ENHANCING THE VANET NETWORK LAYER

ROSTISLAV KITSIS

A THESIS SUBMITTED TO
THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

GRADUATE PROGRAM IN COMPUTER SCIENCE AND ENGINEERING
YORK UNIVERSITY
TORONTO, ONTARIO

June 2016

Abstract

The aim of this thesis is to examine existing VANET network layer functionality and to propose enhancements to the VANET network layer to facilitate vehicular (V2X) communication.

This thesis proposes three enhancements to the VANET network layer which address many of the issues with V2X communication, these enhancements are: a geographic overlay allowing vehicles to localize themselves; an IPv6 addressing strategy which embeds positional information within an IP address allowing for location based routing; and finally a novel position based routing protocol which has two primary advantages over existing protocols, firstly removing unnecessary overhead information and control communication, and secondly support for multiple types of V2X communication models.

The simulation results show that the proposed enhancements are well suited in low and medium vehicular density environments. Based on the observed behaviors the author recommends further modification and study of position based routing protocols.

Acknowledgements

This thesis would not have been possible without guidance and support from my family, friends and instructors.

I would like to thank my family and friends for supporting me throughout my thesis. Your encouragement and support throughout my entire academic career has been invaluable, especially when things were tough I knew I could always rely on my family and friends. Without you, this thesis would not have been possible.

I would also like to thank all of my instructors throughout my academic career; their work motivated me to explore the limits technology culminating in this thesis. I would like to extend special thanks to my adviser Dr. Suprakash Datta for providing guidance throughout the development of this thesis. I would also like to express my gratitude to the members of my examination committee, Dr. Uyen Trang Nguyen and Dr. Marin Litoiu.

Table of Contents

List of Tables

List of Figures

1 Introduction

Communication technologies have become an integral part of North American society in the past 20 years. Starting with a handful of universities the Internet has grown to approximately 3 billion users with more users added each year. Starting with slow wired networks, today communication has moved to the wireless domain where WiFi and cellular communication are a standard part of life in North America. Wireless communication technology has been integrated into two major pieces of everyday technology: computers and hand-held smart phones. With the increasing amount of vehicles on the road every year, the next major area of interest for wireless technology integration is in the vehicle in the form of VANETs; Vehicular Ad-Hoc Networks.

Current trends in technology see vehicles and roads being equipped with wireless communication capabilities in the form of *On-Board Units* (OBU) in vehicles and *Road Side Units* (RSU) deployed along roads. The goal of these trends is to increase driver safety, increase road utilization and finally increase passenger enjoyment while on the road [13]. Broadly this set of applications are referred to as an Intelligent Transportation

System (ITS) where the VANET is a vital part of the system allowing for communication between vehicles and roadside infrastructure [13].

1.1 VANET Communication Models

VANET communication may be seen as falling into one of three models; entirely infrastructure dependent, entirely ad-hoc or hybrid, an illustration of these models may been seen in Figure 1.1.

1.1.1 Infrastructure Model

Vehicles communicating in the infrastructure model exclusively communicate with pre-existing infrastructure such as cellular towers. This communication model is generally seen as undesirable for the following reasons.

1.1.1.1 Cellular Tower Congestion

Cellular tower congestion is a growing issue in Canada and North America in general, as the number of mobile users grows, cellular towers must manage ever increasing amounts of traffic. Classical GSM technology uses a Time Division Multiple Access (TDMA) MAC protocol to assign time-slots during a frame for which a node has exclusive permission to transmit. However, TDMA is quite vulnerable to congestion, once the total number of time-slots has been exceeded, new and existing nodes will experience degradation

in performance. Newer LTE/4G technology uses a more advanced Orthogonal Frequency Division Multiplexing (OFDM) data-link layer to manage channel access. Nodes using OFDM are assigned multiple non-interfering or orthogonal frequencies over which they may simultaneously transmit; once again when the number of nodes exceeds the system limits all nodes will suffer performance degradation. The effects of congestion may be seen during particularly busy periods of communication such as during sporting events or during emergencies.

By adding further reliance on cellular networks, congestion issues will be further compounded and may impact regular cell phone usage.

1.1.1.2 Vulnerability to Failure

A purely infrastructure based model is centralized and is prone to failure should anything happen to the infrastructure [23]. Without infrastructure in place vehicles will not be able to communicate with either infrastructure or other vehicles.

1.1.2 Ad-Hoc Model

Vehicles communicating using the ad-hoc model communicate directly with each other without any additional infrastructure. Vehicles are self organizing, self addressing and entirely distributed [27]. Ad-hoc communication models are extremely resilient to failure; should a vehicles' OBU fail, rendering it unable to communicate with the network,

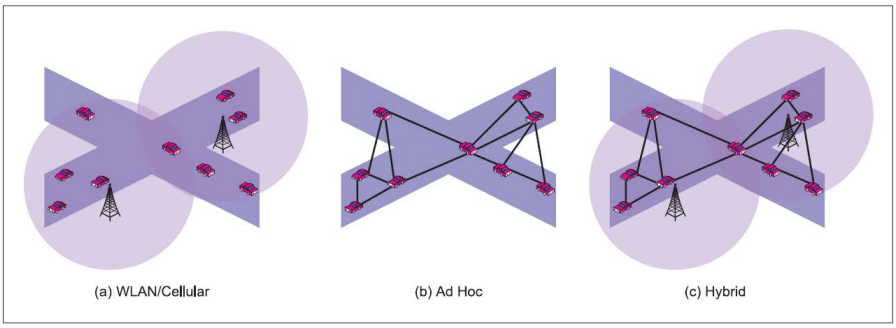

Figure 1.1: VANET Models of Communication

[28]

neighboring vehicles may quickly compensate for the failure and restore network connectivity.

The ad-hoc model is well suited to disseminate emergency information, it directly communicates a vehicles state to all other vehicles in a zone of reference. However, the ad-hoc model suffers from inability to reach other networks, and this prevents vehicles from reaching any internet resources for any data processing.

1.1.3 Hybrid Model

The hybrid model combines both the ad-hoc and infrastructure models, vehicles may communicate directly with each other as well as with infrastructure such as cellular towers or RSUs. The hybrid model benefits from all of the strengths of the ad-hoc and

infrastructure models while minimizing their weaknesses. Vehicles may dynamically react to failures in the network, routing around issues while maintaining a link to outside networks allowing them to access data on the Internet.

1.2 VANET Characteristics

Initially VANETs were seen as a special case of mobile ad-hoc networks (MANET) [13] however past simulations and studies have shown that protocols developed for MANETs do not perform well in VANET environments [27]. The unique characteristics of VANETs pose significant challenges to MANET protocols [27]. These characteristics include the following.

1.2.1 Highly Dynamic Topology

Vehicles move at highly variable speeds ranging from 0-25 m/s causing the network topology to constantly change. Given a highway environment and two vehicles traveling in opposite directions, assuming a radio range of 250m two vehicles will only be within radio range for maximum of 10 seconds. Vehicles may quickly join and leave local networks in very short periods of time [13, 28, 33, 27].

1.2.2 Frequently Disconnected Network

Due to the highly dynamic topology links between nodes frequently appear and disappear, in extreme cases links may appear and disappear in the time taken for nodes to recognize those links. Frequently disconnected networks are particularly problematic on sparse networks where multiple paths cannot be found between a source and destination. The effects of frequently disconnected networks are further exacerbated based on time dependent variables. Vehicular density greatly varies throughout the day, during periods of high density multiple paths may be formed between networks reducing the probability of a network becoming disconnected. During periods of low vehicular density it is unlikely that multiple paths may form between networks, if a link between two nodes is broken it is much less probable that a secondary link exists to bridge the now disconnected nodes [13, 28, 33, 27].

1.2.3 Constrained Mobility and Mobility Prediction

Vehicles are constrained to specific mobility patterns stemming from the underlying infrastructure. Infrastructure such as roads, traffic lights, speed limits, traffic conditions and driver behavior make it feasible to accurately predict future vehicle positions. Accurate simulation results must include realistic trace sources; traces may be generated through trace generators or by collecting mobility traces from vehicles [13, 28].

1.2.4 Unlimited Power and Storage

Nodes in VANETs are vehicles with ample battery power and storage capacity. While battery power and storage are not truly unlimited, when compared to small MANET nodes, VANET nodes essentially contain unlimited storage and battery power. Consequently, power and storage conservation is less of a concern in VANET networks [13, 28, 36].

As a consequence of more battery power being available to VANET nodes and hardware cost being less of an issue, VANET nodes are also assumed to have more powerful processors than those in MANET nodes [36].

1.2.5 Availability of On-board Sensors and Location data

Protocols designed for VANETs are assumed to have full access to all sensors in the vehicle. Information such as the vehicles speed and GPS location may be utilized ITS systems [13, 28, 33, 27].

1.2.6 Hard Delay Constraints

Particularly important during emergency situations, when emergency messages are sent by nodes they must be given the highest transmission priority and be processed immediately. While the goal of achieving high data rates is desirable in VANETs it should not

be the only goal of safety protocols developed for VANETs [25].

1.3 Applications of VANETs

As mentioned earlier VANET applications may be broadly divided into three categories: safety, efficiency and comfort.

1.3.1 Safety Applications

Safety applications primarily attempt to increase the safety of all drivers on the road through communication with vehicles in the local neighborhood, ultimately decreasing the number of collisions on the road. In 2008, Canada ranked 10th out of the 13 countries part of the Organization for Economic Cooperation and Development in terms of fatalities per billion kilometers traveled with 7.18 fatalities per billion kilometers driven [9]. Globally, approximately 1.2 million people are killed each year in vehicular collisions, the majority of which are between 5 and 44 years of age [9]. The cost of these collisions is approximated to be between two and three percent of a countries GDP [9].

VANET safety applications may be used to reduce the number of collisions and ultimately reduce the number of fatalities behind the wheel; VANET safety applications include the following.

1.3.1.1 Real-time Traffic Monitoring

Traffic data may be stored and by both VANET nodes and RSUs; that data may then be queried in real time by other nodes. Using real-time data, vehicles may avoid traffic jams, congestion and accidents [25]. Further utilizing real-time traffic data emergency services may monitor traffic data and react proactively to road-side issues [13].

An example of a system utilization real-time data to increase safety may be seen in [41]. In this system vehicles exchange period messages to allow for queries of the nearest k neighbors around a vehicle. Using this query vehicles may provide drivers with collision warnings near the vehicles vicinity, lane change warnings if an intended lane change may cause a collision and finally pre-crash sensing; warning vehicles and drivers of imminent and unavoidable collisions [41].

1.3.1.2 Post Crash Notification

Once a crash has occurred there are two issues that must be addressed; the collision itself as well as vehicles around the collision [13]. Vehicles involved in a collision may broadcast warning messages allowing vehicles in the vicinity to approach with caution as well as alerting authorities of an emergency. Post crash notification messages may alert the drivers of upcoming issues or may trigger vehicles to autonomously react to upcoming issues.

An example of post crash notification increasing safety is seen in [7]. In this paper authors seek to reduce the chances of chain collisions in highway settings. Chain collisions occur when a collision occurs and other approaching vehicles do not have enough time to avoid the initial collision. Using V2V communication vehicles approaching a collision are notified of the upcoming collision, reducing chain collisions by up to 48%.

1.3.1.3 Hazard Notification

Vehicles may broadcast warnings of potential hazards to other vehicles traveling behind it [25]. Hazards such as landslides, icy roads or dangerous road conditions may sent to other vehicles notifying them of the potential danger.

1.3.2 Efficiency Applications

Efficiency applications aim to improve road utilization and vehicle mobility. Efficiency applications provide the driver with information about road conditions such as congestion, alternative routes and parking availability, allowing drivers to make informed decisions [13]. Efficiency applications may be particularly helpful in spreading out congestion among multiple routes, thus eliminating or reducing bottleneck effects on roads.

One example of such an application may be seen in the StreetSmart application seen in [14]. When vehicles encounter road conditions outside of the expected conditions they generate informational messages notifying other vehicles in the network of the ob-

served conditions. Each node independently forms clusters based on the data, modeling observed road conditions. Using the models, vehicles may the estimate upcoming road conditions allowing drivers to react to the conditions before encountering them.

A second example of efficiency application may be seen in [17], a congestion avoidance system. Following the standardized 802.11p protocol vehicles regularly transmit *Basic Safety Messages* (BSM) messages to all neighbors; neighbors are not required to receive each transmitted message but must receive and process a subset of sent messages. When vehicles approach major junctions where two or more possible routes exist, vehicles send out a congestion request message listing the roads of interest; roads which may be used to reach the final destination. Vehicles respond to the request message with a congestion response message containing observed congestion information on the roads of interest. Based on the response to the congestion request message vehicles may choose the best route.

1.3.3 Comfort Applications

Comfort applications are a broad category of applications encompassing applications which do not improve safety or increase road utilization. Comfort applications make driving more comfortable and enjoyable for drivers and passengers. Examples of comfort applications include internet access, social networking, tourist information, advertisements, leisure information and file sharing [13].

An example of a comfort application may be seen in [26] where P2P file-sharing is used to exchange files in a VANET environment. Files are shared between vehicles and RSUs. In trials using 2 vehicles and a RSU along a 1 KM stretch of road, 25MB files are successfully shared between the vehicles and the RSU.

1.4 Motivation for this Thesis

The motivation for this thesis comes from the following four areas:

Lack of Layer 3 Standardization for V2V communication in VANETs Standardization efforts in VANETs have successfully created standards for the physical layer (layer 1) and data link (layer 2) of the network model in the form of 802.11p and WAVE; these standard are described in Sections 2.2 and 2.3. Current standardization efforts in network layer (layer 3) technologies have succeeded in standardizing the V2I model of communication however layer 3 in the ad-hoc and hybrid communication model is still an open research area. This lack of standardization has particularly been an issue in bringing internet access to VANET nodes using the ad-hoc and hybrid models of communication [23].

Lack of Holistic Views of VANETs Much of the existing research on VANET routing protocols focuses on the specific metrics in specific situation. Metrics such as delays, average hop counts and throughput are calculated based on a given envi-

ronment. While these metrics are essential to evaluating a protocol, protocols must also be evaluated in the presence of multiple types of traffic. VANETs will eventually contain network traffic in the form of multicasts, unicasts and any-casts; a routing protocol must be able to effectively deal with each of these forms of traffic while maintaining quality of service.

Lack of Study in Hybrid Communication Model Enabling Internet Access Internet access in VANETs has been largely un-addressed in a great deal of research [23]; much of the current research focuses solely on V2V traffic while industry study has focused on V2I traffic.

Neighbor Discovery Much of the current evaluation of VANET unicast routing protocols evaluate the protocols by sending multiple streams of data between *n* pairs of nodes. This evaluation determines the performance of the routing protocol however it leaves a question unanswered; how do the vehicles know of their peers existence? Peer nodes must be *k* hops apart where $k \geq 2$, otherwise communication is trivial from a routing perspective, however for $k \geq 2$ a routing protocol must be present to complete multiple hops to discover the peer. This leads to a circular problem, in order to route between two nodes the existence and position of the nodes must be known; however to discover the existence of the nodes a routing protocol is required. Much of the reviewed literature assumes the existence of a

system to discover distant neighbors, however this leads to a similar circular issue. Routing protocols must know of the existence and position of a neighbor discovery system but to learn of the system the routing protocols must already have access to the location system.

1.5 Contributions

This thesis will present contributions in the following areas.

IPv6 Geographic Addressing: We will present an IPv6 addressing strategy building on the IPv6 addressing strategy standardized for the V2I communication model. Our strategy will extend the strategy in two areas, firstly the proposed addressing strategy will no longer depend on infrastructure and secondly the proposed strategy will encode geographic position information within an IPv6 address.

Contention Based Hybrid Routing: We will present a contention based routing protocol enabling both V2V and V2I communication. Using geographic addressing, a structured overlay and contention based addressing the proposed routing protocol will enable safety, efficiency and comfort applications all within a single protocol.

Holistic Analysis: Due to the muli-traffic support of the proposed routing protocol our analysis will examine protocol performance while managing multiple forms of

traffic. Holistic analysis will examine VANETs as an entire system on a macro scale rather then examining singular issues on a micro scale.

Neighbor Discovery and Bootstrapping: Based on the proposed geographic routing strategy and a structured overlay this thesis will propose methods provide VANET nodes with bootstrap or anchoring information in order facilitate initial communication. Using the stored information the proposal will lay out a framework for neighbor discovery and hybrid internet access.

VANET Simulation and Development Moving away from protocol design this thesis presents an analysis of VANET simulation and development. Three areas of VANET simulation are evaluated, network simulation, mobility simulation and VANET network creation. The evaluations performed provide future researchers with a framework for choosing a simulation platform for their development work.

1.6 Organization

The rest of the thesis is organized as follows:

Chapter 2 presents an overview of TCP/IP, an overview of the standardized 802.11p and WAVE protocols, a literature survey of existing MANET and VANET routing protocols and a review of IP addressing in VANETs.

Chapter 3 presents a detailed explanation of our proposed VANET routing protocol and VANET layer 3 extensions.

Chapter 4 presents an overview of the VANET simulation, reviewing the importance of VANET simulators, mobility simulation, and simulation platforms. Upon reviewing VANET simulation platforms the rational behind choosing a particular simulator is given. Finally a detailed description of one of the central implementation challenges encountered is given followed by a description of the applied solution.

Chapter 5 presents a performance evaluation and discussion of the ideas proposed in this thesis. Performance is evaluated in multiple environments with multiple vehicular densities. The performance of ideas presented in this thesis are compared to multiple well established protocols to provide readers with a point of reference.

Chapter 6 presents a summary of the thesis, the conclusions which may be drawn from the performance evaluation and ideas for future research areas. Two primary areas of future work as discussed, local neighborhood estimation and cross layer optimization. Both suggested research areas are suggested to enhance the performance of the proposed protocol based off of evidence gathered during performance evaluation as well as through logging on node conditions during simulations.

2 Network Architecture Review

In this chapter a brief review of existing research is given. First a description of the standard network model is given. Next the first three layers of the networking model are described as pertaining to VANETs beginning with the physical layer 802.11p standardization, followed by the WAVE MAC protocol and finally the WAVE network management protocol. Next existing routing protocols for MANETs and VANETs are reviewed. Finally node addressing in VANETs is reviewed.

2.1 Network Protocol Stack Architecture

Classical network architecture sees the network stack split in distinct layers to maintain separation of concerns. Two network models are commonly referenced in literature; the OSI model and the TCP/IP or Internet model. Both models describe similar processes however the OSI model references seven layers while the TCP/IP model references four layers; both models are shown in Figure 2.1. Each layer is independent of the layers above and below it, the only interaction required between layers is an interface to accept

data from the layer above or to send data to the layer below. Upon passing data down through the network stack each layer adds a layer of encapsulation and passes the data to the layer below, upon receiving data from an upper layer the receiving layer treats the incoming data as a black box; processing is done independent of the contents of the data. Conversely as data is passed upwards in the network stack, layers of encapsulation are stripped off at each layer and processed independently. Due to the finer grain layers of the OSI model, this thesis will refer to network layers in the OSI model however any descriptions given will also be applicable in the Internet model.

Networking functions such as routing, medium access and packet transmission is handled by the bottom three layers while the top four layers are associated with user level functions. The descriptions given in this thesis will focus on the bottom three layers as these are the primary layers used in the thesis.

Physical Layer The physical layer is responsible for transmitting and receiving electrical signals. The physical layer converts the bits received from the above layer into an electrical signal. The physical layer also converts received electrical signals from the medium into bits to pass to higher layers. Issues such as modulation, frequencies usage and connectors are defined at the physical layer.

Data link Layer The data link layer may be broken up into two sub layers; the logical link control (LLC) and media access control (MAC) sub-layers. The LLC layer serves

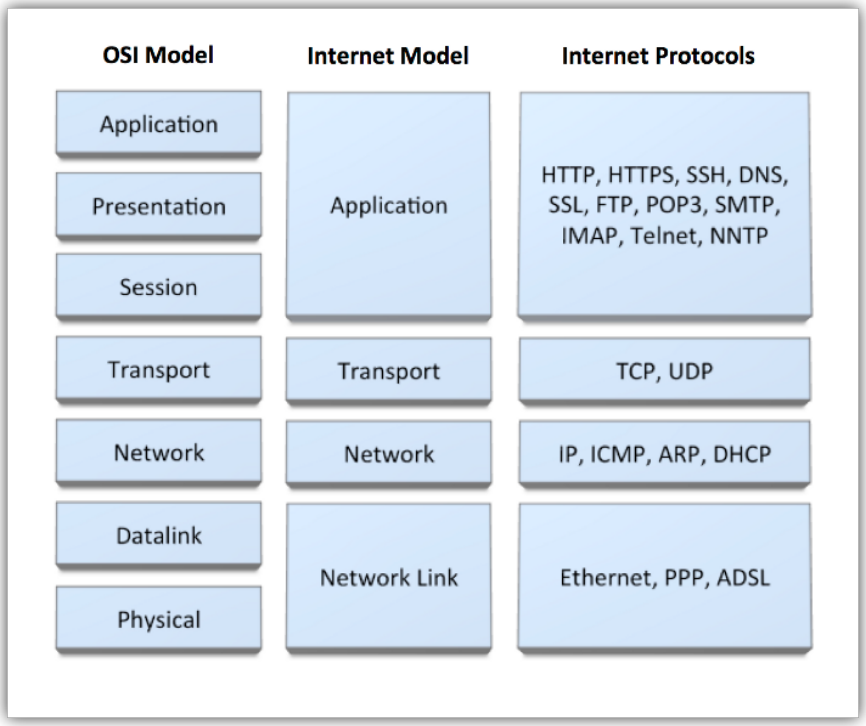

Figure 2.1: Network Stack Architecture

[19]

as the interface between the data link layer and upper layers and provides control functions such as flow control, acknowledgments and error checking. The MAC sub-layer implements MAC protocols such as CSMA/CD and CSMA/CA which control node access to the media. MAC protocols are particularly vital in the wireless medium, due to the broadcast nature of the wireless medium all nodes within radio range may detect a transmission.

Wireless transmission detection falls under one of three categories based on the re-

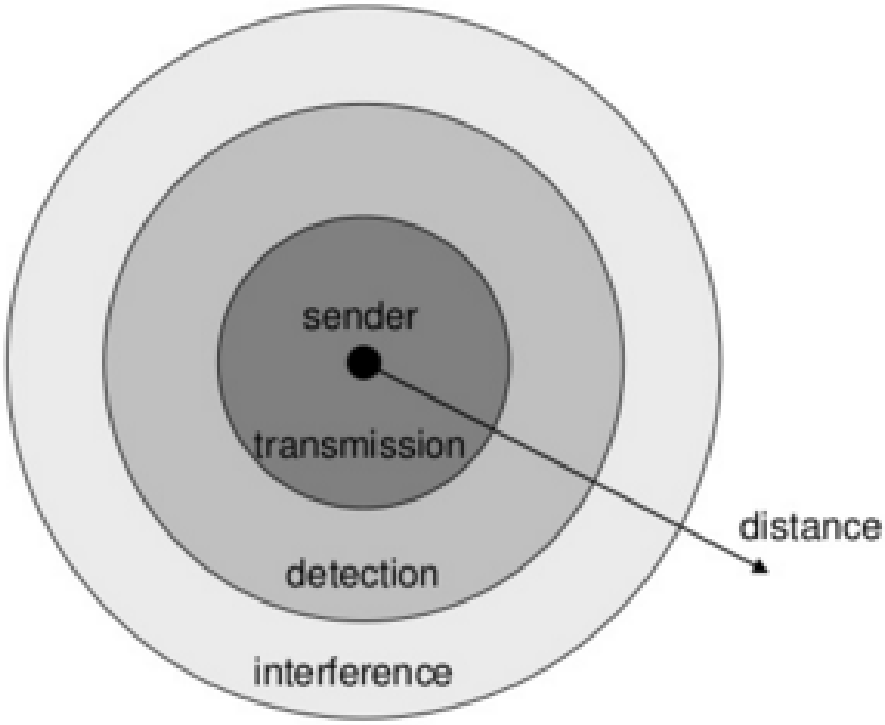

Figure 2.2: Transmission Detection

[38]

ceivers position with respect to the sender, the three categories are shown in Figure 2.2. Nodes within the transmission range may reliably detect and receive transmissions given an effective MAC protocol has scheduled the transmission during a period where no other nodes are simultaneously transmitting. Nodes within detection range will detect a transmission is occurring however they will not reliably receive the transmission correctly. Finally nodes within the interference range neither detect or receive the signal; transmissions appear as background noise[38]. Readers should be aware abstract nature

of Figure 2.2, existing literature commonly transmission detection at this level of abstraction. Abstract diagrams of this nature assume a perfect omni-directional antenna and well defined borders between each detection category, in reality these features less strictly defined, readers interested in further information may find a more detailed explanation of wireless signals in [8].

Network Layer The network layer primarily manages two network functions; logical addressing in the form of an IP address and packet routing. IP addresses are in the form of 32 bit IPv4 addresses or 128 bit IPv6 addresses; each interface with a network layer requires at least one IP address although an interface may contain more than one address. At the time of writing this thesis IPv4 is the dominant logical addressing scheme used around the world, however due to IPv4 address exhaustion IPv6 is slowly becoming the global standard. IPv6 has been slow to become globally adopted, as of April 2016 only 1.6% of all Internet traffic is IPv6. One of the major reasons IPv6 as not been adopted at a faster rate is continued technological innovation preventing address exhaustion from becoming a widespread issue halting the growth of the Internet. One of the main technological innovations is Network Address Translation (NAT), NAT is a many to one translation system translating many user addresses to a single internet routable address. Through the use of address translation an entire household or office may use a single Internet routable IPv4 address instead of a single Internet routable address for

each device.

The encapsulation performed by the network layer is accomplished via the addition of a IP header. Figure 2.3 shows the standard 20 byte IPv4 header format. This thesis will not cover each of the header fields however the following fields are worth mentioning as they are actively used in subsequent sections.

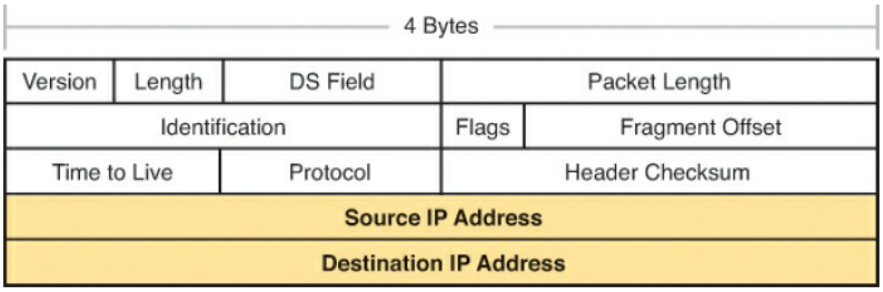

Figure 2.3: IPv4 Header Format

[32]

Time To Live The time to live field (TTL) determines how many times an IP packet may be forwarded. Each time a packet is forwarded the field is decremented by one. Once the field reaches zero the packet is dropped, this prevents packets from being forwarded indefinitely.

IPv4 Source Address The source address defines the IPv4 address of the node which created the packet. While routers may choose to de-encapsulate and modify parts of the IPv4 header, the source address may not be modified at any point in the

routing process except during the NAT process; if it is in use.

IPv4 Destination Address The destination address defines the IPv4 address of the destination node. Similar to the source address, the destination address cannot be modified during the routing process unless NAT is in use.

Figure 2.4 shows the standard 40 byte IPv6 header format. Due to the larger addressing scheme the IPv6 header is naturally larger than the IPv4 header. Additionally the IPv6 header fields are simplified when compared to those in the IPv4 header. Once again only relevant fields of the IPv6 header will be discussed.

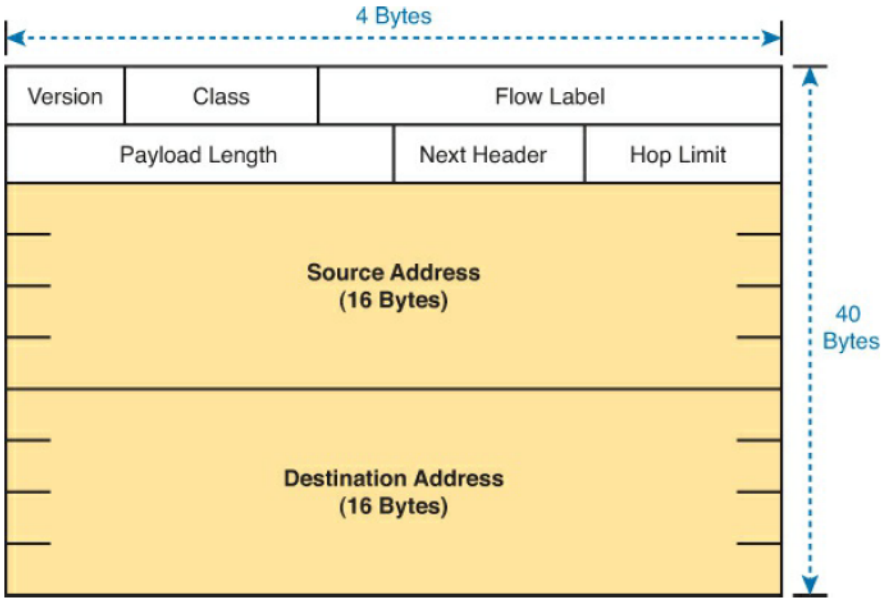

Figure 2.4: IPv6 Header Format

[32]

Time To Live Similar to the IPv4 header the TTL field defines the number of times a packet may be forwarded. Once the TTL field reaches 0 the packet is dropped, this prevents indefinite forwarding of a packet.

IPv6 Source Address Similar to the IPv4 header the source address field defines the IPv6 address of the source node. Due to the abundance of addresses in the IPv6 address space the usage of NAT with IPv6 is continually debated. Consequently the source address may be the address of the original sending node or of a NAT router.

IPv6 Destination Address Similar to the IPv6 header the destination address field defines the final destination of the packet. Once again, due to the debate concerning NAT in IPv6 the destination address may be a NAT router or the actual destination.

2.2 VANET Physical Layer Standardization: 802.11p

The current standard for the physical layer in VANETs is 802.11p, officially standardized in 2007 by IEEE [29]. The 802.11p standard is based member of the classical 802.11 family of protocols, specifically the standard is based off of the 802.11a WiFi standard [39]. The 802.11p standard operates at the 5.9 GHz frequency band; from 5.850 - 5.925 GHZ. The first five MHz of the frequency band is reserved as a guard band, the remainder of the band is divided up into seven 10MHz channels [5], Figure 2.5 shows the 802.11p

channels. Channels are divided into one of three roles; service (SCH), control (CCH) and safety channels.

Channel 178 - the control channel is reserved for controlling transmissions between nodes. Channel 178 is used to control broadcasting on all other channels and for controlling link establishment [5].

Channels 172 and 184 are reserved for safety applications. Delving further into the specifics of each channel; channel 172 is reserved for critical messages vital to preserving life. Channel 184 is reserved for public wide scale safety, this is a higher power channel, theoretically it could be used to reach vehicles up to 1 KM away from the transmitter [39].

The remaining channels 174,176,180 and 182 are used as services channels, these channels may be used for either safety or non-safety messages. Channels 174 and 176 and channels 180 and 182 may be combined into two 20 MHz channels; channels 175 and 181 doubling the bandwidth of a single channel [5].

Abiding by the 802.11 specification, 802.11p utilizes Orthogonal Frequency Division Multiplexing (OFDM) when sending data. 64 orthogonal sub-carriers are defined however only 52 sub-carriers are actually used, the 52 sub-carriers are numbered from -26 to 26 [5]. The 52 sub-carriers are then further divided into 48 data and 4 pilot sub-carriers. Pilot sub-carriers are well know carriers using a set modulation scheme used for synchronization and channel estimation, data sub-carriers are used for actual data transmission,

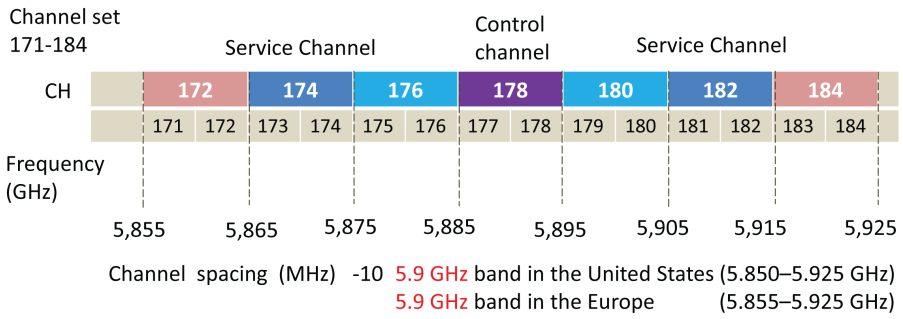

Figure 2.5: 802.11p Spectrum

[37]

they may change modulations between each data burst depending on channel conditions.

Table 2.1 shows a comparison between 802.11a and 802.11p.

2.3 WAVE MAC Protocol: Data Link Layer Standardization

The VANET MAC protocol is standardized in IEEE 1609.4 (WAVE MAC), this standardization defines how a single radio should support concurrent safety and non-safety communication when employing the multi-channel scheme in section 2.2 [10]. IEEE 1609.4 divides time into control channel (CCH) and service channel (SCH) intervals, each of which is 50 ms long [10]. Combining the CCH and SCH interval forms a *Sync Period*; each second contains ten sync intervals corresponding to a 10 Hz sending rate of basic safety messages (BSM). The start of a CCH interval begins at the start of a second synchronized to UTC. This necessitates all nodes to have a way to synchronize

Table 2.1: Comparison between 802.11p and 802.11a

[29]

Parameters	802.11p	802.11a
Frequency	5.9 GHz	5/2.4 GHz
Bandwidth	10 MHz	20 MHz
Data Rates	3,4.5,6,8,12,18,24,27	6,9,12,18,24,36,48,54
Modulation	BPSK, QPSK, 16QAM, 64QAM	BPSK, QPSK, 16QAM, 64QAM
Channel Coding	1/2, 2/3, 3/4	1/2, 2/3, 3/4
Data Sub-carriers	48	48
Pilot Sub-carriers	4	4

their clocks to the beginning of a second. Most commonly time synchronization is accomplished through GPS however clocks may also be synchronized based on communication with other nodes. Due to possible errors in synchronization as well as delays in radio switching the beginning of each CCH and SCH interval is reserved as a guard period. During the guard period nodes refrain from transmitting. Guard period time is determined by two fields in the WAVE standard; *SyncTolerance* and *MaxChSwitchTime*. Sync tolerance is defined by a nodes tolerance to time synchronization errors while MaxChSwitchTime defines the maximum time a radio may be unable to transmit or receive due to channel switching [3]. Figure 2.6 shows the makeup of a guard period, throughout the two sync tolerance periods nodes may continue receiving transmissions but they may not start transmissions however once entering the switching period, a node may neither receive or send transmissions.

Due to the multi-channel nature of the VANET physical standard the MAC protocol must define channel usage. The WAVE standard defines two types of channel usage; continuous and alternating channel access. Continuous channel access sees a node staying on a pre-configured single channel, either a safety channel or the control channel [3]. Using an alternating channel access scheme nodes begin on the control channel, monitoring the channel for service advertisements. Once a service of interest has been found nodes may switch to the advertised service channel immediately or they may switch at the beginning of the next time slot[3].

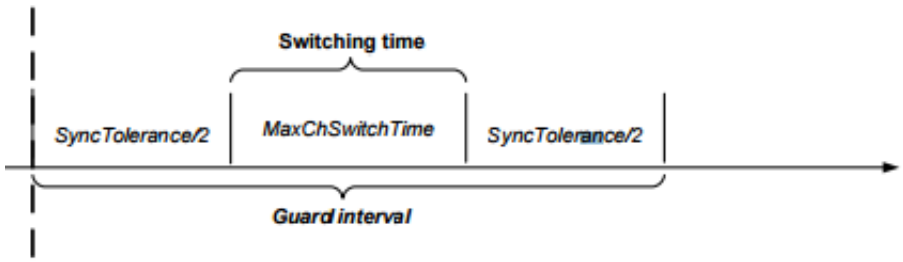

Figure 2.6: Guard Period Specification

[3]

Channel Access Channel access in VANETs is adopted from the 802.11e standard *Enhanced Distributed Channel Access* EDCA [3], designed for contention based prioritized QoS [18]. The EDCA standard builds on top of the existing 802.11 standard of contention window, back off and inter-frame timers seen in CSMA/CA. VANET MAC uses the following inter-frame spaces (IFS):

Short Inter frame Space SIFS The SIFS period is the shortest of all inter-frame spaces. Being the shortest inter-frame sequence nodes waiting SIFS to send are granted the highest sending priority. SIFS is used for signaling once a transmission has begun; the most common case usage of SIFS is to gain access to the channel to send an acknowledgement message.

Point Coordination Function Space PIFS The PIFS is the next shortest inter-frame period after the SIFS. The PIFS is used exclusively by access points while running the point coordination function. The shorter PIFS period gives access points

Table 2.2: AIFS In VANETs

AC	CWmin	CWmax	AIFS[AC]
AC_BK (Background)	15	1023	9
AC_BE (Best Effort)	7	1023	6
AC_VI (Video)	3	1023	3
AC_VO (Voice)	3	7	2

priority over a channel before nodes are able to contend.

Distributed Coordination Function DIFS The DIFS is the next shortest inter-frame period, longer than both SIFS and PIFS. Non-VANET nodes must wait a DIFS period after the channel has become free to begin contending for the channel, however VANET nodes utilizing EDCA have replaced DIFS with an updated contention mechanism allowing for QoS [21].

Arbitration Interframe Space AIFS The AIFS period is a variable size window depending on the priority of a packet. Packets are assigned an *Access Category* AC based on their priority. EDCA allows for four packet priorities, each packet is assigned to a queue based on its priority [21]. Each AC contains a distinct inter-frame period, minimum contention window length and maximum contention length. Table 2.2 details the AIFS spacing and contention window for each queue.

In order to access the channel a node proceeds through the following steps:

Listen to channel Nodes begin by listening to the channel. If the channel has been free for the AIFS period for the packet AC as given in 2.2 then the node may transmit immediately. If another node does begin transmitting before the AIFS period expires the node must perform a backoff operation [6].

Backoff The backoff procedure is to choose a random integer uniformly distributed between 0 and CW; the contention window. The drawn value is then multiplied by the time of a single slot to derive a backoff timer. While the channel is free the backoff timer is decremented, once the timer reaches 0 the node is free to transmit immediately [6].

Stop and Wait CSMA/CA is a stop-and-wait protocol, this indicates that after transmitting a packet the node must pause to receive and acknowledgement (ACK) from the indented receiver. If the ACK is received the transmission is successful and the node may proceed to the next transmission. If an ACK is not received then either the ACK was lost or the original transmission was lost; regardless of the loss case the packet must be retransmitted [6].

Retransmission Procedure Before the packet may be re-transmitted the backoff procedure must first be completed. Each time a transmission fails the CW is doubled up to CWmax or until the re-transmission limit has been reached and the transmission

is abandoned. Using the new CW limit a value is drawn to create a new backoff timer. Once a transmission has been successfully received the CW is reset to the minimum value once again [6].

2.4 WAVE Network Management

The VANET network layer is standardized in IEEE 1609.0 and 1609.3 and are presented in [2] and [1]. The 1609.3 standard primarily focuses on setting up the architecture for inter-layer communication, WAVE message transmission and IP addressing. The standard does an excellent job of building the framework for the VANET network layer however much of the standard focuses solely on the V2I communication model. As a result research in VANET layer three technologies using the V2V and hybrid communication models continues to be an active area.

IPv6 in VANETs IEEE 1609.3 mandates that IPv6 must be implemented all VANET architectures [1]. Infrastructure addressing may be statically assigned by the network administrator. Mobile node addressing may be addressed statically or if supported, dynamically. Dynamic addresses must be calculated via *Stateless Address Auto Configuration* (SLAAC). SLAAC is covered further in depth in 3, at a high level SLAAC creates unique IPv6 addresses based on a 64 bit IP prefix and the interface MAC address. The MAC address for each interface is known to the node however the IP prefix must be learned from

the nearest router. Wired interfaces learn the IP prefix through multicast router solicitation messages and router advertisements. IEEE 1609.3 learns of the IP prefix through *WaveRoutingAdvertisement* messages.

WAVE Routing Routing in the 1609.3 standard primarily follows the V2I routing model; nodes communicate directly with RSU infrastructure. Nodes within a single hop may follow the V2V communication model when sending safety messages, safety messages are broadcast and received by all nodes within transmission range. WAVE routing information is disseminated via *WaveRoutingAdvertisement* messages, these messages have the following format [1]:

- Router Lifetime The router lifetime is a 16 bit value which defines the duration for which the received information is valid.
- IP Prefix The IP prefix is a 128 bit IPv6 address, using the IP prefix nodes may set their IP address to the RSU network.
- Prefix length The prefix length is an 8 bit value which defines the number of significant bits in the IP prefix. The significant bits define with network with the remaining bits equal to the host portion of an IP address.
- Default Gateway The default gateway field is a 128 bit value which defined the IP address of the RSU. The RSU serves as a gateway connecting the VANET network

to other rout-able networks.

- DNS Address The DNS address is a 128 bit value defining the IPv6 address which VANET nodes may use for DNS lookups.

Using the information contained within *WaveRoutingAdvertisement* VANET nodes contain sufficient information to set their IPv6 address and communicate directly with RSUs.

2.5 VANET Literature Review

Routing in VANETs may be broadly divided up into 5 major categories; topology based and geographic routing. Topology, hybrid, clustering, geographic and data fusion. Figure 2.7 shows a variety of VANET routing protocols and their classifications.

Topology Based Routing Topology based routing uses information based on links in the network to form routes. Route selection depends on knowledge of links forming a complete path between the source and destination. Topology based routing may be further divided into reactive and proactive protocols.

Reactive Reactive routing protocols discover routes based on demand. Reactive protocols flood the network with route discovery requests which are subsequently forwarded by their neighbors until the destination is reached or until

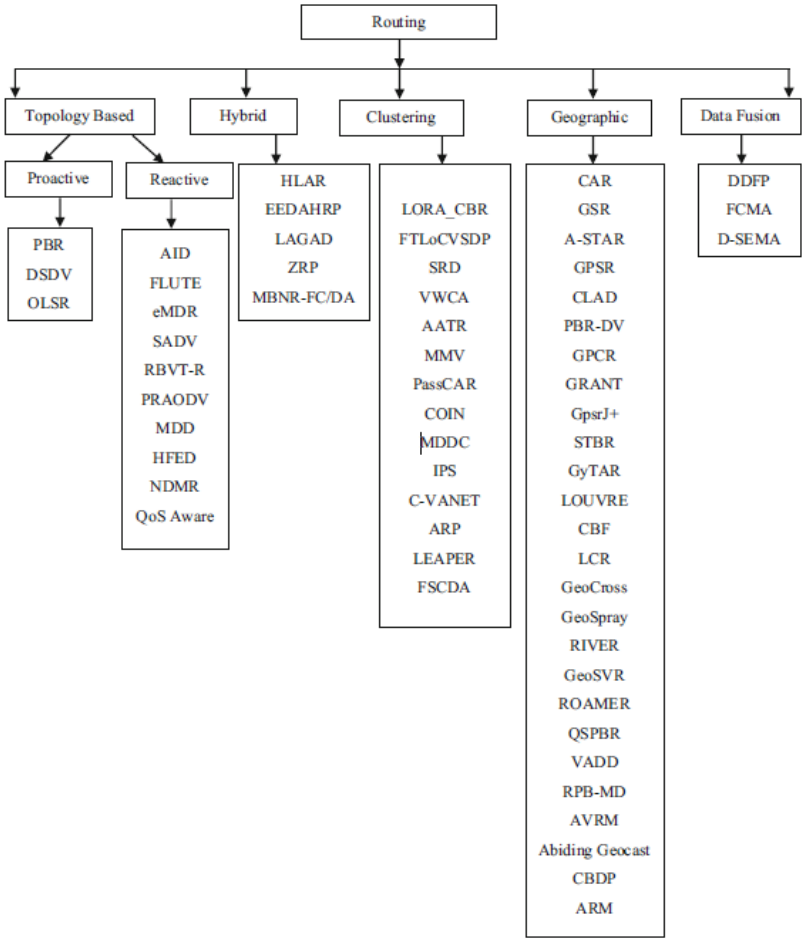

Figure 2.7: Routing Models

[23]

the request is dropped. The path to the destination is then returned to the originating node which may then send the packet.

Proactive Proactive routing protocols keep an up to date routing table which is maintained through regular updates. Should any change occur in the network,

nodes aware of the change will broadcast a message notifying the rest of the network of the change [23].

Hybrid Hybrid routing is a second class of topology based routing protocols which combines both proactive and reactive routing concepts. Routes are initially created using proactive routing principles, later routes are established using proactive routing techniques such as control message flooding [23].

Cluster Routing Cluster routing divides the topology into clusters based on node density in the topology. Each cluster than elects a cluster head (CH) which controls message transmission among nodes in the cluster. The CH is analogous to a hub in the hub-and-spoke LAN network topology. All nodes in the cluster must be connected to the CH and any messages sent in the cluster must traverse through the cluster header. When the CH leaves the cluster a new CH election must take place, transferring all control functions to the new CH [23].

Geographic Routing Geographic routing performs routing decisions based on sender and destination positions. Unlike topology based routing protocols, geographic protocols have no prior knowledge of the topology and do not perform route discovery. Geographic protocols assume all nodes are able to determine their own position as well as the destination position [23]. Source positions may be obtained through GPS data, however an additional system must be implemented to learn of

the destination position. While the source position must be as exact as possible some geographic routing protocols may use a fuzzy destination position. Fuzzy destinations may be given in the form of a *Zone of Relevance* (ZoR); a larger zone where the destination may be is given rather than a strict coordinate. As geographic routing protocols do not exchange link state information, they are extremely resilient to rapidly changing network topologies; this has made geographic routing a promising class of routing protocols for VANETs [28].

Data Fusion Data fusion protocols are a class of routing protocols which reside on top of other routing protocols. Data fusion protocols attempt to combine and reduce data from multiple sources into just the data of interest. The reduced data is then disseminated reducing the total amount of data sent in the network [23].

2.6 MANET Routing Protocols

MANET protocols largely fall under topology based protocols and serve as the foundation of VANET routing protocols. MANET protocols have been extensively tested in VANET environments resulting in mixed results however the general consensus of the literature is that MANET protocols do not perform well in VANET environments. In an attempt to reuse existing work multiple attempts have been made to adapt MANET protocols with varied success; a sample of these attempts will be shown in subsequent

sections.

2.6.1 Ad-hoc On-Demand Distance Vector Routing (AODV)

AODV is an extremely popular MANET routing protocol frequently evaluated in both MANET and VANET environments. The original AODV protocol was presented in [35] more than 17 years ago and is still actively used in evaluations today. AODV is a reactive topology based protocol, as such it does not maintain an active view of the topology but builds a route each time it is needed.

Routing Table AODV maintains a routing table of active routes, each routing entry is composed of:

- Destination Address: The IP address of the destination node.
- Next Hop Address: The IP address of the next hop in the routing process.
- Sequence Number: A unique number for each route, the sequence number is used to distinguish routes.
- Life time: The remaining route lifetime determines how long a route will remain in the routing table before being removed. Each time the route is used the route life time is reset back to the maximum value.

When sending a packet a node must check its routing table for a valid route to the

destination. If a valid route is found, the packet is unicast to the next hop node. If a route is not found, the node begins the route discovery process [35].

Route Discovery Route discovery is managed through two types of control messages; route request (RREQ) and route reply (RREP) messages. Nodes maintain two counters, the *sequence number* and *broadcast ID*. To discover a route a node creates a new RREQ packet containing the source IP address, sequence number, broadcast ID, destination address, destination sequence number and hop count[35]. The node then broadcasts the RREQ to all of its neighbors. Each time a RREQ packet is sent the broadcast ID counter is incremented. The pair of source address and broadcast ID uniquely identify a RREQ.

Neighbors receiving the RREQ check the source address and broadcast ID pair against previously received RREQ, if the pair is found the packet is labeled as duplicate and dropped. Duplicate RREQ may be received due to the broadcast nature of the wireless medium. If the RREQ passes the duplicate check the intermediate node processes the RREQ packet. Processing the packet takes place in one of two ways; firstly intermediate node may send a RREP if it is the intended destination or if it already contains an active route to the destination more recent than the senders query or secondly the node may rebroadcast the RREQ.

If the intermediate node cannot reply with a RREP message it must also send a

RREQ. Before a new RREQ is sent the intermediate node records the destination IP address, source IP address, broadcast ID, expiration time and source sequence number. These fields are used to relay a response back to the source node.

The route discovery process is repeated until the the RREQ has reached its traversal limits in which case it is dropped or the RREQ reaches the destination. Once the destination is reached the route reply process begins.

Route Reply A RREP message is sent back to the originating node once a valid route has been found. The destination node creates a new RREP packet containing the source address, destination address, destination sequence number, hop count and lifetime. The RREP back is routed back to the source node through unicast routing based on the information stored when transmitting RREQ packets. While routing the RREP packet back to the original source each node updates with RREP packet with an incremented hop count, records or updates its own route to the destination and updates sequence numbers to signify newer data [35].

Figure 2.8 shows a simplified view of the RREQ and RREP process in AODV.

Route Errors Due to the possibility of links between nodes degrading into unidirectional links or nodes moving entirely outside of each others transmission range AODV monitors the reach-ability of its next hops. Link status may be monitored proactively or reactively [35]. Proactive monitoring is managed through the pe-

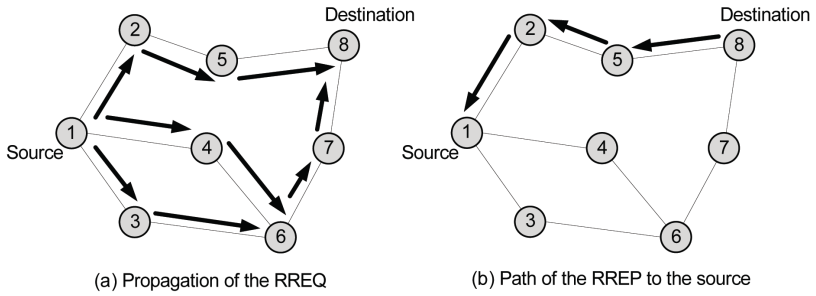

Figure 2.8: AODV Route Discovery

[27]

riodic exchange of *hello* messages. AODV nodes may miss k consecutive hello messages before the node recognizes a link failure; the k value may be configured on a per node basis. Alternatively link failures may be detected based on errors in the data link layer acknowledgments [35]. When a link error is detected routes containing the failed link are removed from the routing table. When a downstream node sends a packet along a route with a failed link, the node closest to the failed link sends a route error notification (RERR) back to the source. Once the source receives a RERR it must find a new route using the route discovery mechanism.

2.6.2 Destination-Sequenced Distance-Vector Routing (DSDV)

The DSDV routing protocol is one of the earliest examples of ad-hoc routing protocols created for MANETS. Like AODV, DSDV is a topology driven protocol, however unlike AODV DSDV is a proactive routing protocol [34] which regularly exchanges link state

updates. Falling under the class of distance vector routing protocols each node maintains a distance measure to each destination rather than maintaining a complete view of the entire topology[32].

DSDV DSDV continues to use the Bellman-Ford algorithm which enables uses hop count as a metric to build routes. The Bellman-Ford algorithm is well suiting in routing protocols designed for wired networks where links rarely fail. Frequent failures of links may cause routing loops to form when using the Bellman-Ford algorithm making it unsuitable for MANET or VANET networks [34]. DSDV augments the Bellman-Ford algorithm through the addition of sequence numbers to address the issues of routing loops forming due to frequently disconnected links, once again making it suitable for MANET environments. Sequence numbers add temporal information to routing information giving nodes the ability to distinguish between stale routes and new routes, this allows them to avoid creating routing loops [34].

DSDV Route Establishment DSDV learns and maintains routes through periodic broadcast messages. Each node maintains a vector containing destination addresses, distances and sequence numbers. Periodically each node sends a route update containing its address and an incremented sequence number as well as the nodes current routing table. DSDV defines two ways nodes may transmit their routing tables; incremental and full dump. Incremental updates only send changes from the previous full dump, reducing

the amount of redundant information transmitted in the network. Full dumps transmit the entire routing table [34]. Each node must independently determine which type of update to send, the authors in [34] suggest sending a full dump when the size of an incremental update reaches the limits of being contained in a single transmission, although exact details may vary by implementation. Each routing entry in both types of updates contains the destination address, the number of hops required to reach the destination and the destination sequence number received when the node learned about the destination. Choosing routes to a destination is based on a combination of sequence number and metric. Routes to destinations are built based on the update with the highest sequence number. If multiple updates with the same sequence number are received the node with the better metric is chosen. Similar to the route update structure DSDV routes are composed of the destination address, the number of hops reported and the sequence number.

Sequence Numbers Sequence numbers are a vital part of the DSDV protocol and are further examined here. Before a node sends an update it first generates a new sequence number by incrementing the old sequence number by two. As a consequence of this all *valid* routes have an even sequence number. When a route become invalid a node detecting the route failure increments the routes sequence number by one, thus all *invalid* routes contain odd sequence numbers [34].

DSDV Route Maintenance Routes are maintained via regular updates as described in the route establishment section. When invalid routes are detected DSDV employs a *route poisoning* mechanism. Route poisoning refers to sending route updates with a metric higher than the maximum possible value. DSDV accomplishes route poisoning through updating the sequence number to an invalid value as described in the DSDV sequence numbers and by setting the hop count to infinity. Nodes receiving the poisoned route detect the route has failed, invalidate the route in their routing tables and continue propagating the poisoned route[34].

2.6.3 Optimized Link State Routing Protocol for Ad Hoc Networks - (OLSR)

OLSR was released after both AODV and DSDV using lessons learned in both in an attempt to optimize MANET routing. OLSR is a link state proactive protocol, routing tables are actively maintained through the periodic exchange topology information at each node [22]. OLSR contains two primary optimizations, firstly the size of control messages is reduced by only declaring links with *multipoint relays* and secondly control packets are only forwarded by a nodes *multi-point relays* [22]. Due to the proactive link-state properties of OLSR the protocol must exchange periodic messages to maintain a view of the topology, OLSR refers to these messages as *hello* messages. Hello messages contain two fields, a list of neighbors with which the node has bi-directional links and a list of nodes with which a unidirectional link has been established but a bidirectional

link has not yet been verified [22]. Exchanging *hello* messages allows each node to learn of all nodes within two hops.

Multipoint Relays Multipoint relays are the primary mechanism through which OLSR optimizes routing in MANETs. Each node independently selects a set of nodes within its neighborhood to re-transmit its routing update messages. These set of nodes are referred to as the *multipoint relays* (MPRs) of a node. Nodes outside of the MPR continue to receive and process update messages however unlike MPR nodes, non-MPR nodes will not forward the message. In addition to maintaining the set of MPRs that will forward messages, each node also maintains a list of nodes for which it is responsible for forwarding routing updates, this list is referred to as the *MPR Selectors* of the node [22]. Both referenced lists may be changed at anytime; changes to MPRs are propagated through the list of bi-directional links contained within *hello* messages.

Multipoint Relay Selection MPR selection is performed by selecting the smallest set of nodes within one hop which may be used as routes to reach all nodes within two hops [22]. OLSR does not require that the MPR is optimal however the MPR should be kept as small as possible.

Route Selection Before routes may be selected nodes must build an *intra-forwarding* database. The intra-forwarding database is built through the exchange of *Topology Con-*

trol (TC) messages, these messages are broadcast messages forwarded throughout the entire network using MPRs. Each TC message contains the sending nodes MPR selector set as well as a sequence number to identify newer data from old data [22]. Each node builds the intra-forwarding from the TC messages giving each node a complete view of the network. Routing tables are built using the intra-forwarding database by finding minimal length paths from the source node to the destination node.

2.7 VANET Routing Protocols

VANET routing protocols are the next generation of routing protocols designed specifically for the unique properties of VANET environments. VANET routing protocols largely fall under the categories of clustering and geographic routing.

2.7.1 Greedy Perimeter Stateless Routing for Wireless Networks (GPRS)

The GPRS is one of the early position based routing protocols explicitly designed with protocol scaling and frequently disconnected networks in mind [24]. Frequently disconnected networks are a problem in distance vector and link state protocols, frequent disconnection may cause inaccuracies in protocol states leading to routing loops forming [24]. Protocol scaling is most greatly effected by the rates of change in the topology and the number of nodes in the network [24]. GPRS assumes each node knows all of its one hop neighbors and their positions, nodes may determine their own positions through a

GPS system and nodes have access to some sort of system containing positions of nodes multiple hops away; though this system is beyond the scope of GPRS [24].

Greedy Routing GPRS employs greedy forwarding when making routing decisions, greedy forwarding is based off of the packet senders position and final destination position. When sending a packet each node chooses the next hop based which neighbor will make the greatest geographic progress forward. Choices are made greedily at each hop until the packet reaches its final destination [24]. If a routing decision cannot be made GPRS switches to perimeter routing until it is able to switch back to greedy routing.

Neighbor Detection and Maintenance Neighbor detection is managed through periodic broadcasts of node positions. Each node broadcasts its own position with an average frequency of B. To avoid synchronization of beacons the exact beacon sending time is randomly chosen from $0.5B$ to $1.5B$ [24].In addition to sending periodic beacons each node piggybacks its current position to each packet allowing it to skip its next beacon transmission. Failed links are detected through the absence of beacon messages, if a beacon is not received within $4.5B$ the link is considered to have failed.

Greedy Routing Failure One of the weaknesses of GPRS is its behavior when the only neighbors in range are further from the destination than the source node. In order to alleviate this issue GPRS employs the *Right Hand Rule*. The right hand rule states that

the next hop will be first sequential node encountered when sweeping counter clockwise from the edge on which the packet was received. The *right hand rule* allows node to successfully route around voids between source and destination nodes. GPRS refers to this type of routing as perimeter routing.

The right hand rule necessitates a *planarized graph*. A planarized is defined as a graph where no two edges cross. Each node must independently generate a planarized view of its edges under the condition that any edges it removes do not lead to the graph becoming disconnected [24].

2.7.2 Geographic Routing In City Scenarios (GPCR)

GPCR builds on the idea of greedy position based routing and repair when a local maximum is reached, the GPCR protocol is presented in [30]. Like in GPRS nodes exchange periodic beacons to build a model of the on hop topology. Additionally all GPCR nodes contain a complete map of their environments detailing roads and junctions. GPCR assumes nodes have access to some sort of location service to determine the final destination of their packets, though the specification of such a system is outside of the scope of GPCR.

GPCR Routing GPCR routing is split into greedy perimeter coordinator routing and local maximum repair. Coordinator routing takes place in two phases, first packets are

greedily forwarded along roads towards junctions. Nodes within the area of a junction are refereed to as *coordinators*. Each node independently determines if it is a coordinator, its coordinator state is added to each beacon message informing surrounding nodes of their neighbors coordinator statues. When choosing the next hop GPCR searches through its list of neighbors for qualified next hop neighbors, qualified neighbors are neighbors which make forward progress towards a junction. If no coordinator nodes are present within the list of qualified nodes the furthest next hop is chosen, however if a coordinator node is present within the list of qualified node the coordinator node takes precedence over all other nodes as the next hop. Should more than one coordinator node be present a random coordinator node is chosen [30]. Coordinator nodes are the only nodes which may make true routing decisions, non-coordinator nodes may only forward packets along the same road segment, however coordinator nodes may switch the road segment packets travel along.

GPCR Repair Strategy Due to the greedy nature of GPCR a repair strategy is required when a local maximum is reached. The GPCR repair strategy uses elements of GPRS in its recovery strategy. Roads form a natural planar graph, when in repair mode GPCR uses the previously described right hand rule to route packets towards a coordinator node. Coordinator nodes may then route the packet onto another road segment to escape a local maximum [30].

2.7.3 Geographic DTN Routing with Navigator Prediction for Urban Vehicular Environments (GeoDTN+Nav)

GeoDTN+Nav builds on both GPRS and GPCR and improves upon them by predicting network partitions and improving reach-ability through store and forward techniques, GeoDTN+Nav is presented in [12]. GeoDTN+Nav introduces a *Virtual Navigation Interface* (VNI), the VNI is a uniform interface to exchange the following information:

1. Route Information Route information represents a vehicles current route, routes may be composed of detailed paths, final destinations or directions.

2. Confidence Confidence relays the probability that the route information will remain valid. Depending on the type of vehicle confidence may fall between 0 and 100.

GeoDTN+Nav Route Selection and Recovery GeoDTN+Nav selections routes via three mechanisms. Firstly greedy forwarding as described within GPCR routing is used to route between junctions. Secondly when a local maximum is reached recovery mode is entered, the recovery mode is similar to that presented in GPCR however the started location is noted. If recovery mode ends up routing the packet to where it first began its recovery attempt, the third mode is entered; DTN routing. The transmitting node checks its neighbor database built based on information received on the VNI. The node

calculates a score for each VNI entry, the node with the highest score is selected as the DTN carrier is selected and the packet is sent to that node. The DTN node carries the packet until it is able to switch back to greedy forwarding.

2.7.4 Contention Based Forwarding (CBF)

Contention based forwarding is a position based routing protocol, however unlike position based protocols such as GPRS and GPCR where nodes choose the next hop, neighbors compete to forward packets. The CBF algorithm is presented in [15]. CBF routing offers the following advantages over previous greedy forwarding strategies:

1. Accurate Information

   When selecting the next hop each node uses the most accurate and up to date information based on when the packet was sent and the current nodes position. This allows nodes to make decisions based on the most current data available and allows nodes which have just entered transmission range to instantly participate in packet routing [15]. Protocols such as GPRS and GPCR use periodic beacons to learn of the local topology, however due to the highly dynamic properties of VANETs topology information quickly becomes stale and outdated. When nodes base their routing decisions on this information non-optimal decisions may be made due to changes in the topology. Additionally, nodes entering the topology in-between beacons are excluded from routing decisions.

2. Elimination of Beacon Overhead

   Contention based forwarding entirely eliminates beacons commonly used to maintain a view of the local topology. The elimination of beacons saves network bandwidth normally used for beacon transmission, reduces memory required to store neighbor information and increases protocol scalability as the number of nodes in the network has no effect of the number of transmissions [15].

CBF Route Selection CBF route selection is performed in three steps, first the originating node broadcasts the packet it is attempting to send to all neighbors within transmission range. Second, all neighbors compete for the right to become the next hop; this period is referred to as the *contention period*. Finally a node wins the contention and gains the right to re-transmit the packet, the winning node *suppresses* all other nodes ensuring it is the only node transmitting the packet [15]. This process continues until the packet has reached the final destination.

Node Transmission Suppression Node suppression is the heart of the CBF protocol, effective contention and suppression strategies are required to ensure multiple nodes do not simultaneously re-transmit the same packet and or cause a broadcast storm. Node suppression requires two components, a contention strategy and a suppression strategy. CBF uses timer based contention; timer based contention is broken up into two categories, random and progress based. Random based contention timers simply choose a

random value as a backoff value, this strategy is extremely simple however it suffers from equal treatment among nodes. Treating nodes equally may cause problems if nodes near the source win the contention period, thus making little progress. Progress based contention is much more suitable in CBF, back-off timers are set based on forward progress towards the destination. Nodes making the most forward progress have the shortest back-off timer allowing them to win the contention period and ensure the most progress is made at each hop [15]. Once the contention timer has been set a suppression strategy must be selected, CBF proposes three suppression strategies.

1. Basic Suppression

   Basic suppression is the simplest of suppression strategies, all nodes receiving the packet broadcast are eligible to become the next hop. Each node receiving the broadcast starts a contention timer, the first node whose contention timer expires sends the packet and suppresses all other nodes. Simple based suppression is suffers from the possibility of up to three duplicate packet transmissions due to the possibility of contending nodes being outside of each others transmission range [15].

2. Area-based Suppression

   Area based suppression preemptively removes nodes from contending as a possible next hop forwarder. Only nodes within a specified area may contend to become the

next hop, all other nodes are may not contend. CBF uses a *Reuleaux triangle* with width of the transmission range originating from the sending node towards the destination. The reuleaux triangle loses potential forwarding nodes however all forwarding nodes are with transmission range eliminating duplicate transmissions due to re-transmitting nodes being outside of each others transmission range [15].

3. Active Selection

   Active selection seeks to eliminate duplicate transmissions caused by duplicate transmissions taking place before transmission suppression takes place. Active selection uses a similar procedure as the RTS/CTS scheme performed in the 802.11 family of MAC protocols. Before the packet is transmitted a node sends a *request to forward* (RTF) via broadcast to all of its neighbors. The RTF packet contains the sending node position and final destination position. Each neighbor starts a contention timer based on is forward progress, upon expiring the node sends a *clear to forward* CTF packet offering forwarding services to the sending node. Each other node receiving the CTF packet stops its contention and is considered suppressed. The sending node listens for CTF packets, chooses the best forwarding node and unicasts the packet to that node to forward [15].

2.8 VANET Node Addressing

VANET node addressing is a critical area of any area of network research however it has garnered relatively little research interest. Current research in routing protocols assumes an addressing scheme is already in place and functioning to provide addresses for routing processes. Section 2.4 describes the addressing scheme specified in WAVE standard, this system uses modified router advertisement messages as seen in wired IPv6 networks to advertise network prefixes to nodes. Using the advertised prefix nodes may use SLAAC to generate a unique IPv6 address. This type of architecture suffers from two main issues, firstly RSUs must use high powered transmissions to reach all nodes within the area and secondly any issues with the RSU will stop nodes from obtaining IP addresses effectively stopping all VANET communication. These two issues combine to make VANETs extremely dependent on RSU availability. The following section presents an alternative addressing strategy.

2.8.1 Alternative IPv6 Addressing Scheme

An alternative to the WAVE addressing scheme is presented in [40], this addressing scheme introduces the following terminology:

- Access Router (AR) The access router serves as the primary gateway between the VANET and global internet.

- Vehicular Domain (VD) The VD is defined as a geographic area covered by multiple access pointer all connected to a single AR.

- Border Access Point (BAP) The BAP defines access points located on the border of multiple VDs. BAPs belong to multiple VDs.

- Common Access Point (CAP) The CAP defines an access point which belongs to a single VD.

- Associated Access Point (AAP) The AAP is access point to which an OBU uses as a gateway to eventually reach the internet.

Each BAP and CAP is assigned an AP ID value, values range from $[1, 2^i - 1]$ where i is determined by the size of a VD. When setting i to 4 the maximum area of a VD is 4 km^2. Each BAP and CAP is responsible for managing its own address space, address spaces contain $2^n - 1$ addresses where n is determined by the maximum number of nodes with a geographic area. An individual IPv6 address in this addressing scheme is 128 bits where each address is composed of a global routing prefix is $128 - i - n$ bits, AP ID *i* bits and an OBU ID *n* bits.

IPv6 Address Acquisition IP address acquisition may be performed in one of two methods; either from a BAP or CAP. Both BAPs and CAPs send periodic advertisement packets, when attempting to obtain an IP address nodes listen for these advertisements.

If multiple advertisements are received, the OBU responds to the advertisement with the strongest signal.

- **Strongest Advertisement from a CAP:** If an advertisement comes from a CAP, the OBU sends a request packet to the CAP. Each CAP tracks the addresses it has previously assigned, using a hashing function presented in [40] a new OBU ID is generated. The CAP then generates and sends a response packet containing the global prefix, AP ID and OBU ID back to the OBU. Combining the prefix, AP ID and OBU ID the OBU generates a new IP address.

- **Strongest Advertisement from a BAP:** If the advertisement comes from a BAP, the OBU sends a request packet to the BAP. Instead of responding with a response containing an IP address the BAP and OBU periodically exchange *fresh* packets. The purpose of these packets is to determine the next VD the node will enter. Once the future location of the node has been estimated the BAP returns a response packet containing addressing information based on the address information of the next CAP the node is set to encounter.

3 Proposed Layer 3 Enhancements: Zone Based Geographic IPv6 Addressing and Contention Based Multi-protocol Hybrid VANET Routing

The background VANET literature and research review sheds light on several areas where the state of the art requires further research. The first area of research is node addressing, the current IEEE standard employs centrally controlled addressing. Nodes must contact an RSU which will assign the node an address, as stated in previous sections this raises two issues. First the addressing strategy is centralized and consequently more prone to failure, should an RSU fail nodes have no way to address themselves. Secondly the addresses assigned contain no inherent location information. A great deal of VANET applications reference the node location in some manner; without location data embedded within a nodes address VANETs must employ multiple system to exchange location information with neighboring nodes. Our proposal seeks to decentralize and embed geographic addressing information into IPv6 addresses with a geographic, zone

based addressing strategy.

The second area of research is hybrid V2V and V2I routing. The current IEEE standard defines V2I communication methods however it does not clearly define V2V communication methods. Research trends have largely focused on the V2V communication model. Research on V2V communication has been shown to enhance driver safety and road efficiency however pure V2V communication removes the possibility of communicating with outside networks. Our proposed routing protocol will combine elements of both V2V and V2I routing to create a hybrid of the two communication models allowing both V2V and V2I communication. The proposed protocol allows neighboring nodes to exchange information to facilitate safety and efficiency applications. Additionally neighboring nodes are used to route packets towards road side infrastructure allowing VANET nodes to reach both distant VANET networks as well as existing networks on the Internet. Building on the hybrid design, the proposed protocol will differ from a great deal of existing work by simultaneously routing multiple types of traffic.

Three forms of communication are considered for their unique characteristics, while these are not an exhaustive list of the forms of communication each form of communication possesses unique characteristics highlighting elements of the proposed routing protocol. The first form of communication considered is safety messaging, further broken down into control and emergency messages. Safety messaging is based on V2C communication following a broadcast communication model, both control and emer-

gency messages are broadcast to all nodes within a certain area. The second form of communication considered is Internet communication. Internet communication is based on directed unicast transmissions divided into communication towards a known gateway and communication towards a predicted node position. The third of communication considered is infrastructure assisted V2V communication in the form of geographic queries. Geographic query communication combines directed unicast towards roadside infrastructure and VANET based contention.

Rather than completely replacing the existing IEEE standards our proposal seeks to build on top of the existing standards to extend VANET functionality. Our proposal will exist at the network layer enhancing routing, node addressing and providing gateway bootstrapping functionality. Further building on current standards, our proposal bridges the gap between V2V and V2I communication methods, provides geographic based addressing and builds a platform for future multi-protocol VANET research.

Our proposal for layer 3 functions in VANETs are split into three areas; zone assignment, IP addressing and hybrid routing. Zone assignment is reviewed in section 3.1, next the proposed IP addressing scheme is reviewed in section 3.2 and finally the proposed hybrid routing is reviewed in section 3.3.

3.1 Zone Assignment

Our proposal divides the entire land mass of the earth into discrete geographic zones. Ideally each zone contains its own address space and its own RSU responsible for gateway functions and distant neighbor location services. Zone definition is vital when examining the ”big picture” of VANETs on the global internet especially when addressing nodes in VANETs, zones give geographic context to communication between VANETs and external networks. Using the geographic information routing protocols and applications may make better decisions by identifying source of the transmission. The total land area of earth is 148,326,000 km^2 thus any zone assignment strategy must cover this entire area. Zone assignment must balance location granularity with address space, communication range and cost of infrastructure. Each VANET node is expected store its own version of the RSU database.

3.1.1 Zone Definition

Balancing the three factors of zone assignment in section 3.1 this proposal assumes each zone covers an area of $1km^2$. Based on that assumption, at least 148,326,000 zones are required to cover the land area of the earth. Using equation 3.1 to find the number of bits needed to uniquely represent each zone requires 27.144 bits; the number of bits must be an integer value bringing the required number of bits up to 28.

$$\log_2 n = b \tag{3.1}$$

Each zone is defined by the following fields:

- Bottom Left Corner: The bottom left corner is a 128 bit field defining the coordinates of left most corner of the zone. Coordinates are based on a Cartesian coordinate system, each coordinate is composed of a 64 bit x and 64 bit y value.

- Top Right Corner: The top right corner is a 128 bit field defining the coordinates of the top right corner. The coordinate is defined as a 64 bit x value and a 64 bit y value.

- RSU IP Address: The RSU address field is an optional 128 bit field which defines the IPv6 address of the RSU for the zone. The RSU address for the current zone may be used as the gateway address to reach other networks.

- RSU Position: The RSU position field is a 128 bit field defining the x and y coordinates of the RSU. This field serves as a *boot-strapping* mechanism providing anchoring information to nodes.

- Zone ID: The zone ID defines the geographic ID of the zone. As stated above 28 bits are required to identify a zone, this is expanded to 32 bits to make the field a multiple of eight in order to facilitate storage of the ID.

Each zone is fully defined by 68 bytes. The entire set of all zones definitions may be stored in a database approximately 10 GB in size. Based off of the properties of VANETs described in section 1.2, each VANET node can easily accommodate storage of a 10 GB database. As each node is expected to store its own version of the zone database certain optimizations may be used to reduce the size of the database, one such optimization would be to only store relevant entries. Vehicles are unlikely to travel extreme distances from their original sale location and extremely distance entries are unneeded in the zone database. As an example; vehicles in North America are examined, vehicles in North America are unlikely to ever leave the continent. The North American continent covers approximately 16% of the total land area of the Earth. Using the 16% value as an example, a vehicle in North America may store all zones within North America with a database less than 2 GB in size.

3.1.2 Determining Zone Membership

As nodes travel through the network each node must independently determine its zone based on its position. Figure 3.1 shows an example of a node attempting to determine its current zone. Based off of the information in the zone database a node is able to determine the four vertices of each zone. A well known property of quadrilaterals is that when a point is inserted anywhere inside of the quadrilateral four triangles may be form using the inserted point as one of the vertices in each triangle. The total area of the four

formed triangles must equal the area of the quadrilateral. This property of quadrilaterals is used when determining if a node is in a particular zone; for each potential zone the nodes current position is used as a vertex to form four triangles based off of the zone edges. The area of each triangle is computed; if the area of the triangles is equal to a zone area, the node is within the checked zone. The equation of a triangle may be determined based on its three vertices A = (A_x, A_y), B = (B_x, B_y) and C = (C_x, C_y) using formula 3.2.

$$A = \left| \frac{A_x(B_y - C_y) + B_x(C_y - A_y) + C_x(A_y - B_y)}{2} \right| \tag{3.2}$$

Periodically each node compares its current position to the current zone. If the nodes position does not match the current zone. nodes must begin the re-addressing process as show in section 3.2.

3.2 IP Addressing

Our proposal uses IPv6 addressing as mandated by the IEEE 1609.3 standard, addresses are generated via Stateless Address Auto Configuration (SLAAC) as specified in the same standard. Our proposal builds upon IPv6 addressing in two areas; firstly it creates a unique VANET identifier making VANET traffic easy to identify and manage, secondly geographic information is embedded into each nodes IP address giving a sense of where a

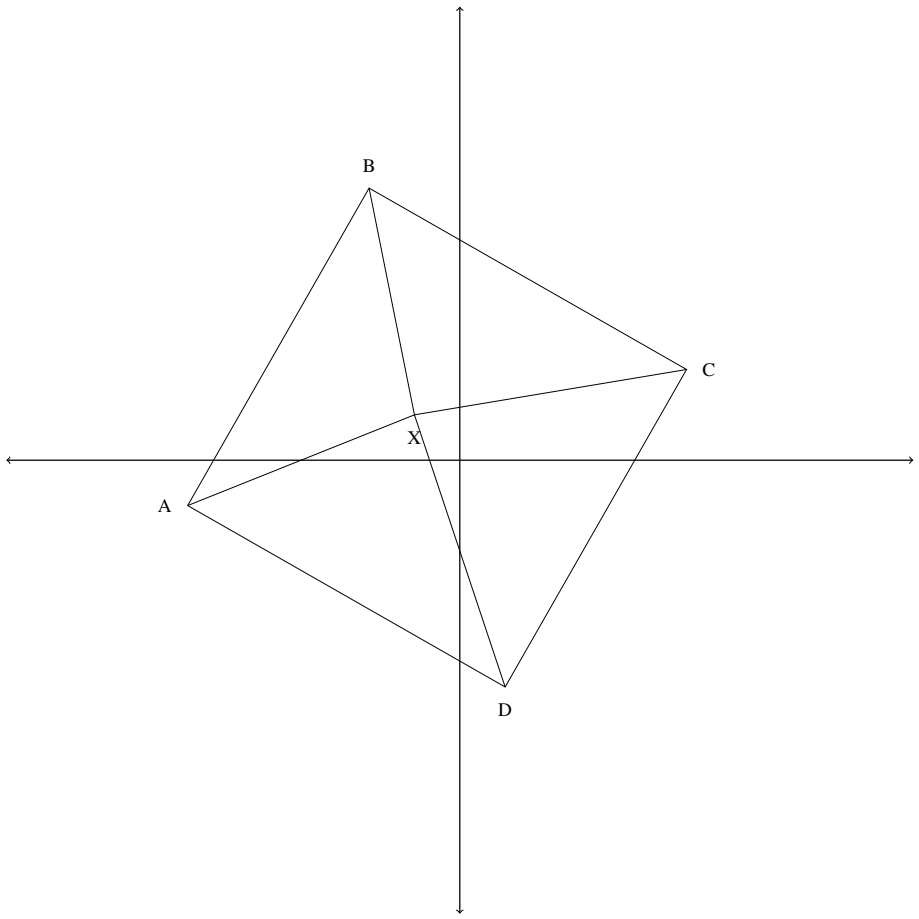

Figure 3.1: Zone Membership

node is located from simple examination of the address. Moreover our proposed addressing strategy scales extremely well for integration into the global Internet, the proposed addressing strategy is hierarchical in nature allowing for the crucial route summarization to be used in internet routers. Figure 3.2 shows the format of a IPv6 address in our proposal.

The proposed IPv6 address is composed of three sections a 36 bit VANET identifier, a

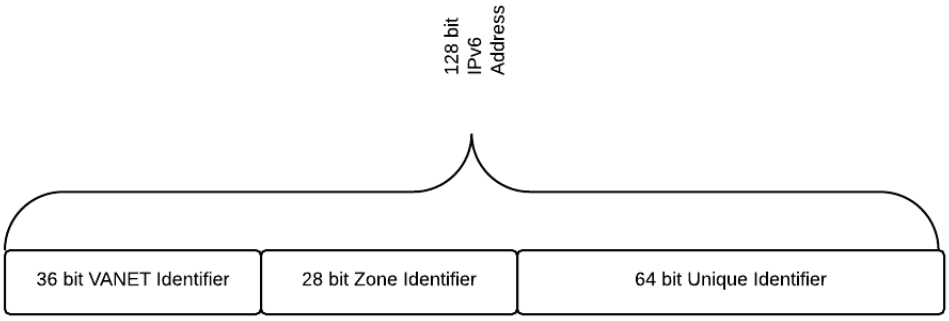

Figure 3.2: Proposed IPv6 Address Model

28 bit zone identifier and a 64 bit unique host identifier. The VANET identifier is a common identifier shared across all VANET nodes. End nodes communicating with VANET nodes may easily identify they are communicating with VANET nodes; this opens up the possibility of QoS functionality and application specific behavior based on traffic source. The zone identifier identifies the nodes current zone, this directly embeds geographical coordinates with a granularity of $1km^2$. The zone identifier field is common to all nodes within a zone. Currently deployed addressing proposals cannot directly determine the geographic location of the source of a transmission, databases exist which attempt to map an IP address to a geographic area however these databases are extremely inaccurate with mobile nodes. The final 64 bits uniquely identify each node; the generation of this field is reviewed in section 3.2.1.

3.2.1 Stateless Address Auto Configuration (SLAAC)

Stateless Address Auto Configuration is a self addressing mechanism introduced in the IPv6 standard allowing nodes to generate their own address based off of a 64 bit IPv6 prefix and a 48 bit MAC address. The 64 bit prefix is generated by combining the 36 bit VANET identifier and the 28 bit zone identifier. The final 64 unique host identifier is generated through the following procedure:

- Determine the interface MAC Address: The MAC address for the interface being addressed is extracted.
- Split into halves: MAC addresses are 48 bits in length; this step breaks the MAC address into two 24 bit halves.
- Generate New bits: The MAC address is extended to 64 bits by adding 16 bits between the two halves of the previously split MAC address. The bits added in hexadecimal are FFFE.
- Regenerate MAC Address: The two halves of the MAC address and the additional bits added into the previous steps are merged into a single 64 bit value.
- Invert Universal bit: Bit 7 of a MAC address is commonly referred to as the universal bit. MAC addresses burned into networking equipment are required to have a 0 in the 7th position. The SLAAC procedure inverts this bit to a 1. The exact

reason for this last step is beyond the scope of this thesis however from a high level this inversion is primarily for convenience.

Once the 64 bit host field has been generated the IPv6 prefix and host fields are combined to form the unique 128 bit IPv6 address the node uses in routing and communication. Based on this addressing scheme each zone supports up to $2^64 - 2$ nodes; this is far beyond maximum possible number of nodes within a zone however the unique host space is not reduced as this would invalidate the SLAAC procedure, work against IEEE standards and work against current trends in IPv6.

3.2.2 IPv6 Route Summarization and Hierarchical Addressing

One of the primary concerns when creating any addressing strategy is the scalability of the strategy. Scalability is vital in addressing strategies due to growth in routing tables on internet routers, routing tables must be kept as small as possible to ensure route lookups are performed as fast as possible. Furthermore addressing strategies must be devised such that routing tables are able to fit into internet router memory, IPv6 supports enough addresses to address each atom on the surface on the earth 100 times over; without summarization at the forefront of addressing strategy design routing tables would be overwhelmed with more routing entries that could possibly fit into a routing table.

Addressing strategy scalability is primary achieved through summarization and hierarchical addressing. Hierarchical addressing is achieved by subdividing network address

space into smaller logical units until each network is addressed. Route summarization may then take place on internet routers; distant routers only require the high level address to route the packet in the correct direction, downstream routers closer to the final destination contain higher granularity routes allowing them to make more specific routing decisions until the final destination is finally reached.

Example Hierarchy One example to illustrate hierarchical addressing would be to subdivide the zone identification field of addresses into an 7 bit region ID followed by 21 bits usable for zone identification in each region. This allows for the creation of 128 worldwide regions with approximately 2.1 million zones per region. 128 regions approximately lines up with the number of countries however it falls short of a region for each country however multiple smaller countries may be combined into a single region. Internet routers distant from VANET destination areas would only need to store a maximum of 128 routes for each region. The remaining zone bits may then be used to further subdivide regions or to address zones.

3.2.3 Address Space Usage

One of the concerns with addressing strategies is effective use of addressing space. While the IPv6 is exceedingly large and it is unlikely to be exhausted in the near future, care must be taken to avoid unnecessarily wasting large amounts of address space. Based on

current recommendations in IPv6 usage addresses are broken up into a 64 bit network prefix and a 64 bit host field. A 64 bit network prefix allows for up to 2^{64} networks while our proposed addressing strategy reserves 2^{28} VANET zones. Our addressing strategy reserves less than 0.00000000146% of the total IPv6 address space for VANETs.

3.3 Multi-protocol Contention VANET Routing Protocol (CVR)

Our proposed routing protocol (CVR) builds on top of the CBF presented in section 2.7.4 as well as the spatio-temporal multicast protocol presented in [11]. The proposed routing protocol proposes several novel ideas; first the RSU is actively involved in the VANET allow it to increase the total functionality of the VANET, secondly the routing protocol handles a variety of communication types and models; this allows a greater variety of traffic to be managed by a single protocol.

Routing decisions are primarily split into three types of communication; internet access via a hybrid V2V and V2I model, emergency V2V messages and finally hybrid V2V and V2I zone queries. Due to the unique constraints of emergency messages as opposed to internet and query messages, these two classes of messages are managed in discrete units. Each units cached data may be accessed by another unit through common interfaces however each individual unit manages its own routing decisions and its cached data.

3.3.1 Safety Message Routing

Safety messages are the first class of messages routed by our protocol. Due to the real time constraints and priority concerns of safety messages, safety message routing is given priority over all other types of communication. Safety messages are given absolute priority over all types of messages at the routing layer; incoming messages of any type are first processed by the safety routing unit and outgoing traffic is first generated and queued by the safety routing unit.

3.3.1.1 Safety Protocol

The safety message protocol uses a special type of multicast referred to as a *spatiotemporal* multicast or *Mobicast*. Mobicast routing is specifically designed to *Zone Of Relevance* (ZoR) based on node conditions at the time of message generation. The proposed protocol builds on the ideas in [11] where nodes involved in an emergency dynamically generate an elliptical ZoR based off of their current velocity and position. Next, using a modified version of contention based routing the emergency message is propagated throughout the ZoR. This modified version of contention based routing necessitates the exchange of periodic messages informing all neighbors of their positions.

3.3.1.2 Periodic Hello Message Exchange

The authors of the mobicast protocol given in [11] require the exchange of periodic hello messages. While our system overall does not require periodic hello messages due to the contention based routing built upon, the original authors sentiment is maintained. Periodic messages are sent with a period of [T - 1.5T], after sending a hello message, nodes choose a random delay from the range of [T - 1.5T] to wait before sending the next hello message. A range of delays is used to avoid message synchronization and serial collisions. Each transmitted hello message contains the nodes ID and position. Each node maintains a cache containing neighbor information; neighbor information consists of the neighbors ID, its most recent position and the time at which the latest update was received. The neighbor cache actively clears stale cache entries with a last update greater than 5T. Stale cache entries indicate two nodes have move outside of each others transmission range.

3.3.1.3 ZoR Generation

ZoR generation begins with an onboard sensor detecting an emergency event such as an imminent collision or a sudden stop. The detection of an emergency event triggers the generation of a ZoR; the ZoR is defined by formula 3.3 where a is the length of the major axis, b is the length of the minor axis, V_e is the node where the event was encountered,

V_i is any point within the ellipse and x,y refer to the coordinates of a point. Figure 3.3 shows the ZoR generated with the event vehicle at the center of the ellipse. Two virtual vehicles are generated at the left and right apex of the ZoR ellipse, these virtual vehicles are used as targets when disseminating the emergency message. The *a* and *b* values are vital forming the ZoR; the *a* value may based on the type of emergency message, the authors of [11] suggest the following formula: $a = V_e * \frac{1}{10} * L_m$ where L_m is the length of a vehicle. The *b* value is suggested to be the width of a lane however our protocol expands this value to multiple lanes to increase message dissemination.

$$Z_t(Vi) = \frac{(x_t^{V_i} - x_t^{V_e})^2}{a^2} + \frac{(y_t^{V_i} - y_t^{V_e})^2}{b^2} - 1 = 0 \tag{3.3}$$

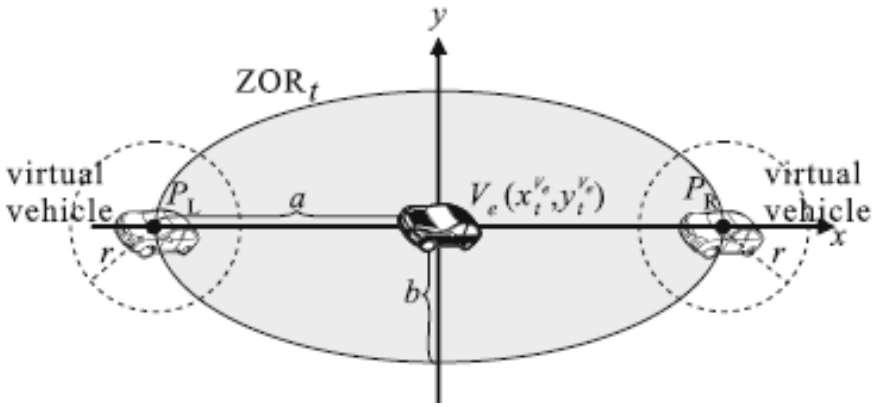

Figure 3.3: Zone of Relevance Generation

[11]

3.3.1.4 Emergency Message Dissemination

Emergency messages are disseminated through the following steps:

1. Event node sends Emergency Message: Beginning the message dissemination process the event node multicasts the emergency message. The message is composed of three fields; first the ID of the node generating the message, the authors of [11] do not explicitly specify the format of the ID however we have adopted the unique IPv6 address of the node as the ID. Second, a description of the ZoR is included, the ZoR is described by the apex coordinates and the radius lengths. Finally the third field contains the actual message containing the emergency details.

2. Surrounding Vehicles Process Message All vehicles within transmission range of the original transmission receive the transmission and begin processing the message. Processing is performed in 5 steps

   (a) Determine is node within ZoR Based on the description of the ZoR in the transmission each node determines in it is within the defined ZoR. If the node is indeed within the ZoR it continues processing the packet, otherwise the packet is dropped.

   (b) Determine forwarding direction Once it has been determined that the node is within the ZoR the node determines which direction is should forward

the message in; this is accomplished by determining which apex the node is nearest to.

(c) Determine effective re-transmission Each node checks if it would be effective to forward the message. Effective message forwarding is determined by two factors; first the node must have a potential neighbor in the ZoR that could be a candidate to forward the message once again, and secondly, any candidate that could forward the message must be closer to the apex than the current node.

(d) Begin proportional backoff If packet forwarding is deemed effective the node begins a backoff timer based on the distance from the apex multiplied by a random value; the equation is given in equation 3.4.

$$R_{time} = \frac{d_{V_j}}{d_{V_i}} * r_{time} \tag{3.4}$$

(e) Transmit or suppress packet Once the backoff timer expires a node is free to forward the emergency message. All nodes detecting the forwarded message must suspend their contention efforts.

3.3.2 Contention based VANET Routing Protocol (CVR)

The CVR routing protocol proposed in this thesis is capable three main types of communication; pure V2V, hybrid V2V and V2I internet and finally hybrid V2V communication assisted by V2I infrastructure. Due to the extensive amount of research work on pure V2V communication using multiple routing strategies, the two hybrid forms of communication are the primary focus of this thesis. The basis of our routing protocol builds on the ideas in contention based routing, nodes do not require knowledge of their local neighborhood to make routing decisions and all nodes compete to become the forwarding node through a contention mechanism. The first form of hybrid V2V and V2I VANET communication for internet access is covered in this section while the second V2V method employing V2I assistance is reviewed in section 3.3.3.

3.3.2.1 Hybrid Internet Access: Node to Internet

Hybrid internet access is achieved through a combination of V2V and V2I communication, and zone based gateways. Each RSU serves as an *intelligent* gateway serving as the border between the VANET network and other networks on the Internet. Aside from standard gateway functions each RSU performs two additional functions; mobility prediction and inter-RSU forwarding. Mobility prediction leverages mobility constraints in VANETs in order to predict a nodes position when attempting to send reply packets into

the VANET. Inter-RSU forwarding further builds on mobility prediction to handle cases where nodes move between zones between sending a transmission and receiving a reply. Inter-RSU forwarding allows an RSU to forward a reply to another RSU if the node is predicted to have entered that RSUs zone. Inter-RSU forwarding attempts to reduce the number of hops a packet must make over the less reliable VANET channels by forwarding over reliable point-to-point links to reach a gateway as close to the target node as possible. A complete internet query is accomplished through the following steps:

1. Generate Request: The sending node begins the communication process by creating a standard TCP or UDP packet and passing it down the network stack until it reaches the network layer.

2. Attach Headers and Transmit Packet: Before the packet is transmitted two headers are attached to the standard IP packet. The first of these headers identifies the traffic type and the second contains relevant information to be used in routing logic on receiving nodes. The first header is a type 3 header as described in section 3.4, this identifies the packet as a packet being sent to the internet and it also informs receiving nodes that the next header contains information required to route the packet. Once both headers have been attached the packet is multicast to all neighbors within transmission range.

3. Receive Packet and Begin Contention: All nodes within transmission range receive the packet and strip the type and information headers. As the type 3 header was attached in the previous step the receiving node will detect the transmission is a type 3 message triggering internet destination processing. Type 3 packet processing is split into four phases; active forwarding queue management, effective validation validation, contention and suppression. Active forwarding queue management attempts to spread forwarding duties among eligible nodes instead of focusing all packet forwarding through a single node. Before beginning contention nodes check the number of active contention cycles currently running. If the number of active contention cycles is below a set threshold, the effective validation and contention processes may begin, otherwise the node does not enter contention. Effective validation builds on the ideas presenting in CBF presented in section 2.7.4. Three conditions must be passed for contention to be deemed effective, these conditions are:

   (a) Zone verification: The first step in contention management is to verify the current node is in the same zone as the sending node. If both nodes are in the same node this check passes and the effective contention checking continues. If the current node is not in the same zone the contention process is abandoned and the packet is dropped. The current iteration of the protocol does not allow inter-zone routing however future work could enable this form

of routing.

(b) Forward Progress: The next check determines if packet forwarding will make forward progress towards routing the packet to the gateway. Nodes determine forward progress by checking the sender position field in the information header. The distance between the the transmitting node and the current node is calculated; if the distance between the current node and the RSU is less than the transmitting node and the RSU passes this check and continues processing, otherwise the packet is dropped.

(c) Target Not Overshot: The final check determines if the current node has passed the target location. This condition prevents a packet from being continually routing around an RSU while missing suppression messages.

If all three conditions are met packet forwarding is deemed effective and the contention process may continue. Contention is based on a backoff timer proportional to the current nodes distance to the gateway as compared to the transmitting node. The contention based timer is set by adapting formula 3.4 with a constant r_{time} value as a backoff. Based on the result of the contention formula each node begins a contention timer and adds the packet to a cache to forward the packet once the contention phase has ended.

4. Packet Forwarding and Suppression

Eventually one of the transmission timers set in the previous step expires and the node updates the type 3 header attached to the packet and begins its transmission. Once the packet has been transmitted the transmitting node begins a *hold down timer* for T seconds. The hold down timer keeps packet metadata temporarily cached to ensure the transmission is not erroneously forwarded again.

All other nodes receiving the forwarded packet have their contention efforts suppressed; each node stops its contention timer. In addition to dropping the packet from its cache, these nodes cache the packet metadata and start a hold down timer for T seconds to prevent them from forwarding the same packet twice.

5. RSU Acknowledgment Contention, packet forwarding and suppression are repeated until the packet is dropped or until it reaches the intended RSU. When the RSU receives an internet bound packet from the VANET the packet is processed in 3 phases.

   (a) Cache Transmission Details The RSU caches the packet origin source address, position, velocity and timestamp. These cached details are used to assist in routing replies.

   (b) Remove Headers and Route Onto Internet Global internet routers are not aware of the headers attached in our proposed protocol and do not require them to make routing decisions. Without removing headers internet band-

width would be wasted transmitting unneeded information and more importantly, end nodes would incorrectly remove headers due to an unknown packet structure. Once the additional headers are removed the packet is reformed with the standard IP payload and IP header, then it is routed onto the Internet.

(c) Send Acknowledgment Building on suppression mechanisms in our protocol and the CBF protocol in section 2.7.4 RSUs send a type 5 RSU acknowledgement packet with the VANET to Internet header. All nodes receiving the RSU acknowledgement packet check the information header and check their contention cache for scheduled duplicate packets. If a cached packet is found its contention timer is stopped and the transmission is suppressed.

3.3.2.2 Hybrid Internet Access: Internet to Node

Packets sent onto the internet are assumed to receive a response, this section describes the logic in routing the response from an RSU back to the originating VANET node. We assume a client-server paradigm where VANET nodes are clients and servers are static internet nodes. The opposite scenario does not appear to be reference in literature though it is not a fundamentally impossible scenario; as this case is rarely studied our proposal does not consider this case. The internet reply is routed into the VANET through the following steps:

1. Receive Internet Packet and Predict Destination Zone: The RSU receives the reply from the internet and removes the standard IP header for further processing. Based on the destination address the RSU runs a lookup on its cache to determine if it has at least one entry for the destination; the most recent entry is returned. Once an entry has been returned mobility prediction is performed; mobility patterns may be modeled in one of three models, the model type is determined by the data in the cache entry.

   The first prediction model is used if the velocity contained within the cached entry is a non-zero value, the future node position is predicted by calculating the time difference between when the packet was sent and the current time. Using the time difference the future position is calculated by adding the velocity multiplied by the time difference to the initial position.

   The second prediction model is used if the latest cache entry contains a zero velocity value indicating the node was stopped during the transmission. The second prediction model searches the cache for additional entries from this node, if no additional entries are found the third model is used. If additional entries are found the velocity values of these entries are averaged to generate an average velocity value. The average velocity value is then used in the same way as the first model to predict the nodes position.

The third and final prediction model is used if the velocity is zero and no previous entries are found no prediction may take place. The node is predicted to stay in the same position as there is not enough information to make a better decision.

Based on the nodes predicted position the nodes zone is also predicted, the prediction determines if the RSU routes the packet into the VANET or if it routes the packet towards another RSU.

If the node is predicted to be in another zone the RSU creates a new Internet to VANET information header and a type 7 RSU-to-RSU type header. Both of these headers are attached to the reply and the packet is sent to the RSU for the calculated zone. Upon arriving at the next RSU, the RSU removes the type 7 header, modifies the standard IP header destination address by modifying the zone ID portion of the address to its zone ID, attaches an I2V information header, attaches a type 4 header and routes the packet into the VANET. If the node is predicted to be in the same zone, RSU to RSU forwarding is skipped, a I2V information header and a type 4 header is attached and the packet is routed into the VANET.

2. Gateway to VANET node Routing: Gateway to VANET node routing is accomplished identically to VANET to gateway routing except for the target destination. Nodes attempt to route the packet towards the predicted position as opposed to a known gateway position. The packet is continually routed until it is dropped due

to ineffective transmissions, exceeded hop counts or if it reaches the destination node.

3. Destination Node Receives Packet: When the destination node receives the packet it performs two functions to finish the communication and routing process. First the node passes the next network layer and secondly the node sends an acknowledgement to suppress any remaining transmissions. The acknowledgement is sent via a short acknowledgement packet with a type 6 header and the received information header.

3.3.3 Geographic Query (GeoQuery) Routing

The second form of hybrid routing is referred to as geographic query routing. Geographic query routing referrers to a form of routing where a node attempts to query a specific distant geographic area for information, however two challenges are encountered. Firstly a sending node may not know the IP address of a node within the geographic area it is attempting to query. Secondly, the issue of distance is encountered, distant locations require large numbers of hops to reach. Each hop decreases the chances of a packet ultimately reaching its final destination due to increases in contention and chances of collision. Our proposal solves these two issues through V2I assistance and reply contention. A complete GeoQuery is accomplished through the following steps:

1. Upper Layer Query Formation:

   As with all routing processes the upper layers in the network stack form the request. The upper layers are not required to specify the final destination as that will be determined by reply contention, however the query position must be specified. Due to the structure of the network stack as previously described, the network layer does not have access to upper layer data. This presents an problem as the routing protocol requires the geographic position of the query however it is specified in the upper layers. In order to solve this issue we provide a very limited interface allowing upper layers to specify GeoQuery coordinates as the packet is passed down the network stack.

2. Contention Based Routing Towards RSU: Once the routing protocol receives the packet from the upper layer it forms and attaches a type 8 type header and GeoQuery information header. Due to the similarity between this stage and V2I routing described in internet routing we do not go into detail routing the packet towards the gateway RSU. Upon receiving the GeoQuery packet the RSU transmits the GeoQuery packet towards the correct RSU zone and it also transmits an acknowledgement packet with a type 10 type header; this suppresses any remaining contention attempts. Upon arrival at the query zone RSU the RSU forwards the GeoQuery request packet into the VANET.

3. GeoQuery Reply Contention: Nodes within range of the RSU receive the transmitted GeoQuery request and begin contention by setting a timer based on equation 3.4. Upon expiration of the contention timer the node sends a packet containing a type 12 type header and the original GeoQuery request header; additionally the node passes the request packet up the network stack for further processing. Upon receiving suppression packet nodes suspend their contention processes and stop transmission. Once the upper layers have successfully processed the request a reply is generated and sent back to the routing protocol. Due to mobility prediction and caching on RSUs the replying node does not specify the predicted position of destination node. The format of the reply packet sent uses a type 9 type header identifying the packet as a reply and uses a GeoQuery reply information header containing routing information. The reply is routed towards the RSU which then routes the packet towards the destination zone.

4. Contention Based GeoQuery Reply Routing: The GeoQuery reply routing is quite similar to internet based replies presented in the previous section. Upon receiving the reply packet the RSU searches its cache for entries on the destination node, using cached information the RSU predicts the nodes final position based on mobility prediction methods described in earlier sections. If the node is predicted to have moved outside of the RSU zone the reply is forwarded to the correct zone, otherwise processing continues normally to forward the packet. Once a reply has

reached the correct zone the RSU updates the GeoQuery reply header and forwards the reply backed into the VANET. The VANET routes the reply packet towards the predicted position using contention based routing logic identical to that described in I2V routing in the previous section. Upon receiving the reply packet the destination node sends an acknowledgment to suppress all other forwarding attempts. Acknowledgments are sent using a type 11 type header along with the original GeoQuery reply information header. All nodes receiving the acknowledgment stop their contention and forwarding efforts and start a hold down timer after which the cached packet is dropped.

3.4 Packet Headers

The proposed routing protocol identifies two classes of headers; information headers and type headers. Figure 3.4 shows the structure of all packets sent in our proposal. Transmitted packets are divided into four sections; the standard variable length IP payload, an information header using in processing packets in our protocol, a 1 byte type header identify type of packet while routing the packet in the VANET and finally a standard 40 byte IP header. The outer IP header ensures underlying mechanisms do not require modification in order to process VANET packets. Once the IP header is removed our protocol is free to remove and modify the type and information headers. The type and information headers are exclusively used when communicating with VANET based sys-

tems; when communicating with non-VANET aware nodes these headers are removed from the packet before being routed forward. The removal of these headers ensures that non-VANET aware nodes do not need to be modified in anyway to integrate VANET nodes into the global Internet. Figure 3.4 shows the packet structure packets transmitted by our routing protocol.

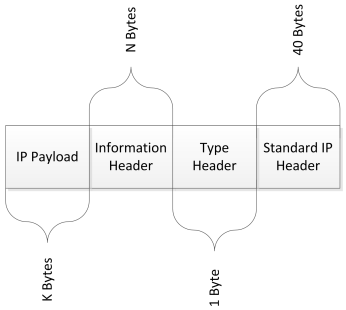

Figure 3.4: Packet Structure

A type header may hold up to 256 values, each corresponding to a type of information header, we use fourteen of these values. Type headers identify the preceding information header and routing logic process the packet. The following fourteen type headers are defined:

1. Type 1: MobiCast Hello: Type 1 headers indicates the information header contains information on a current neighbor. Type 1 headers prompt nodes to updating their cached neighbor table.

2. Type 2: MobiCast Control: Type 2 headers indicates the packet contains emer-

gency event messages and the information header contains information on the emergency ZoR. This type of packet triggers processing to first process the emergency message and second to determine if the node should continue to forward the emergency message.

3. Type 3: Hybrid Internet Routing (VANET to RSU): Type 3 headers indicate the current packet is to be routed towards the nearest gateway, additionally it indicates the information header contains data required to route the packet. This header triggers contention based forwarding logic to route the packet further towards an RSU.

4. Type 4: Hybrid Internet Routing (RSU to VANET): Type 4 headers indicate the current packet is a reply from the internet originating from an RSU and is sent into the VANET. Similar to type 4 packets, this header indicates the information header contains information required to make routing decisions. This header triggers contention based forwarding logic in order to route the packet towards a geographical location.

5. Type 5: RSU Acknowledgment: Type 5 headers indicate a packet has successfully reached its gateway RSU and the RSU is sending an acknowledgement. This packet triggers contention suppression in nodes which may be attempting to forward a packet which has already reached its gateway.

6. Type 6: VANET Acknowledgment: Type 6 headers are similar to type 5 headers except these headers are sent by VANET nodes when they have successfully received a message. This header triggers contention suppression in nodes which may be attempting to forward a packet which has already reached its destination.

7. Type 7: RSU Redirect: Type 7 headers are specifically sent between RSUs. These headers indicate the information header attached contains transmission information required for an RSU to forward the received packet into the VANET towards a node which has newly entered the zone.

8. Type 8: VANET GeoQuery Request: Type 8 headers indicate the current packet has been sent from a VANET node to query a geographic location. Type 8 headers indicate the information header contains data needed to route the packet towards the query location.

9. Type 9: VANET GeoQuery Response: Type 9 headers indicate the current packet has been sent from a VANET node in response to a type 8 GeoQuery message. This header indicates the attached information header contains data needed to route the packet back to the source VANET node.

10. Type 10: RSU GeoQuery Acknowledgment: Type 10 headers indicate an RSU has received a GeoQuery request to route to the request zone. Packets with this header trigger suppression of GeoQuery contention in all VANET nodes still attempting

to forward a query.

11. Type 11: VANET GeoQuery Reply Acknowledgment: Type 11 headers indicate a VANET node has successfully received a reply to its GeoQuery message.

12. Type 12: VANET Sending GeoQuery Reply: Type 12 headers indicate a node believes it has won the contention phase of the GeoQuery contention process and will be sending a reply message. Packets containing this type of header suppress all other nodes contending to send the reply.

13. Type 13: GeoReply RSU Redirect: Type 13 headers are sent between RSUs and indicate a sending node is predicted to have changed its zone between the RSU receiving the request and reply; this headers functionality is extremely similar to type 7 headers.

14. Type 128: Unknown Transmission: This final header type is a catch-all default type header which indicates an error has occurred. This type of error may occur if bit level corruption occurs during packet transmission or if processing logic has failed. This header is not actively used but all nodes must check for occurrence of an unknown message type to prevent error propagation throughout the network.

3.4.0.1 Information Headers

Behind the type header is an information header, these headers provide the information required for neighbors to make routing decisions. The following information headers are defined:

1. Neighbor: The neighbor information header is 50 bytes in length and contains neighbor data which would be added to all receiving neighbors caches. The 50 bytes are split into the following fields:

   (a) Node ID: The node ID is a 16 byte field containing the IPv6 address of the transmitting node.

   (b) Node Position: The node position is an 16 byte field containing the node position. The 16 bytes are split into 8 bytes containing the x position on a Cartesian plane. The next 8 bytes are similarly split into the y position.

   (c) Node Velocity: The node velocity is an 18 byte field containing the node velocity. Similar to the position field, the 18 bytes are split into 8 bytes containing the x velocity, the next byte contains the sign of the x velocity. The next 9 bytes are similarly split into the y velocity and the corresponding sign.

2. Emergency Message Dissemination Control: The emergency message dissemination information header is 80 bytes in length and contains information required to determine if the packet should be further disseminated after processing the message. The 80 bytes are split into the following five 16 byte fields:

   (a) Sending Node ID: The sending node ID identifies the source of the emergency message, the ID is in the form of an IPv6 address.

   (b) Transmission Source ID: The transmission source ID identifies the node which last transmitted the packet, initially this value matches the sending node ID however as nodes forward packets this field is updated by the transmitting node.

   (c) Left Apex Coordinate: The left apex coordinate identifies the left apex of the ellipse defining the ZoR. The coordinate value is split into a 64 bit x coordinate and a 64 bit y coordinate.

   (d) Right Apex Coordinate: The right apex coordinate identifies the right apex of the ellipse defining the ZoR. The coordinate value is split into a 64 bit x coordinate and a 64 bit y coordinate.

   (e) Ellipse Center: The ellipse center defines the coordinates of center of the ZoR and the position of the emergency. The coordinate value is split into a 64 bit x coordinate and a 64 bit y coordinate.

3. Internet Routing - VANET to Internet (V2I): The V2I information header is 94 bytes in length and contains information required by nodes to make routing decision to forward the packet towards the zone gateway. The 94 bytes are split into the following fields:

   (a) Origin Position: The origin position field is 16 bytes in length containing the x and y coordinates of the node at the time that the packet was generated.

   (b) Origin Velocity: The origin velocity field is 18 bytes in length containing the x and y velocity and the velocity signs of the node at the time that the packet was generated.

   (c) Origin Packet Time-stamp: The origin packet time-stamp is a 8 byte value which contains an exact time-stamp of when the packet was created.

   (d) Delay Tolerance: The delay tolerance field is a 1 byte field specifying if the packet may tolerant longer carrying delays. This field is not currently used, however it is maintained to allow for future protocol expansion.

   (e) Sending Node Position: The sending node position is a 16 byte value containing the x and y coordinates of the last node transmitting the packet. Initially this field is equal to the origin position however each re-transmitting node modifies this field to its own position.

(f) Sending Node Velocity: The sending node velocity field is 18 bytes in length containing the x and y velocity and the velocity signs of the node at the time of transmitting the packet. Initially this field is equal to the origin velocity however each re-transmitting node modifies this field to its own velocity.

(g) Gateway Address: The gateway address field is a 16 byte field containing the IPv6 address of the sending nodes gateway. This field is used to make routing decisions as well as transmission suppression.

(h) Hop Count: The hop count field is an 8 byte value which is incremented after each hop. This field is used to drop packets to ensure packets are not routed indefinitely.

4. Internet Routing - Internet to VANET (I2V): The I2V information header is quite similar to the V2I information header however it has been slightly reduced in size to 76 bytes by removing unnecessary information. The I2V information header is composed of the following fields:

(a) Origin Position: The origin position field is a 16 byte field containing the x and y coordinates of gateway beginning the transmission into the VANET.

(b) Original Time-stamp: The original time-stamp field is a 16 byte field containing the time-stamp of the original request. The time-stamp field is used to differentiate multiple transmission to a destination node.

(c) DTN Tolerance: Similar to the V2I information header the DTN tolerance field is a 1 byte field determining is a packet may be switched to a store and forward protocol. This field is not actively used however it is included to allow for future expansion.

(d) Sender Position: The sender position field is a 16 byte field containing the x and y coordinate of the latest sender. Initially this position will be set to the RSU address however as the packet is routed towards its destination it is updated by the forwarding node.

(e) Sender Velocity: The sender velocity field is an 18 byte field containing the x and y velocity and signs of the latest sender. Initially this position will be set to 0 as the RSU is stationary however as the packet is routed towards its destination it is updated by the forwarding node.

(f) Predicted Position: The predicted position field is a 16 byte field containing the x and y coordinates of the predicted node position. This field is used as a target towards which packets are routed.

5. Geographic Query Request: The geographic query request information header is quite similar to the Internet Routing - VANET to Internet (V2I) information header however it has been extended to include additional position query coordinates. The geographic request header is 110 bytes long, adding 16 bytes to the V2I informa-

tion header. The initial 94 bytes are identical to those in the V2I header and will not be covered again, however the next 16 bytes are reviewed here. The additional 16 bytes contain the geographic coordinates of the query location. The query coordinates are split into two eight byte fields, the first contains the x coordinate and the second contains the y coordinate.

6. Geographic Query Reply: The geographic query reply information header is structurally identical to the I2V header presented above. The geographic reply header is a 76 byte header containing required to route the reply back to the request source. As the query reply header fields are identical to those in the I2V header they will not be covered in depth in this section. Readers may inquire as to why two identical headers are included; this is primarily explained in two points. Firstly, separation of concerns; if changes are made to the I2V header GeoQuery routing will not be affected, conversely changes to the query reply header will not affect internet routing headers. Secondly, simplification of implementation, while the I2V header could have been re-used routing logic would become more complex as each routing decision would require checking on the type of transmission. The increased complexity would increase processing time and latency of transmissions. A separate header increases duplication of efforts however it decreases function complexity and allows for future changes without effecting other routing functionality.

4 Evaluating Simulation Frameworks: Network Simulation, Mobility Simulation and Ns-3 Challenges

This chapter covers the importance of network simulation, the details of selecting a network simulator, node mobility generation and finally a central ns-3 challenge is reviewed to give readers interested in using ns-3 for future research insight into development.

Section 4.3.3 reviews the importance of network simulators, provides a description of ns-3 which was used to develop and evaluate our proposal. Section 4.4 reviews the Open Street Map project and its usage to create networks in our evaluation. Section 4.5 reviews the Simulation of Urban Mobility project and its usage to generate vehicular mobility trace files to simulate vehicular mobility in ns-3. Finally vital implementation challenges and solutions to creating new protocols which exchange data based on packet headers are covered in section 4.6.

4.1 Network Simulation and Simulator Evaluation Criteria

Accurate network simulation is vital to research efforts on any layer of the network stack. One of the major reasons accurate network simulations are required is due to the extreme difficulty of creating non-simulated networks, for example creating a VANET without a network simulator would be prohibitively difficult. Once network simulation has been chosen the primary focus must shift to simulation accuracy. A network simulator must accurately model the real world, without an accurate model of the world the results obtained from the simulator will not carry over once the research is implemented in a real network.

Network simulation has been an active research area for over 20 years, as such a multitude of simulation platforms have been developed. Over the years many of these platforms have evolved, merged or have been abandoned, today a handful of extremely accurate and well developed simulators exist for research purposes. One of the first steps in beginning research development work is choosing a network simulator. Each simulator has its own strengths and weaknesses, before choosing a simulator researchers must evaluate the strengths and weaknesses with respect to the research area. This thesis evaluated simulators based on ease of use and modification, mobility simulator support, documentation and community support.

Due to the extreme importance of mobility simulation in VANETs, mobility simula-

tion is covered in 4.2 before network network simulator evaluations are presented.

4.2 Mobility Simulation

Mobility simulation is an extremely important aspect of VANET simulation and modeling. Classic mobility models such as random way-point or random mobility models are inappropriate as they do not capture the unique elements of VANET node mobility. VANET movement does not conform to standard mobility patterns, movement is based off of road intersections, inter-vehicle interaction and road signs. All of these elements must be incorporated into a mobility simulator and as a consequence, VANET node mobility simulation is exceedingly complex and requires its own dedicated simulator. Rather than duplicating the efforts of past work by re-writing mobility simulators in network simulators designers follow one of the following three mobility simulation models:

4.2.0.2 Isolated Mobility Model:

Isolated mobility models completely separate network simulation and mobility simulation. Mobility scenarios must be input to the mobility model, based on those scenarios the mobility model generates a mobility trace file detailing node movements at each timestep. The trace file is then input into the network simulator which parses and plays back the the specified movements. Many current research platforms employ this model

as it is the simplest of the three models and provides accurate node behavior during evaluation. Network and mobility research work is entirety decoupled, as long as trace files may be generated the two areas of research are not required to collaborate and work may proceed independently. This model has the disadvantage of being inflexible, once trace file has been loaded by the network simulator node movements cannot be modified. Ns-3 follows the model leveraging the mobility simulator reviewed in 4.5 to generate trace files [20].

4.2.0.3 Embedded Mobility Model:

Embedded mobility models attempt to merge a network simulator with a mobility simulator into a single simulator. This model allows two way communication between the network simulator and mobility simulator. Due to the complexity of maintaining both mobility and network simulation simulators using model are limited in functionality. This class of mobility model is currently uncommon in research areas [20].

4.2.0.4 Federated Mobility Model:

Federated mobility models attempt to combine both isolated and embedded mobility models. Network simulation and mobility simulation maintained through separate simulators joined together by a third piece of software acting as a bridge between the two simulators. The federated model allows work to proceed independently on both sim-

ulators as long as the interfaces used by the bridging software are maintained. Using the bridge, this model allows two way communication between network and mobility simulators [20].

4.3 Network Simulators

This section reviews several network simulators considered during the simulator evaluation phase of this thesis. Each evaluation focuses on the criteria outlined in section 4.1.

While the final implementation and evaluation of the proposal is performed in ns-3 several other simulators were considered before choosing ns-3 as the final simulation platform. During evaluation of simulators ease of use and modification, VANET support, mobility simulator support, documentation and finally accuracy were considered. Sections 4.3.1, 4.3.2, 4.3.2.1 and 4.3.3 provide an analysis and evaluation of the considered network simulators.

4.3.1 QualNet and GloMoSim

QualNet was one of the first simulators evaluated, QualNet is the commercialized version of the GloMoSim simulator. At the time of writing GloMoSim is no longer under active development and has fallen outside of active usage. While QualNet is a commercial product, it is open source and allows extension for research purposes. In the

area ease of use QualNet contains analysis and network architecture tools making network setup and result analysis much simpler than on other platforms. However, QualNet supports from several major ease of use deficiencies, firstly the licensing model and secondly the development environment. Being a commercial product QualNet requires a license, licenses are managed through a centralized licensing server. Early in the simulator evaluation licensing difficulties were encountered making evaluation extremely difficult. The licensing requirement further complicates development work as it prevents multiple installations of QualNet to run simulations running in parallel. The second major deficiency is the lack of an IDE on Linux or OSX based systems where development would take place due to support requirements from mobility simulation tools. Examining the ease of modification, QualNet is entirely open source and may be freely modified, however QualNet does not provide tools to assist with modification efforts. Mobility simulation is a severe deficiency in QualNet, QualNet does not support an interface the SUMO mobility simulator further discussed in 4.5. SUMO has become the de-facto standard for mobility simulation for VANETs, a lack of SUMO support is an extreme blow to the usability of QualNet for VANET simulation. Examining QualNet documentation, QualNet provides extremely detailed on all aspects of the platform. The final area of investigation is the community support, QualNet features professional support however there does not appear to be a large collaborative online community. Due to the ease of use and mobility simulation deficiencies QualNet was not used in this thesis.

4.3.2 Ns-2

Ns-2 is one of the oldest simulators still in use, ns-2 has been actively used by the research community for over 20 years. Ns-2 is entirely open source and is a free product, due to the extensive use of ns-2 its accuracy has been confirmed multiple times. Due to the rich history of ns-2 it was considered as a candidate for the simulation platform however ns-2 has started to become outdated and support for ns-2 has dwindled. As one of the goals of this thesis is to develop a platform for future VANET research work, ns-2 was not used due to its near end of life status.

4.3.2.1 OMNET++

OMNET++ and its commercial version OMNEST were the initial choices for implementation and evaluation work on this thesis. Like ns-3 OMNET++ is a discrete event simulator designed for research. Similar to ns-3 much of the development work on OMNET++ has been network oriented though OMNET++ is a more general event simulator and has been applied to a greater variety of fields. OMNET++ uses C++ as its modeling language, networks are defined using a proprietary scripting language referred to as *Network Description* (NED). The defining feature of OMNET++ is its modularity, both at the network model level and at the simulator layer. Network models and components are defined by simple modules, these modules are combined to form compound mod-

ules defining a single component, a component such as a network interface will be a compound module composed of multiple simple and compound modules. Similarly the OMNET++ simulator is extremely module, OMNET++ alone is simply an event simulator it requires modules referred to as frameworks to run atop of the simulator to perform any meaningful simulation, notable frameworks evaluated throughout the course of this thesis include the INET framework and the VEINS framework. OMNET++ is entirely open source and may be freely modified making it ideal for research work, it is actively used and developed by researchers world wide, it features its own IDE and provides a great deal of VANET support with the VEINS and INET frameworks.

The VEINS framework is particularly important for VANET research, it provides a federated interface between SUMO and OMNET++. The federated mobility simulation provides researchers with powerful tools when evaluating protocol performance in VANETs, using feedback from developed protocols and applications, mobility patterns may be actively modified during simulations. Federated mobility models also expose more information about the network to VANET nodes, features such as road IDs, intersection state and vehicle status are accessible to simulations.

Due to the great deal of VANET support in OMNET++, OMNET++ was originally chosen as the simulation platform however it was abandoned early in the development process. Two major issues were encountered which prompted the change of platform: incomplete documentation and difficulties integrating frameworks. At the time of initial

evaluation large pieces of the documentation were missing or slated to be completed at a later unspecified date. The lack of documentation made it quite difficult to begin development work in OMNET++ however this issue was somewhat mitigated through the examples and tutorials provided by OMNET++. The major point of difficulty which ultimately necessitated a change of platform was the inability to integrate the VEINS framework with the INET framework. The VEINS framework provides VANET support while the INET framework provides network simulation support, these two frameworks are capable of working in tandem however due to the lack of documentation they could not be successfully integrated into a project. Without the frameworks integrated, research could not proceed using the OMNET++ simulator.

4.3.3 Ns-3

The proposed routing protocol is implemented and simulated in ns-3 version 3.24, detailed ns-3 documentation, examples and the simulator itself may be accessed at [31]. Ns-3 is the spiritual successor to ns-2, it attempts to improve the architecture, models and educational components defined in ns-2. While ns-3 is the successor to ns-2 the two projects are not cross compatible, C++ components of ns-2 have been ported to ns-3 however OTcl models in ns-2 are entirely incompatible with ns-3.

Ns-3 is discrete event network simulator primarily designed for research and educational purposes. Ns-3 was officially released in 2008 and has been actively updated since

release, a new stable release is distributed every 4 months. At its core ns-3 is an event simulator and may be used for any number of applications however the majority of its use and development has focused on IP network simulation. Ns-3 provides support and models for network layers one, two, three and four. All provided models are entirely open source and may be freely modified.

Evaluating ns-3, falling under ease of use ns-3 provides multiple tools to assist researchers with development efforts. Firstly the ns-3 build system is directly compatible with the populate Eclipse IDE, compatibility with Eclipse immediately grants researchers access to all of the debugging tools offered by Eclipse. A second ease of use tool provided by ns-3 is an integrated interface with gdb; a stack trace utility. Stack tracing capability provides researchers an extremely powerful tool when investigating errors such as segmentation faults. Falling under ease of modification ns-3 provides several tools to assist researchers. The first tool is the ns-3 logging system, ns-3 integrates a multi-level logging system allowing researchers to log simulation events ranging from vital to trivial. The logging system is entirely customization and may be tailored to the required usage. The second ease of modification feature again comes from the IDE, due to the compatibility with a fully featured IDE modification of the ns-3 source code is greatly simplified through the use of IDE tools such as *content assist*. Examining mobility simulator support ns-3 supports the isolated mobility simulation model, traces may be generated by SUMO and input into ns-3. While the isolated model of ns-3 is inferior to that of OM-

NET++ it was deemed acceptable as the proposal does not contain hard requirements on modification of mobility patterns during simulations. Examining ns-3 documentation, a great deal of documentation detailing the ns-3 architecture is provided, however the greatest strength of ns-3 documentation is its online API and source code. The entire ns-3 API and source code is available organized in several search-able and navigate-able formats online. Examining the final area of community support, ns-3 boasts a large online community. Using the provided forums researchers may tap into the knowledge of many researchers around the world for ns-3 support. As ns-3 meets all evaluated criteria it was chosen as final the simulation platform.

4.4 Open Street Map Database (OSM)

The OSM database is a community driven database providing detailed topology information on real world maps [16]. The OSM database provides researchers with an interface through which researchers may export maps in an xml style output file. The exported file may then be used as the underlying network when creating mobility traces using a VANET mobility simulator.

4.5 Simulation of Urban Mobility (SUMO)

The SUMO simulator is an open source traffic simulator, based on input parameters and an underlying network, SUMO simulates vehicular movements along that network. SUMO has been extensively tested and is commonly used in both research and real world traffic modeling. Using the network file generated by the OSM database SUMO models vehicular movements along that network, the movements are extremely detailed including details such as acceleration and stopping times. Mobility is further modeled by including details such as traffic lights, stop signs, roundabouts and other various conditions encountered on roads [4].

4.6 Ns-3 Challenges and Solutions: Header Management

This section reviews one of the major challenges encountered during the implementation and evaluation of the proposal. Readers interested in extending the work in this thesis or in using ns-3 in their own research work should take particular note of this section. One of the major challenges encountered during the development of this thesis was header management. Before the challenge can be effectively described a high level view of routing in ns-3 must be reviewed.

4.6.1 Ns-3 Routing Overview

Routing in ns-3 is primarily handled through two common interfaces, the *Route Input* and *Route Output* functions. The route output function is used when a node attempts to send a packet from an upper layer. The route input function is used when a packet had been received on one its interfaces and has traversed up the network stack to the routing layer, the route input function is responsible for either passing the packet further up the network stack or to make a routing decision to forward the packet towards its final destination.

4.6.2 Header Management: Challenge Description

Following the standard network stack, packets are assumed to be fully formed when received from an upper layer, however ns-3 does not follow this paradigm. When the route output function is called by the transport layer it passes an incomplete packet. The incomplete packet contains the raw packet data however the transport layer header has not been attached at this stage, the transport layer header is only attached when the route output function returns a valid route. This creates a major issue in any protocol which uses headers to pass data between routing protocols. If headers are attached within the route output function they will be attached behind the transport layer header, other nodes will not be able to read these headers. Based on this challenge only two effective

solutions were found, first the core of ns-3 could be modified to change the header order or any protocols using headers to transfer data must intercept the packet once it has been fully formed. The first solution is a viable solution however it is extremely undesirable, firstly modifying the ns-3 core makes any developed work incompatible with the rest of the research community and secondly modifying the ns-3 core moves the simulator further away from accurately simulating the network stack. This leaves the second option of intercepting the fully formed packet.

4.6.3 Header Management: Solution Description

Without modifying the ns-3 core the only other function within the routing protocol with access to a packet is the route input function. The route input function is only called when a packet is received from a lower layer, as a result the packet must have passed through the route output function on some node creating a well formed packet. Using the fact that packets passing through the network stack are only fully formed after passing through the route output function but are received fully formed by the route input function our proposal uses the following interception strategy.

When packets are generated by a node they are first encountered by the route output function, here the packets are tagged to identify their origin and are sent to the loopback interface. Packets traverse the network stack and are sent back up the network stack by the loopback interface where they are eventually received by the route input function.

Using the tag and the received interface the route input function identifies that packet as originating from the current node. The route input function receives the fully formed packet, removes the attached tag, attaches the correct routing headers and correctly routes the packet towards the destination. Using the route input function causes extra processing however it allows fully formed packets to be intercepted and successfully modified.

5 Performance Analysis

This chapter reviews the performance of our proposal, specifically focusing on the routing elements. Following each performance evaluation, results are analyzed and discussed.

5.1 Evaluation Metrics

The development of any protocol at any layer of the network stack must establish metrics to evaluate the proposal against existing work. Due to the lack of standardization in V2V and hybrid routing protocols designed for VANETs, little work has been done to create standard VANET routing libraries in popular network simulators. Due to the lack of VANET protocol support in popular simulators we evaluate our protocol against AODV, DSDV and OLSR as described in Chapter 2. These protocols were originally designed for MANETs however each of these standards has over a decade of refinement and optimization, based on the available library of protocols these are the best candidates for comparison for our protocol.

The following evaluation metrics are used:

1. Average Round Trip Time (RTT)

   The RTT represents the time between sending a packet and receiving a reply, we calculate the average of this value over all successfully received packets. The value should be kept as low as possible. The RTT is calculated through Equation 5.1.

$$\frac{\sum_1^n ReceiveTime - SendTime}{NumberofSuccessfulTransmissions} \tag{5.1}$$

2. Successful Packet Delivery Ratio (PDR)

   The PDR represents the percentage of successful transmissions; a successful transmission is defined as complete request-reply cycle.

$$\frac{PacketsSuccessfullyReceived}{TotalPacketsSent} \tag{5.2}$$

3. Average Hop Count - VANET to RSU

   This metric represents the average number of times a packet must be forwarded before it reaches an RSU. This metric should be as low as possible however it must be interpreted in conjunction with the PDR. A low hop count with a low successful packet delivery ratio indicates distance node could not reach the RSU, giving this metric a deceptively low value as further nodes will never actually have their routing metrics recorded.

4. Average Hop Count - RSU to VANET

   This metric represents the average number of times a packet originating from an RSU must be forwarded before it reaches the target VANET node.

5. Network Throughput

   Protocol throughput represents the total number of bytes successfully transmitted through the network. Network throughput is calculated by Equation 5.3, routing protocols should aim to maximize network throughput as this would represent more data successfully routed through the network.

$$\frac{NumberOfReceivedPackets}{Simulationtime} xPacketSize \tag{5.3}$$

5.2 Scenario Description

The evaluation of the proposed enhancements take place over two geographic regions, an urban and rural region. Each evaluated region is approximately 4 KM^2 and is split into four geographic zones each 1 km^2. An RSU is placed at the center of each zone serving at the gateway from the VANET to other networks, both distant VANETs and the Internet. Urban environments are classified by dense roads spaced apart at regular intervals. Urban roads form quadrilateral regions, forming regular rectangular regions. Density patterns range from low traffic, low density conditions to high traffic, high density peri-

ods. Rural environments are classified by sparse roads at irregular intervals. Vehicular density patterns are expected to reach low and medium density conditions however high density conditions are unlikely to occur. The proposed enhancements are evaluated in three ways, firstly they are evaluating routing pure Internet bound traffic, using the hybrid routing model packets are routing towards the current zone RSU to reach the nodes Internet gateway. The second form of evaluation takes place with the proposed protocol routing both safety control messages, safety message dissemination and Internet bound traffic. The third form of evaluation combines pure Internet routing, safety message routing and GeoQuery routing. Providing a basis for comparison OLSR, DSDV and AODV are evaluated in each scenario alongside the proposed protocol. Providing an additional comparison point, AODV is evaluated using both active neighbor discovery through the broadcast of hello messages as well as passive neighbor discovery without hello message broadcast. Both urban and rural scenarios are evaluated for successful packet delivery ratios, packet round trip times, and average hop counts both to and from the zone RSU. Table 5.1 shows a summary of the simulation parameters.

Following the evaluation of the two geographic scenarios a throughput analysis is performed to determine the limits of the proposed protocol as compared to the best performing existing protocol. Evaluation is performed in a medium density urban environment where network traffic is continuously doubled until the limits of the evaluated protocols is reached. The average throughput is calculated based on the number of pack-

Table 5.1: Simulation Parameters

Field	Value
Packet Sending Rate	1-2 per second
MAC Protocol	802.11a
Percentage of Nodes Transmitting	10-100%
Mobility Pattern	SUMO generated random trips 10 Intermediate points per trip
Packet Payload Size	1024 bytes
Simulation Time	120 seconds
Vehicular Densities	25,50,100 per KM^2
Contention Constant	10 μs
Transmission Power	25 dB

ets traversing the network both towards the RSU from the VANET and from the RSU back into the VANET.

5.3 Urban Environment - Low Density

This section presents the evaluation of the proposed protocol in a low density urban environment. Each node sends 1-2 packets per second; inter-packet delays are varied between each transmission to ensure transmissions do not become synchronized. Each simulation runs for 120 seconds. Nodes do not begin sending packets until 5 seconds into the simulation allowing them to begin their mobility before transmitting.

5.3.1 Successful Packet Delivery Ratio

Figure 5.1 shows the average successful packet delivery ratio. Successfully transmissions are defined by a node sending a request and receiving a reply to that request. Table 5.2 further expands the evaluation shown in Figure 5.1 by providing the average standard deviation and the average successful packet delivery ratio over all trials for each protocol.

Based on Figure 5.1 and Table 5.2 several observations are made, first it can be seen that the proposed protocol outperforms all other evaluated protocols. The next highest performing protocol in this scenario is AODV with periodic messages disabled, though the proposed protocol achieves a 26% improvement over AODV. Secondly it can be seen that the standard deviation is quite high over all tested protocols. Due to the similarity

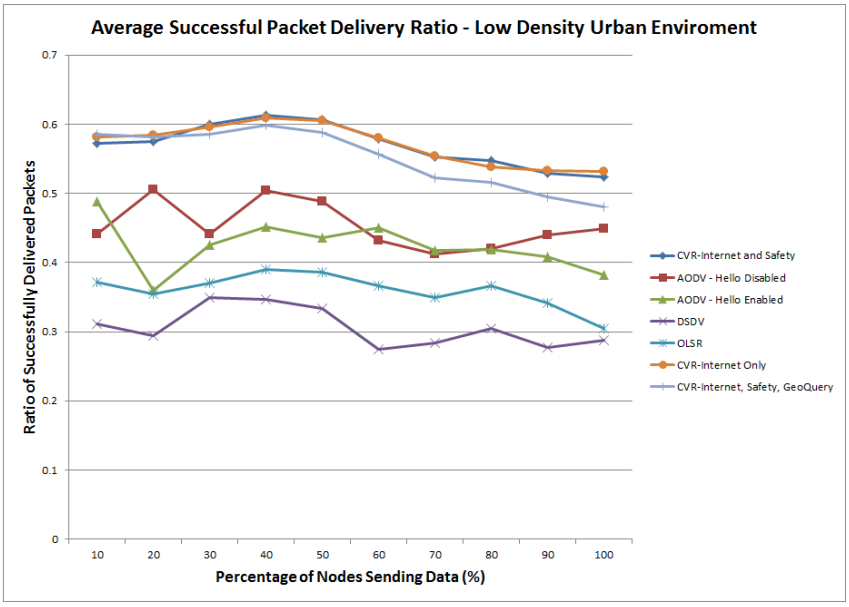

Figure 5.1: Low Density Urban Environment Packet Delivery Ratio

between the standard deviations between all protocols it appears that the deviation value is a property of the mobility pattern rather than a property of the protocols themselves. Thirdly a comparison of pure Internet traffic, Internet and safety and finally Internet, safety and GeoQuery performed is examined. Comparing pure Internet traffic to Internet and safety traffic very little difference in the ratio of successfully received packets is seen. This indicates that the safety message traffic is sparse enough that it does not have a significant effect on other forms of traffic. Comparing pure Internet traffic and a combination of Internet, safety and GeoQuery traffic a more pronounced difference in performance is seen. GeoQuery traffic causes an increase in contention as the contention process must take place at both the source and destination. The increase in contention

Table 5.2: Urban Environment Low Density PDR Summary

Protocol	Average PDR Over All Trials	Standard Deviation
CVR Internet and Safety	0.57	0.18
AODV Hello Disabled	0.45	0.20
AODV Hello Enabled	0.42	0.21
DSDV	0.30	0.20
OLSR	0.36	0.20
CVR Internet Only	0.57	0.18
CVR Internet, Safety, GeoQuery	0.55	0.18

creates a larger strain on network resources resulting in a lower successful delivery rate, at the largest point of difference a 4% difference is seen between pure Internet traffic and a combination of all traffic types. The fourth and final takeaway from these results is the relatively low average delivery rate, this indicates that the low density environment prevents nodes from successfully forming complete links to their respective RSU.

5.3.2 Average Round Trip Time (RTT)

Figure 5.2 shows the average RTT for a request, the RTT represents the time between sending a request and receiving the reply. Table 5.3 provides further information, show-

ing the average RTT over all trials and the average standard deviation.

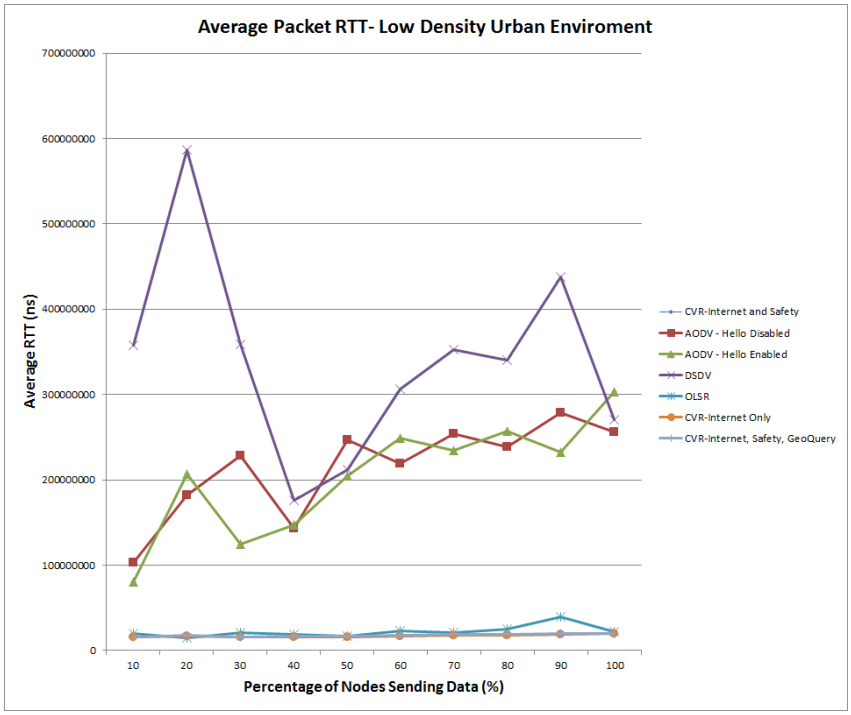

Figure 5.2: Low Density Urban Environment RTT

Based on Figure 5.2 and Table 5.3 several observations are made, firstly compared to all protocols other than OLSR the proposed routing protocols RTT is approximately 10 times less than all other protocols. One possible reason for this might be the route discovery procedure in both AODV and DSDV, packets are delayed until routes are found. The proposed routing protocol does not delay packets until routes are found and therefore it is not effected by these effects. The second observation is the gradual rise in RTT, this is especially noticeable in AODV simulations where the RTT value nears tripling between 10% of nodes transmitting and 100%. The proposed routing protocol does not appear

Table 5.3: Urban Environment Low Density Average RTT Summary

Protocol	Average RTT Over All Trials	Average Standard Deviation
CVR Internet and Safety	17204106	8518217
AODV Hello Disabled	214844329	223349548
AODV Hello Enabled	204037541	215326110
DSDV	340046416	688736888
OLSR	22084008	27225306
CVR Internet Only	16954549	7362230
CVR Internet, Safety, GeoQuery	17867537	9021076

to suffer from increases in RTT remaining near constant throughout all trials, similarly OLSR does not appear to suffer from increased RTT. The addition of safety messages and GeoQuery messages has a minimal effect on the proposed protocol RTT.

5.3.3 Average Hop Count

The average hop count is divided into two metrics, average hop count from the node towards the RSU and average hop count from the RSU towards VANET nodes. Figures 5.3 and 5.4 show the average number of hops required to reach a VANET node from an RSU and the average number of hops required to reach an RSU from a VANET node.

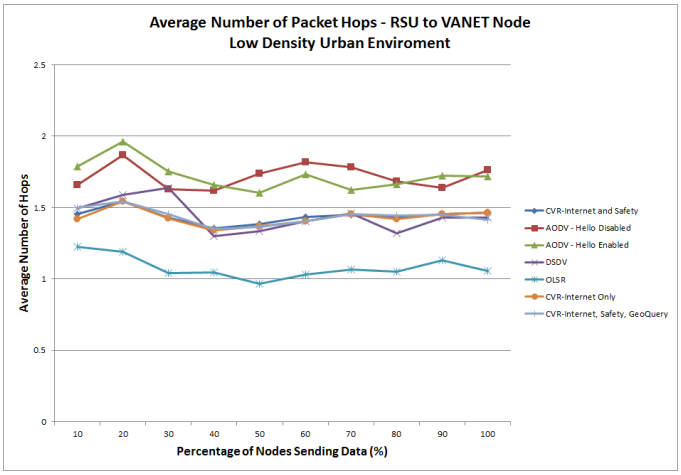

Figure 5.3: Low Density Urban Environment Average Hop Count RSU to VANET

Based on the geometry of a zone and the placement of an RSU the optimal routing solution will use a single hop, a single hop allows a node to reach an RSU from the furthest point in a geographic zone. VANETs are non-ideal environments and it is expected that the ideal 1 packet hop value will not be reached. All evaluated protocols achieve a packet hop value of approximately 1.5 indicating all protocols find near ideal routes. OLSR achieves the lowest hop count value however due to its low successful delivery rate, this corresponds to OLSR simply not creating effective multi-hop paths.

5.4 Urban Environment - Medium Density

This section presents the evaluation of the proposed protocol in a medium density urban environment. Each node sends 1-2 packets per second, inter-packet relays are varied

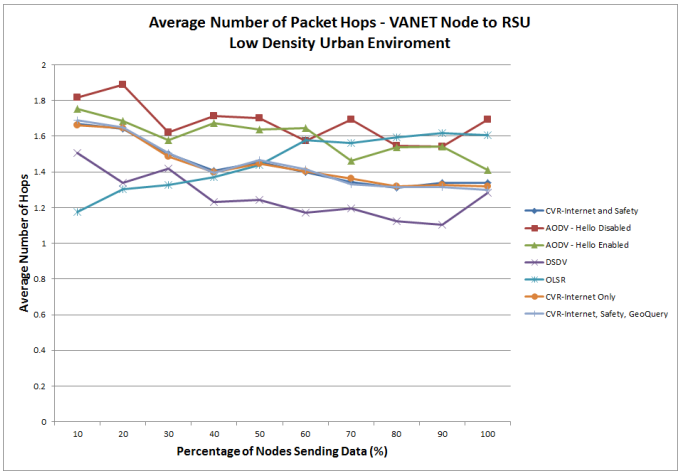

Figure 5.4: Low Density Urban Environment Average Hop Count RSU to VANET

between each transmission to ensure transmissions do not become synchronized. Each simulation runs for 120 seconds. Nodes do not begin sending packets until 5 seconds into the simulation allowing them to begin their mobility before transmitting.

5.4.1 Successful Packet Delivery Ratio

Figure 5.5 shows the average successful packet delivery ratio. As before, successfully transmissions are defined by a node sending a request and receiving a reply to that request. Table 5.4 further expands on the evaluation showing the average standard deviation and average packet delivery ratio over all trials for each protocol.

Examining Figure 5.5 and Table 5.4 several trends are seen. First the proposed contention based protocol outperforms all evaluated protocols, even when routing both internet and safety traffic the proposed protocol outperforms the next best protocol by 10%.

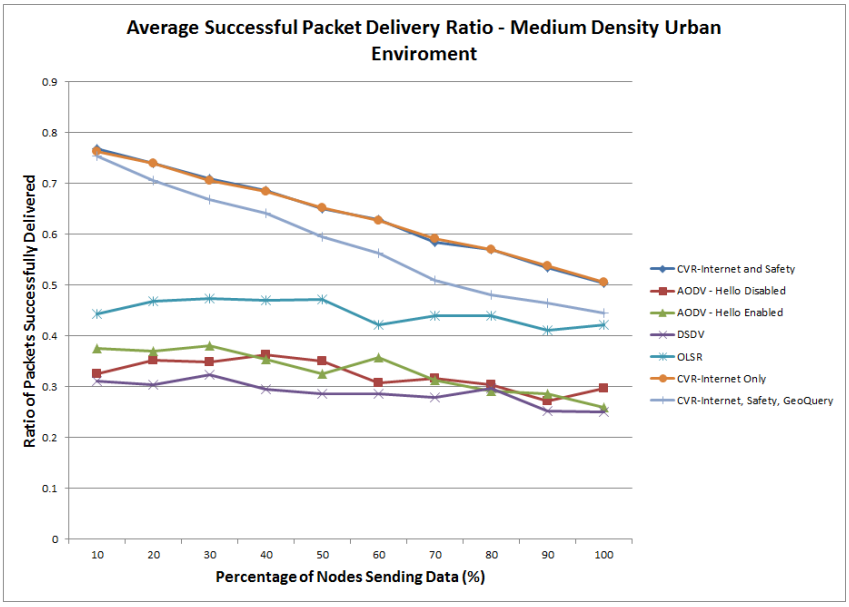

Figure 5.5: Medium Density Urban Environment Packet Delivery Ratio

Further increasing the routing load of the proposed protocol through the addition of GeoQuery messages sees a decrease in the proposed routing protocol performance due to an increase in contention. When 100% of nodes are transmitting the increased GeoQuery load causes a decrease of 5% in the delivery rate bringing the performance just above that of OLSR. The proposed protocol performs especially well when few nodes are transmitting packets, the vehicular density is high enough to successfully create multi-hop paths between VANET nodes and RSUs while the low amount of nodes transmitting data keep contention low resulting in a high level of success. As the number of transmitting nodes increases the performance of the contention based routing protocol drops, though even at the point where all nodes within the simulation are transmitting the performance is

Table 5.4: Urban Environment Medium Density PDR Summary

Protocol	Average Delivery Ratio Over All Trials	Average Standard Deviation
CVR Internet and Safety	0.64	0.14
AODV Hello Disabled	0.32	0.20
AODV Hello Enabled	0.33	0.21
DSDV	0.29	0.22
OLSR	0.44	0.22
CVR Internet Only	0.64	0.14
CVR Internet, Safety, GeoQuery	0.59	0.15

still approximately 10% higher than the next highest protocol. Examining the average standard deviation, once again the standard deviation is quite high however the proposed protocol has reduced the standard deviation by 4% when compared to the low density environment while other protocol deviations have remained high. This indicates that the geographic topology continues playing a role in causing the high standard deviation however a higher node density has made the proposed routing protocol more consistent. Examining the data from an overall view it can be seen that the performance appears be to split into 3 classes; high, medium and low. The high performance class contains the proposed protocol, the medium class contains OLSR and all other protocols fall under

low performance. This trend appears to line up with neighbor management, the proposed protocol performs very little neighbor management and performs comparatively well. OLSR performs optimized neighbor management, only maintaining a subset a neighbors, this appears to be an advantage to OLSR. The remaining protocols perform meticulous neighbor management which appears to cause significant performance degradation.

5.4.2 Average Round Trip Time (RTT)

Figure 5.6 shows the average RTT for a request, the RTT represents the time between sending a request and receiving the reply. Table 5.5 further expands the evaluation and shows a summarized view of the RTT and average standard deviations over all trials.

Based on Figure 5.6 and Table 5.5 similar trends to the low density simulation are seen. Once again the RTT for the proposed protocol is quite low, approximately 2-3 times lower than all protocols other than OLSR. While 60% or less of all nodes are transmitting the RTT of OLSR and the proposed routing protocol are almost identical, past 60% OLSR RTT outperforms the proposed protocol however the trade off for the lower RTT is a decrease in the number of successfully received packets. As seen in the low density scenario the addition of safety messages slightly increases the RTT, safety messages are sparse enough that they cause a minimal increase in the delay. The addition of further increases the load on the channel causing an increase the in RTT.

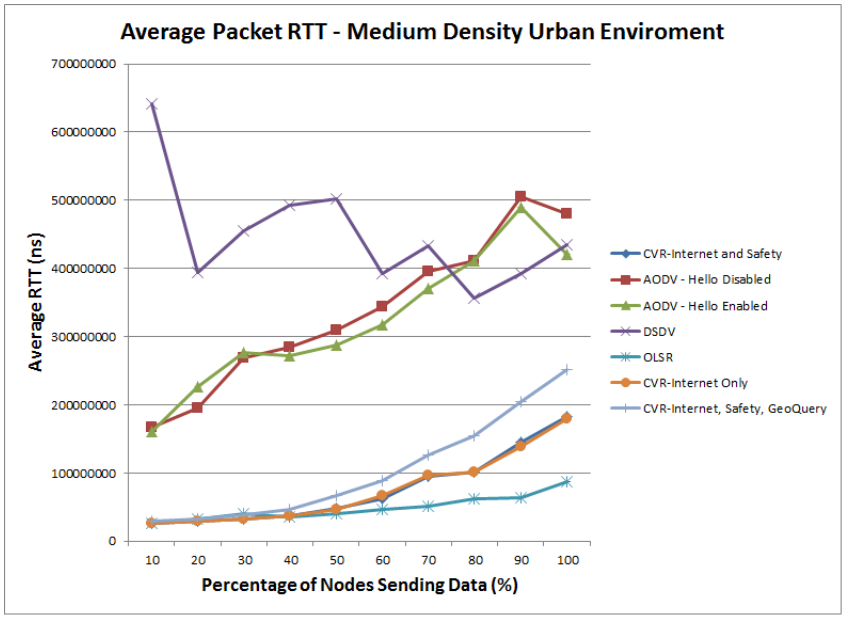

Figure 5.6: Medium Density Urban Environment RTT

5.4.3 Average Hop Count

Once again the average hop count is divided into two metrics, average hop count from the VANET node towards the RSU and average hop count from the RSU towards VANET nodes. Figures 5.7 and 5.8 show the average number of packet hops required to reach a VANET node from an RSU and the average number of hops required to reach an RSU from a VANET node.

Similar to the low density scenario the average number of packet hops are between 1 and 2 hops. The most distant nodes in a zone should be able to reach an RSU within one hop given ideal node placement. All evaluated protocols packet hop counts lie between

Table 5.5: Urban Medium Density RTT Summary

Protocol	Average RTT Over All Trials	Average Standard Deviation
CVR Internet and Safety	75813731	55740086
AODV Hello Disabled	335959149	291125642
AODV Hello Enabled	323016607	261738781
DSDV	449339476	618840087
OLSR	48305912	45610152
CVR Internet Only	75051152	55845089
CVR Internet, Safety, GeoQuery	103463033	76457282

1 and 2 hops indicating all evaluated protocols find effective routes.

5.5 Urban Environment - High Density

This section presents the evaluation of the proposed protocol in a high density urban environment. Each node sends 1-2 packets per second, inter-packet relays are varied between each transmission to ensure transmissions do not become synchronized. Each simulation runs for 120 seconds. Nodes do not begin sending packets until 5 seconds into the simulation allowing them to begin their mobility before transmitting.

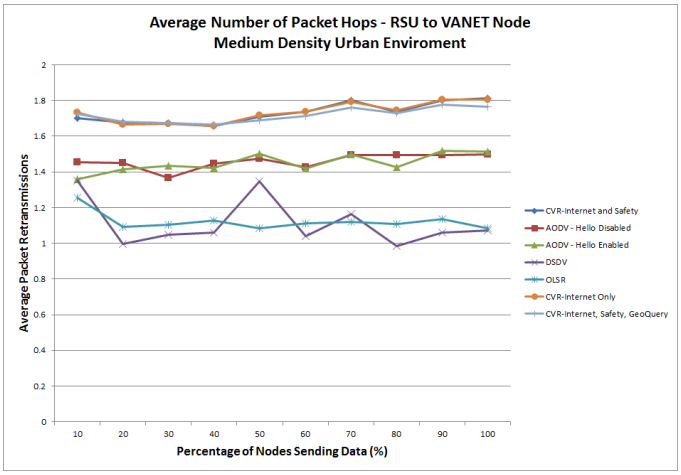

Figure 5.7: Medium Density Urban Environment Average Packet Hop Count RSU to VANET

5.5.1 Successful Packet Delivery Ratio

Figure 5.9 shows the average successful packet delivery ratio. As before, successfully transmissions are defined by a node sending a request and receiving a reply to that request. Expanding on the results of Figure 5.9, Table 5.6 shows the average standard deviation and average packet delivery ratio over all trials for each protocol.

Examining the performance evaluation given in Figure 5.9 and Table 5.6 several trends are seen. Examining the proposed protocol routing pure Internet and Internet and safety messages the ratio of successfully received packets begins over 20% higher than all other protocols. As the number of transmitting nodes increases the average ratio of successfully received packets gradually decreases, between 45-50% the ratio of

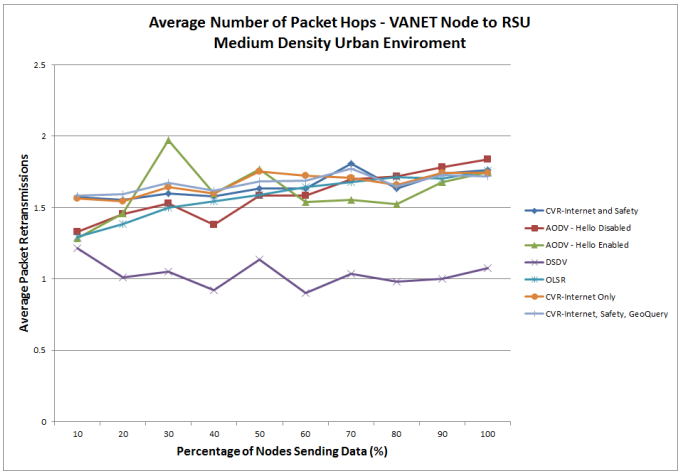

Figure 5.8: Medium Density Urban Environment Average Packet Hop Count VANET to RSU

successfully received packets reaches parity with OLSR. Between 50 and 100% the proposal protocol experiences rates within 1-3% of OLSR performance. This set of results shows that at high densities the proposal protocol route discovery performs extremely well as it may form multiple paths to the destination. As the number of transmitting nodes increases the performance suffers as contention processes are disrupted. Possible proposal improvements to increase performance are discussed in section 6.3. The addition of GeoQuery messages to safety and Internet messages further saturates the channel causing an increase in contention difficulties. Comparing the results of the proposed protocol to AODV and DSDV it can be seen that the proposed protocol outperforms AODV and DSDV in all evaluation environments.

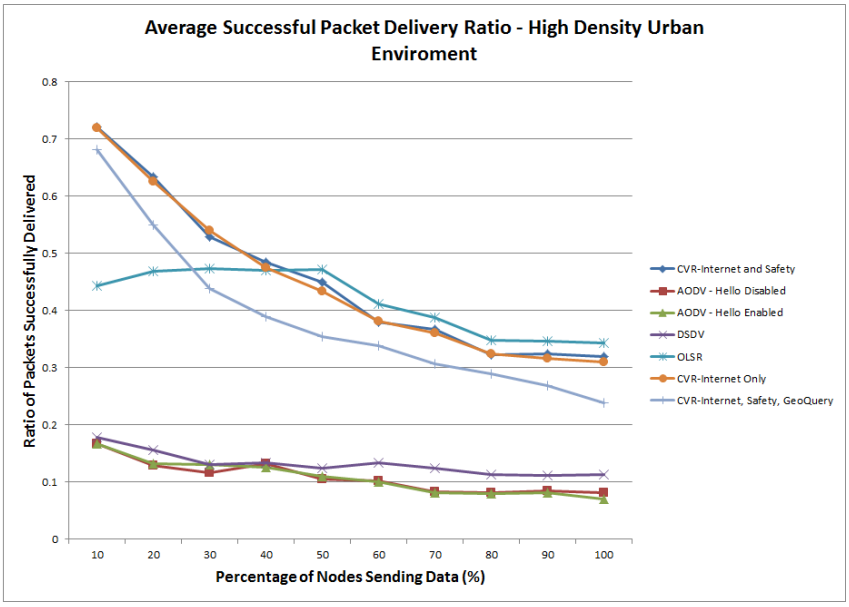

Figure 5.9: High Density Urban Environment Average Packet Delivery Ratio

5.5.2 Average Round Trip Time (RTT)

Figure 5.10 shows the average RTT for a request, the RTT represents the time between sending a request and receiving the reply. Table 5.7 shows a summarized view of the RTT and average standard deviations over all trials.

Based on Figure 5.10 and Table 5.7 the primary trend seen is the increasing RTT. Relating the increasing RTT to the decreasing successful delivery rate more evidence is seen showing the MAC protocol as the underlying cause the reduction of the ratio of successfully received packets. As the delay increases contention cannot occur correctly thus causing unnecessary forwarding, further increasing congestion in the channel.

Table 5.6: Urban Environment High Density PDR Summary

Protocol	Average PDR Over All Trials	Average Standard Deviation
CVR Internet and Safety	0.45	0.12
AODV Hello Disabled	0.10	0.13
AODV Hello Enabled	0.10	0.13
DSDV	0.13	0.16
OLSR	0.41	0.21
CVR Internet Only	0.45	0.11
CVR Internet, Safety and GeoQuery	0.39	0.15

5.5.3 Average Hop Count

Once again the average hop count is divided into two metrics, average hop count from the node towards the RSU and average hop count from the RSU towards VANET nodes. Figures 5.11 and 5.12 show the average number of packet hops required to reach a VANET node from an RSU and the average number of hops required to reach an RSU from a VANET node.

Examining the packet hop counts it can be seen that the proposed protocol experiences an increase in the number of hops when compared to the low and medium density scenarios. The average number of hops increases by approximately one hop. This in-

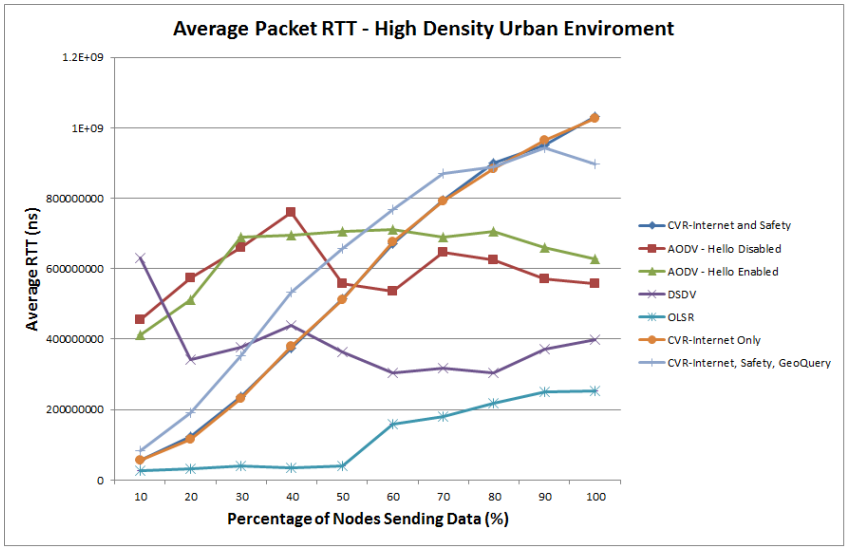

Figure 5.10: High Density Urban Environment RTT

crease is once again attributed to the overwhelmed channel. Due to the difficulties in the contention process, non-optimal nodes are able to win contention periods, these nodes are further from the target position thereby creating longer routes.

5.6 Rural Environment - Low Density

The following section presents the evaluation of the proposed protocol in a low density rural environment. Each node sends 1-2 packets per second, inter-packet relays are varied between each transmission to ensure transmissions do not become synchronized. Each simulation runs for 120 seconds. Nodes do not begin sending packets until 5 seconds into the simulation allowing them to begin their mobility before transmitting.

Table 5.7: Urban Environment High Density RTT Summary

Protocol	Average RTT Over All Trials	Average Standard Deviation
CVR Internet and Safety	565301138	341734901
AODV Hello Disabled	593773077	874529993
AODV Hello Enabled	640639375	420486078
DSDV	384678666	801239865
OLSR	123464021	90717455
CVR Internet Only	563662184	336759667
CVR Internet, Safety and GeoQuery	618158009	76457283

5.6.1 Successful Packet Delivery Ratio

Figure 5.13 shows the average successful packet delivery ratio. As before, successfully transmissions are defined by a node sending a request and receiving a reply to that request. Table 5.8 further expands on the results and shows the average standard deviation and average successful packet delivery ratio for each protocol.

Examining the results of the low density rural environment successful packet delivery ratio several trends similar to those seen in the urban environment are seen. All protocols maintains similar performance to that seen in the urban environment. The proposed protocol performs between 10 and 30 % better than all other evaluated protocols. The

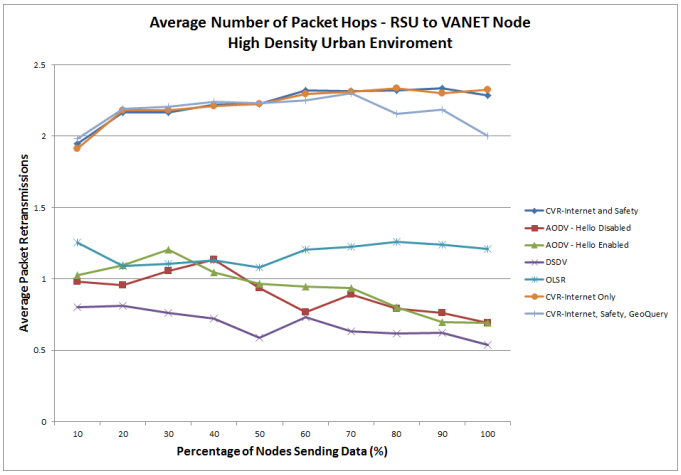

Figure 5.11: High Density Urban Environment Average Packet Hop Count RSU to VANET

addition of safety and GeoQuery messages lowers the successful packet delivery ratio by up to 5 % however even with reduced performance the proposed protocol continues to perform better than all of the tested alternatives. A notable difference is seen between the rural and urban environments, there appears to be an increase in the proposed protocol performance. Comparing the urban environment results to the rural environment results the proposed protocol achieves between a 5 and 15% higher successful packet delivery ratio. The largest difference is seen with few vehicles transmitting data, as the number of vehicles transmitting data increases the performance of the proposed protocol in the rural environment approaches that of the urban environment. The addition of safety messages does not appear to have a notable affect on the ratio of successfully received packets. However, when GeoQuery and safety messages are both introduced the channel

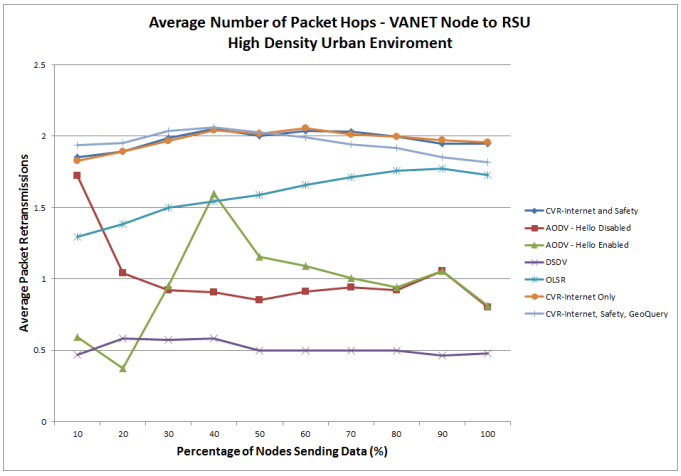

Figure 5.12: High Density Urban Environment Average Packet Hop Count VANET to RSU

becomes more congested causing difficulties with contention.

5.6.2 Average Round Trip Time (RTT)

Figure 5.14 shows the average RTT for a request, the RTT represents the time between sending a request and receiving the reply. Table 5.9 shows a summarized view of the RTT and average standard deviations over all trials.

Examining the results of the RTT in a rural environment the trends seen are extremely similar to those in the urban environment. The proposed protocol and OLSR both show RTTs of 5-10 times lower than all other evaluated protocols. The inclusion of safety control and emergency messages has little effect on overall RTT of the proposed protocol. Further increasing the routing load through the addition of GeoQuery messages the RTT

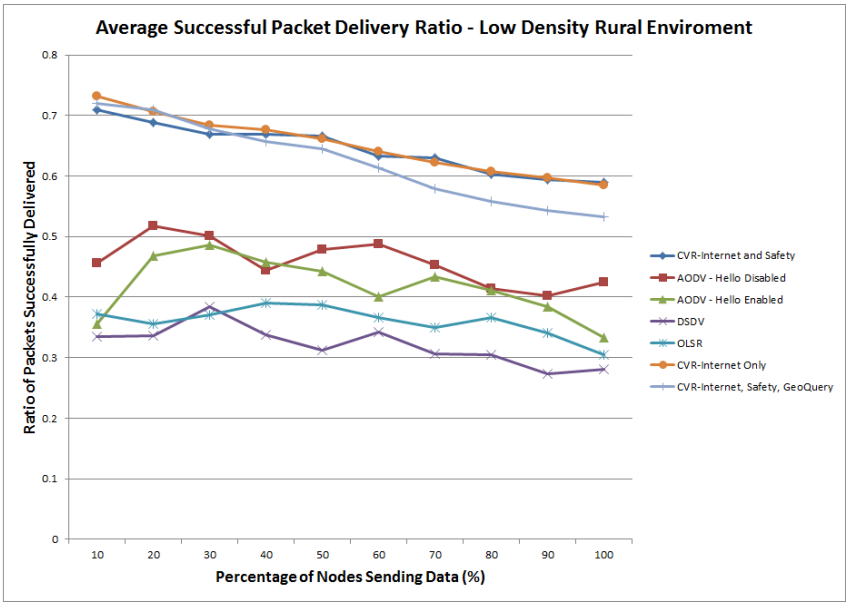

Figure 5.13: Low Density Rural Environment Packet Delivery Ratio

largely left unchanged.

5.6.3 Average Hop Count

Similar to the urban evaluation the average hop count is divided into two metrics, average hop count from the node towards the RSU and average hop count from the RSU towards VANET nodes. Figures 5.15 and 5.16 show the average number of hops required to reach a VANET node from an RSU and the average number of hops required to reach an RSU from a VANET node.

Similar to the urban results all protocols found routes between one and two hops. Once again this result indicates all protocols discovered equally optimal routes to the

Table 5.8: Low Density Rural Environment PDR Summary

Protocol	Average PDR Over All Trials	Average Standard Deviation
CVR Internet and Safety	0.64	0.18
AODV Hello Disabled	0.45	0.18
AODV Hello Enabled	0.41	0.20
DSDV	0.32	0.20
OLSR	0.36	0.20
CVR Internet Only	0.65	0.19
CVR Internet, Safety and GeoQuery	0.58	0.15

destination.

5.7 Rural Environment - Medium Density

This section presents the evaluation of the proposed protocol in a medium density urban environment. Each node sends 1-2 packets per second, inter-packet relays are varied between each transmission to ensure transmissions do not become synchronized. Each simulation runs for 120 seconds. Nodes do not begin sending packets until 5 seconds into the simulation allowing them to begin their mobility before transmitting.

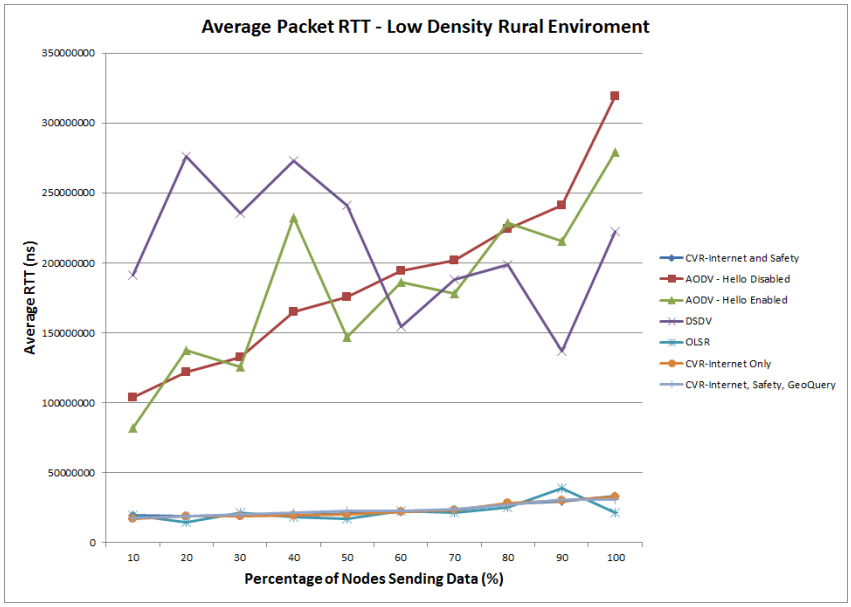

Figure 5.14: Low Density Rural Environment RTT

5.7.1 Successful Packet Delivery Ratio

Figure 5.17 shows the average successful packet delivery ratio. As before, successfully transmissions are defined by a node sending a request and receiving a reply to that request. Expanding on the results, Table 5.10 shows the average standard deviation and average packet delivery ratio for each protocol over all trials.

Examining the results of the medium density rural environment a mix between the urban medium and high density environments is seen. As seen with previous simulations the successful delivery rate begins much higher than all other protocols. As the percentage of nodes transmitting increases the successful packet delivery ratio decreases. The base routing protocol reaches parity with OLSR when all nodes within the simulation

Table 5.9: Low Density Rural Environment RTT Summary

Protocol	Average RTT Over All Trials	Average Standard Deviation
CVR Internet and Safety	23451825	11552181
AODV Hello Disabled	187903858	155048767
AODV Hello Enabled	181246829	193600391
DSDV	211755240	557282319
OLSR	22084008	27225306
CVR Internet Only	23087581	11346825
CVR Internet, Safety and GeoQuery	17867537	9021076

are transmitting. Adding GeoQuery traffic to the simulation causes the proposed routing protocol performance to drop below OLSR; at the highest number of transmissions a 4% difference is observed. Excluding OLSR all other evaluated protocols perform quite poorly in a medium density environment;

5.7.2 Average Round Trip Time (RTT)

Figure 5.18 shows the average RTT for a request, the RTT represents the time between sending a request and receiving the reply. Table 5.11 shows a summarized view of the RTT and average standard deviations.

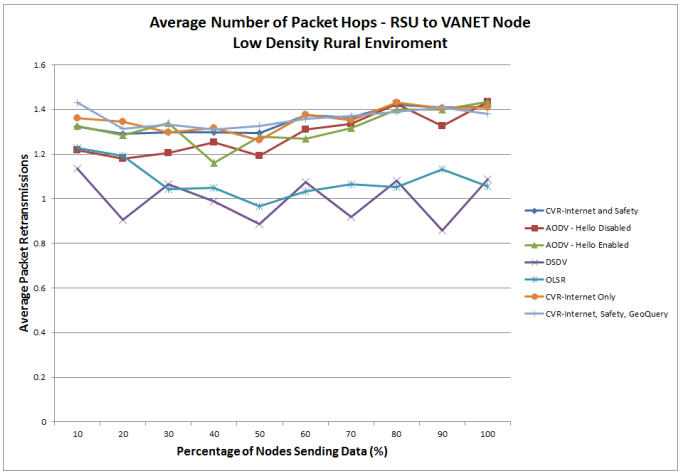

Figure 5.15: Low Density Rural Environment Average Packet Hop Count RSU to VANET

Examining the RTT of the medium density rural environment an RTT trend similar to that of a high density urban environment is seen. As the number transmitting nodes increases the channel becomes more congested increasing wait time per packet. Due to the increased wait times contention becomes more difficult causing more transmissions and further increases channel congestion. The AODV evaluation appears to also suffer from increasing wait times, this seems to indicate that AODV links are unstable and new routes must constantly be discovered.

5.7.3 Average Hop Count

Similar to the urban evaluation the average hop count is divided into two metrics, average hop count from the node towards the RSU and average hop count from the RSU towards

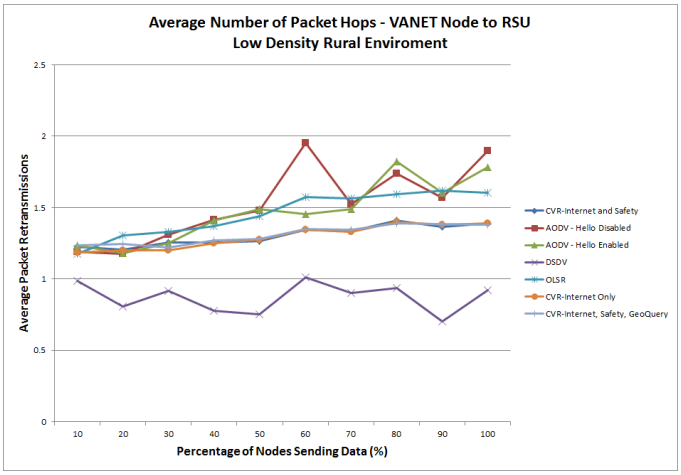

Figure 5.16: Low Density Rural Environment Average Packet Hop Count VANET to RSU

VANET nodes. Figures 5.19 and 5.20 show the average number of hops required to reach a VANET node from an RSU and the average number of hops required to reach an RSU from a VANET node.

Examining the results of the medium density rural environment a parallel is seen between the high density urban environment. As the channel becomes more congested packets spend increasing periods waiting in a transmission queue before the node gains access to the channel. This causes errors in contention cause inefficient nodes to win congestion periods. When inefficient nodes win contention periods, longer routes are formed as seen in the observed results.

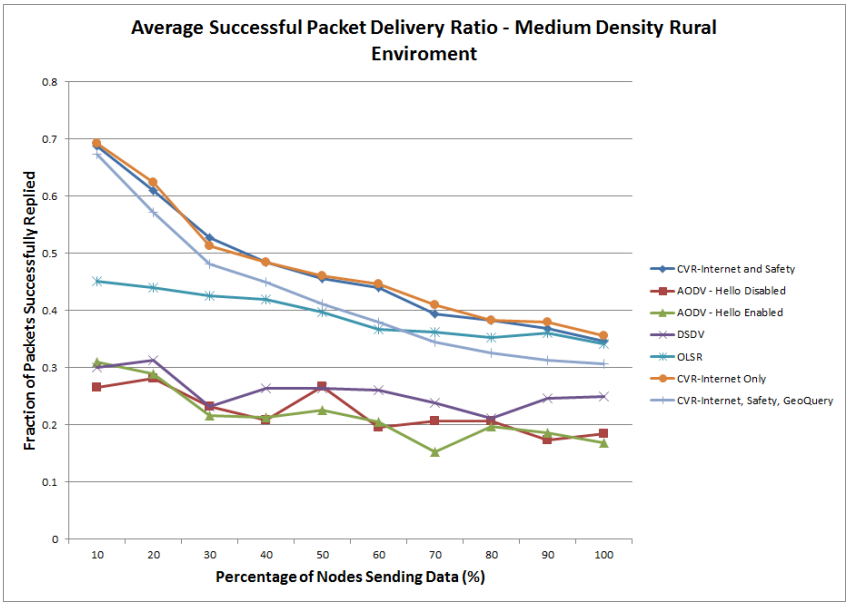

Figure 5.17: Medium Density Rural Environment Packet Delivery Ratio

5.8 Throughput Analysis

The following section presents a comparative analysis of total network throughput simulating OLSR and the proposed CVR protocol. OLSR is chosen as the protocol to compare against due to its superior performance to all other existing protocols. Throughput is calculated using Equation 5.3, the urban medium density simulation is run multiple times using varying total network traffic rates. Total network traffic begins at 20 packets generated per second and is doubled after each simulation up to 5120 packets per second. As with all previous simulations the packet payload is maintained at 1024 bytes. Figure 5.21 shows the throughput analysis of both OLSR and the proposed protocol.

Examining the two sets of results two distinct trends are seen, firstly from 20 to 1280

Table 5.10: Medium Density Rural Environment PDR Summary

Protocol	Average PDR Over All Trials	Average Standard Deviation
CVR Internet and Safety	0.46	0.14
AODV Hello Disabled	0.22	0.17
AODV Hello Enabled	0.21	0.17
DSDV	0.25	0.21
OLSR	0.39	0.22
CVR Internet Only	0.47	0.19
CVR Internet, Safety and GeoQuery	0.42	0.15

packets per second and secondly from 2560 to 5120 packets per second. In the first trend both protocols experience quadratic increases in throughput as the total network traffic increases. The proposed CVR protocol begins at approximately 100 kbps and increases up to 1300 kbps where it appears to reach a plateau. OLSR performs similarly though at a lower throughput, OLSR begins at 50 kbps and increases linearly up to 800 kbps. Within the range of the first trend the CVR protocol shows throughput increases between 50 and 60%. Examining the second trend at 2560 and 5120 packets per second OLSR begins to plateau at 850 kbps. The CVR protocol appears to run into difficulties, dropping the throughput to 600 and finally 200 kbps.

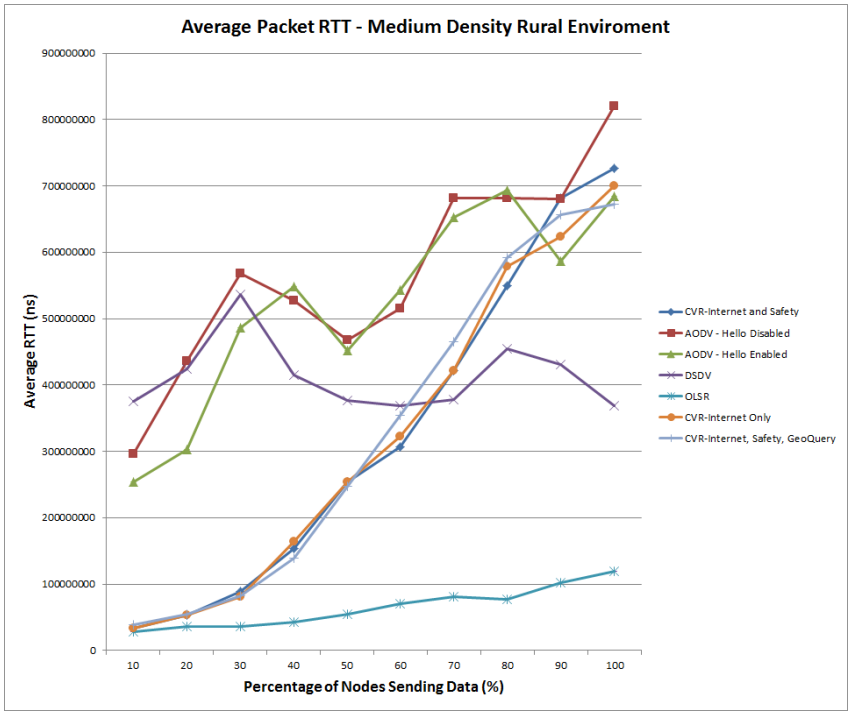

Figure 5.18: Medium Density Rural Environment RTT

The drastic drop in throughput prompts an examination of what could cause such a drop. Two possible explanations are explored, first the increased traffic may be interfering with route establishment causing packets to be dropped. The second possibility is that contention is not correctly taking place, node contention is at the heart of the proposed protocol and errors in contention could cause duplicate transmissions, flooding the channel with unnecessary traffic and interfering with correct transmissions. Following an empirical analysis of traffic patterns it was found that the MAC layer could not keep up with the higher traffic rates, the MAC layer would routinely discard packets as it could not access the medium before a packets maximum queue time expired. The MAC layer

Table 5.11: Medium Density Rural Environment RTT Summary

Protocol	Average RTT Over All Trials	Average Standard Deviation
CVR Internet and Safety	326940487	265284203
AODV Hello Disabled	567624641	527981232
AODV Hello Enabled	520633073	557696334
DSDV	412896862	492063613
OLSR	65013286	70001304
CVR Internet Only	323237406	251346825
CVR Internet, Safety and GeoQuery	103463033	262879886

problems propagated up to the network layer where contention was effected. Due to the MAC layer dropping packets from its queue contention between nodes was interrupted and duplicate transmissions were sent. Duplicate transmissions create even greater congestion issues resulting in further delays. When the MAC layer experiences difficulty it drops packets from its queue, thus breaking the contention process on other nodes. This causes all nodes contending to unnecessarily forward the packet, further congesting the channel and cause the MAC layer to drop more packets. A proposed solution to this issue is given in 6.3.

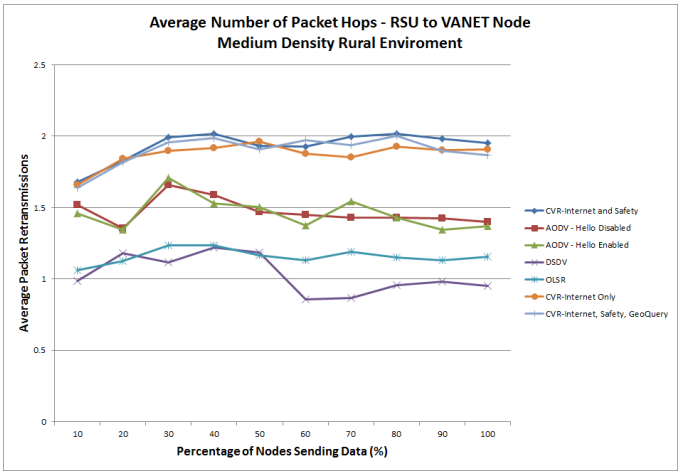

Figure 5.19: Medium Density Rural Environment Average Packet Hop Count RSU to VANET

5.8.1 Summary and Discussion of Performance Evaluation

Upon examining all performance evaluations in two different environments under a variety of conditions several strengths and weaknesses are seen. Beginning with the strengths, the proposed protocol performs quite well at low and medium densities in both urban and rural environments when compared to existing work. By considering the system in a holistic fashion and using cooperation between nodes and RSUs a much higher successful packet delivery ratio is achieved. Examining GeoQuery traffic in particular, this form of communication is only possible through cooperation with VANET nodes, RSUs and general Internet infrastructure. Even at an average of 1.5 hops existing protocols only achieve a 50% success rate at the highest point, a distant GeoQuery using an ex-

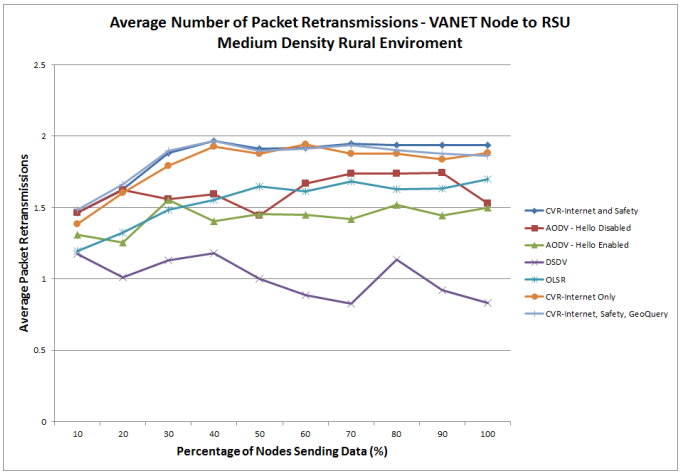

Figure 5.20: Medium Density Rural Environment Average Packet Hop Count VANET to RSU

isting protocol has an extremely poor chance of success. Examining the weaknesses of the proposed protocol the main issue seen is the propagation of MAC errors into the network layer. Due to the MAC layer queuing mechanism packets may experience extremely long queuing times. The long queuing time may cause errors with the contention process creating unnecessary network traffic and creating ineffective routes. Section 6.3 discusses several ways in which the effects of channel congestion could be counted by the proposed routing protocol.

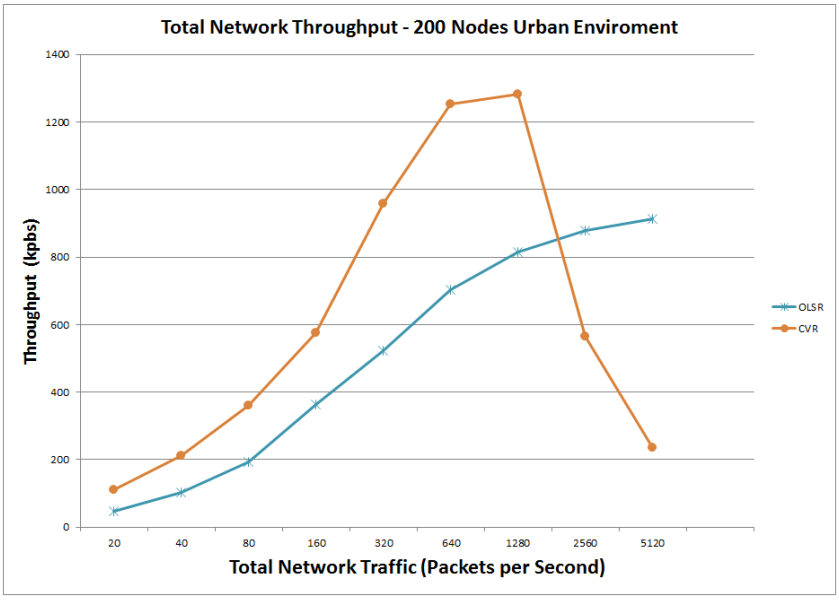

Figure 5.21: Throughput Analysis

6 Conclusion and Future Work

This chapter presents a summary of the proposals in the thesis, provides an analysis of the conclusions that may be drawn from the collected data, outlines future work and finally identifies several areas of system limitations stemming from ns-3 limitations as well as from simulations characteristics required to evaluated the proposals outlined in this thesis.

6.1 Thesis Summary and Contributions

This thesis provides contributions to VANET research in five areas, IPv6 geographic addressing, contention based hybrid routing, holistic network analysis and finally neighbor discovery and bootstrapping.

Zone Definition and Self Organizing Geographic Addressing The proposed IPv6 geographic addressing strategy places a geographic overlay over the surface of the earth dividing the land surface of the Earth into 1 KM^2 zones. Each zone is identified by a unique zone identifier, zone identifiers are 28^2 bits long. Based off

of the geographic zone overlay three addressing aspects are addressed, firstly using a constant VANET identifier, zone ID and MAC address each VANET node creates its own unique IPv6 address. Secondly, leveraging the zone ID a hierarchical addressing strategy is presented enabling efficient integration of VANETs into the Internet. Finally a geographic addressing strategy for VANET nodes is presented, each node sets its IPv6 address based on a VANET identifier, zone ID and MAC address. The VANET identifier ensures any node receiving a packet from the VANET may immediately identify the packet a originating from a VANET, this identification may be used for QoS functions or for VANET specific functionality. The zone ID segment of the IPv6 address provides receiving nodes the ability to localize the origin of a transmission to within 1 KM, in addition to the routing protocol presented the position may be used for data collection and processing on a geographical scale.

Multi-protocol Hybrid Contention VANET Routing The proposed contention based hybrid routing protocol differs from existing work in two primary areas. The first area of difference is seen in neighbor management, unlike many existing protocols contention based routing does not actively maintain a list of its neighbors. Instead of selecting a neighbor as the next hop nodes multicast their packet to all neighbors, each neighbor follows a contention procedure to forward the packet towards the final destination. Nodes continually compete to forward packets until

the final destination is reached, if an RSU is used to route traffic the RSU attempts to track node mobility to ensure reply packets are forwarded a nodes current position rather than a past position. The second area of difference is multi-traffic support. Almost all of the existing work focuses on a narrow scenario routing a single form of communication, the proposed protocol provides support for V2V, V2I and hybrid communication. In addition to providing support for various forms of communication the proposed protocol is developed in a modular configuration, elements of the proposed protocol such as safety message support may be entirely replaced by a new protocol without effecting any of the routing decisions made for all other forms of communication. Building on the proposed protocols support for multiple forms of traffic this thesis presents a network performance analysis from a holistic view. Rather than focusing on a single form of traffic, multiple forms of traffic are simultaneously routed to analyze the interactions the different forms of communication.

Geographic Bootstrapping The fourth area of contribution lies in providing a bootstrapping method to facilitate neighbor discovery and enable internet access through V2V routing. Building on the geographic overlay and zone ID process nodes may be pre-loaded with positions and addresses of RSU gateways within a large geographic region. The pre-existing knowledge allows nodes to use RSUs as internet gateways, additionally RSUs may be used in neighbor discovery applications al-

lowing nodes to discover distant nodes through a centralized query location.

VANET Research Platform Exploration The final area of contribution is in extending ns-3 as a simulation platform for VANET research. Ns-3 is still a relatively new simulator and while it is exceptionally stable, thoroughly tested and is continually extended it contains peculiar behavior which may cause difficulties for future researchers. This thesis presents one the ns-3 peculiarities in header management. Any research work using headers to communicate routing information via headers can greatly benefit from the header management reviewed in this thesis.

6.2 Conclusion

This thesis presents a contention based VANET routing protocol enabling multiple forms of communication through V2V, V2I and hybrid communication. Using a grid overlay the surface of the Earth is divided into multiple zones for the purposes of hierarchical and scalable node IPv6 addressing and information bootstrapping. Using zone information each node independently generates a unique IPv6 address, and if available identifies its zone RSU gateway. The RSU gateway is used by nodes to reach non-VANET networks or to reach distant networks. Communication is not strictly limited between VANET nodes and the RSU, each VANET node may communicate with its neighbors directly. The routing of multiple forms of communication gives the proposed protocol the unique

property of allowing protocol evaluation when examining the system as a whole rather than as discrete units. The routing method used is a contention based protocol, nodes are not required to keep a list of their neighbors and do not rely on exact neighborhood information to make routing decisions. Rather nodes compete for the right to forward a packet, this has the advantage of using each competing nodes most current data rather than relying on quickly outdated neighbor information.

Based on the evaluation the proposed contention based protocol, addressing and zone overlay provide a promising avenue for VANET research. Through cooperation of the system as an entire unit, network performance as low and medium densities is greatly improved. MAC layer issues present a challenge to contention based routing as outlined in this thesis however several future research areas are outlined to solve these issues. Overall the contention based routing protocol with infrastructure support appears to be a promising network layer extension with multiple areas for future research.

6.3 Future Work

Based on the performance evaluation two primary areas are identified for future research work. The first area of future research is local neighborhood estimation and the second area of future research layer 2 contention coordination.

Local Neighborhood Estimation One of the additional suppression mechanisms intro-

duced in this thesis performed suppression based a nodes active contention processes. Nodes exceeding an active contention parameter are automatically suppressed. This additional suppression mechanism prevents a single node from winning all contentions and becoming a bottleneck. The active contention suppression mechanism spreads forwarding nodes out among multiple nodes reducing bottleneck effects. Currently the proposed routing protocol uses a static active contention threshold determined through multiple rounds of testing. The current threshold is optimized towards the conditions in the tests however different conditions could lead to different requirements in the contention threshold. An appropriate future research area would be examine dynamically assigning the active contention threshold based off of the observed routing traffic. A dynamically assigned active contention threshold may optimize each nodes routing protocol based on its observed conditions.

Cross Layer Contention Coordination One of the observed areas where difficulties arise for the proposed contention based routing protocol is when the MAC layer becomes overwhelmed with transmissions and cannot clear its transmit cache. The overwhelming effects of the MAC layer propagate to the network layer in the form of missed suppression packets, causing more transmissions to be sent forming a negative feedback loop. A future area of research would be to explore integrating the MAC layer into contention. MAC layer integration could be explored in two ar-

eas; active MAC layer cache management and MAC contention piggybacking. The MAC layer performs contention to gain access to the medium through the RTS-CTS mechanism. The RTS-CTS mechanism may be modified to include packet forwarding details, using the added details contention could be resolved before the actual packet is transmitted. One method to modify the RTS-CTS mechanism would be to include a nodes position and velocity into RTS and CTS messages. Using the modified RTS and CTS messages source nodes may actively select its preferred forwarder. A second method to modify the RTS-CTS mechanism would be to modify the CTS backoff timer to reflect the nodes contention value. The modified CTS timer would ensure the node with the best contention value wins the contention period, a modified CTS timer would once again place contention control with forwarding nodes. The second area of research is active cache monitoring. Even after winning a contention period and passing a packet down to the MAC layer nodes monitor the contention phase for that packet. If a transmission is detected which would normally suppress a packet for which forwarding has already begun the node could actively remove that packet from its data-link layer transmission queue, saving channel resources by eliminating an unnecessary transmission.

6.4 System Limitations

Due to the unique geographic aspects of this thesis as well as the current state of the ns-3 simulator several system limitations are encountered which influenced the performance evaluation.

Geographic Simulation Area and Node Density Based on the mobility prediction, geographic overlay and GeoQuery message routing outlined in chapter 3 a minimum of two geographic zones are required to evaluate all proposed enhancements. While two geographic zones would allow evaluation of all proposed enhancements it would have the undesirable effect of skewing mobility in a particular direction as the nodes would be able to move double the distance in one direction; therefore a minimum of four geographic zones are required for an accurate simulation. In addition to a minimum number of geographic zones, node density must be maintained. Each additional geographic zone simulated increases the number of nodes required to maintain the average node density, consequently this greatly increases the number of events processed. Increases in either node density or simulation area greatly increase the run time of a simulation. As a reference point; simulations running AODV or DSDV in high density urban environments took upwards of 72 hours to complete. The high run times make simulating higher densities or larger geographic areas a soft system limitation; given more time and resources it

is technologically possible to run larger simulations.

Simulation Time In addition to node density and simulation area the amount of time simulated is also limiting factor. Increasing the amount of time simulated also increases the number of events required to be processed and increases the total time required to finish the simulation. The total increase is highly variable depending on the number of nodes in the simulation and the number of events those nodes generate. Using the previously mentioned AODV and DSDV run times of 72 hours in a high density urban environment, an increase of each second to simulate increased simulation run time by 36 minutes. Based on this limiting factor, a simulation time of 120 seconds was chosen to limit simulation run time while allowing enough time for mobility effects to be captured in the simulation. 120 seconds allows nodes to move through at least one geographic zone , this captures node readdressing, mobility prediction and mobility effects on packet forwarding. Additionally 120 seconds allows the simulation to capture the effects of nodes moving in and out of range of each other as well as the effects of nodes stopping and accelerating changing how dynamically the local network topology changes. The simulation time is a soft system limitation, given more time and more resources the simulations could be set to simulate longer periods of time.

Serial Processing The current state of the ns-3 simulator does not allow for parallel pro-

cessing to take place while simulating wireless connections; currently ns-3 only supports parallel processing across point-to-point links. Due to parallel processing being unavailable the advantages of multiple cores cannot be utilized; this greatly increases simulation run time as events must be processed serially. Processing events serially may support any node density or simulation time, however increases in either of these areas will also increase the number of event to process thus increasing simulation run time.

Mobility Adjustments As discussed in Chapter 4 ns-3 uses the isolated mobility model when using SUMO to generate mobility traces. SUMO generates mobility trace files which are input to simulations, once a trace file has been input a nodes mobility may not longer be changed throughout the simulation. The isolated mobility model imposes a hard system limitation on adjusting node mobility, specifically node mobility may not be modified once the trace file has been loaded by the simulation. This system limitation leads to two limiting areas for mobility adjustment. The first mobility adjustment limitation is the inability adjust mobility based on VANET communication. This limitation stops ns-3 from being used in VANET research designed to modify node mobility patterns based VANET communication as well as observed conditions. The second mobility adjustment limitation is the inability to accurately simulate emergency situations. During emergency situations VANET nodes may stop or break away from mobility constrains set by roads. The

ns-3 and SUMO interface currently does not have a way to coordinate emergency situations, thus during emergency situation simulations in ns-3, vehicles continue their standard mobility patterns.

7 Appendix

A Creating Mobility Trace Files

Mobility traces are created using using a combination of the OSM database and SUMO mobility simulator as described in previous section. This section describes the exact procedure used to create mobility traces for all of the simulations in this thesis. Mobility traces are created through the following steps:

1. Extract OSM Map: Using the interface in [16] a geographic area is selected for export. Extra care must be taken when selecting the export area, the OSM database uses latitude and longitude coordinates when defining areas making it difficult to specify an exact area. The exported file will be in an osm format.

2. Convert to a Network This step extracts network information from the osm file. The extracted network will be used by SUMO to set vehicle mobility along the exacted network. The conversion command is listed below:

```
netconvert --osm-files <Path_To_OSM_File>
--output-file <Name>.net.xml
--geometry.remove --roundabouts.guess --ramps.guess
```

```
--junctions.join --tls.guess-signals --tls.discard-simple
--tls.join --remove-edges.isolated --remove-edges.by-vclass
 rail_electric,rail,bicycle,pedestrian
```

The netconvert utility is a provided by SUMO to convert various map formats to compatible network files, here it is used to convert an osm map file to an xml network file. The passed options attempt remove irregularities in the map such as disconnected edges or edges on which vehicles may not travel.

3. Create Vehicle Trips: Based on the output network random trips are created using the randomTrips python script provided by SUMO. The randomTrips script is run using the following command:

```
python $SUMO_PATH$/sumo-0.22.0/tools/trip/randomTrips.py -n
 <name>.net.xml -e E -l -validate -p P
 --intermediate N
```

The randomTrips script creates E/P random source destination pairs, each pair represents a vehicular source and destination. In order to ensure nodes remain in the simulation throughout the entire simulation N intermediate waypoints are defined. The randomTrips script outputs its source-destination pairs in an xml file; trips.trips.xml.

4. Generate Routes: Using the previously defined source-destination pairs routes must be found between each source-destination pair. Routes are found using the duarouter utility. The duarouter utility parses the network and trip output files and finds the shortest path between each source-destination pair while passing through the intermediate waypoints. Duarouter is run using the following command:

```
duarouter -n <name>.net.xml -t trips.trips.xml -o
<name>.rou.xml --ignore-errors
--error-log routeErrors.txt
```

5. Create SUMO configuration file: SUMO provides a standardized template for inputting. The template is copied and modified to include the generated network and route files. Any name may be used for the template however the standard naming convention is ¡name¿.sumo.cfg.

6. Create Trace File: Creating the trace file runs simulates the input mobility patterns on the underlying network. Each simulation step is output into an xml file in a raw format. The trace file is created using the following command:

```
sumo -c <name>.sumo.cfg --fcd-output <name>Trace.xml
```

7. Convert to ns-3 Readable Format: The final step in the mobility generation process is to convert the raw output in the last step into a format readable by ns-3. Trace

conversion is performed using the traceExporter python script, the traceExporter script is run using the following command:

```
python $SUMO_PATH$/sumo-0.22.0/tools/traceExporter.py
--fcd-input <name>Trace.xml
--ns2mobility-output <name>Mobility.tcl
```

The final *.tcl file is in a format readable by ns-2 and ns-3, when running simulations this file is parsed and simulated.

B Running Ns-3 Simulations

This section reviews running running ns-3 simulations and provides a high level view of the ns-3 command line, readers interested in a full analysis of all ns-3 features and options are directed to [31].

Ns-3 employs the *waf* framework to manage its build environment, the *waf* framework manages building, linking and running simulation C++ files.

B.0.1 Compiling

Compiling ns-3 programs is fully managed through the *waf* build component. Compiling ns-3 is accomplished through two steps, configuration and build. The first step in compiling ns-3 is the configuration phase, configuration sets the build mode and build options. Ns-3 supports two types of builds, an optimized and debug build. Debug builds enable ns-3 logging functionality and performs minimal compiler optimization. Optimized builds disable ns-3 logging and flags the compiler to optimize the compiled code as much as possible. The build options allow users to enable *waf* tests and allows users

to specify if examples should be built as well at the main model files.

Once configuration has been completed a set of dependency tests are run to ensure ns-3 will successfully run. If all required tests are successfully completed the configuration is saved and ns-3 may be compiled. To actually compile ns-3 the *waf* script is invoked, using the saved configuration it will check all ns-3 files for modifications, build and link all eligible files.

B.0.2 Simulation Parameters

Simulation parameters are entirely configurable by users creating simulations. Ns-3 provides exceptional support for specification of simulation parameters through command line arguments, users specify an attribute, a variable to store the value and a description, the ns-3 framework manages validation and command line management. In addition to managing command line parameters ns-3 provides an informational help command specifying all available parameters and their default values, below is the information section for one of the simulations files used to run a single simulation.

```
ThesisRoutingNsMobility [Program Arguments] [General Arguments]


Program Arguments:

    --nVeh:             Number of vehicle nodes [2]

    --nRSU:             Number backbone nodes [2]
```

```
    --nRsuRow:          Number of RSU in a row [4]
    --nSendPerc:        Percentage of vehicular nodes
                         acting as sources [0.1]
    --trace:            Location of the mobility trace []
    --simTime:          Simulation time [20]
    --sFreq:            Number of packets sent per second [1]
    --pSize:            Size of packet to send [1024]
    --maxPacketCount:   Maximum number of packets to send [200]

General Arguments:
    --PrintGlobals:                 Print the list of globals.
    --PrintGroups:                  Print the list of groups.
    --PrintGroup=[group]:           Print all TypeIds of group.
    --PrintTypeIds:                 Print all TypeIds.
    --PrintAttributes=[typeid]:     Print all attributes of typeid.
    --PrintHelp:                    Print this help message.
```

The listing shows two main pieces of information, program arguments and general arguments. Program arguments are the parameters specified by the user while general arguments are ns-3 specific arguments. The program arguments section provides users

with three pieces of information, the parameter to set, a description of the parameter and the default value in square brackets. The general arguments provided give users a starting point to explore the attributes of any of the classes referenced in the simulation. Through setting and modifying the program parameters users may exactly specify the parameters of their simulation.

Bibliography

[1] IEEE standard for wireless access in vehicular enviroments (WAVE) - networking services. Technical report, IEEE, dec 2010.

[2] 1609.0-2013 - IEEE guide for wireless access in vehcular enviroments (WAVE) - architecture. Technical report, IEEE, 2013.

[3] 1609.4-2016 - IEEE standard for wireless access in vehicular enviroments (WAVE) – multi-channel operation. Technical report, IEEE, mar 2016.

[4] Simulation of urban mobility, 2016.

[5] Abdeldime M.S. Abdelgader and Wu Lenan. The physical layer of the IEEE 802.11p WAVE communication standard: The specifications and challenges. *WCECS*, 2, oct 2014.

[6] Katrin Bilstrup, Elisabeth Uhlermann, Erik G. Strom, and Urban Bilstrup. Evaluation of the IEEE 802.11 MAC method for vehicle-to-vehicle communication. *Vehicular Technology Conference*, pages 1–5, 2008.

[7] Subir Biswas, Raymond Tatchikou, and Francois Dion. Vehicle-to-vehicle wireless communication protocols for enhancing highway traffic safety. *IEEE Communications Magazine*, pages 74–82, 2006.

[8] Bruce A. Black, Philip S. Dipiazza, Bruze A. Ferguson, David R. Voltmer, and Frederick C. Berry. *Introduction To Wireless Systems*. Prentice Hall, 2008. 17-70.

[9] Transport Canada. Road safety in canada, 2015.

[10] Qi Chen, Daniel Jiang, and Luca Delgrossi. IEEE 1609.4 DSRC muli-channel operations and its implications on vehicle safety communications. *IEEE Vehicular Networking Conference*, pages 1–8, oct 2009.

[11] Yuh Shyan Chen, Yun Wei Lin, and Wing Ling Lee. A mobicast routing protocol in vehicular ad-hoc networks. *Mobile Networks and Applications*, 15(1):20–35, feb 2010.

[12] Pei-Chun Cheng, Kevin C. Lee, Mario Gerla, and Jerome Harri. GeoDTN+Nav: Geographic DTN routing with navigator prediction for urban vehicular enviroments. *Mobile Networks and Applications*, 15(1):61–82, feb 2010.

[13] Felipe Domingos da Cunha, Azzedine Boukerche, Leandro Villas, Aline Carneiro Viana, and Antonio A. F. Loureiro. Data communication in VANETs: A survey, challenges and applications. *INRIA Research Report, RR-8498*, pages 3–26, 2014.

[14] Sandor Dornbush and Anupam Joshi. Streetsmart traffic: Discovering and disseminating automobile congestion using VANETs. In *2007 IEEE 65th Vehicular Technology Conference - VTC 2007 Spring*, pages 11–15, 2007.

[15] Holger Fler, Jrg Widmer, Michael Ksemann, Martin Mauve, and Hannes Hartenstein. Contention-based forwarding for mobile ad hoc networks. *Ad Hoc Networks*, 1(4):351 – 369, 2003.

[16] Open Street Map Foundation. Openstreetmap, 2016.

[17] Melvut Turker Garip, Mehmet Emre Gursoy, Peter Reiher, and Mario Gerla. Scalable reactive vehicle-to-vehicle congestion avoidance mechanism. *12th Annual IEEE Consumer Communications and Networking Conference (CCNC)*, pages 943–948, 2015.

[18] Chong Han, Mehrdad Dianati, Rahim Tafazolli, Ralf Kernchen, and Xuemin Shen. Analytical study of the IEEE 802.11p MAC sublayer in vehicular networks. *IEEE Transactions on Intelligent Transport Systems*, 13(2):873–886, jun 2012.

[19] Vic Hargrave. TCP/IP network programming design patterns in C++, 2013.

[20] Hannes Hartenstein and Kenneth Laberteaux. *VANET Vehicular Applications and Inter-Networking Technologies*. John Wiley & Sons, 2009. 141-144.

[21] Geert Heijenk, Martijn Eenennaam, and Anne Remke. *Quantitative Evaluation of Systems: 11th International Conference, QEST 2014, Florence, Italy, September 8-10, 2014. Proceedings*, chapter Performance Comparison of IEEE 802.11 DCF and EDCA for Beaconing in Vehicular Networks, pages 154–169. Springer International Publishing, Cham, 2014.

[22] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot. Optimized link state routing protocol for ad hoc networks. *IEEE International Multi Topic Conference*, pages 62–68, 2001.

[23] M. Milton Joe and B. Ramakrishnan. Review of vehicular ad hoc network communication models including WVANET (web vanet) model and WVANET future research directions. *Wireless Networks*, pages 1–18, 2015.

[24] Brad Karp and H. T. Kung. GPSR: Greedy perimeter stateless routing for wireless networks. In *MobiCom '00 Proceedings of the 6th annual International conference on Mobile computing and networking*, pages 243–252. ACM, 2000.

[25] Vishal Kumar, Shailendra Mishra, and Narottam Chand. Applications of VANETs: Present & future. *Communications and Network*, 5:12–15, feb 2013.

[26] Kevin C. Lee, Seung-Hoon Lee, Ryan Cheung, Uichen Lee, and Mario Gerla. First experience with cartorrent in a real vehicular ad hoc network testbed. *Mobile Networking for Vehicular Enviroments*, pages 109–114, 2007.

[27] Kevin C. Lee, Uichin Lee, and Mario Gerla. Survey of routing protocols in vehicular ad hoc networks. *Advances in Vehicular Ad-Hoc Networks: Developments and Challenges*, 2010.

[28] Fan Li. Routing in vehicular ad hoc networks: A survey. *IEEE Vehicular Technology Magazine*, 2(2):12–22, jun 2007.

[29] Yunxin (Jeff) Li. *Quality, Reliability, Security and Robustness in Heterogeneous Networks: 7th International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness, QShine 2010, and Dedicated Short Range Communications Workshop, DSRC 2010, Houston, TX, USA, November 17-19, 2010, Revised Selected Papers*, chapter An Overview of the DSRC/WAVE Technology, pages 544–558. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[30] Christian Lochert, Martin Mauve, Holger Fubler, and Hannes Hartenstein. Geographic routing in city scenarios. *ACM SIGMOBILE Mobile Computing and Communications Review*, 9(1):69–72, jan 2005.

[31] NS3. ns3, 2016.

[32] Wendell Odom. *Cisco CCNA Routing and Switching 200-120 Official Cert Guide Library*. Cisco Press, 2013.

[33] Vijan Paul, Md. Ibrahim, and Md. Abu Naser Bikas. VANET routing protocols: Pros and cons. *International Journal of Computer Applications*, 20(3):28–34, apr 2011.

[34] Charles E. Perkins and Pravin Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. *SIGCOMM*, pages 234–244, 1994.

[35] Charles E. Perkins and Elizabeth M. Royer. Ad-hoc on-demand distance vector routing. *Mobile Computing Systems and Applications*, pages 90–100, feb 1999.

[36] Bhuvanshwari S., Divya G., Kirithika K.B., and Nithya S. A survey on vehicular ad-hoc network. *International Journal Oof Advanced Research in Electrical, Electronics and Instrumentation Engineering*, 2(10):4993–5000, 2013.

[37] Prasan Kumar Sahoo, Ming-Jer Chiang, and Shih-Lin Wu. SVANET: A smart vehicular ad hoc network for efficient data transmission with wireless sensors. *Sensors*, 14(12), 2014.

[38] Ghanshyam Singh. Ece 618: Mobile & wireless communications, sharda university. Web, nov 2015. Set of slides by professor Singh going over wireless transmissions.

[39] Fernando A. Teixeira, Vinicius F. e Silva, Jesse L. Leoni, and Daniel F. Macedo. Vehicular networks using the IEEE 802.11p standard: An experimental analysis. *Vehicular Communications*, 1:91–96, 2014.

[40] Wang Xiaonan and Zhong Shan. Research on IPv6 address configuration for a VANET. *Jourbal of Parallel and Distributed Computing*, feb 2013.

[41] Bo Xu, Ouri Wolfson, and Hyung Jo Cho. Monitoring neighboring vehicles for safety via V2V communication. In *Vehicular Electronics and Safety (ICVES)*, pages 10–12, jul 2011.