes' positions and their transmitter and receiver coverage patterns, transmission power levels and co-channel interference levels, a wireless connectivity in the form of a random, multi-hop graph or "ad hoc" network exists between the nodes. This ad hoc topology may change with time as the nodes move or adjust their transmission and reception parameters [14].

A vehicular ad hoc network (VANET) is simply a mobile ad hoc network that uses cars and trucks along highways and road systems as nodes in a network to create a mobile network. VANETs turn every participating car into a wireless router or node. As cars travel out of the signal range of the network, other cars join in, connecting the vehicles to one another in a very dynamic fashion.

2.2 Communication Protocols in VANETs

Some of the characteristics of the communications protocol for VANETs were previously mentioned in section 1.3. Because of the stringent delay requirements of safety traffic, the transmission delay of the protocol must be very low, at least less than 200 ms. It must be able to support mobility in an ever changing constellation of nodes where the same set of nodes are almost never present for any set period of time. The protocol must be able to effectively coordinate tens, possibly hundreds, of sources of broadcast traffic. Because of these conditions, and the lack of centralized control, it makes sense to only consider protocols that are broadcast in nature. The problem of hidden nodes, explained in section 2.3, also plays a significant role in the selection of a broadcast protocol. This is due to the forgoing of two-way handshaking due to the considerable amount of unnecessary overhead it would cause. Given these criteria, the two types of broadcast protocols to be considered are CSMA/CA based and repetition based.

2.2.1 Repetition Based Protocols

The fundamental idea behind repetition-based broadcast is the repeating of a message several times in an interval shorter than or equal to its lifetime to ensure high probability of reception. In repetition-based broadcast protocols, time is divided into frames, the maximum length of which must not be greater than the lifetime of a safety message. Each frame, in turn, is divided into timeslots with length equal to the transmission time of a single packet [3]. An example of how the scheme allows two nodes to transmit packets can be seen in the figure below.

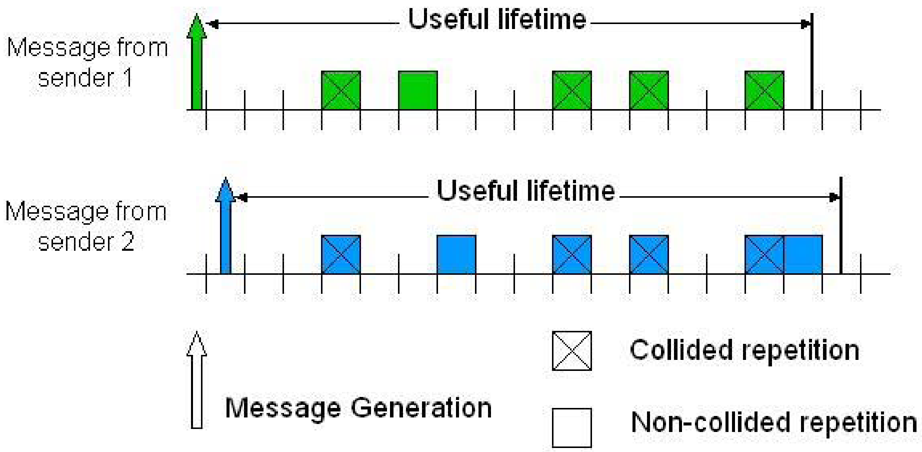

Fig. 2.1 Concept of Repetition Based Transmission [11]

The segmenting of the useful lifetime into transmission slots can be observed in the figure. In each timeslot when a node is not transmitting it is listening for incoming packet transmissions. Some of the flavors of repetition-based protocols include:

1. *Asynchronous Fixed Repetition (AFR)* – In AFR, as well as in all other fixed repetition protocols, the design parameter is the number of repetitions *k*. The protocol randomly selects *k* distinct slots out of the *n* slots constituting the lifetime. The protocol is so called since the number of repetitions is fixed. The radio does not listen to the channel before it sends a packet with AFR.
2. *Asynchronous p-persistent Repetition (APR)* – The p-persistent repetition protocol determines whether to transmit a packet in each of the n slots in the lifetime with probability $\frac{k}{n}$, where *k* is again a configuration parameter of the protocol. The average number of repetitions of a message is *k*. However, for each realization, the exact number of repetitions is different. Like AFR, the radio does not listen to the channel before it sends a packet.

3. *Synchronous Fixed Repetition (SFR)* – This protocol is the same as AFR except that all the slots in all the nodes are synchronized to a global clock.
4. *Synchronous p-persistent Repetition (SPR)* – The SPR protocol is the same as the APR protocol except that all the slots in all the nodes are synchronized to a global clock.
5. *Asynchronous Fixed Repetition with Carrier Sensing (AFR-CS)* – AFR-CS generates repetitions in the same way as the AFR protocol. Prior to transmitting a packet, this protocol senses the channel. Upon finding the channel idle, the packet is transmitted. If the channel is busy, the packet is dropped and transmission is deferred to the next selected time slot for transmission. Hence the selected number of retransmissions, *k*, will most likely not be the actual number of transmissions.
6. *Asynchronous p-persistent Repetition with Carrier Sensing (APR-CS)* – This is similar to AFR-CS except that the slots for message repetitions are selected in the p-persistent manner, mimicking APR.

These descriptions are detailed in [11]. It is important to note that both of the two previously listed types of repetition-based protocols do include some characteristics of CSMA/CA protocols in that they sense the channel. Then, depending on whether they sense the channel to be busy or idle, they decide whether or not to transmit.

2.2.2 CSMA/CA

Carrier Sense Multiple Access (CSMA) is a probabilistic Medium Access Control (MAC) protocol in which a node verifies the absence of other traffic before transmitting on a shared transmission medium, in this case a 10 MHz band in the 5.9 GHz range. *Carrier Sense* describes the fact that a transmitter uses feedback from a receiver that detects a carrier wave before trying to send. It attempts to detect the presence of an encoded signal from another station before attempting to transmit. If a carrier is sensed, the station waits for the transmission in progress to finish before initiating its own transmission. *Multiple Access* describes the fact that multiple stations send and receive on the medium. Transmissions by one node are generally received by all other stations using the medium. Carrier Sense Multiple Access With Collision Avoidance (CSMA/CA) is a modification of CSMA. *Collision avoidance* is used to improve the performance of CSMA by attempting to be less “greedy” on the channel. If the channel is sensed busy before transmission then the transmission is deferred for a pseudo-random interval. This reduces the probability of collisions on the channel.

2.3 IEEE 802.11p

IEEE 802.11p is a recently approved amendment to the IEEE 802.11 standard to add wireless access in vehicular environments (WAVE). The specification seeks to accomplish two things:

• Describes the functions and services required by WAVE-conformant stations to operate in a rapidly varying environment and exchange messages either without having to join a Basic Service Set (BSS) or within a WAVE BSS [15].

• Defines the WAVE signaling technique and interface functions that are controlled by the IEEE 802.11 MAC [15].

To understand the 802.11p protocol it is best to briefly describe the original 802.11 protocol. The 802.11 protocol itself, like any 802.x protocol, covers the MAC and physical (PHY) layers. The MAC layer defines the distributed coordination function (DCF). For 802.11 this function is the CSMA/CA mechanism, discussed in section 2.2.2. One concern with this scheme is the hidden node problem. When two nodes are far enough away from one another they will not be able to sense each other's transmission. However, there may be nodes in-between the two that can receive both transmissions. Since both nodes may perceive the channel as open while the other is actually transmitting, they transmit. This causes any nodes in the middle to receive two messages at the same time, forcing them to drop either or both. A simple drawing of the hidden node problem can be seen below.

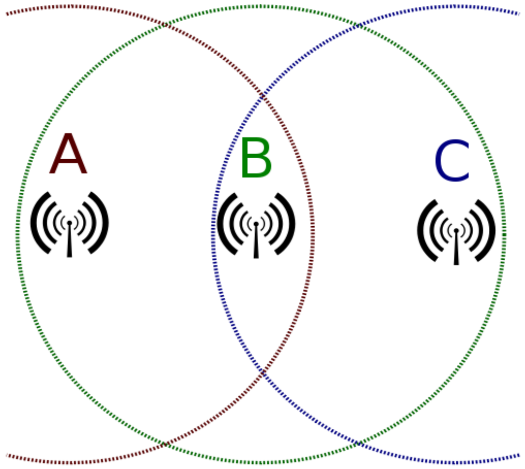

Fig. 2.2 Hidden Node Problem

Another problem with the CSMA/CA scheme, in addition to that of hidden nodes, is that if two nodes sense the channel at the same time and then transmit, a collision will occur. These collision situations must be identified so the MAC layer, rather than the upper layers, which would cause even more delay, can retransmit the packets. Thus the

CA mechanism is coupled with a positive acknowledge (ACK) scheme as follows: A node wishing to transmit will sense the medium. If the medium is busy, the transmission is deferred. If the medium is idle for a specified amount of time called the distributed interframe space (DIFS) in the standard, the node is allowed to transmit. The receiving node checks the cyclic redundancy check (CRC) of the received packet and sends an ACK packet. Receipt of this ACK by the transmitter indicates that no collision occurred. If the sender does not receive an ACK, it will retransmit the frame until receiving an ACK, or throw the packet away after a given number of retransmissions. According to the protocol, a maximum of 7 retransmissions are allowed before the frame is dropped [16]. To combat the hidden node problem, IEEE 802.11 standards employ a virtual CS mechanism. A station wanting to transmit a packet first transmits a short control packet called a request to send (RTS), which includes the source, destination, and duration of the intended packet and ACK transaction. The destination station responds with a response called clear to send (CTS), which includes the same information. All other stations that receive either the RTS and/or CTS can set their virtual CS indicator, called a network allocation vector (NAV), for the given duration and use this information together with the physical CS when sensing the medium. The physical layer carrier sensing is called clear channel assessment (CCA). The CCA is combined with the NAV to indicate the busy state of the medium [16]. For instance, in the Fig. 2.2, if A has a transmission to send, it will first send an RTS out, which will be received by B. C will not hear the RTS, but, provided B is free to receive the transmission from A and sends a CTS, C will hear this. Thus, C will then indicate it's virtual carrier sensing as busy, and will defer from transmitting a message that might have otherwise collided with A's transmission. It should also be noted that, due to the fact that the RTS and CTS are short frames, the mechanism also reduces the overhead of collisions, since these short transmissions allow faster recognition of collisions than would be possible for the transmission of an entire packet. A simplified algorithm of the IEEE 802.11 CSMA/CA scheme can be seen below.

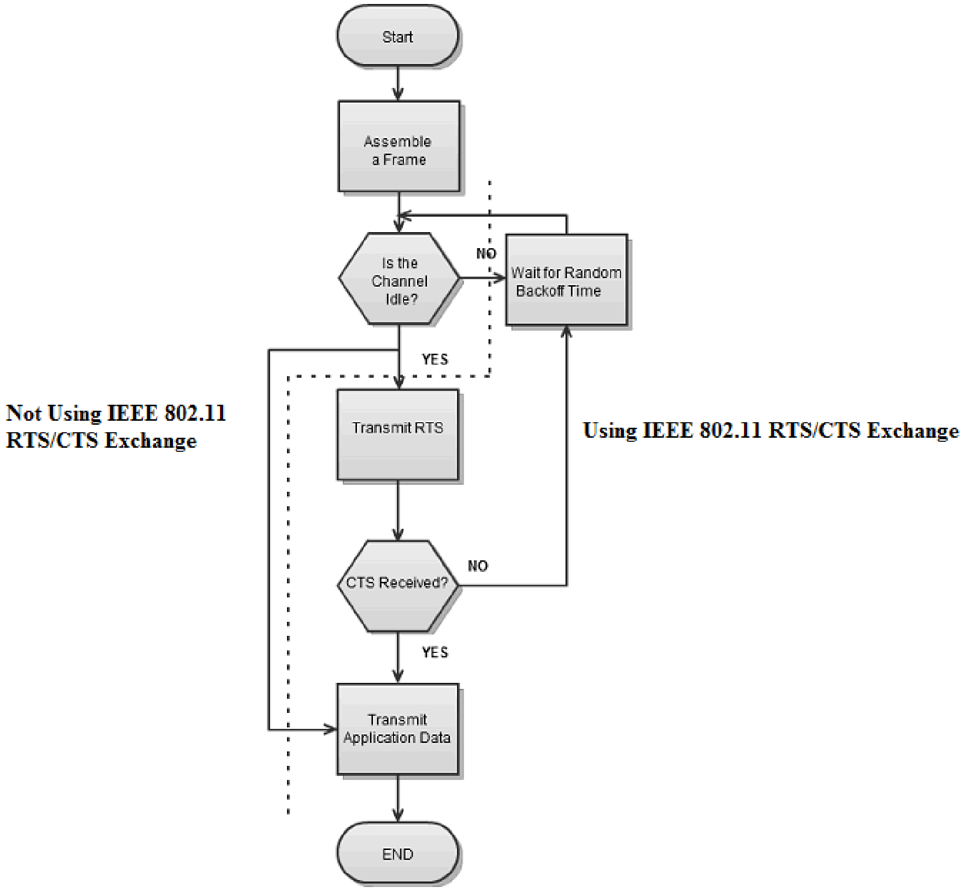
Start
Assemble a Frame
Is the Channel Idle?
NO
Wait for Random Backoff Time
YES
Not Using IEEE 802.11 RTS/CTS Exchange
Transmit RTS
Using IEEE 802.11 RTS/CTS Exchange
CTS Received?
NO
YES
Transmit Application Data
END

Fig. 2.3 Simplified CSMA/CA Algorithm [35]

Another important aspect of the IEEE 802.11 standard is the random wait time chosen from a backoff window. When one node transmits its packet, other neighboring nodes, which also have a packet to send, find the channel busy and defer their transmission for a random time. This random time is: $i * t_s$ where i is a random integer uniformly selected from the backoff window $\{0,\cdots,cw-1\}$, t_s is the unit time slot duration, and *cw* is the contention window. The contention window has a minimum and maximum value established. If an ACK is not received from a transmitted message, the *cw* will usually be doubled when retransmitting the message. This action allows fairness in congested scenarios. The IEEE 802.11 standard specifies other characteristics of transmissions, such as modulation and coding rates. However, to maintain the focus of this research these characteristics will not be discussed in detail here.

The physical layer of IEEE 802.11p is based on IEEE 802.11a Orthogonal Frequency Division Multiplexing (OFDM), so that existing 802.11a WI-FI chip architectures can be used as the basis for inexpensive WAVE implementations and deployment. The use of existing WI-FI chip architectures has great advantages for economies of scale in the production of WAVE devices, taking advantage of the large market for consumer WI-FI [17]. Early testing of existing 802.11a chipsets by Atheros and companies showed adequate performance of the PHY at vehicle speeds, so changes to the 802.11 PHY as part of Amendment PHY have been relatively minor [17]. These changes include adjusting the frequency range because DSRC operates at 5.9 GHz while the 802.11a band stops at 5.825 GHz, and also using a 10 MHz channel.

There have been many small changes to the MAC for IEEE 802.11p, but below is a short summary of those changes. For one, all IEEE 802.11p radios are by default in the same channel and configured with the same Basic Service Set Identification (BSSID). This was done so communications may begin in a very short period of time, like when two vehicles are approaching one another at rapid speeds [18]. A WAVE BSS (WBSS) is a type of BSS consisting of a set of cooperating stations in WAVE mode that communicate using a common BSSID. A WBSS is initialized when a radio in WAVE mode sends a WAVE beacon, which includes all necessary information for a receiver to join [18]. A radio joins a WBSS when it is configured to send and receive data frames with the BSSID defined for that WBSS. Conversely, it ceases to belong to a WBSS when its MAC stops sending and receiving frames that use the BSSID of that WBSS. A station shall not be a member of more than one WBSS at one time. A station in WAVE mode shall not join an infrastructure BSS or IBSS, and it shall not use active or passive scanning, and lastly it shall not use MAC authentication or association procedures [18]. A WBSS ceases to exist when it has no members. The initiating radio is no different from any other member after the establishment of a WBSS. Therefore, a WBSS can continue if the initiating radio ceases to be a member [18].

2.4 Universal Geocast Scheme for Vehicle Ad hoc Networks

This section aims to give a more in-depth view of how the Universal Geocast Scheme presented earlier attempts to successfully transmit data in VANETs. The proposed algorithm is based on retransmitting a packet during its useful lifetime. Each retransmission is carried out in a single or multi-hop fashion based on the geometry of the surroundings and amount of useful lifetime of the packet that remains. Note that the useful lifetime (or acceptable delay to deliver a packet) is assumed to be the time interval between the generations of two subsequent data packets, which is 200 ms for a 5 GHz. GPS device. Calculations done in previous work [12] and simulations discussed in section 6.2 prove that this time frame provides enough opportunities for each vehicle to retransmit its packet several times within its useful lifetime.

2.4.1 Proposed MAC

The MAC of this scheme is similar to that of regular IEEE 802.11. As mentioned previously in section 1.4, because of the size of these safety messages and their broadcast nature there are no RTS/CTS or ACK messages. A vehicle with a status update to transmit, first listens to the channel. If the channel is idle, the packet is sent and if the channel is busy, the transmission is deferred for a random amount of idle channel time, and then sent. Without an ACK exchange, the vehicle has no way of knowing if all intended receivers have received the packet. The MAC will retransmit the packet. To allow for fairness between transmission opportunities of contending neighbors, it increases the size of the interval from which the backoff window is chosen. That is, a vehicle, after transmitting a copy of its packet, backs off and waits for its next turn by choosing a random integer from the interval $\{0,\cdots,\lfloor 2^{(i*X+Y)}cw \rfloor -1\}$, where i is the retransmission number of the packet, and X and Y represent variables that take into account the environment in which the vehicle is transmitting. X and Y are derived from research conducted in [12]. In this previous research the backoff interval is actually represented by $\{0,\cdots,\left\lfloor 2^{(i+\frac{k}{k_j})} cw \right\rfloor -1\}$ where k represents the vehicular density as observed by the transmitting vehicle and k_j is the jam density, originally calculated by [12] to be 250 vehicles/mile/lane. The research done in this current project was the first to adequately test these values. Simulation results showed that even when the k/k_j value is set equal to 1, the largest value it can be, the backoff interval was not large enough, even when the vehicular density was not at a maximum level. Individual packets were being retransmitted too many times causing congestion and ultimately decreasing the reception ratio. The larger the backoff interval, the higher the likelihood of a large backoff period being chosen for a packet before being transmitted. The longer that packets wait prior to their transmission, the less number of times they actually get transmitted within their useful lifetimes. Initial simulations showed that multiplication of the retransmission number *i* by integer values, represented by variable *X*, provided better performance results than the addition of the fraction represented by $\frac{k}{k_j}$ to the retransmission number *i*. Further simulations proved that by also adding an integer value, represented by variable *Y*, to the retransmission number the results would be improved even further. The purpose of *X* and *Y* are still to represent the vehicular density, and a part of this research worked towards finding the optimal values for these variables, which will be discussed in Chapter 6. The basis of the fairness involved in this scheme is a vehicle that has already had a chance to transmit its packet would have to, on average, wait a longer time for its next retransmission of the same packet in comparison to a node that has not yet had a chance to transmit. Also the vehicular environment variables *X* and *Y* take into account the vehicular density of the area in which the transmissions are being made, so in situations where more vehicles are trying to transmit, less transmissions are made per vehicle, helping to keep the channel from being overly congested. The fairness of this MAC protocol was proved mathematically in [10]. Below is a simplified algorithm of the proposed MAC.

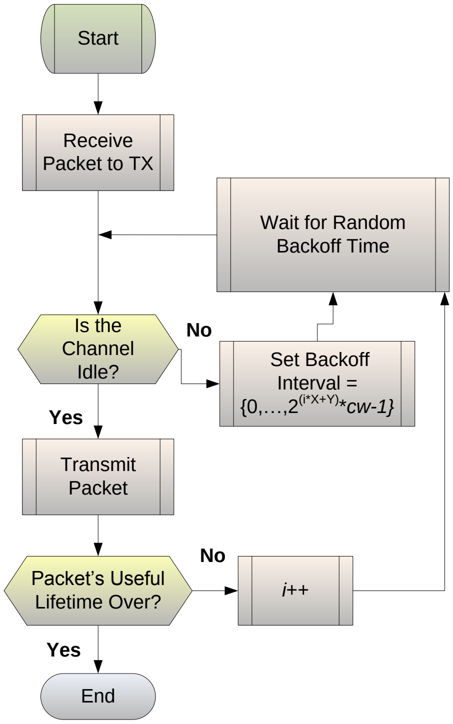
Start
Receive Packet to TX
Wait for Random Backoff Time
Is the Channel Idle?
No
Set Backoff Interval = {0,…,2^(i*X+Y)*cw-1}
Yes
Transmit Packet
No
Packet's Useful Lifetime Over?
i++
Yes
End

Fig. 2.4 Simplified Algorithm of Proposed MAC

2.4.2 Proposed Power and Hop Control Scheme

It has long been established that single-hop, long range communication decreases the throughput of wireless networks due to the increased contention for media access [12]. Multi-hop communication is a solution for dense areas, but it would bring about unwanted delay. This scheme takes into consideration the surrounding area of a vehicle wishing to transmit a message using GPS and digital maps and then decides whether or not a packet should be sent via single or multi-hop. Each vehicle, upon generating a packet, will deploy the channel access algorithm described in the last subsection to gain access to the channel and retransmit its packet for as many times as possible within its useful lifetime. Prior to each retransmission, a vehicle would decide whether to send this copy through single or multiple hops. The scheme considers three factors to decide between single and multiple hops. The first is the geometrical properties of the neighborhood. The more buildings and obstructions in the area, the higher the probability the packet should be sent multi-hop. The second is the vehicular density. The lower the vehicular density, the higher the probability of sending the message single-hop because there is not much contention for channel access in such sparse areas. The third factor is the time past from the generation of the packet. That is, the lower the amount of time left to the end of a packet's useful lifetime, the higher the probability of single-hop transmission. This characteristic is a result of the fact that sending a packet whose useful lifetime is nearly coming to an end via multiple hops renders it useless even if it does reach the intended destination, but after the deadline. The probability models and fundamental geometrical definitions used to determine these probabilities can be seen in [12]. This research was focused on finding some of these probability and geometry values and determining ways of assessing them via simulations. Once the vehicle has decided whether to send a packet using either a single-hop or a multi-hop approach, the power of this transmission can be decided. This power, P_i is either P_i^1 or P_i^2 where P_i^1 is the transmission power required to reach the furthest vehicle within the geocast range of the vehicle, in a single transmission. P_i^2 is the transmission power to reach the furthest vehicle within its geocast range, to which it has LOS. Note that in the latter case, the packet needs to go through additional hops (within its useful lifetime) to reach all its other intended receivers, whereas in the former case, the packet is sent in just one single-hop transmission to reach all vehicles in the geocast range. Also note that if the farthest vehicle within the geocast range is not in LOS, then P_i^1 must be large enough such that it can either overcome penetration loss through buildings or be diffracted around them. For instance, Figure 6.14 in the results section demonstrates that the power level necessary for a NLOS single hop transmission with a 72.8 meter range results in a LOS transmission with a range of over 900 meters.

Packets must include in their headers their time stamp, so other vehicle can determine the end of their useful lifetimes. Upon receiving multi-hop packets, vehicles sort them in descending order of their time stamp in what is called the *priority stack*. The responsibility of forwarding the multi-hop packet is now incumbent on the vehicles that can see regions not in the LOS region of the original sender, but inside its geocast range. These vehicles have an additional phase in their transmission policy. First they need to determine whether they are sending their own or someone else's packet. Next they need to decide whether the packet is going in a single or via multiple hops. Before gaining channel access, such a vehicle decides to transmit the packet that resides on top of its priority stack with probability t_1, where t_1 is its time stamp; or transmits its own packet with probability $1 - t_1$. This way, it transmits someone else's multi-hop packet whose lifetime is coming to an end, with a higher probability than its own packet. Note that in the case of an RSE forwarding packets at intersections, there is no need for this phase in vehicles, since the RSE will handle this. If a vehicle transmits someone else's packet, others hear this transmission and omit the corresponding packet from their own list. This happens because they are in more or less the same geographical area and hear each other's transmissions. The next time the vehicle has a turn to transmit, it chooses the next packet waiting to be forwarded for additional hops and transmits it with the corresponding probability. If a vehicle gains enough opportunities to transmit all the packets in its priority stack, it could retransmit the previously transmitted ones in case their useful lifetime has still not finished. A packet whose useful lifetime is over gets discarded from the priority stack. When a vehicle is forwarding other vehicles' packets it must take into account the original geocast range when determining transmission power.

CHAPTER 3

PREVIOUS RESEARCH

There has been a massive amount of research performed in the area of V2V and V2I communications. In fact, ACM (Association for Computing Machinery) MobiCom, the Annual International Conference on Mobile Computing and Networking, that has been held every year since 1995, now hosts an annual international workshop on VehiculAr Inter-NETworking (VANET). The sole purpose of this workshop is to present and discuss recent advances in the development of vehicular inter-networking. A search of scholarly papers for inter-vehicle communications results in a seemingly endless source of literature.

One of the most prominent groups currently researching the topic is the University of California, Berkeley, which has an entire research department devoted to transportation studies. Participants in their PATH program have done a lot of work specializing in transportation safety and communication. This work will later be used for comparison against our proposed Universal Geocast scheme.

3.1 U.C. Berkeley PATH Program

Established in 1986, the California Partners for Advanced Transit and Highways (PATH) is administered by the Institute of Transportation Studies (ITS) at the University of California, Berkeley, in collaboration with Caltrans [19]. One segment of this research is the Transportation Safety Research Program, specializing in vehicle-highway cooperation and communication, and “science of driving” investigations on driving behavior. One specific group project is entitled the Intersections and Cooperative Systems. This group's research includes crossing path vehicle crashes, safety aspects of cooperative driver-assist systems, Vehicle Infrastructure Integration (VII) with Expedited VII and VII California [19].

A paper written by the PATH team, "Medium Access Control Protocol Design for Vehicle-Vehicle Safety Messages" describes their research in the design of wireless local area networks to enable active vehicle safety systems. The protocol design is based on rapidly re-broadcasting each message multiple times within its lifetime in combination with the 802.11 DCF in a single-hop fashion[11]. They propose six different variations and after simulations they identify the best and most easily implemented of these designs. Their best design is used for comparison versus our new approach. The PATH work also uses NS-2 for simulations along with a mobility model designed at Berkeley. The two performance measurements they use are Probability of Reception Failure (PRF) and Channel Busy Time (CBT). PRF determines if a randomly chosen receiver in the range of a message fails to receive the message during the message lifetime. CBT is defined in terms of several parameters. For a given time period *T* in the control channel, let T_{safety} be the total length of the time periods within T that is occupied by safety messages. Then, $CBT = \frac{T_{Safety}}{T}$.

The types of protocols used by the PATH group are repetition-based protocols used in combination with the 802.11 DCF. The group states that its best design protocol is AFR-CS, which was described in Section 2.2.1, point number 5 as a repetition-based protocol. The PATH state machine for the AFR-CS MAC can be seen in the figure below.

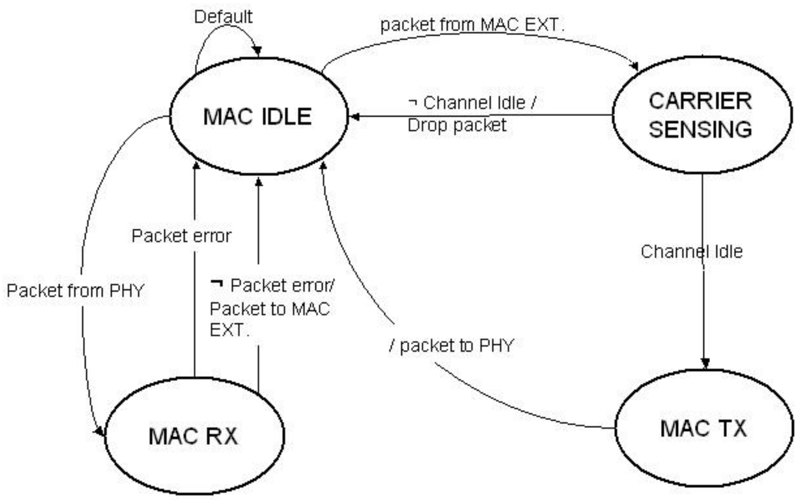

Fig. 3.1 MAC Layer State Machine of the AFR-CS protocol [11]

Setting the number of repetitions *k* configures AFR-CS. The protocol randomly selects *k* distinct slots among the total *n* slots during the lifetime. Whenever a packet is passed down from the MAC Extension, the MAC transitions from the IDLE to the CARRIER SENSING state. In the CARRIER SENSING state, the system checks the channel status using carrier sensing. If the channel is busy, the system drops the packet and transitions back to the MAC IDLE state. If the channel is idle, the system transitions to the MAC TX state and passes the packet down to the physical layer (PHY). It then transitions back to the MAC IDLE state. In MAC IDLE, if PHY sends a packet, the system transitions to the MAC RX state and checks the integrity of the packet. If the packet is corrupted, it is dropped and the system transitions back to the MAC IDLE state. Otherwise, the packet is passed up to the MAC Extension layer, and the system transitions back to the MAC IDLE state.

The PATH paper describes the simulations and mathematical models used to find the ideal fixed number of repetitions *k*. The optimal number of repetitions depends on the message rate, range, vehicular traffic density, and packet transmission time [11]. The paper also discusses the improved performance of repetition-based protocols that take advantage of a CSMA protocol, like the one simulated in this research. The PATH project also describes the optimization of modulation and coding, although this was outside the scope of this research. The summary of their work states that they are able to achieve loss rates between 1/100 and 1/1000 for this protocol with less than 50% CBT. Since this research uses slightly different measures of performance it was decided to simulate the PATH protocol in NS-2 for the most active performance comparison. The simulation of the PATH scheme will be discussed in section 5, which describes the changes made to NS-2.

The main differences between the PATH scheme and the one proposed in this research are the PATH scheme uses a set number of retransmissions randomly distributed over the lifetime of the packet sent via single-hop broadcasts. The scheme in this research has no set number of retransmissions, instead choosing to alter the backoff mechanism, and messages may be sent either by single or multi-hop broadcast, depending on the current status of the vehicle and the message being sent.

3.2 Radio Propagation Models

Recently, there has been a lot of research in the area of radio propagation models due to the new allocation of 75 MHz in the 5.9 GHz band for Dedicated Short Range Communications by the FCC. This work has generated a lot of research into VANETs and associated communication protocols. To effectively simulate wireless communications one must be able to effectively model the attenuation of the radio waves in their transmission environment.

In the November 2009 issue of the IEEE Communications magazine included an article entitled “Field Evaluation of UHF Radio Propagation for an ITS Safety System in an Urban Environment”. For this research an experiment was conducted where a roadside antenna was set up in an urban area in Tokyo with eight-story buildings on both sides of the street. A van equipped with a roof antenna was driven around the area receiving a 792.5 MHz signal transmitted by the roadside antenna. The receiving signal strength and packet reachability (the number of successfully received packets divided by the number of transmitted packets) were collected every 10 ms. Using this data it was possible to create receiving signal strength and packet reachability distribution charts for two different transmitter heights. The 5m transmitter height chart of 80% reachability can be seen below.

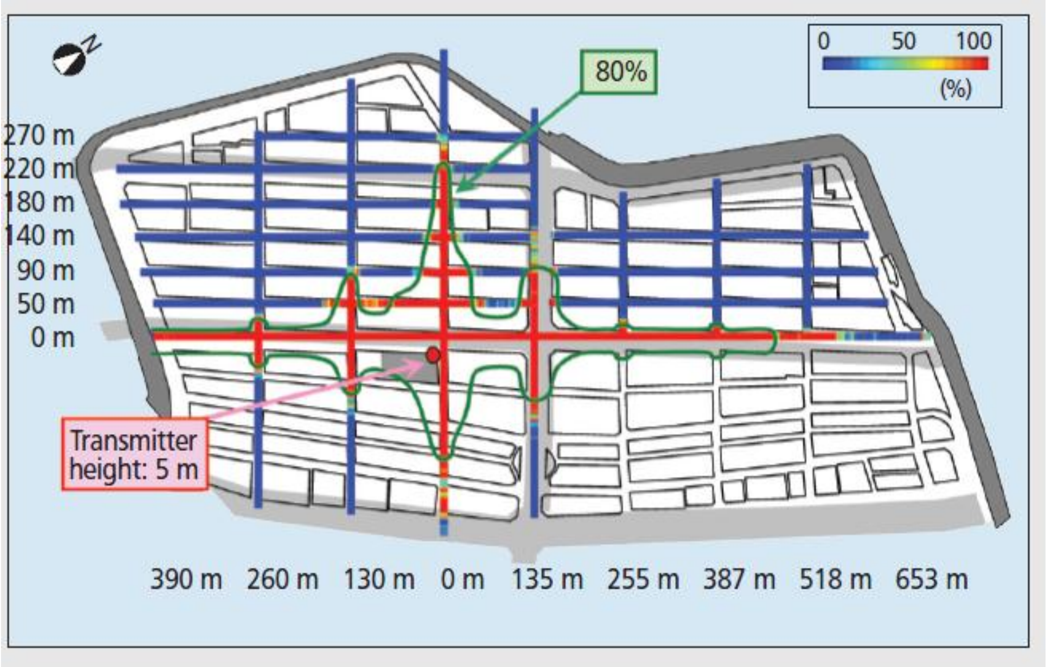

Fig. 3.2 Packet Reachability Distribution Chart for 5m Transmitter Height [21]

An analysis was made of this data, using existing radio propagation models to create a new propagation model that reflects real-world conditions. The baseline for these propagation models was free-space and ground reflection two-ray models, which are two of the most popular radio propagation models [21]. The paper discusses a propagation model from the Kwansei Gakuin University, called the University Kangaku model. This model is dedicated to vehicle-to-vehicle communication and is based on ray tracing simulation results. One important aspect of this new propagation model is that it has line of sight (LOS) and non-line of sight (NLOS) equations for calculating the attenuation loss of a signal. The NLOS equations are based on diffraction of the signal around the building, rather than attenuation of the signals going through the building, like some radio propagation models incorrectly simulate. The work goes on to expand on the Kangaku model to make it even more realistic, getting the models' calculations to agree with experimental results. However, the new propagation model's equations were not completed for all scenarios. The results of the Kangaku model were also very close to those of the actual experiment, which were compared to the new propagation model. Though the experiment was conducted using a 792.5 MHz frequency for transmissions, the Kangaku equations for LOS and NLOS attenuation do take the frequency of the signal into account. The report also states that the applicable scope of the Kangaku model is from 400 MHz to 6 GHz, which includes the 5.9 GHz band proposed by the FCC for vehicular communications in the U.S. [21]. The Kangaku model represented a big improvement on the free-space and two-ray ground reflection models. The equations for all scenarios in the model have been made available. When determining the attenuation of a signal, the Kangaku model takes into account a lot of additional variables that the freespace and two-ray ground reflection models did not. These parameters include the distances of both the transmitting and receiving nodes to the center of the intersection, the width of the roads, and the distance from both nodes to the side of the road. From these parameters the equations of the propagation model can be identified. Because of the additional accuracy of this model in radio propagation simulations it was added into NS-2 in our research, as discussed in section 5.

3.3 Mobility Models

In order to effectively simulate the true dynamic nature of VANETs, the vehicles in the simulation would need to actually be mobile. Research has proven that a critical aspect when testing VANETs protocols is the use of mobility models that reflect the real behavior of vehicular traffic as closely as possible [34]. Though the network simulator used in this research, NS-2, does have the capability of generating mobility, it was much easier to incorporate a mobility model in order to organize and coordinate the movements of the hundreds of vehicles that would be involved in each simulation.

The mobility model that was chosen for this research is the "Simulation of Urban Mobility" (SUMO). SUMO is an open source, highly portable, microscopic road traffic simulation package designed to handle large road networks [41]. In being microscopic, the simulator is meant for tracing the movements of individual cars, rather than just the traffic flow in general. It is mainly developed by employees of the Institute of Transportation Systems at the German Aerospace Center [41]. SUMO uses various car following models that describe the dynamics of each individual vehicle as a function of the positions and velocities of the neighboring vehicles [41]. For the mobility traces in this research the Krauss car following model is used.

In order to use SUMO, the road network and traffic flows need to be set-up in XML, which is used for all SUMO files. To first set-up the road network, nodes must be positioned mapping out the points at which roads will start, end, and be connected to one another. After this, edges are set up, which are basically the roads that connect the individual nodes together. The edges also need to be connected to one another. The number of lanes, maximum speed, connections between lanes/roads, and stop light information are set-up in this file. Then a file needs to be created that describes the flow of traffic along these roads. The flow files establishes the type, including size and maximum speed, and number of vehicles that will be traveling along a particular route, and how long the flow of traffic will continue for. The different routes are also set-up in these files. These files can then be used along with a tool called TraceExporter, in order to create mobility trace files that can be used with the NS-2 program. These tracefiles contain the starting location of every vehicle that will be in the simulation. They also map out the movement of each individual vehicle on a per second basis, and control when nodes begin to send data and to stop sending data, to simulate the vehicles entering and leaving the scenario. The Tcl file used to set up and run the NS-2 simulation just needs to include these files in order to control the vehicle movements and activity. Additionally, once the nodes and edges of a SUMO simulation have been set up, one can go back and create new flow files to be used with them to produce essentially the same mobility models but with different vehicular densities. This proved to be a useful feature and will be discussed further in the results section.

CHAPTER 4

SIMULATIONS USING NS-2

There were a number of network simulators to choose from including NS-2, NetSim [36], OPNET [39], and GloMoSim [40]. The chosen simulator needed to be able to simulate a VANET, include enough complexity to reflect real world VANET transmissions, be readily available, and also allow for customization to simulate the specifics of the network protocols being tested. Since this research also involves discussion of mobility models it is important to note that NetSim is the network simulation tool developed by Tetcos in association with the Indian Institute of Science, Bangalore. It is not the microscopic traffic simulation software package originally developed under the name "Urban Traffic Control System" and combined with FRESIM to create CORSIM [36]. NS-2 is best suited to meet the needs of this research as discussed in the proceeding section, 4.1.

4.1 Network Simulator 2

NS-2 is an open source simulation tool that runs on Linux [22]. It is a discrete event simulator targeted at networking research with a focus on network protocols. It provides support for the simulation of routing, multicast and broadcast protocols along with IP protocols [23]. Simulations can take place over wired and/or wireless environments, including satellite communications [22]. NS-2 can be used for traffic models and application simulation such as FTP, Web, telnet, and CBR. It can also be used for simulating transport layer protocols such as TCP (Reno, Vegas, etc.) and UDP along with multicast protocols like SRM. Various types of routing procedures such as ad hoc routing and direct diffusion may also be simulated, along with various queuing protocols such as RED and drop-tail. The physical media used in NS-2 can be either wired (point-to-point, LANs), wireless (Freespace, Two-Ray ground, and other propagation models included) or satellite [23].

The way that NS-2 works, as a discrete event simulator, is that it models the world as a series of events. The simulator keeps track of a list of events that need to be processed. Once an event is completed, the next event in the scheduler is processed until completion and then the event after that one is handled, until all of the events are completed. Each event happens in an instant of virtual, or simulated, time, while the actual processing of the event or events may take any arbitrary amount of real time. In this way, with a simple single thread of control there are no locking or race conditions to account for [23]. NS-2 uses a split programming model, where two programming languages are used to provide adequate flexibility without inhibiting performance [25]. The low level tasks, such as event processing and packet forwarding through a router require high performance and are infrequently modified once set up. A compiled language such as C++ best implements these operations. However, setting up the dynamic configuration of protocol objects and placement and specification of traffic sources or node placement and movement are often changed. These simulator needs are better met using a flexible and interactive scripting language, such as Tcl [25]. Thus, in NS-2 C++ is used to implement the simulation kernel (the core set of high-performance simulation primitives) but the definition, configuration and control of the simulation is defined via oTcl, an object-oriented variant of Tcl [25]. Tcl files are used to set up all of the parameters for a simulation in NS-2. The number of nodes and the configuration of nodes including the routing protocols, link layer and MAC types, interface queue and physical layer types and other type of variables are all set in the Tcl files. The output from the simulation and the name of the target output file are also specified in the Tcl file.

The user base for NS-2 includes over 1,000 different institutions in over 50 countries, consisting of more than 10,000 users [23]. A review of 151 wireless network research papers from an ACM symposium over a five-year period reported that 76% of the papers used network simulation [24]. This finding helps to demonstrate the wide use of network simulation. Among the different tools used for network simulation, NS-2 is one of the most often used [24]. In the previously cited review, NS-2 was shown to be used the most used simulator. A total of 44% of researchers who used network simulators used NS-2 [24].

NS-2 is frequently updated, facilitating its wide use. In 2007, a research collaboration between Daimler Chrysler Research, Engineering and Design and the University of Karlsruhe’s Institute of Telematics overhauled and updated the simulation of IEEE 802.11 protocols using NS-2 [26]. This project revamped the medium access control (MAC) and physical layer (PHY) models of the simulator for the 802.11 protocols of wireless communications using a clean and modular architecture. Simulator additions included cumulative SINR, preamble and PLCP header handling and capture, and frame body capture features for the PHY. These additions improved accuracy and provided more insight to researchers [26]. These changes allowed NS-2 to model IEEE 802.11 transmission and reception processes realistically and correctly. This work is the starting ground for the research to be completed in this thesis.

4.2 Mobile Networking in NS-2

Up until 1998 NS-2 was unable to accurately simulate the physical aspect of wireless scenarios. NS-2 provided support to model wireless LANs, but this code was not complete because it could not take position and distance factors into account. Every node in the simulation would receive the same transmission at the same time with the same power level. This made it impossible to simulate ad hoc networks.

4.2.1 CMU Monarch Project

At this time the CMU Monarch project was trying to compare the performance of various multi-hop wireless ad hoc network routing protocols [27]. Finding NS-2 unable to perform this type of simulation, the Monarch project created an extension to NS-2 to simulate wireless mobile networks [27]. The freespace and two-ray ground radio propagation models (loss equations for both can be seen in Fig. 3.4) were created to simulate the attenuation of radio waves over a given distance. At short distances the freespace model is used where the power of the signal attenuates at a rate of $1/r^2$, (r is the distance between antennas) and at longer distances $1/r^4$ is used as the ground reflection model. All of the wireless nodes in a given scenario are linked together with the same physical channel object. With these, the position of a mobile node could be calculated as a function of time, and then used by the radio propagation model to calculate the propagation delay from one node to another and to determine the power level of a received signal at each mobile node [27]. This information is used to determine which nodes in a simulation set-up will receive the transmission and at what power level. If the power level is above a preset configuration (receive threshold), the packet will then be passed to the MAC layer, where the packet may begin the receive process. The MAC would then insure that the receive state of the node was idle. After a scheduled amount of time, determined by the size of the packet and the transmission rate in the MAC, the packet can be counted as received. Should other packets arrive during the receive time when the MAC receiving mechanism is not idle, received packet drops, incoming packet drops, or both may occur, depending on the calculated power level of the incoming packets and the currently received packet. If the scheduled receive timer expires without any calculated disruptions, the MAC will check the packet for errors, perform destination address filtering, and pass the packet up the protocol stack.

The link layer, which includes the previously mentioned MAC layer, was designed to implement the IEEE 802.11 protocols. Thus, the MAC performed the Distributed Coordination Function (DCF) technique by performing both physical carrier sensing and virtual carrier sensing. RTS/CTS and ACK four way handshakes would be implemented for unicast packets, but only carrier sensing for broadcast packets. Packet buffering, in a drop-tail fashion, was also implemented in the Link Layer for packets awaiting transmission by the network interface.

4.2.2 Overhaul of IEEE 802.11 in NS-2

Subsequent publications ([28], [29], [30]) appearing starting around the year 2001, showed that accuracy was lacking in the way NS-2 modeled packet interferences and packet reception. The basic assumptions in the wireless simulations did not cover the effects occurring in a real world set-up. These publications showed that the results obtained from simulations would change dramatically when more appropriate models were used. Several research projects attempted to improve the simulations while still using the original wireless structure ([31], [32]), but dramatic improvements would only be seen by completely revising the wireless simulations. This change was implemented by the Daimler Chrysler Research, Engineering and Design and the University of Karlsruhe's Institute of Telematics collaboration. Instead of patching up the existing NS-2 implementation, this project focused on a complete redesign of IEEE 802.11 modeling. The main problem with the previous implementation of 802.11 was that most of the PHY functionalities were mixed up in the MAC. As a result, it was very difficult, if not impossible, to model everything correctly at both the physical and logical levels. The overly complex MAC module was also a big challenge for the users to understand and extend in their research. Beyond this, there were also many instances of over-simplification or inaccuracies in the IEEE 802.11 modeling [26]. Instead of putting everything inside the MAC, all functionalities of the IEEE 802.11 radio are now cleanly and properly separated between the MAC and PHY. A diagram of this can be seen below.

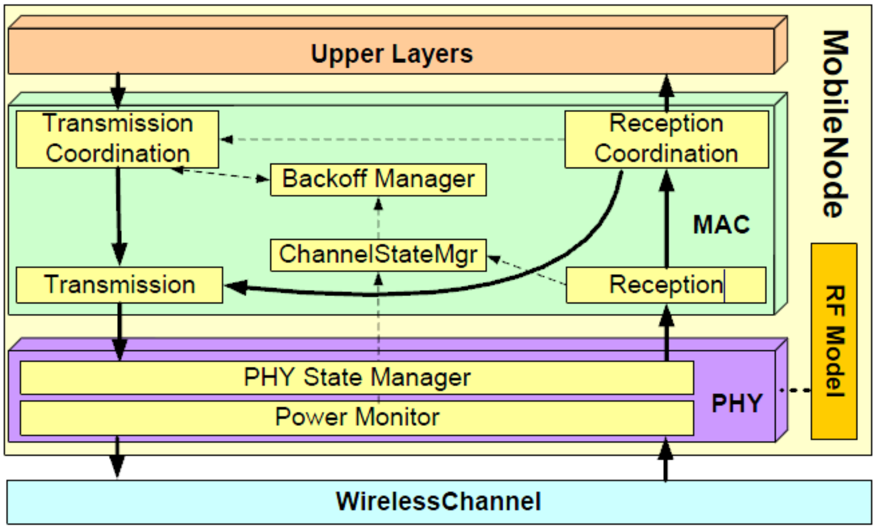

Fig. 4.1 Revised IEEE 802.11 Simulation Architecture [26]

The MAC module now only operates at the logical level. It depends on the PHY to handle actual transmissions, receptions and physical channel sensing [26]. The focus of the MAC design is to correctly and cleanly model all the complexities in the IEEE 802.11 CSMA/CA mechanism, as described in section 2.2. The PHY module handles all physical layer related issues, such as channel sensing, signal-to-interference-plus-noise ratio (SINR) tracking and Physical Layer Convergence Procedure (PLCP) state management. While the work on the PHY is part of the overall IEEE 802.11 modeling, its design is sufficiently generic so that it is able to support the implementation of different MAC designs on top [26]. This was an important aspect of the project, as this thesis would otherwise need to make changes to the structure and function of the MAC. The new MAC now has six separate modules, which can be seen in the figure above. They are the transmission, reception, transmission coordination, backoff manager, channel state manager, and reception coordination modules. The transmission and reception modules now have direct interfaces with the PHY layer, meaning there are functional calls from the MAC to PHY and vice versa. The transmission module passes along transmissions from the transmission coordination module and the reception module applies address filtering to successfully received frames and passes them on to the reception coordination module. The channel state manager keeps track of both the physical and virtual carrier sensing. It depends on input from the PHY to keep track of physical carrier sensing, and input from the reception module to coordinate virtual carrier sensing. The backoff manager module works closely with the channel state manager. It maintains the backoff counter to support collision avoidance, but needs input from the channel state manager to know when the channel is idle or busy. The backoff manager also assists the transmission coordination module to run both the regular backoff and post-transmission backoff, but is not aware of the difference between the two. The reception coordination module takes control and data frames meant for this node from the reception module. It signals the transmission coordination module when CTS and ACK frames arrive. It is responsible for handling the CTS and ACK responses when RTS and data frames arrive. It also filters the data frames before passing them to the upper layers. As it passes data frames to the upper layers, duplicate data frames are discarded and the ACK process is initiated where applicable. Finally, a transmission coordination module is applied. This module manages channel access for packets passed down from the upper layers. A picture of the state machine can be seen in the figure below.

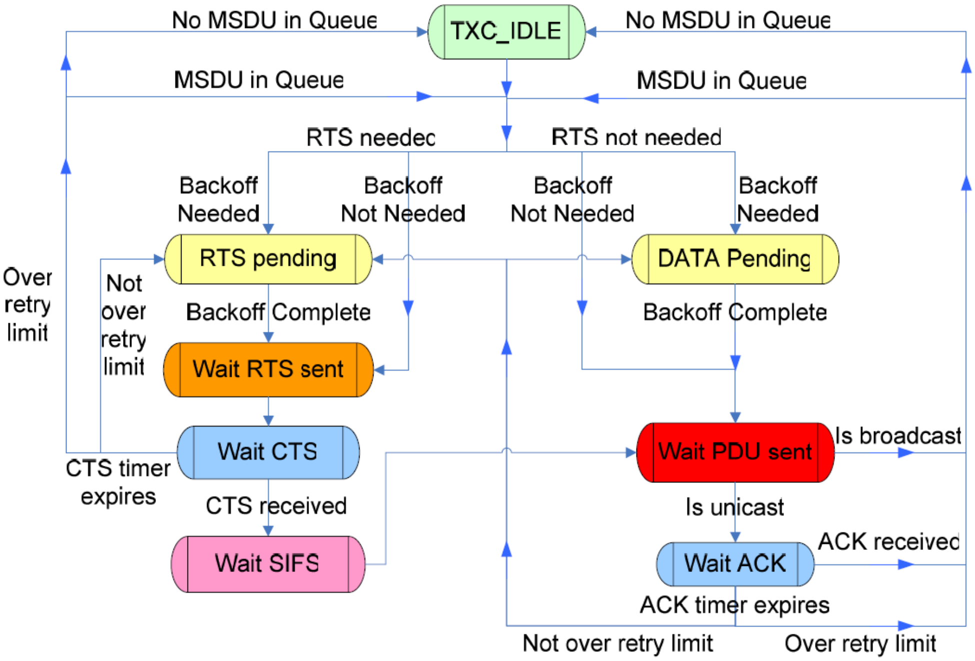

Fig. 4.2 Transmission Coordination Module State Machine [26]

When the transmission coordination module moves out of the TXC_IDLE state because of a packet coming down from the upper layer, it first checks if a RTS frame should be generated. Afterwards, it starts a backoff process at the backoff manager if one isn't already running and moves into the RTS Pending or Data Pending state according to the RTS decision. If the transmission coordination module is in the RTS Pending or Data Pending state, it instructs the transmission module to transmit the RTS or data frame respectively as soon as receiving a signal indicating Backoff Done from the backoff manager. In the case of a broadcast message, like the ones transmitted in VANETs, there are no RTS/CTS or ACK messages. It is also important to not overlook the post-TX backoff, not shown in the state machine. Once a transmission has been successfully sent, a random number is selected from the backoff window, which is decremented when the channel is idle via increments of slot time, which is the maximum theoretical time for a frame to travel a network. The new transmission coordination module is especially important because this is where a majority of the changes to increment the new Universal Geocast Scheme take place. This work allowed NS-2 to model 802.11 transmissions, receptions, and packet drops in a more realistic manner. Specific mechanisms of the overhauled 802.11 standard will be shown in detail in section 5.2 when discussing changes made to the protocol to implement the new communications scheme.

4.3 NS-2 Trace Output and Acquisition of Data

There are a number of ways to collect data from NS-2 simulations. Generally, trace data is either displayed directly during execution of the simulation, or (more commonly) stored in a file to be post-processed and analyzed [33]. There are two primary but distinct types of monitoring capabilities currently supported by the simulator. The first, called traces, record each individual packet as it arrives, departs, or is dropped at a link or queue. Trace objects are configured into a simulation as nodes in the network topology, usually with a Tcl “Channel” object hooked to them, representing the destination of collected data (typically a trace file in the current directory). The other types of objects, called monitors, record counts of various interesting quantities such as packet and byte arrivals, departures, etc. Monitors can monitor counts associated with all packets, or on a per-flow basis using a flow monitor. In this research, the traces of individual packets were used. The packets have a unique ID (established in their packet headers via C++) that are used to keep track of them. The packets can be monitored at the Agent (upper layers), router, MAC, and PHY levels. In the simulations in this research the dropped packets will be at the PHY level, primarily because of collisions in the channel and nodes being out of transmission range from one another. In the new wireless trace format, which is used in this research, there is a lot of information in each line of the trace, which correspond to events happening to the packet as various layers. The first field in the trace gives the event, which could be one of four things: Send, Receive, Drop, or Forward. The trace goes on to list the timestamp of the event, the unique ID of the packet, the X, Y, and Z location information of the event, the network trace level (AGT, RTR, MAC, or PHY), the reason for the drop if the packet was dropped, MAC level information such as source and destination Ethernet address', the type of packet, next and previous hop information, etc. The C++ files that handle the trace output can be changed so that any information deemed necessary can be output into the trace files.

Given the amount of nodes in a simulation, the frequency of transmissions, and the length of the simulation itself, these trace files can be very cumbersome. That is why it is necessary to develop scripts that can sort through them, extracting pertinent information. For instance, using Perl or Awk, a script can be written that searches through the trace files and adds up all of the dropped packets or successfully received packets. The scripts can use this along with other information to determine things like throughput, receive ratio, packet delay, etc.

CHAPTER 5

NS-2 MODIFICATIONS

In order to accurately simulate the proposed communications protocol, the C++ files that control the Agent, MAC, and PHY layer functions had to be changed. The Agent level represents the upper layers of the network. This level is where packets are first generated and ultimately received for a positive reception of the message. In Fig. 4.1, the MAC layer is seen as connected to a block titled “Upper Layers”. These upper layers consist of the Agent, which handles the routing protocol, the Link Layer (LL) which includes the Interface Queue (IfQ), and the Address Resolution Protocol (APR). A diagram of these upper layers can be seen below.

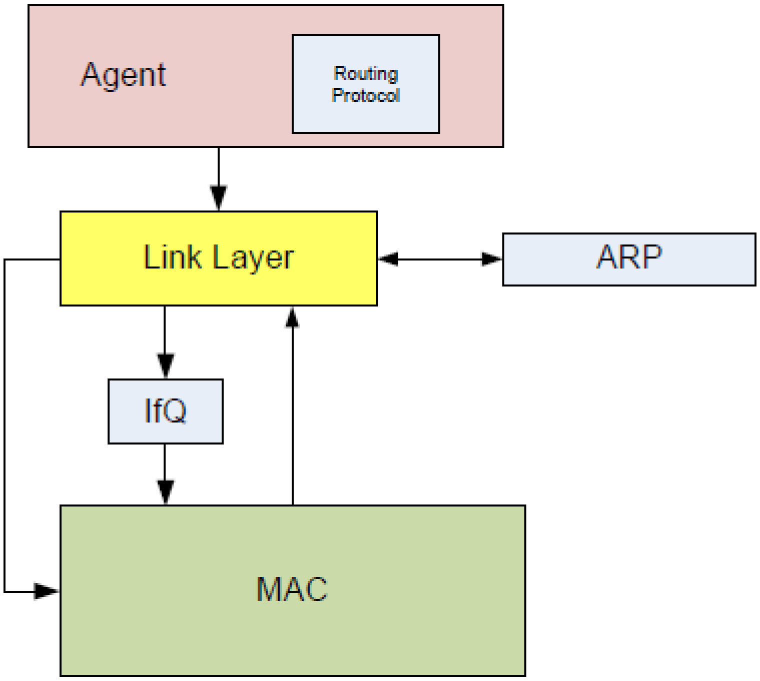

Fig. 5.1 Upper Layer Schematic

Because the transmissions of this protocol are broadcast in nature, the ARP is not used and neither is the routing protocol. The Link Layer can handle protocols such as packet fragmentation and reassembly, and reliable link protocol all while performing the task of setting the MAC destination address in the MAC header of the packet. These protocols are not used in the proposed scheme, and the address is simply set to that of a broadcast packet. Since the Agent layer first creates the actual packet to be sent, it initializes a lot of the packet's variables and information, so changes would need to be made to the Agent layer. Obviously changes were required in the MAC layer, as this is where the proposed protocol would be carried out. Changes were also necessary in the PHY layer, so it was better able to interface with the changes made to the MAC layer. Also, a new propagation model was introduced into the simulator, which would require a completely new source and header file. Other small changes were made in various other areas of the simulator in order to ensure proper function of the proposed protocol.

5.1 Changes to the Agent

Agents represent endpoints where network-layer packets are constructed or consumed, and are used in the implementation of protocols at various layers [33]. Every node that is generated in a simulation needs to have an agent attached to it. For example, to send Transmission Control Protocol (TCP) traffic, two nodes would need to be created. One node could have a TCP agent attached to it, and the other would have a TCPSink attached to it, making one the transmitter and one the receiver. Or a TCP/FullTcp agent could be attached to both of them so 2-way TCP traffic could be sent. At a minimum, 2-way end system agents, like the ones used in this research, must be able to allocate space for new packets, and also have functions to send and receive packets. The agents are capable of modeling higher layer protocols, such as the TCP just mentioned or UDP. In the proposed protocol, broadcast packets are both sent and received by all nodes in the simulation. As part of the previously mentioned IEEE 802.11 overhaul (section 4.2.2), a new broadcast agent was created, called PBCAgent (Periodic BroadCast Agent). Similar to the Ping Agent in NS-2, this agent was created to allow users to test the new IEEE 802.11 implementation [26]. In the Tcl file, users can specify the size of the packets, the broadcast interval, the broadcast variance, and the modulation scheme. The PBCAgent, specified in the file pbc.cc/.h will allocate space to create a new packet to be transmitted at the specified broadcast interval. It also fills in some of the headers of the packet, which are created in the packet allocation process. Information filled in the packet headers include the unique ID of the packet, its size, source IP address, and the timestamp of when the packet was created.

The new protocol assumes that all vehicles, simulated as nodes in NS-2, have GPS devices used to give them a position fix, and that their position information would be transmitted along with the packet. This feature was implemented in the Agent. In all mobile node scenarios, a GOD (General Operations Director) object is created that contains the global state information. It stores information that an omniscient observer would have, such as the total number of nodes and connectivity information. The current use of the GOD object is to store an array of the shortest number of hops required to reach from one node to another, which is done in pre-simulation since on the fly calculations would be time consuming. No node should have access to all of the information contained in the GOD object, but partial information may be obtained when needed [33]. In order to simulate the GPS device, functions were created in the GOD object to allow the PBCAgent to be able to access and record its own X, Y, and Z coordinates. The Agent now places this information in the PBC header of the packet, where the MAC can access it. Additional space also had to be allocated in the PBC header to allow for this new information.

5.2 Changes to the MAC

The majority of the changes that were made to NS-2 were made here, in the MAC files. The files, Mac802_11Ext.cc/.h were created as part of the overhaul of IEEE 802.11. They handle all of the MAC functions as previously mentioned.

5.2.1 Packet Queues

One of the first changes to the MAC was the addition of two new queues. One queue is for the node's own packets (My Queue or MQ) and another queue is for forwarding other node's packets (Their Queue or TQ). If a node is in a position to forward other node's packets and it receives a multi-hop packet it will place it into TQ. A spot in the queue would consist of the original packet, the time of expiration for the packet representing the end of its useful lifetime, and a pointer to the next queue spot. The MAC keeps track of the heads of each of the queues through pointers at all times. Functions were created to enqueue and dequeue packets from these queues. One of the experiments performed was to have all forwarding of multi-hop packets be performed by the RSE. In order to do this a variable was created in the MAC to represent the address of the RSE. When attempting to enqueue a packet into the TQ for forwarding, a node would first have to check its address against that of the designated RSE. If the packet does not have the same address as the RSE, it cannot enqueue the packet and thus cannot forward the packet. The respective dequeue function would simply return the packet at the head of the queue and erase it from the queue. The dequeue function would check to make sure that the packet had not yet expired. The enqueue function would also first check the packet's expiration. Next, the enqueue function would sort the packet into the queue based on its timestamp and retransmission number.

5.2.2 Enqueuing, Retransmission Number, and Backoff

A function called 'handlemsgfromup' would be called whenever a packet was received from the upper layers for transmission. This function was altered such that it would first enqueue the packet to be transmitted, before it went through the rest of the transmission process. No matter which queue the packet came from (MQ or TQ) this function will then enqueue it back to its proper queue. This function would also determine the size of the interval from which the backoff window would be chosen. If the vehicle is going to forward a packet from another node, it will use a fixed backoff window, referred to as the intersection backoff window. The reason for this is that a vehicle forwarding packets will need to make more transmissions than vehicles not forwarding other packets. By not increasing the size of the backoff window for each transmission, the vehicle will, on average, have to wait less time between transmissions. Vehicles that are sending out their own transmissions would use the interval discussed previously of $\{0,\ldots 2\text{^}(i*X+Y)*CW\}$. The number, $2\text{^}(i*X+Y)*CW$ would be passed to the backoff function which would then choose a random number and start the process of counting down until the proper time to transmit. In order to do this, the function would need to know how many times a packet had been retransmitted. A place was added in the common header of each packet to keep track of its retransmission number. Before sending a packet to the transmit function, which would pass the packet onto the PHY layer, it would increment the retransmission number. The receiving function of the MAC would set the retransmission to 0, because each node only cares about the times that they have retransmitted it, not retransmissions of the same packet by other nodes. Also, the values of *X* and *Y* would have to set by the user. Global variables were set up so that these could be input via the Tcl file for different simulations. This variables would be bound to the variables in the C++ file used in determining the size of the backoff interval.

5.2.3 Power Calculation

Before a packet can be transmitted from the MAC, the power for the transmission must first be calculated. The 'calc_power' function performs this task by using the LOS and NLOS equations taken from the Kangaku propagation model. This function takes into account whether the packet is being sent multi or single-hop, the surrounding area of the node, the intended broadcast range of each packet, and the original source location of the packet if it was created by another node. Section 2.4.2 describes the difference in power for either a single hop or multi-hop transmission. For a single hop transmission the transmission power is increased such that the packet will be received by all nodes within an 80 meter range, regardless of whether or not they are in line of sight of the transmitting vehicle. For a multi-hop transmission the power level is set to have a LOS broadcast range of 80 meters. Note that if there are no objects obstructing the LOS, the single-hop and multi-hop transmission powers are the same. The transmission power of the packet is then stored in the packet's common header where it can be accessed by the PHY layer, where the actual physical simulation of the transmission takes place.

For comparison, experiments would need to be run using either the Kangaku radio propagation model or the Free Space model. To implement this, the MAC power calculation functions were updated to be able to calculate the power of transmission necessary for both radio propagation models. The power calculation function was also enabled to be able to tell which radio propagation model is currently being used in the simulation being run through inputs in the Tcl file.

5.2.4 Packet Reception

When a packet being sent via multi-hop is received by another node, the node must first check to see if it is in a position to forward the packet. A function named inter_check was set up so that a node can see if it is in an intersection. Being in an intersection is the prerequisite to perform the function of retransmitting other nodes' packets. The enqueue function of the queue containing other nodes' multi-hop packets checks if the packet is multi-hop, and that the node receiving it is in an intersection. In the case of RSE only forwarding, the enque will also check to make sure the address of the RSE and the its current address are the same. The enqueue function also checks to make sure the packet has not expired. When a packet is being sent via multiple hops, once one node has effectively forwarded this packet, other nodes in the area that hear the transmission do not retransmit it. When the MAC receives a packet from another node, it must check to see if this packet is inside its queue of other nodes' packets. A search function was set up that uses the packets' unique ID, and the original source node, which are both found in the packets headers, to efficiently see if it has already received the packet and if it remains inside of its forwarding queue. If the packet is in the queue, it is removed and the pointers are rearranged accordingly. A packet may have been dequeued from TQ and forced to wait for the backoff to finish before being transmitted. The search function must also check to determine if there is a backoff running. If so it must compare the packet waiting to be transmitted with the one just received. If they are the same packet, the backoff mechanism must be reset, the packet deleted, and the MAC set to be able to transmit any other packets waiting in its respective queues.

5.2.5 Check_Queue Function

In the normal operating mode of the MAC, the Interface Queue (IfQ), which is part of the LL in the layer above the MAC, passes packets down to the MAC for transmission. Upon passing a packet down, the IfQ will block itself. This information means that even if the Interface Queue has other packets from the Agent to send, it will just store them until the MAC is ready and requests another packet. The packet is passed down to the MAC with a handler for the IfQ from which it came. Once the MAC has gone through the transmission process it calls its Check Queue function which uses the handler passed down to it from the Interface Queue to unblock it. It was decided from simulation results that it is best for the MAC to give new packets just arriving from its node priority over all other packets. As a result, the Check_Queue function is used to determine from which MAC queue to transmit. If the IfQ does not have a new packet for the MAC, which only happens every 200ms to simulate a new GPS location, the MAC will then send a packet from either the MQ or TQ. This function looks at the head of the TQ, and extracts the time left until that packet expires. It then turns that number into a percentage of the packet's lifetime that is left. A random number is then picked from 0 to 1, and if that number is greater than the percentage of lifetime left for the head packet of TQ, that packet gets sent. Otherwise, the head packet from MQ gets sent. The smaller the percentage of lifetime left, the higher the probability of sending the packet. Note that if there are no packets in the TQ, only packets from MQ will be sent. Also note that there should only be one packet in MQ at any given time. The node only creates a new packet after the 200ms since the previous packet has passed. When sending from MQ, the expiration of the packet is checked before dequeuing so the old packet will be dropped and the new packet will actually be dequeued.

5.2.6 Transmit Function

The final function called by the MAC before a packet is transferred to the physical layer for actual transmission is the Transmit function. The main purpose of the transmit function is to simply pass the packet to be transmitted down to the physical layer once the proper backoff has been completed. This is why UGS implements the single or multi-hop decision here. When dealing with such large backoff windows generated by UGS from the $\{0,\ldots 2\text{^}(i{*}X+Y){*}CW\}$ interval, it is important to take the amount of time a packet must wait for the backoff to finish into consideration. This is because the amount of useful lifetime a packet has remaining is one of the main deciding factors.

When deciding whether or not the packet should be sent single or multi-hop, the first thing taken into consideration is environment surrounding the vehicle. If a vehicle is not within broadcast range of an intersection, then the packet will be transmitted via a single hop. If a vehicle is within broadcast range of an intersection, then the amount of useful lifetime the packet has left will be used probabilistically to make the decision. Similar to the way the Check_Queue function examines the remaining lifetime of a packet, so does the Transmit function. The function checks the packet and extracts the time left until that packet expires and turns this into a percentage based on the amount of useful lifetime that has already expired. This can be represented by the equation $PercentExp\,ired = \frac{t_{now} - t_{made}}{lifetime}$ where t_{now} is the current time, t_{made} is the time the packet was made and *lifetime* refers to the useful lifetime of a packet, in this case 200ms. Note that in the transmit equation a check is performed to ensure that the useful lifetime of the packet has not expired. A random number is then chosen from 0 to 1 and if the amount of useful lifetime of the packet that has expired is greater than the random number chosen, the packet will be sent single-hop instead of multi-hop. For example, if a packet is 25 ms old, then 1/8 of its useful lifetime has expired. If the random number picket from the range 0-1 is greater than 1/8 then the packet will be sent multi-hop.

Given the exponential growth of the backoff window as the retransmission number increases, the majority of transmissions happen within the first half of a packets useful lifetime. The large backoff windows that successfully control congestion by constricting the number of times that each packet is retransmitted will cause a significant amount of the packet's lifetime to be spent waiting for the backoff to finish decrementing. Due to the large backoff window there are often times when a packet is transmitted with more than half of its useful lifetime remaining, only to be never transmitted again. If a multi-hop packet is unsuccessfully retransmitted by an intersection node it will never reach nodes that are NLOS, dramatically reducing the reception ratio of the packet. Experimentation proved that increasing the likely hood of at least one single hop transmission per packet would provide an increase in reception ratio. The probability scheme was adjusted such that there is a higher probability of a packet being transmitted via multiple hops on its first transmission, with increasing likelihood, as its useful lifetime expired, to be transmitted a single hop in later transmissions.

As discussed in section 2.4.2, if a packet is being sent mutli-hop, it will be sent with the transmission power P_i^2, which is the power level for a LOS broadcast range of 80 meters. Per the Kangaku radio propagation model [21] if the frequency, road width, and height of the transmitting and receiving antennas do not change this power level will be constant, as it is in this research. When a packet is sent via a single-hop, the power must be increased to diffract around buildings and allow the packet to reach any receiver within the geocast range, whether there is line of sight to it or not. In this case the distance of the transmitting vehicle to the intersection and the farthest distance down roads in the geocast range to which the transmitter does not have line of sight must be known. This distance can be calculated by knowing the distance to the intersection from the transmitting vehicle and simple geometrical equations. Figure 6.14 in the results section gives an example of the power level necessary to reach vehicles that are NLOS.

5.3 PHY layer and Tcl files

The physical layer is implemented by the files wireless-phyExt.cc/.h. The only change that needed to be made in these files was to tell the transmission function to check the common header of the packet to find the power level of the packet to be transmitted. The transmission function then informs all of the other nodes on the same channel, which is all of the other nodes in the simulation, that a packet has been transmitted from the location of that node and at the power level for that packet. The other nodes in the simulation then reference the radio propagation model with this information to determine, given there own location, at what signal strength they will receive this transmission. Variables entered in the Tcl file such as modulation scheme, noise floor, etc. are used in this process, as is the current state of the node itself (receiving another packet, transmitting a packet, or idle). The power level of the PHY packet transmission is also normally defined in the Tcl file.

The Tcl file, as explained earlier, defines all of the parameters of the simulation. The type of Agent to be used, the MAC, the PHY, the routing protocols, and others are all defined in the Tcl file. In order to incorporate intersections and the use of LOS and NLOS equations, some new variables had to be introduced to the Tcl file. The most important of these was the intersection location information. The X and Y locations of the intersection are necessary in order for the new Kangaku propagation model to determine the power level of received packets. It is also crucial for the MAC to know these values so it can figure out the power level for transmitted packets in order for all intended receivers to get them. The width of the roads also has to be entered in the Tcl file. One of the parameters of the Kangaku propagation model is the distance from a vehicle to the side of the road. This value can be calculated from the intersection and the road width. Note that the simulations up to this point have only involved one intersection. Some other variables implemented in the MAC were linked to variables in the TCL file so the parameters of the simulation could be changed faster. This included the integer values for the vehicular density variables *X* and *Y*. The set intersection backoff window and the address of the designated RSE could also be entered from the TCL file.

5.4 Kangaku Propagation Model

A new set of files, kangaku.cc/.h were created in order to simulate the new Kangaku propagation model. These files were based off of the existing propagation files where information about the packet being transmitted and the receiving node are passed to the applicable functions in the file and a calculated reception power is returned. The Kangaku propagation model also requires that the intersection X and Y location along with the road widths be passed to it in the Tcl file. From this information the Kangaku model is able to then calculate all of the necessary parameters mentioned in section 3 which it then uses inside the LOS or NLOS equations.

5.5 Implementation of PATH's Proposed Scheme in NS-2

The simulation of the PATH communication scheme in NS-2 was considerably less complex than the simulation of the scheme proposed in this thesis document. There was no hop scheme, and no set of queues to maintain. To implement the PATH scheme, changes were made to the Mac802_11Ext.cc/.h files that were created as part of the overhaul of IEEE 802.11 described in section 4.2.2. The MAC function that receives a packet for transmission from the upper levels divides the useful lifetime of the packet into slots. In the same way a user can specify *X* and *Y*, the vehicular environment variables for the universal geocast scheme described in section 5.2, the user can specify the retransmission number, *T*, of this protocol in the Tcl file. The MAC function takes this number and randomly selects *T* slots during the packet's useful lifetime to transmit the packet. Since the actual time of the transmission is then known, the repetitions can be scheduled in this function. The same single-hop power calculation function that is used in UGS is also used in the PATH MAC. This is especially important in intersection simulations where the power must be calculated for the transmission to reach every node in an 80 meter range of the transmitting vehicle. This power calculation will change in NLOS communications based on both the distance of the transmitter to the intersection and the farthest distance down roads in the geocast range to which the transmitter does not have line of sight. When the scheduled time to transmit arrives, the channel is sensed. If the channel is idle, the packet is sent. If the channel is busy, the MAC does nothing except wait until the next scheduled transmission time. At the end of the packet's useful lifetime it is dropped. The original implementation of this scheme in NS-2 was performed by Mohammad Nekoui. This research updated the implementation by allowing it to use the new Kangaku radio propagation model by including the power calculation function for single-hop transmission in intersection simulations.

CHAPTER 6

RESULTS

This chapter discusses the methods for the collection of results, the results of the proposed protocol, those of the PATH protocol, and a comparison between the two. For both the highway and intersection scenarios each vehicle produces an original message every 200 ms, which is subsequently retransmitted based on the backoff strategies described in sections 2.4 and 3.1. The broadcast range of the messages is 80 meters at a frequency of 5.18 GHz and the packets are 200 bytes in size with a useful lifetime of 200 ms. For single hop transmissions, the power is calculated for each transmission such that every vehicle within an 80-meter radius of the transmitting vehicle will receive the broadcast. Only vehicles within the 80-meter broadcast range of the sender count towards calculated results. For each individual simulation, the reception ratio, average number of transmissions per packet, and average delay per packet and the maximum packet delay were recorded.

6.1 Data Collection Methods

The output of NS-2 is a very long trace file that documents each packet as it progresses through the different layers from the transmitting node to the various receiving nodes. Depending on the number of vehicles in the simulation and the amount of time that the simulation is run for, these trace files can be millions of lines long and take up hundreds of megabytes. To save time, rather than write this trace output directly to a file, it is run through an awk script, and only the results of the awk script are actually written to a file. One of the results taken from this trace output is the reception ratio of each packet, and the average for the entire simulation. An example of the reception ratio of a vehicle can be seen in the figure below.

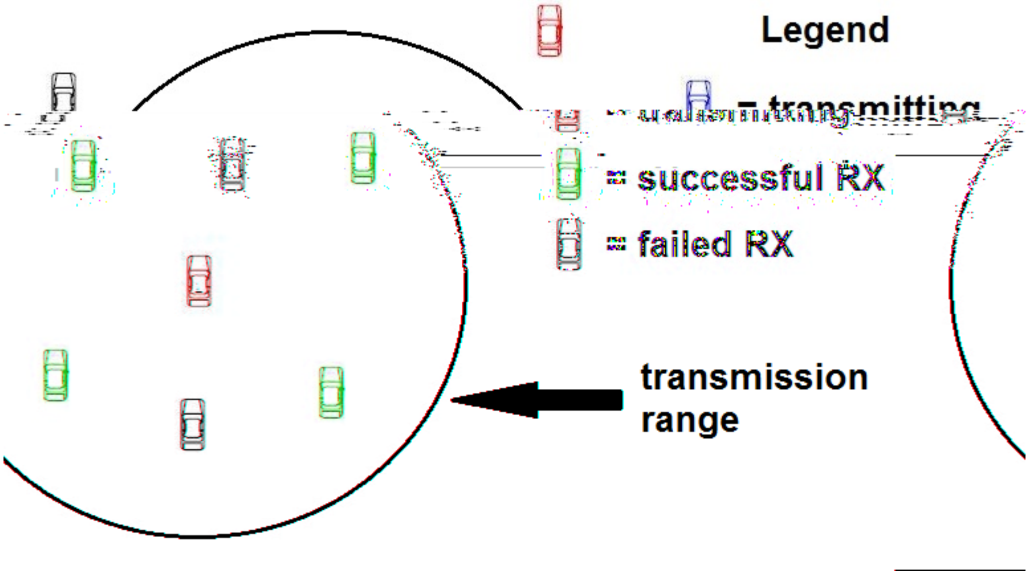

Fig. 6.1 Reception Ratio Example

In this picture the circle represents the transmission range of the blue vehicle in the center, which is transmitting. The red vehicles successfully receive the transmission, while the vehicles without color do not. The reception ratio of this simple example would be 4/6 or approximately 0.667. Even though the red vehicle outside of the transmission range does successfully receive the transmission, it is not counted in the reception ratio. To extract the reception ratio from the trace output, an awk script was created to run with the Tcl file and the results of different simulations were recorded in a spreadsheet.

The awk script works as follows: It searches through each line of the trace file to find the original transmissions of packets. The first time a packet is sent, its location,unique ID, and time the packet was created are saved into an array. The trace output keeps track of every single other nodes response to the packet that is transmitted. For the majority of other vehicles the packet is dropped because it is simply out of the broadcast range of the packet. Other drops occur due to collisions, or from one vehicle not being able to receive the packet because it is trying to transmit a packet of its own. Of course some amount of vehicles actually receive the message. The awk script sums together the number of vehicles that are within the intended broadcast range (80 meters in this research) of the transmitting vehicle as these are the number of vehicles intended to receive it. The awk script then traverses the rest of the ouput and checks which nodes properly receive these packets at the Agent level and records a count of successful receives. The awk script uses the original source location and the locations of the receiving nodes to ensure that no receptions outside of the intended broadcast range are included in the receive ratio. The awk script then divides the number of vehicles that successfully receive the packet by the previously calculated number of vehicles that should have received the packet to calculate the reception ratio for the packet.

The awk script also keeps track of the time that a node actually receives a packet and compares it to the previously recorded time the packet is actually created. The amount of time it takes for each vehicle that receives a packet to receive it is summed together and then divided by the number of vehicles that receive the packet to calculate the average delay. The awk script also keeps track of the largest amount of time it takes for a packet to be successfully received and records this as the maximum packet delay. It is important to note that packets that are not successfully received are ignored when calculating the average and maximum packet delays.

Similarly, the awk script also keeps track of the number of times each original packet is retransmitted. This is used to calculate the Average Packet Transmissions (APT), which is the average amount of times that each packet is transmitted.   .

At the end of the simulation the awk script will write to a file, for each individual packet, the reception ratio, average delay, and the number of times that particular packet was transmitted. This was done to check for any anomalies during the simulation to find opportunities for improvement. At the end of the file the awk script will write the average recepction ratio, average and maximum delay, and packet transmissions (APT) for all transmissions made during the simulation. This information was then recorded into spreadsheets. Parameters may be changed, or the simulation might stay the same to check for consistent measurements using the same parameters and the same awk script parameters.

Alternative awk scripts were also used to determine the number of collisions each packet was involved in, the packets original source and distance from the intersection vs. the reception ratio, the number of other packets each vehicle may or may not forward, etc. This information allowed for optimization of the protocol and preliminary investigation into determining the values of *X* and *Y* in the backoff scheme.

After doing numerous simulations and experimenting with different configurations it became readily apparent that the number of times a packet is retransmitted has a large effect on the results of the simulation, especially in congested simulation scenarios. This observation relates to the previously mentioned *X* and *Y* variables that contributes substantially in determining the size of the interval from which the backoff window is selected. The more congested a simulation, the better results from having a smaller number of retransmissions per packet. To decrease the number of transmissions, the backoff time needs to be increased and increasing the variables *X* and *Y* accomplishe this goal. The effect was first noticed while doing simulations of the PATH scheme, to be discussed shortly, where the maximum retransmission number is selected before the simulation begins. Different maximum retransmission numbers needed to be determined for different vehicle concentrations to optimize the PATH scheme. The proposed scheme does not have a maximum retransmission number set prior to starting the simulations. Thus, it became necessary to have the awk script also keep track of the number of times that each packet was transmitted, as explained previously in this section.

Another variable that had an impact on the results was the set intersection backoff window, for vehicles forwarding other vehicles packets. As previously discussed in section 5.2.2 this allows nodes engaged in forwarding other nodes packets more opportunities to transmit. However, since this backoff window does not increase with the number of times a packet has been retransmitted, it needs to be reasonably large. Otherwise, packets that are forwarded from intersection nodes will have a disportionate number of extra transmissions, greatly increasing the congestions in the simulation.

6.2 Highway Simulations

These results are presented for a simulation representing a four-lane highway. In this scenario there are no RSE units, and all vehicles are considered to be within LOS of one another. The four lanes are parallel to one another and vehicles all travel in the same direction at varying speeds, designated by the Krauss car-following model used in SUMO, with a maximum speed of 30 m/s (67 mph). A figure representing the highway simulation can be seen below.

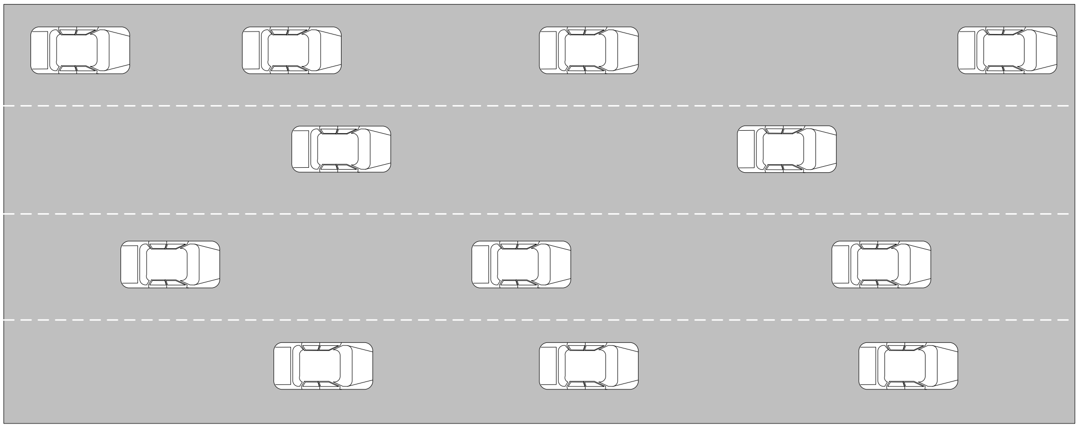

Figure 6.2 Four Lane Highway Scenario

Like the highway simulation presented in the PATH research [11] it is straight without any entrances or exits and is 1800 meters in length. The highway simulated is actually 2000 meters in length, but like the intersection simulation, the edge effects are accounted for by ignoring the first and last 100 meters of the highway. The edge effect is caused by the highway being limited in length. For nodes at the beginning of the highway they contend with less interference because there are no nodes behind them. Nodes at the end of the highway have no nodes in front of them, so they too contend with less interference. To account for this, nodes are simulated in the first and last 100 meters of the highway, but results are only collected from the nodes in the middle 1800 meters. In the same manner the simulations are run for 20.4 seconds and the first and last 0.2 seconds of transmissions are also ignored. Though the Kangaku propagation model is employed, only the LOS equation is used since there are no barriers in the roadway to LOS. The packets include 200 bytes and have a transmission range of 80m due to the congestion of the scenario. Because all vehicles are within LOS of one another the scheme broadcasts messages in a single-hop fashion. Each individual simulation was run at least twice and the average of these results is presented.

Different vehiclular densities are simulated for separate simulations. In the PATH paper the maximum flow per lane for highways was described to be 2200 vehicles/hour, which amounts to an average of 240 nodes on the 1800 meter highway at any given time, with an average inter-vehicle spacing of 30 meters [11]. This is the maximum flow density, and unless otherwise noted, all simulations are run using this.

6.2.1 Universal Geocast Scheme Highway Simulation

The following results are specific to the UGS scheme. Using the maximum vehicular density, simulations were run and the values of *X* and *Y* were varied to see their effect on the results.

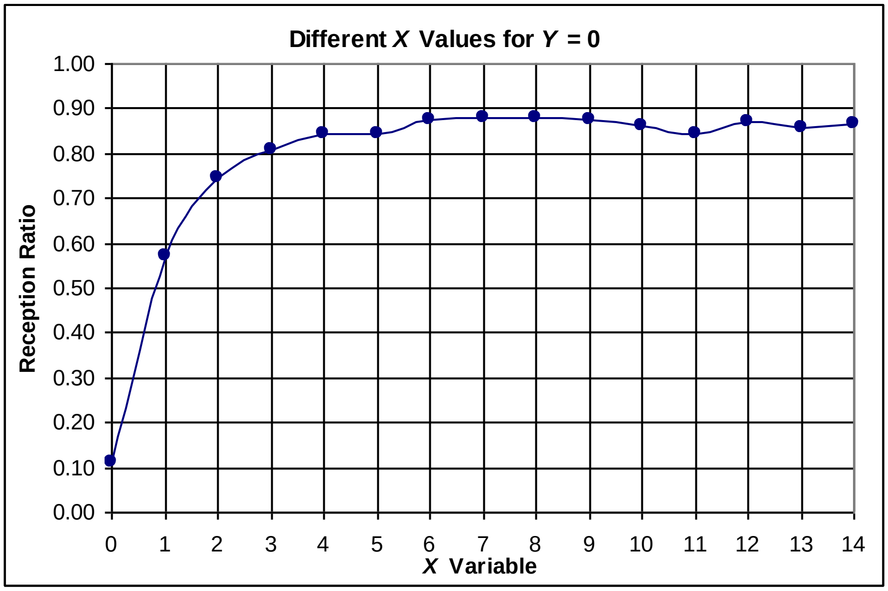

Figure 6.3 Reception ratio for *X* variable when *Y*=0

This figure shows that when a small value of *X* is used, the scheme performs poorly, having a very small reception ratio. This is the result of too many retransmissions caused by a backoff window that is too small. The effect of the *X* variable can also be seen in the figure below, which plots the average packet transmissions (APT) vs. *X*.

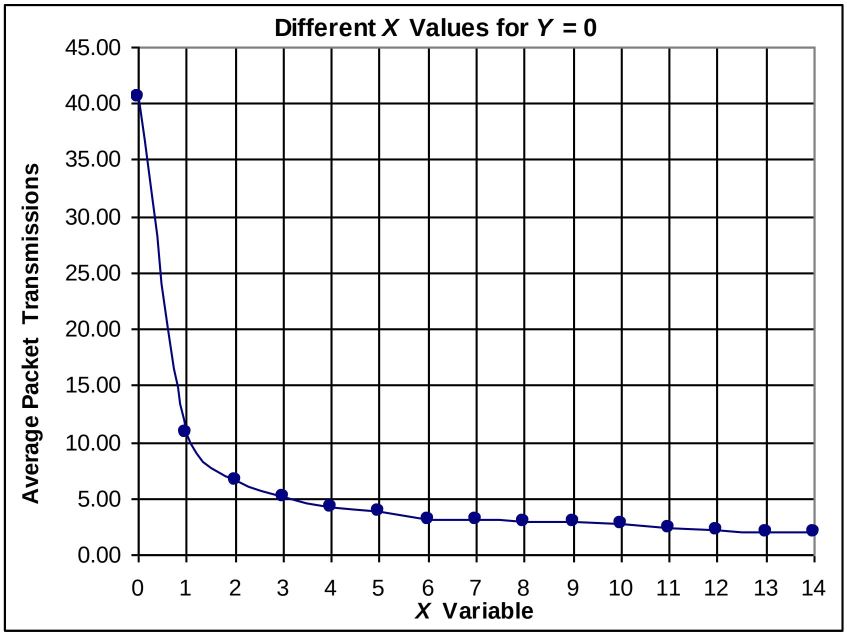

Figure 6.4 APT for *X* variable only

When *X* is 0, the APT is 40.5, which is too many retransmissions per packet to avoid over congesting the channel, causing a rection ratio of only 0.113. A dramatic decrease is seen when the *X* variable is simply set to 1 because now the section in the backoff window equation utilizing *X* and *Y*, which is 2^(*i***X*+*Y*), does not get set to 1 by having both *X* and *Y* equal 0. Instead, the backoff window is doubled for each retransmission, which, on average, will double the amount of wait time before a packet can be retransmitted after each transmission. As the *X* variable increases high enough, the APT starts to settle around the number 2. The reason for this is because most times the backoff is implemented post-TX, as described in section 4.2.2. The node will sense an idle channel and transmit. Upon completion of the transmission, the node will then backoff, to ensure there is at least one backoff interval between two consecutive transmissions. The node will use the backoff interval of the last transmitted packet, which will be the original minimum of 15. The *Y* variable will increase the initial backoff window minimum.

In the figures below the *X* variable is set to 0 and a sweep of different *Y* values is performed. One figure shows reception ratio while the other shows the APT.

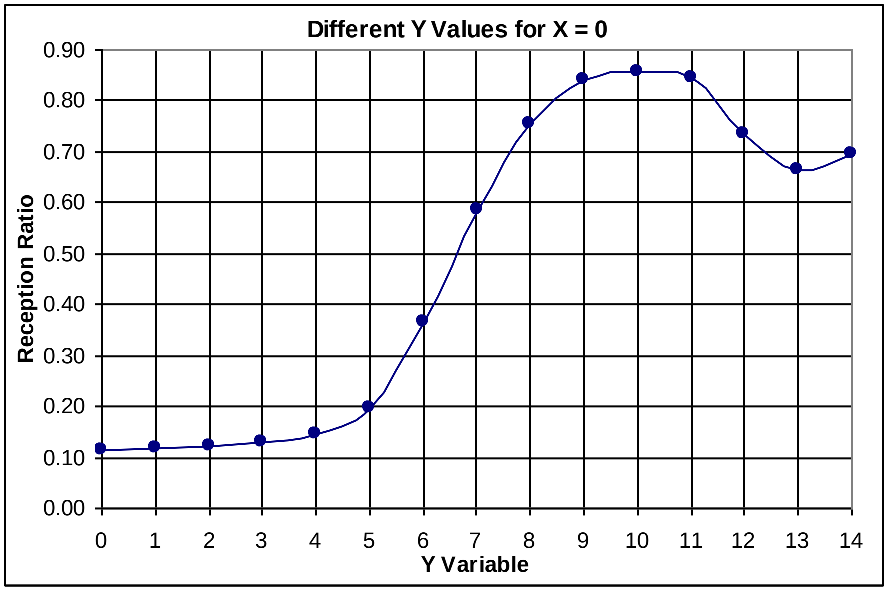

Figure 6.5 Reception ratio for *Y* variable when *X* = 0

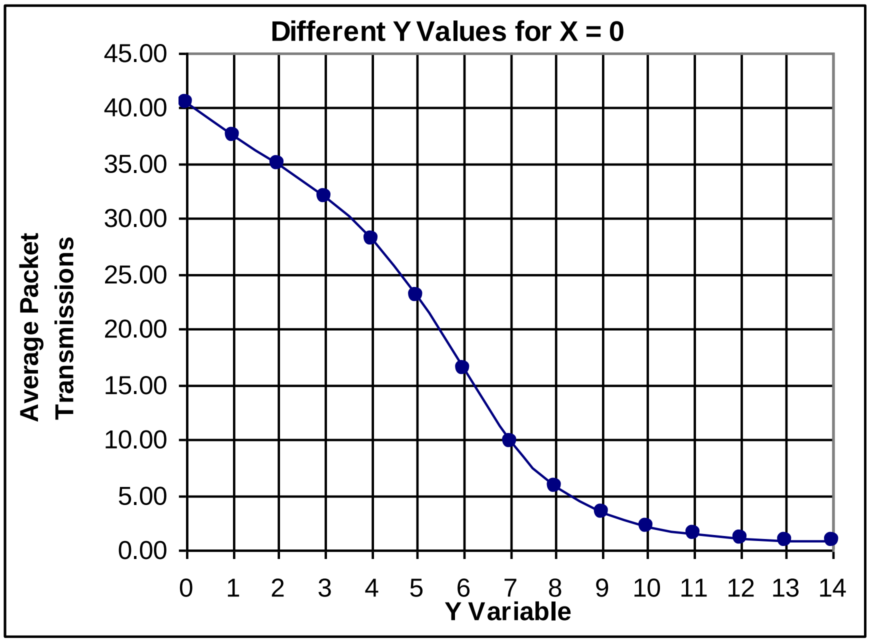

Figure 6.6 APT for *Y* variable only

The figure shows that when *X* is 0 and *Y* is small, the APT is very high, between 35 and 40, which corresponds to a lower reception ratio due to over congestion of the network. When *Y* is equal to 10, the best reception ratio is found for when *X* = 0. This corresponds to an APT of 2.1. Being that this is just an average, some individual packets are transmitted as many as 6 times, while others may only be transmitted once, but the majority of packets are transmitted 2 times. As *Y* is increased even more, the backoff window becomes so large that some packets spend their entire useful lifetime waiting for the backoff to finish and are never sent. This explains the decrease in reception ratio for high *Y* values.

From these experiments, the best values of *X* and *Y* for the highest reception ratio were determined. To ensure the most appropriate values of *X* and *Y*, one variable would be set to a value that produced a relatively high reception ratio while the other variable was again incremented. Figure 6.7 indicates than an *X* value of roughly 9 is ideal for increased reception ratio when used with a *Y=8* value and figure 6.8 indicates a *Y* value of 8 is ideal for an *X* value of 9.

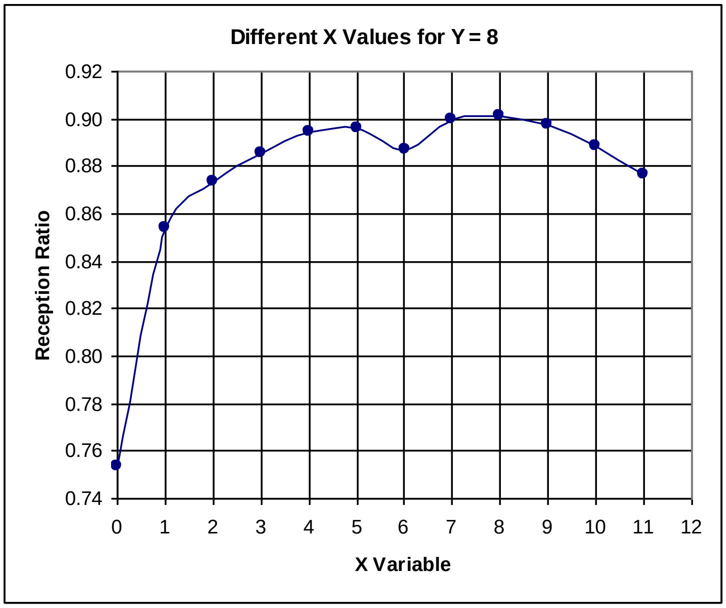

Figure 6.7 Reception ratio vs. *X* for UGS in Highway Simulations

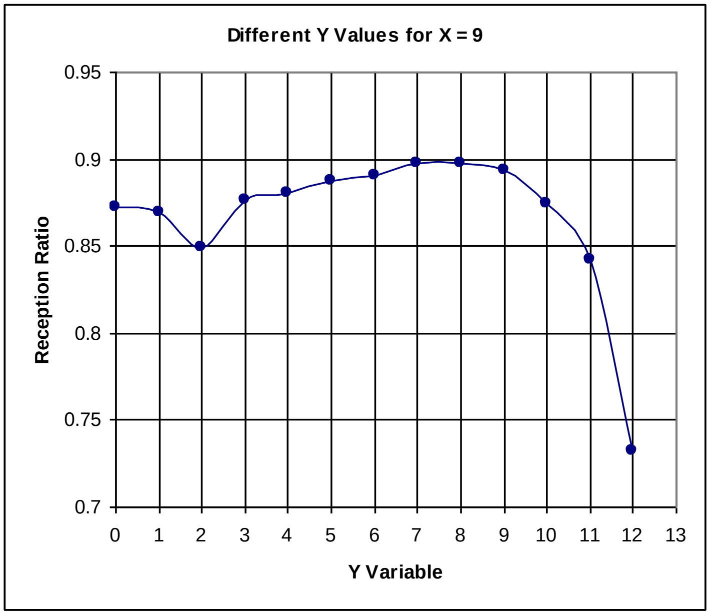

Figure 6.8 Reception ratio vs. *Y* for UGS in Highway Simulations

An important factor in the performance of both PATH and UGS schemes is vehicle density. Taking this maximum density into account, as described previously in section 6.2, simulations were repeated considering traffic densities of 75%, 50%, and 25% of the maximum. These densities may also be referred to as flows, since they are configured by altering the flow of traffic in SUMO to achieve the correct vehicular density. An 1,800 meter long highway simulation at 100% maximum flow will contain 240 vehicles, on average. Flows representing 75%, 50%, and 25% of the maximum flow will contain 180, 120, and 60 vehicles, respectively. Below is a table displaying the results of the UGS scheme for the different percentages of the maximum flow of vehicles.

Max Flow %	**UGS**		
	Recp Ratio	**APT**	**delay (ms)**
100	0.898	2.01	8.28
75	0.922	2.01	6.34
50	0.962	2.01	4.78
25	0.999	2.12	0.72

Table 6.1 UGS with different vehicular densities

As the vehicular density decreases there is less contention for the channel, less collisions, and a better reception ratio. The *X* and *Y* values for these simulations were the optimal values discussed previously in this section where $X = 9$ and $Y = 8$.

6.2.2 PATH Project Highway Simulation

These simulations are nearly identical to the ones performed using UGS. The only difference is that there are no variables *X* and *Y*. Instead, the PATH scheme needs to be told the number of times to attempt a transmission of a packet, which is labeled variable

T for set Transmission variable. This is the same PATH variable discussed for the implementation of the PATH protocol in section 5.5. A plot of the results of the PATH scheme for different T values can be seen in the figure below.

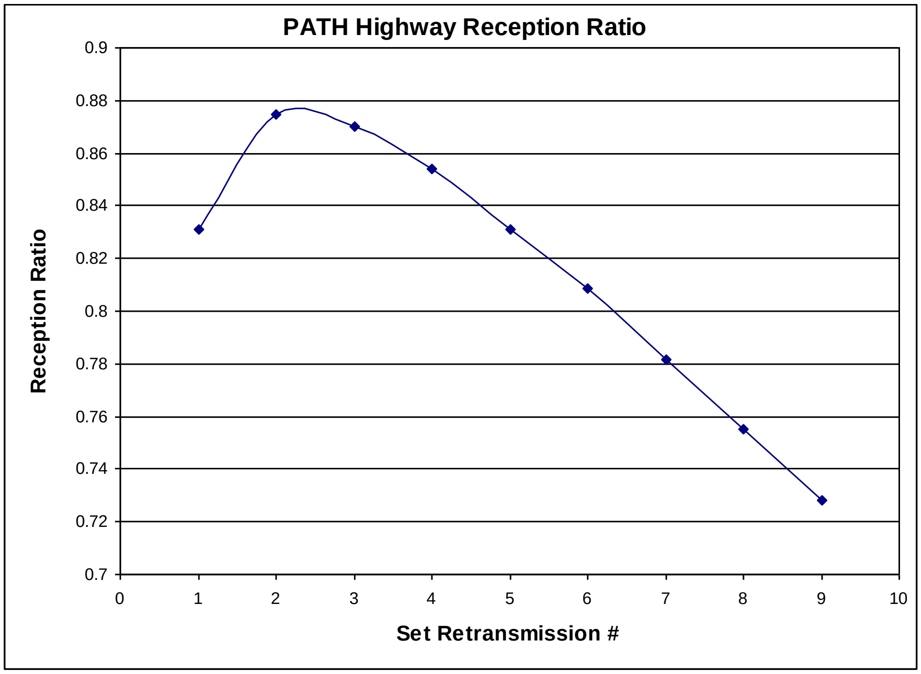

Figure 6.9 PATH Reception Ratio for Highway Simulations

A table of this information, including the APT, can be seen below.

	PATH	
Set TX Num	**Recp Ratio**	**APT**
1	0.831	0.94
2	0.875	1.77
3	0.870	2.53
4	0.854	3.25
5	0.831	3.92
6	0.809	4.55
7	0.782	5.17
8	0.755	5.76
9	0.728	6.35

Table 6.2 PATH Results with different set TX Numbers

Like in simulation using UGS, the best results are found when the scheme uses variables that allow the APT to be closest to 2. In the case of the PATH scheme this happens when the set TX number, *T*, is equal to 2. Notice how the APT is not equal to the set TX number. This is due to the PATH scheme finding the channel busy when attempting a scheduled transmission and differing to the next scheduled transmission time. Given the impact that vehicular density has on the results, the simulations were repeated for 100, 75, 50, and 25% of the maximum vehicular densities. A figure of the reception ratios of the PATH scheme utilizing various set transmission numbers in the different vehicular density simulations can be seen below.

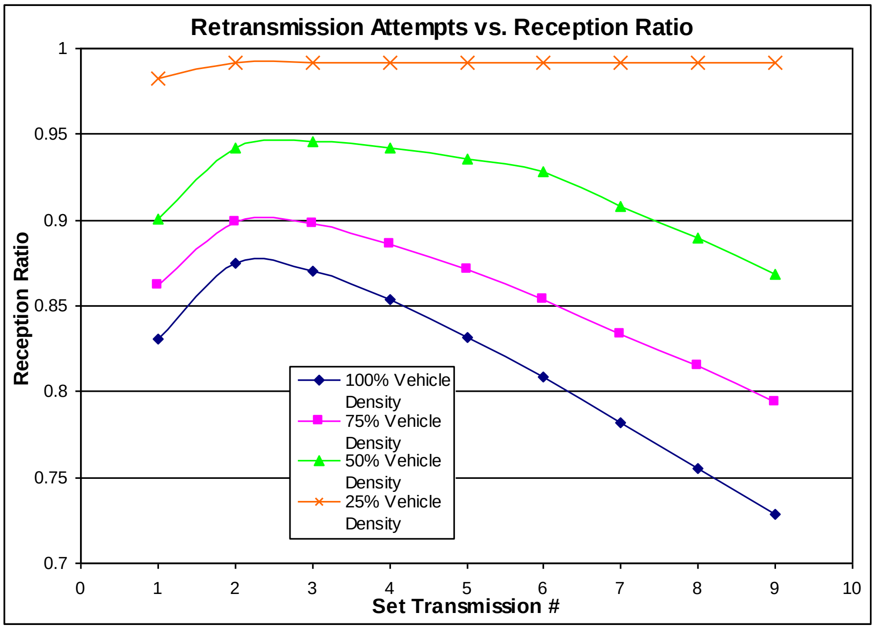

Figure 6.10 Highway reception ratio for PATH

As the density of the simulations decreases, the best performing set number of transmissions may increase. It is important to note that the ideal number of transmissions will likely change if the broadcast range, lifetime, packet size, or any of the other variables regarding packets are changed.

6.2.3 Comparison of UGS vs. PATH in Highway Simulations

This section will compare the results of UGS to PATH in highway simulations when the optimal values are chosen for both scheme's variables to produce the best results. Given that the highway is long enough to demonstrate the interference caused by each node attempting to send and receive messages within its own 80 meter transmission range, the length of the highway should not have a large impact on the reception ratio and other measured values of the simulation. To ensure this, simulations were also performed using a highway 3000 meters in length, again ignorming the first and last 100 meters to account for edge effects. As expected there were no significant differences between the results of the different length highway simulations.

	PATH	UGS	UGS % Improvement
1800 meter	0.875	0.898	2.66
2800 meter	0.866	0.890	2.86

Table 6.3 Reception Ratio vs. Highway Length at 100% Vehicular Density

The table below shows the reception ratio improvement of UGS over PATH at different vehicular densities on the 1800 meter highway.

Max Flow %	PATH	UGS	UGS % Improvement
100	0.875	0.898	2.66
75	0.899	0.922	2.58
50	0.946	0.962	1.79
25	0.992	0.999	0.73

Table 6.4 Highway Reception Ratio Comparison

Though the percent improvement in performance decreases as the vehicular density decreases, UGS shows an improvement of 2.6% at the maximum density. The other important metric in VANETs is delay. The table below compares the average and maximum packet delay of the two schemes in highways.

	Average Delay		Maximum Delay	
Max Flow %	**PATH in ms**	**UGS in ms**	**PATH in ms**	**UGS in ms**
100	81.27	8.28	199.47	195.07
75	79.50	6.34	199.47	190.26
50	63.27	4.78	199.47	137.49
25	9.15	0.72	195.76	33.66

Table 6.5 Average and Maximum Highway Packet Delay of UGS vs. Path

UGS took less than 10% of the amount of time, on average, to successfully transmit and receive a packet than PATH did. The figure below gives a visual representation in the difference in delay.

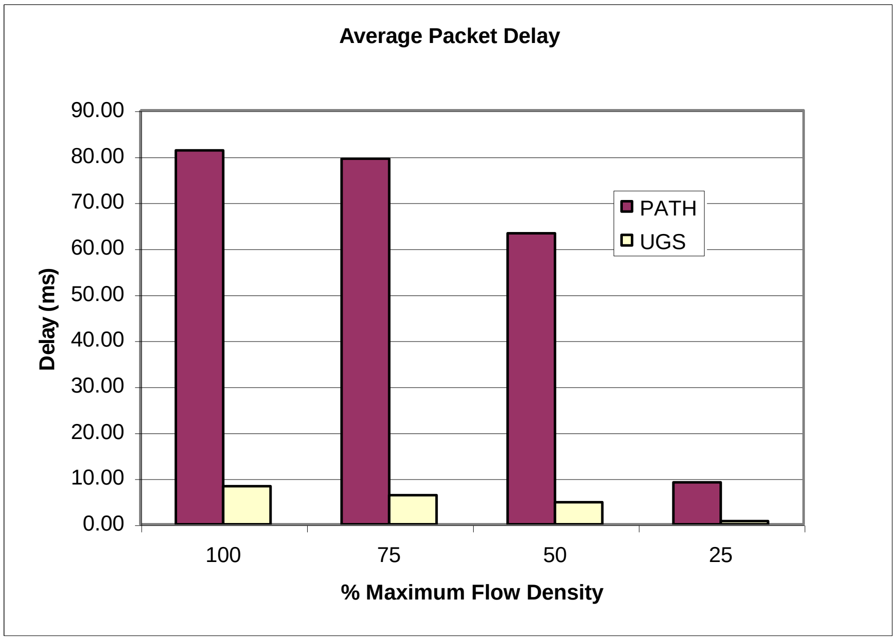

Figure 6.11 Average Highway Packet Delay of UGS vs. Path

There is also a dramatic decrease in the maximum packet delay time in UGS over PATH as the vehicular density decreases. The figure below is meant to represent the lifetime of a packet and give an example of when either scheme will transmit or re-transmit its packet.

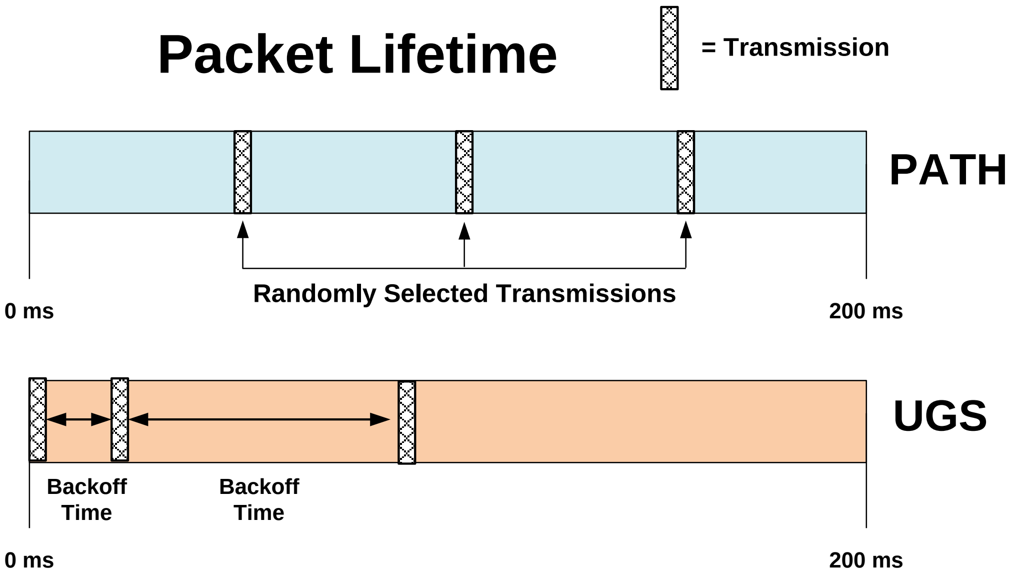

Figure 6.12 Packet Lifetime vs. Transmission Example

The large difference in average delay of the two protocols can be attributed to the PATH protocol picking random times to transmit, despite the channel state. UGS will transmit immediately upon finding the channel idle. Only as the number of transmissions increases does the backoff, and thus wait time, increase. Also, in the less vehicular dense simulations the channel is more likely to be idle a higher percent of the time. This means that the UGS has a higher probability of finding the channel idle and since the backoff is only decremented when the channel is idle the backoff is finished faster, leading to a lower maximum delay value.

The final set of simulations for the highway scenario was done using the FreeSpace propagation model to reference against when the Kangaku propagation was used. Simulations were only run for one vehicular density because the FreeSpace and Kangaku radio propagation models were expected to produce similar results.

Radio Prop Model	PATH	UGS
Kangaku	0.875	0.898
FreeSpace	0.872	0.891
% Diff	0.34	0.75

Table 6.6 Highway reception ratio of FreeSpace vs. Kangaku

This table demonstrates that the results of using either radio propagation model are very close to one another. This was expected since the highway scenario does not have any NLOS communications, the largest difference between the Kangaku and FreeSpace models.

6.3 Intersection Simulations

The intersection scenario is shown in Figure 6.11. In addition to vehicles, the intersection contains a roadside equipment (RSE) unit located near the center of the intersection.

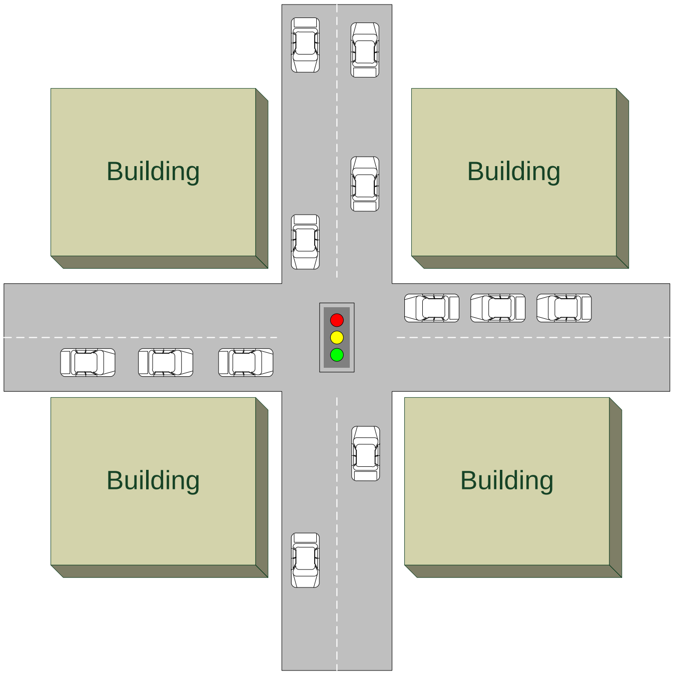

Figure 6.13 Intersection Model for Simulations

Vehicles may be within line-of-sight (LOS) or not (NLOS) depending upon their placement in the intersection. Two identical lengths of road with two lanes each cross in the middle where a traffic light is located. Unless otherwise noted the roads are 900 meters long and located on all four sides of the intersection (representing two 1800 meter roads that cross in the middle). These roads are actually 1000 meters in length, but to avoid edge effects, as in highway simulations, the beginning 100 meters of every road is ignored. For these four roads the beginning is described as the side of the road farthest from the intersection, in whatever direction is applicable. Buildings are assumed to be located on all four corners of the intersection. Vehicles wait at the intersection stoplight until they are given a green light, and continue straight across until they reach the other end of the road. Vehicles that are not in the intersection can transmit their own packets. Vehicles within the intersection can both transmit their own packets and retransmit packets received from other vehicles. The Kangaku radio propagation model is also used in the intersection. Figure 6.12 shows how a packet sent at a power level to overcome NLOS attenuation at a range of 72.8 meters will produce a much longer LOS broadcast resulting in unwanted interference.

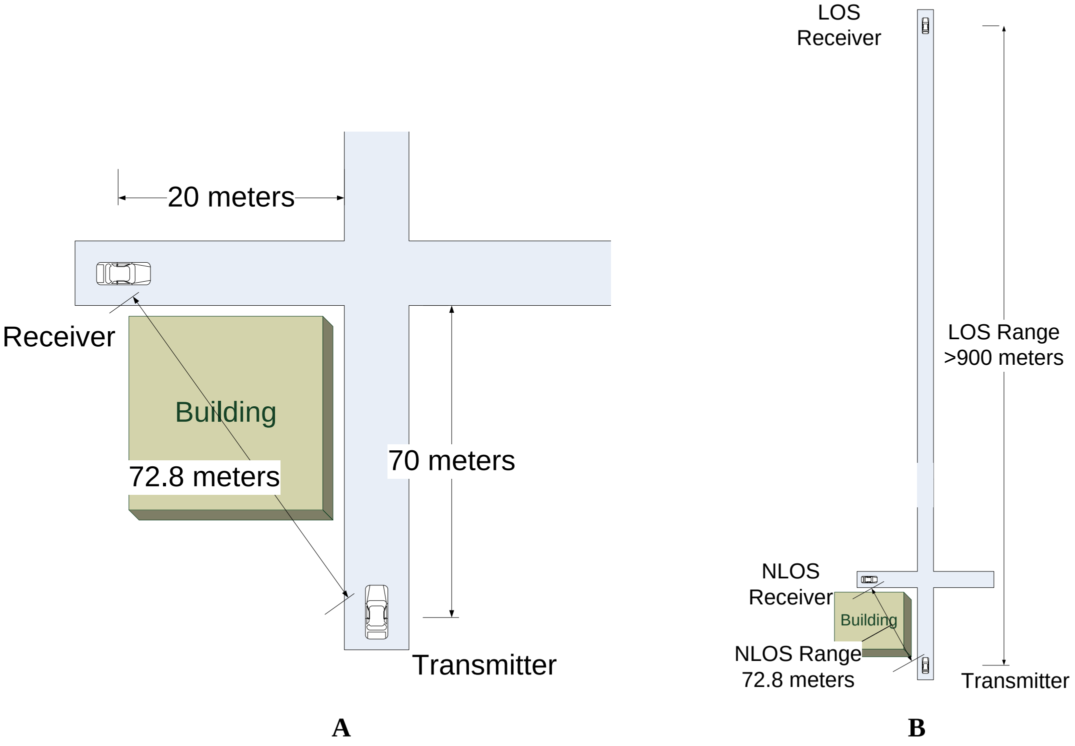

Figure 6.14 Comparison of NLOS and LOS Range (not to scale)

Also in intersections an important factor in the performance of both PATH and UGS schemes is vehicle density. The PATH paper [11] indicated that the maximum flow density for highway simulations is an inter-vehicle spacing of roughly 30 meters. For the 900-meter road intersection scenario, the average number of active vehicles evaluated during 20-second simulations at any given time is 337, leading to an average inter-vehicle spacing of 23.7 meters. Given the slower speeds in intersection simulations and the nature of vehicle movement with a traffic light, this average inter-vehicle spacing is assumed to be the maximum vehicular density of the intersection simulations. Taking this maximum density into account, simulations were repeated considering traffic densities of 75, 50, and 25% of the maximum.

6.3.1 Universal Geocast Scheme Intersection Simulation

A sweep was also performed of the intersection scenario to evaluate the best possible values for the *X* and *Y* variables. Though a number of simulations were performed with different *X* and *Y* values, similar to the highway simulations the ideal *X* value was found to be 9 and the ideal *Y* value 8. The same procedure of setting one of the variables and running simulations as the other variable was incremented was performed.

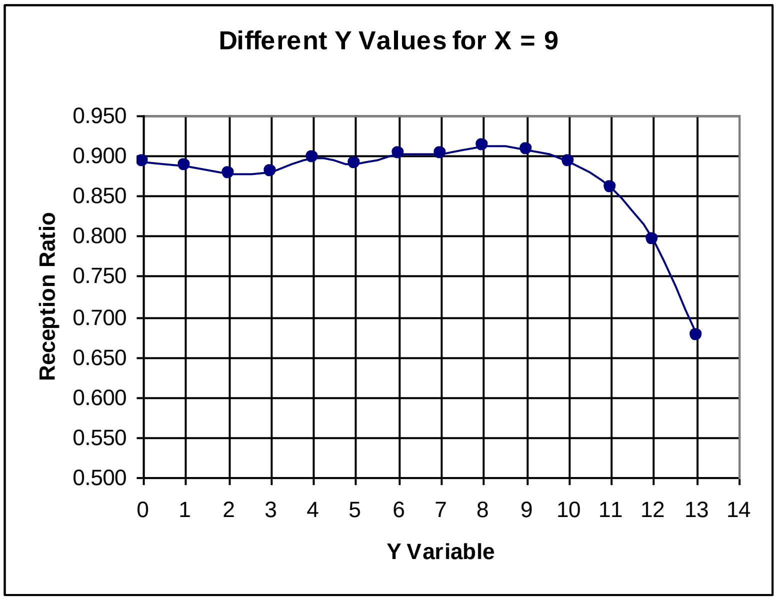

Figure 6.15 Reception ratio vs. *Y* for UGS in Intersection Simulations

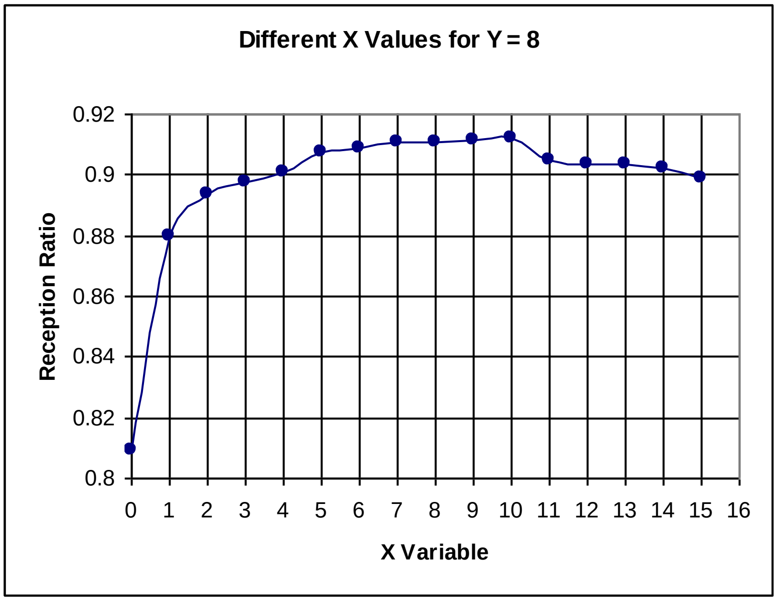

Figure 6.16 Reception ratio vs. *X* for UGS in Intersection Simulations

As in the highway simulation the ideal setting for the UGS backoff variables is *X* = 9 and *Y* = 8.

Another important variable that impacts UGS is the set CW for intersection node forwarding, especially when an RSE is used. The figure below indicates the reception ratio of UGS in a 900-meter road length intersection scenario under varying intersection *CW values*.

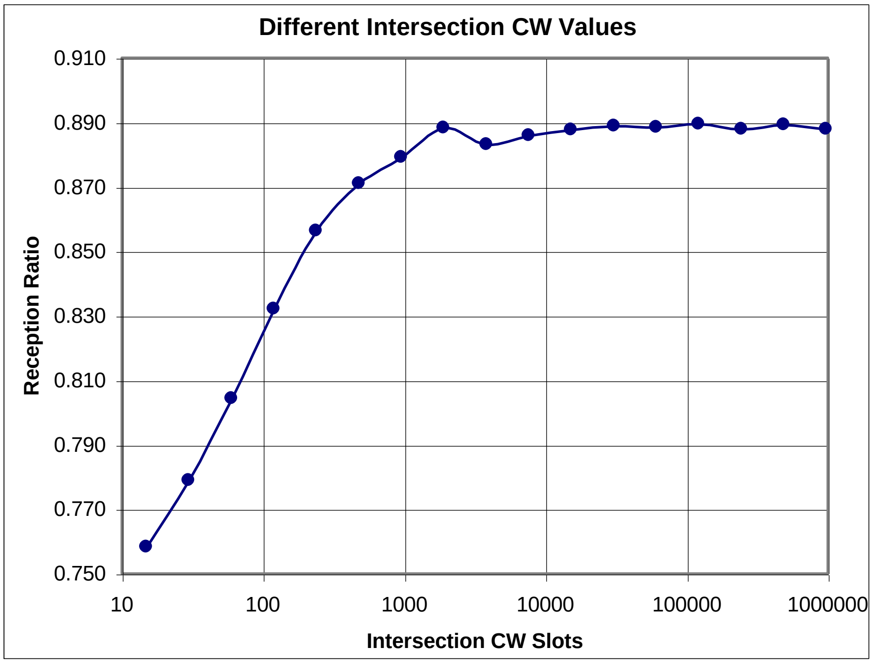

Figure 6.17 Effect of Intersection CW on Reception Ratio

The table below presents the average number of times each packet is forwarded and how many unique packets are forwarded per vehicle as a function of *CW* at the RSE. This does not represent all of the packet forwarding as other intersection vehicles also contribute, though it does represent a significant example of how the set intersection cw effects packet forwarding.

Intersection CW	Average Forwards per Packet	Unique Packets per Second
15	16.00	64
60	5.28	133
480	1.25	166
1200	1.01	87
4800	1.05	21
10000	1.08	12

Table 6.7 RSE Retransmissions

When the backoff is large enough, each packet will only be forwarded once. In order to account for some packets not being sent at all and other being sent many times, when the post-Tx backoff is running, no matter how large, if a packet arrives that has not been sent, the backoff is interrupted. This is done to allow for fairness between the different packets being sent. For instance, if an RSE has just forwarded a packet that has already been sent a number of times, upon transmitting the packet it will start the post-TX backoff, which after multiple transmissions will be very large. If the RSE then receives another packet to forward that has not yet been sent, it is not fair to make the newly arrived packet wait for the previously sent packet's backoff to finish. Thus the backoff is interrupted and the new packet forwarded. In very congested simulations like the ones presented in this research, one forward from an intersection node is all that is needed to adequately increase the reception ratio of that packet without over burdening the already heavily congested channel. The performance of UGS was also evaluated for when both the RSE and vehicles in the intersection forward other vehicles' packets and when the RSE alone does all of the packet forwarding. The table below shows the reception ratio of either scenario at various set intersection cw values.

Intersection cw	Vehicles & RSE	RSE Only	% Difference
15	0.757	0.757	0.0
960	0.879	0.881	0.2
3840	0.883	0.885	0.3
15360	0.888	0.886	0.2
30720	0.889	0.887	0.2

Table 6.8 RSE only vs. RSE & vehicle forwarding

This table demonstrates that there is less than 0.3% difference in reception ratio between the two, meaning that either form can be an acceptable method of performing packet forwarding.

6.3.2 PATH Project Intersection Simulation

A plot of the results of the PATH scheme in intersections for the maximum flow density for different T values can be seen in the figure below.

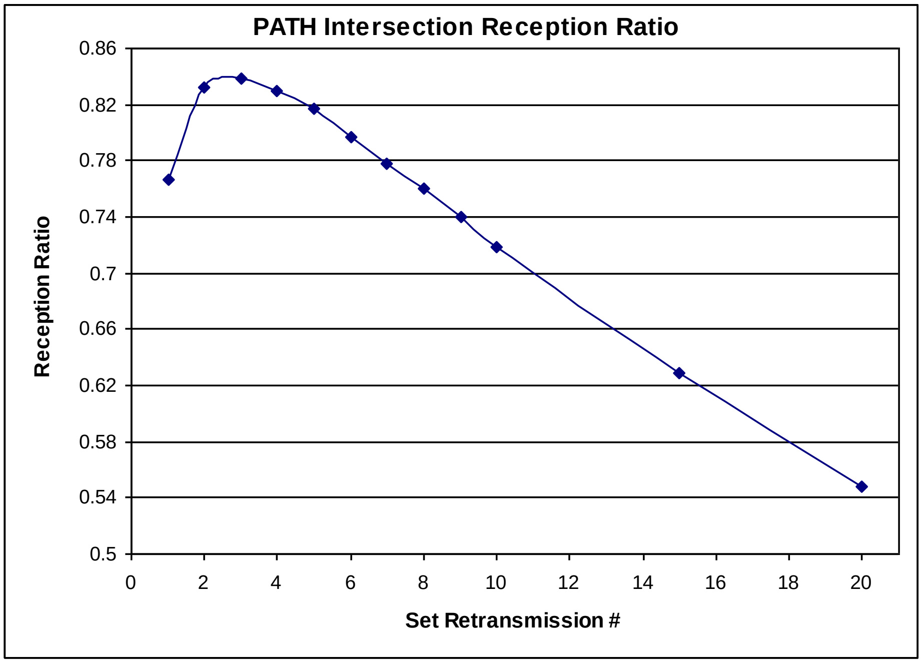

Figure 6.18 PATH Reception Ratio for Intersection Simulations

Just like the highway simulation the intersection simulation were repeated with varying vehicle densities of 100, 75, 50, and 25%. A figure of the reception ratios of the PATH scheme utilizing various set transmission numbers in the different vehicular density for the intersection simulation can be seen below.

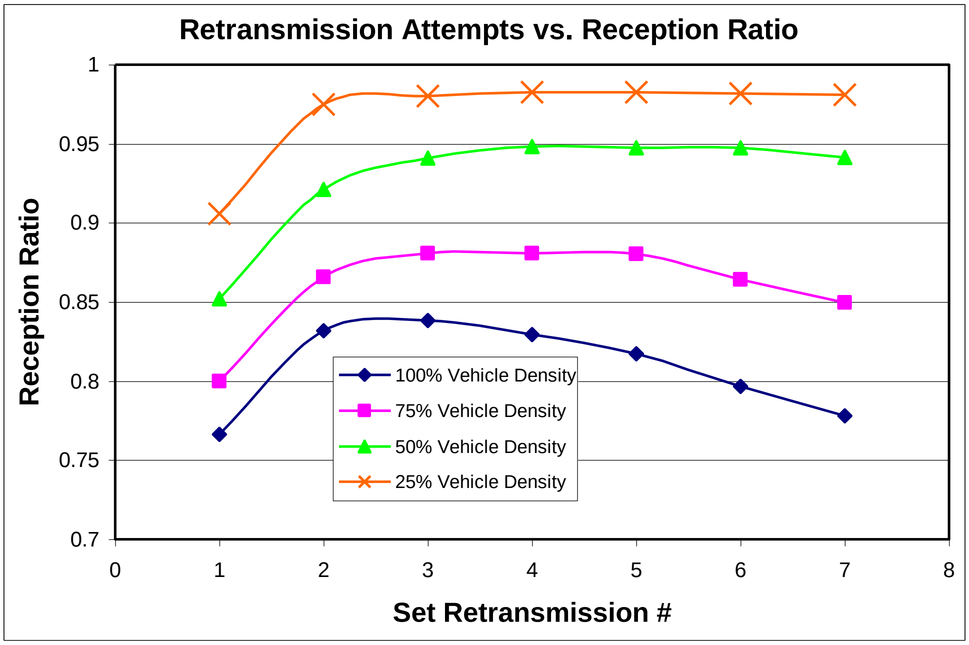

Figure 6.19 Intersection Reception Ratio for PATH vs. Vehicle Density

In this case the best reception ratio is found when the set retransmission number is between 2 and 4, depending on the vehicular density of the scenario.

6.3.3 Comparison of UGS vs. PATH in Intersection Simulations

Intersection scenarios utilizing 500, 800, 900, and 1,200 meter roads on all four sides of an intersection were evaluated in four separate experiments. As shown in Figure 6.17, as the length of the roads increased, the reception ratio of UGS versus PATH improved. UGS has a decrease in performance by 1% for 500 meter roads, but an increase of 4.6%, 6.2%, and 6.5% for 800, 900, and 1200 meter roads, respectively. The longer roads allow for improved interference modeling of single-hop transmissions. The added road length increases inter-vehicle transmission interference, especially when NLOS vehicles within the 80 meter broadcast range are considered.

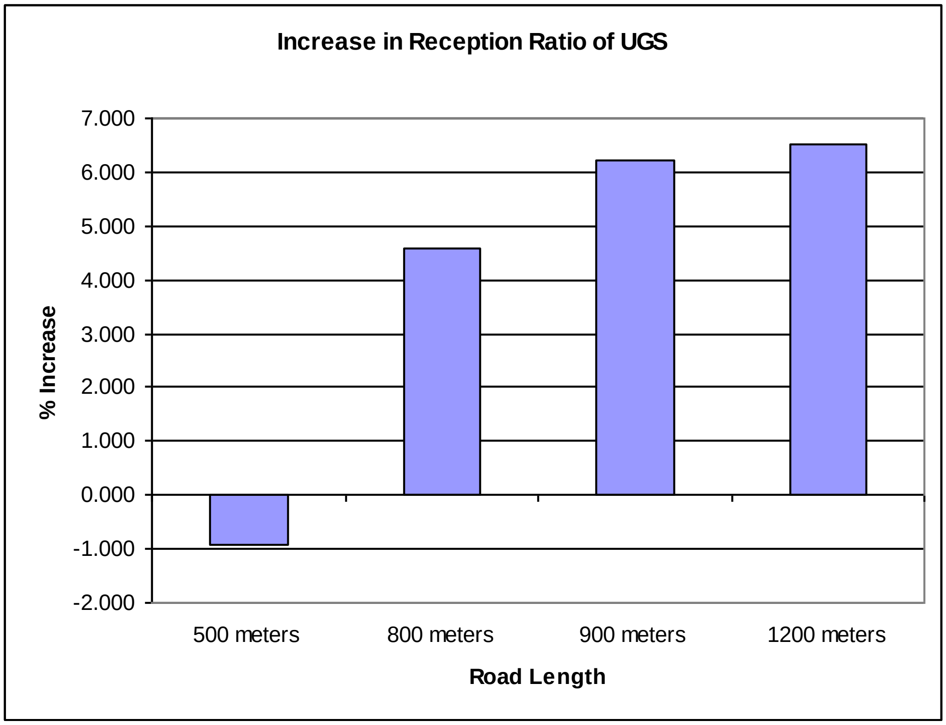

Figure 6.20 Reception Ratio Increase of UGS vs. PATH for Different Road Lengths

As Table 6.9 demonstrates, as vehicle density decreases, the advantage of the UGS scheme is reduced. The last two columns represent the performance increase of UGS with and without the use of an RSE located in the center of an intersection to forward packets. In the densest simulations, UGS has an increased performance over the PATH scheme by over 6%.

Max Flow %	PATH	UGS	UGS with RSE	UGS % Improvement	
				Without RSE	With RSE
100	0.839	0.891	0.892	6.2	6.3
75	0.881	0.919	0.920	4.4	4.4
50	0.948	0.963	0.960	1.5	1.2
25	0.983	0.981	0.984	-0.2	0.2

Table 6.9 Intersection Reception Ratio

The table also shows that there is less than 0.4% difference in reception ratio for when an RSE is used versus when it is not used. This information and the 0.3% difference in performance shown in Table 6.8, of RSE only vs. RSE & vehicle forwarding demonstrates that the UGS is able to effectively function with or without an RSE. This is important because it means that the UGS does not need to have an infrastructure of RSE units set up to make it effective, and can be employed via vehicle transceiver units only.

Again, the other important metric in VANETs is delay. The table below compares the average and maximum packet delay of the two schemes in intersections.

	Average Delay		**Maximum Delay**	
Max Flow %	**PATH in ms**	**UGS in ms**	**PATH in ms**	**UGS in ms**
100	76.28	13.01	199.47	199.41
75	74.47	12.74	199.47	199.43
50	63.42	8.33	199.47	199.49
25	46.62	5.37	199.47	197.12

Table 6.10 Average and Maximum Intersection Packet Delay of UGS vs. Path

The table demonstrates the decrease in delay of the UGS. In the maximum flow simulations UGS, on average, only took 17% of the time that the PATH scheme did in successfully transmitting packets to intended receivers. Again, a figure is include to visually represent the average delay between the two schemes.

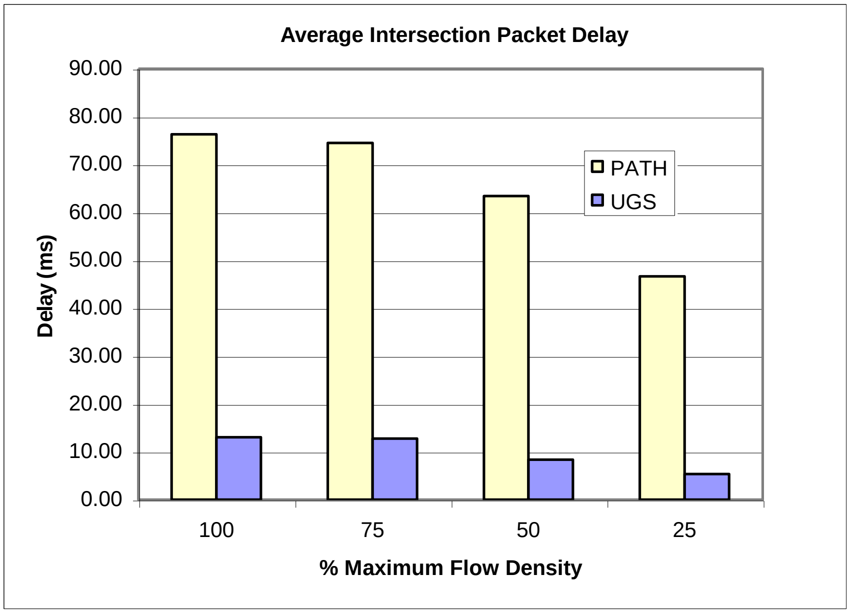

Figure 6.21 Average Intersection Packet Delay of UGS vs. Path

This happens for the same reasons explained in section 6.2.3. The maximum delay of both scheme is closer to equal for the intersection simulations. This is most likely because of the extra delay incurred due to multi-hop transmissions for some packets.

One additional problem incurred when using the PATH scheme is the number of packets that are never sent. In the maximum vehicle density scenario, due to the congestion, the set number of retransmission attempts for PATH to provide the best reception ratio is 3. As stated in section 3.1, these 3 set retransmissions are the randomly selected *k* distinct time slots as potential transmission slots out of the *n* slots constituting the entire lifetime of the packet. If the channel is currently busy when this scheduled time slot arrives, the packet will not be sent and is deferred to the next selected time slot. If during all of the selected time slots the MAC finds the channel to be busy, the packet is simply dropped and never sent out at all. This is accounted for in the reception ratio, but by not sending some packets, the congestion is decreased and other packets are able to be received by more intended targets. Table 6.11 shows the average number of total packets sent in a given simulation, the average number of Packets Never Sent (PNS), and the percent of packets that are never sent. In the table "Dropped" is a reference to packets being dropped by the MAC and never transmitted at all.

Max Flow %	Total Pkts Sent	PATH		UGS	
		Pkts Never Sent	PNS %	Pkts Never Sent	PNS %
100	33071	595	1.80	0	0.00
75	24656	404	1.64	0	0.00
50	16662	69	0.41	0	0.00
25	8398	2	0.02	0	0.00

Table 6.11 Intersection Packets Never Sent from MAC

The final set of simulations for the intersection scenario was done using the FreeSpace propagation model to reference against when the Kangaku propagation was used. Simulations were only run for the maximum vehicular density with the 900 meter intersection roads. Compared to the highway simulations, there is a much larger difference between the reception ratios for the intersection simulations, especially for UGS. A table of the difference between reception ratios using the different radio propagation models for the intersection with the maximum vehicle density can be seen below.

Radio Prop Model	PATH	UGS	UGS with RSE
Kangaku	0.839	0.891	0.892
FreeSpace	0.825	0.833	0.822
% Diff	1.67	6.56	7.76

Table 6.12 Intersection Reception Ratio of FreeSpace vs. Kangaku

In all intersection simulations there is a decrease in performance when switching from the Kangaku radio propagation model to the FreeSpace. This happens because the NLOS equations used by Kangaku help to break up the vehicles into two nearly separate collision domains, one being the East-West road and the other the North-South road. Without this barrier between them vehicles from either road more freely share the same medium, causing more collisions. This is why UGS sees a larger decrease in performance. The point of sending messages in a multi-hop fashion is to decrease the power they are sent at, thus decreasing the interference they impose on other vehicles attempting communications. Without NLOS attenuation, the multi-hop transmissions easily reach other vehicles on the opposite road, and the additional transmissions by intersection nodes forwarding the packets only increase the congestion. This explains why simulations that use a designated RSE to forward packets have the largest decrease in performance when switching to the FreeSpace radio propagation model.

CHAPTER 7

CONCLUSION

7.1 Conclusions

This research describes a detailed evaluation of a new multi-hop inter-vehicle communication protocol, which has been optimized for obstructed intersections. This modified Universal Geocast scheme performs 6% better in intersections and 2.6% better in highways in terms of reception ratio, than an accepted, previously-published approach [11] for periodic inter-vehicle messages which do not use RTS/CTS and ACK messages. The Universal Geocast Scheme also provides an 82% decrease in delay in intersection simulations and a 90% decrease in delay in highway simulations over the previous approach. The UGS repetition-based scheme allows for both LOS and NLOS packet transfer. Our results have been generated using a modified NS-2 simulator, a recently-developed radio propagation model, and a traffic mobility simulator.

7.2 Further Work

Now that the Universal Geocast Scheme has been evaluated as a successful candidate for inter-vehicle communication, the next step would be to use the scheme in real life experiments. Currently work is being conducted in the UMass Transportation Department to allow for systems to be installed on cars to collect pertinent information and then use transceivers to broadcast this data to other vehicles. Another group of students is working to implement the Universal Geocast Scheme through the use of FPGA hardware. The work with FPGAs could be used in conjunction with the

Transportation Departments work to allow the vehicular information to be sent using the Universal Geocast Scheme, so actually results may be collected.

BIBLIOGRAPHY

[1] USDOT Transportation Statistics, http://www.rita.dot.gov/publications/transportation_vision_2030/pdf/entire.pdf, accessed Nov. 2010

[2] USDOT RITA information on ITS http://www.its.dot.gov/ accessed Nov. 2010

[3] Farnoud, F.; Valaee, S.; *Reliable Broadcast of Safety Messages in Vehicular Ad Hoc Networks*, IEEE INFOCOM 2009

[4] Draft standard *IEEE 802.11p wireless access for vehicular environments*, http://grouper.ieee.org/groups/802/11/.

[5] USDOT RITA Connected Vehicle Program http://www.its.dot.gov/connected_vehicle/connected_vehicle.htm accessed Apr. 2011

[6] USDOT Research and Innovative Technology Association, http://www.rita.dot.gov/ accessed Nov. 2010

[7] USDOT National ITS Architecture, http://www.iteris.com/itsarch/html/entity/paents.htm accessed Nov. 2010

[8] IntelliDrive Homepage, http://www.intellidriveusa.org/ accessed Nov. 2010

[9] Morrell, S.; Marcq, D.; LaBarge, J.; Swochak, A.; *Project Crossroads: Vehicular Ad Hoc Network for Collision Warning* UMass Amherst, Senior Design Project 2008

[10] Nekoui, M.; Pishro-Nik, H.; *Reliable inter-vehicle communications for vehicular ad hoc networks* Proceedings of The Fourth Annual International Wireless Internet Conference (WICON 2008), 2008

[11] Xu, Q.; Mak, T.; Ko, J.; Sengupta, R.; *Medium Access Control Protocol Design for Vehicle-Vehicle Safety Messages* IEEE Transactions on Vehicular Technology, Vol. 56, Nr. 2, March (2007), p. 499-518.

[12] Nekoui, M.; Pishro-Nik, H.; *A Universal Geocast Scheme for Vehicular Ad Hoc Networks* Consumer Communications and Networking Conference (CCNC), 2010 7th IEEE p. 1-5

[13] Wireless ad hoc networks wiki, http://en.wikipedia.org/wiki/Wireless_ad_hoc_networks accessed Nov. 2010

[14] RFC 2501 *Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations*, Jan. 1999 http://datatracker.ietf.org/doc/rfc2501/

[15] Status of Project IEEE 802.11 Task Group p http://grouper.ieee.org/groups/802/11/Reports/tgp_update.htm accessed Nov. 2010

[16] Xu, S.; Saadawi, T.; *Does the IEEE 802.11 MAC protocol work well in multihop wireless ad hoc networks?* Communications Magazine, IEEE Jun. 2001 Vol.39 Issue: 6, p. 130 - 137

[17] Dickey, S.; Dulmage, J.; Huang, C.; Sengupta, R.; *ITS Band Roadside to Vehicle Communications in a Highway Setting* California PATH Research Report, Caltrans report number: CA10-0817, April 2010 http://www.path.berkeley.edu/PATH/Publications/PDF/PRR/2010/PRR-2010-17.pdf

[18] Jiang, D.; Delgrossi L.; *IEEE 802.11p: Towards an International Standard for Wireless Access in Vehicular Environments* IEEE Vehicular Technology Conference, May 2008, p. 2036-2040

[19] U.C. Berkeley ITS California Partners for Advanced Transit and Highways http://www.path.berkeley.edu/ accessed Nov. 2010

[20] ACM SIGMOBILE *The Seventh ACM International Workshop on VehiculAr Inter-NETworking (VANET 2010)* http://www.sigmobile.org/workshops/vanet2010/

[21] Sai, S.; Niwa, E.; Mase, K.; Nishibori, M.; Inoue, J.; Obuchi, M.; Harada, T.; Ito, H.; Mizutani, K.; Kizu, M.; *Field Evaluation of UHF Radio Propagation for an ITS Safety System in an Urban Environment* IEEE Communications Magazine, Nov. 2009 Vol. 47 Issue: 11, p. 120-127

[22] Haddad, I.; Gordon, D.; *Network Simulator 2: a Simulation Tool for Linux*, Linux Journal, Oct. 21, 2002 http://www.linuxjournal.com/article/5929

[23] Haldar, P.; Chen, X.; *Tutorial: Internet simulations with the ns simulator, 2002.* http://www.isi.edu/nsnam/ns/ns-tutorial/tutorial-02/slides/NS_Fundamentals_1-1.pdf

[24] Ivanov, S.; Herms, A.; Lukas, G.; *Experimental validation of the ns-2 wireless model using simulation, emulation, and real network* Proceedings of the 4th Workshop on Mobile Ad-Hoc Networks (WMAN '07), Berne, Switzerland, March 2007.

[25] Breslau, L.; Estrin, D.; Fall, K.; Floyd, S.; Heidemann, J; Helmy, A.; Huang, P.; McCanne, S.; Varadhan, K.; Xu, Y.; Yu, H.; *Advances in Network Simulation.* IEEE Computer, 33 (5) pp. 59-67, May, 2000.

[26] Chen, Q.; Schmidt-Eisenlohr, F.; Jiang, D.; Torrent-Moreno, M.; Delgrossi, L.; Hartenstein, H.; *Overhaul of IEEE 802.11 modeling and simulation in ns-2* Proceedings of the 10th ACM Symposium on Modeling, analysis, and simulation of wireless and mobile systems, p. 159–168, October 2007.

[27] Maltz, D.; Johnson, B.; Hu, Y.; Jetcheva J.; *A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols*" Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking 1998 http://courses.csail.mit.edu/6.885/spring06/papers/BrochMJHJ.pdf CMU

[28] Takai, M.; Martin, J.; Bagrodia, R.; *Effects of wireless physical layer modeling in mobile ad hoc networks* Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing (MOBIHOC 2001), pages 87–94, New York, NY, USA, 2001

[29] Kotz, D.; Newport, C.; Gray, R.; Liu, J.; Yuan, Y.; Elliot, C.; *Experimental Evaluation of Wireless Simulation Assumptions* Proceedings of the MSWiM'04, Oct. 2004

[30] Torrent-Moreno, M.; Jiang, D.; Hartenstein, H.; *Broadcast Reception Rates and Effects of Priority Access in 802.11-Based Vehicular Ad-Hoc Networks,* Proceedings of the 1st ACM International Workshop on Vehicular Ad Hoc Networks (VANET), Philadelphia, PA, USA, Oct. 2004.

[31] Henderson, T.; Roy, S.; Floyd, S.; Riley, G.; *ns-3 project goals*, Proceedings of the 2006 workshop on ns-2: the IP network simulator, Pisa, Italy, October 2006

[32] Lacage, M.; Henderson, T.; *Yet another network simulator*, Proceedings of the 2006 workshop on ns-2: the IP network simulator, Pisa, Italy, October 2006

[33] NS-2, *The Manual*, http://www.isi.edu/nsnam/ns/doc/ accessed Nov. 2010

[34] Jérôme, H.; Fethi, F.; Christian, B.; *Mobility models for vehicular ad hoc networks: a survey and taxonomy*, IEEE Communications Surveys and Tutorials, Vol. 11 No. 4, December 2009, ISSN: 1553-877X , p. 19-41

[35] Carrier Sense Multiple Access with Collision Avoidance wiki. http://en.wikipedia.org/wiki/Carrier_sense_multiple_access_with_collision_avoidance accessed Nov. 2010

[36] CORSIM Microscopic Traffic Simulator http://mctrans.ce.ufl.edu/featured/tsis/ accessed Dec. 2010

[37] Antoniotti, M.; Gollu, A.; *SHIFT and SMART-AHS: A Language For Hybrid System Engineering Modeling and Simulation*, October 1997 http://path.berkeley.edu/smart-ahs/DSL97.html

[38] California PATH Smart AHS http://path.berkeley.edu/smart-ahs/index.html accessed Dec. 2010

[39] OPNET Application and Network Performance Simulator http://www.opnet.com/ accessed Dec. 2010

[40] GloMoSim Global Mobile Information Systems Simulation Library http://pcl.cs.ucla.edu/projects/glomosim/ accessed Dec. 2010

[41] Simulation of Urban Mobility (SUMO), Institute of Transportation Systems at the German Aerospace Center http://sourceforge.net/apps/mediawiki/sumo/index.php?title=Main_Page Accessed Dec. 2010