2011

ON THE INTEGRATION OF VEHICULAR AD-HOC NETWORKS AND VISION-BASED DRIVER ASSISTANCE

Kyle Charbonneau

ON THE INTEGRATION OF VEHICULAR AD-HOC NETWORKS AND VISION-BASED DRIVER ASSISTANCE

(Spine title: On the Integration of VANETs and VBDA)

(Thesis format: Monograph)

by

Kyle Charbonneau

Graduate Program in Computer Science

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science

The School of Graduate and Postdoctoral Studies
The University of Western Ontario
London, Ontario, Canada

THE UNIVERSITY OF WESTERN ONTARIO
School of Graduate and Postdoctoral Studies

CERTIFICATE OF EXAMINATION

Examiners:

Supervisor:

....
Dr. Steven Beauchemin

....
Dr. Michael Bauer

....
Dr. Xianbin Wang

....
Dr. Michael Katchabaw

The thesis by

Kyle Charbonneau

entitled:

On The Integration of Vehicular Ad-Hoc Networks and Vision-Based Driver Assistance

is accepted in partial fulfillment of the
requirements for the degree of
Master of Science

....
Date

....
Chair of the Thesis Examination Board

Abstract

Vehicular ad-hoc networks (VANETs) allow for short range wireless communication to share information between vehicles. Vision-based driver assistance (VBDA) uses computer vision to obtain information about nearby objects. The goal of both systems is to create a model of the environment surrounding the vehicle in order to make decisions. With unique strengths and weaknesses the two systems complement each other well. A simulation environment for both VANETs and VBDA is created to test both systems alongside one another. They are evaluated and then combined to build the best possible model of the environment with the goal of improving vehicle safety under adverse conditions.

Keywords: Vehicular Ad-Hoc Networks, Vision-Based Driver Assistance, Cooperative Collision Warning System, IEEE 802.11p, Wireless Access in Vehicular Environments

Contents

List of Figures

List of Tables

List of Appendices

Chapter 1

Introduction

Wireless communication between vehicles through vehicular networks presents an interesting research area with endless possibilities. Vehicular networks are still relatively new having yet to appear in production vehicles yet there is a large body of research and standards that exist to help guide their implementation. There are also clear motivations for why they should exist, one example being to reduce the number of accidents by providing early warning to the driver. Unlike existing wireless computer networks or wireless sensor networks, vehicular networks have unique challenges that have yet to be tackled on such a large scale. Furthermore, beyond the challenges in simply implementing vehicular networks there is the additional challenge of keeping the network secure from malicious users due to the possibility of life and death consequences while on the road. The gaps in the existing standards present numerous opportunities for research into vehicular networks that could potentially change the way we drive.

There is also ongoing research exploring the use of various kinds of on-board sensors to augment the safety and information gathering features of vehicles. Digital video cameras along with computer vision algorithms are one example. Multiple cameras can be used to keep a constant watch three hundred and sixty degrees around the vehicle, a task that is impossible for a human driver. The motivations behind implementing on-board sensor based driver assistance is similar to the motivations for vehicular networks. It is again an interesting research area and possibly closer to reality than vehicular networks since we already have driver assistance systems based on RADAR or LIDAR available in production vehicles.

Both technologies present us with ways to make driving safer and their own unique strengths and weaknesses. This thesis will examine the interplay between the two technologies.

1.1 What Are Vehicular Networks?

Before we begin discussing specifics about vehicular networks it is important to define what they are. In the past few decades we have seen the proliferation of various wireless communication networks. These networks support two-way communication between nodes over varying distances. Examples include high speed wireless computer networks based on the Institute of Electrical and Electronics Engineers (IEEE) 802.11 wireless standards that have become common in homes and businesses as well as the many variations of cell phone networks from early analogue networks to today's 3G and 4G digital networks.

It makes sense to base vehicular networks on existing standards in order to reuse existing designs and reduce the costs associated with developing them. Vehicular networks would provide short range communication, a few hundred meters at most, between computers located in each vehicle. Wireless computer networks are a better starting point than cell phone networks due to the similar transmission range and data throughput requirements. To participate in a vehicular network each vehicle would have to have a computer and wireless radio. This system can be referred to as the On Board Unit (OBU). The computer would receive input about the current state of the vehicle, such as speed or turn signal status, input from a Global Positioning System (GPS), providing location and timing data, and finally input from a wireless radio. The OBUs will autonomously communicate amongst each other to share data without any driver interaction. Since operating a motor vehicle is a dangerous task the driver should only be alerted of something high priority that requires immediate action, such as an imminent collision.

Ideally the lower level protocols, operating at the physical, data link and network layers, would be well defined to ensure that vehicles from all manufacturers can communicate. At the same time there should be some freedom in the higher level protocols, operating at the transport and application layers, to allow services beyond what are envisioned today. This freedom to develop new services was what allowed the internet as we know it to develop so effectively. Security is also an important issue, since drivers may be making split second decisions that could put them at danger based on the warnings they receive, and should be taken into account in the design of vehicular networks.

Finally, a distinction must be made between Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) networks. V2V networks will be formed between multiple vehicles in an ad-hoc manner. As such, V2V networks can be referred to as Vehicular Ad-Hoc Networks (VANETs). Messages can be sent between vehicles without a central Access Point (AP) mediating communication. These messages can be related to safety, traffic or any other service. V2I networks on the other hand will involve one or more vehicles and a piece of roadside infrastructure, or Road Side Units (RSUs). RSUs would likely be built into existing road related features, such as traffic lights or lamp posts, or could be purposely built to serve a specific purpose. The infrastructure will have one or more services it provides to nearby vehicles and will most likely be connected to another type of network as well, such as a Local Area Network (LAN) or the internet. This extra network connection can allow it to provide information that would be either unavailable in a strict VANET, for example weather information, or difficult to acquire in a strict VANET, for example traffic information from outside the reach of V2V communication.

Given this general definition of vehicular networks their operation in relation to this thesis will be described in greater detail over the following chapters.

1.2 What About Vision-Based Driver Assistance?

The use of on-board cameras and computer vision algorithms again aims to obtain data about the world surrounding the vehicle. We have seen similar driver assistance technologies on production vehicles for some time now. Backup cameras, that display what is immediately behind the vehicle in an otherwise blind spot to the driver, have become common place. Some high end vehicles, such as the Lexus LS, even have cameras that point in all directions to provide the driver with a three hundred and sixty degree view around the vehicle while maneuvering in tight spaces. Adaptive Cruise Control (ACC) actual takes control away from the driver as the vehicle follows either a set speed or if there is a vehicle in front then a set distance from it. Some variations on the technology will even brake to avoid a collision.

Vision-Based Driver Assistance (VBDA) works in much the same way as existing driver assistance technologies but instead of providing an enhanced view to the driver, computer vision algorithms are instead used to process the images and locate objects around the vehicle. By using stereo pairs of cameras objects can be located in 3D space and with it their size and position relative to the vehicle can be found. Objects can be identified by type, for example other vehicles, pedestrians or stationary obstacles. By comparing the position in subsequent time intervals we can estimate the trajectory of moving objects. This information forms a model of the world that can be used to make decisions with.

VBDA would again run autonomously on the OBU. Any potential hazards or collisions could identified in real time and the driver could be warned before it is too late. Since the OBU does not suffer from inattention or slow reaction times it can provide a definite improvement over the driver's eyes alone.

1.3 Motivation

The motivation behind developing VANETs and VBDA is to increase safety and convenience for drivers. During 2008, the most recent year detailed statistics are available for, 2187 people were killed and 178 833 people were injured in motor vehicle accidents in Canada alone [3]. The worldwide death toll for automobile collisions was estimated at 1.2 million in 2004 and is only expected to rise. Beyond the human toll of automobile collisions the economic toll is staggering, estimated at $518 billion US per year worldwide by the World Health Organization (WHO) [4]

Clearly reducing the number and severity of automobile collisions is a worthwhile goal. In addition, vehicular networks present us with the ability to gather and distribute traffic data as well. If they can be used to reduce the severity of traffic congestion and lower commute times there are benefits in the form of additional free time for individuals, less wear and tear on vehicles and less fuel consumed along with the associated reduction in greenhouse gas emissions. A 2002 study by Transport Canada found that the annual cost of traffic congestion in urban areas within Canada is a staggering $2.3 to $3.7 billion [5].

The use of VANETs and VBDA together presents an interesting case. In relation to safety they both work towards the same goal in a similar fashion. By building a model of the world surrounding the vehicle the OBU can anticipate problems and warn the driver before a collision occurs. Despite the overlap in the abilities of VANETs and VBDA it makes sense to look at them together.

VANETs provide a breadth that anything vision based cannot. Vision is inherently limited by line of sight. If an object can't be seen then it can't be detected. VANETs on the other hand allow communication over hundreds of meters. While line of sight does help improve transmission quality, radio waves will pass through and or bounce around obstacles. If all vehicles periodically broadcast their location and trajectory we can place them relative to our own vehicle. By extending their current trajectory into the future we can predict where they will be and predict collisions.

Conversely, VBDA provides more timely information about the world surrounding the vehicle. A refresh rate of 30Hz or higher on each camera is expected giving us at most tens of milliseconds of delay between updates. This provides extremely low latency information. It is not possible to broadcast this frequently in a VANET, as demonstrated by S. Rezaei et al. due to the limited bandwidth of wireless communication [6]. Furthermore, wireless communication is inherently unreliable. Packets are frequently lost due to attenuation and interference. There is simply no guarantee of successful wireless communication in the short time span it takes for a collision to occur at high speeds.

The technology required to implement both VANETs and VBDA is also tightly coupled. Computer vision algorithms require a significant amount of processing power from a computer system along with cameras installed in the vehicle. Vehicular networks again require computer processing power and a wireless radio unit. The shared computer system would also need to access vehicle sensor data and GPS data. As the cost of both digital video cameras and wireless radios have plummeted in recent years the adoption of both technologies together becomes very attractive as the main cost lies in the computer system itself.

Most importantly, people have shown they are willing to adopt new technologies that improve automobile safety. These technologies often add extra cost on an already expensive purchase so there is a risk they could be rejected by consumers. However, history has shown that the majority of consumers are willing to pay the extra price for safety. Examples of these technologies include seat belts, Anti-Lock Braking System (ABS), traction control, and ACC. Overall, VANETs and VBDA have the potential to improve safety and convenience for drivers worldwide in a revolutionary way.

1.4 Challenges

There are numerous challenges in implementing any sort of driver assistance system. There are too many for all of the challenges to be encompassed by the work of this thesis. As such, a general outline of the challenges that are faced along with which ones will be examined in the thesis follows.

Computer vision is a difficult problem. There currently exist effective computer vision algorithms that can locate objects in 3D space. However when faced with real world conditions, such as poor weather or partially occluded objects, even the best computer vision algorithms may fail. Many systems used in vehicles today for similar purposes rely on RADAR or LIDAR as they can work more reliably under adverse conditions. Using some combination of vision, RADAR or LIDAR sensing the world around the vehicle is definitely possible. This is a large problem area outside the scope of this thesis. As we've seen with the Defense Advanced Research Projects Agency (DARPA) Urban Challenge vehicles can use various on-board sensors to identify objects with enough precision to navigate in an urban environment completely autonomously. As such, it is assumed the computer vision algorithms work well enough to provide VBDA within the parameters that are laid out in subsequent chapters.

By their very nature VANETs will have a dynamic topology. Vehicles are designed for rapid transportation after all. While some infrastructure will certainly be involved in vehicular networks its not feasible to install it on every segment of road necessitating VANETs. Creating an ad-hoc network with millions of nodes nationwide is a difficult task that requires careful consideration. There has been considerable research into Mobile Ad-Hoc Networks (MANETs) however most of it has been centered around wireless sensor networks whose mobility is limited in comparison. While VANETs are similar to MANETs they have unique considerations due to their potential for high relative velocity of nodes, up to 500km/h, and set network topologies, as vehicles must follow existing road networks [7].

Being able to communicate with unknown nodes surrounding you within a short time span is crucial. As such, the overhead requirements to facilitate the transmission of messages must be low. Furthermore, fast reliable transmission is crucial for safety applications so packet error rates and latency must be low. Thankfully, unlike many wireless sensor networks electrical and computing power are not an issue. Vehicles can easily house and power a modest computer system. The effective implementation of vehicular networks and their simulation is a central topic in this thesis and will be covered in great depth.

Initially the adoption of VANET technology will be low. There will be at a point in time when the first and only production vehicle rolls off the assembly line equipped with it. From there the percentage of vehicles equipped will rise steadily. It will take many years for adoption to near 100% and until that day VANETs must still be useful to the drivers who have opted in.

The effect of lower adoption rates will examined in the experiments performed.

The integration of V2V and V2I communication is also crucial. While VANETs based on V2V technology will the focus of this thesis V2I communication is also important and will be touched on briefly.

Security is another concern within VANETs. There are a wealth of potential attacks and exploits a malicious user could attempt. In general, nodes with permission to be part of the VANET will be given credentials to participate. Nodes without credentials will be ignored. Furthermore, it should be as difficult as possible to modify the OBU or wireless radio of any legitimate vehicle to transmit arbitrary data. A more thorough description of security in VANETs will follow in Section 2.2.5 however the scope of the problem places it outside the reach of this thesis.

Beyond making any one driver assistance technology work in isolation we must combine both VANETs and VBDA. Both technologies gather a model of the world which is used to make decisions. However, there are no guarantees that these models will match perfectly. VANETs suffer from lost packets and latency while drivers can make sudden changes to their course. Vehicles rarely follow a set path for very long, often accelerating, turning or changing lanes. GPS coordinates are also not entirely accurate and introduce further errors. VBDA is relatively accurate in determining distance to an object under good conditions but determining the precise size and location of a vehicle is again difficult due to limitations on what can be seen. Bad weather and obstructions make this job even more difficult. As such, both sources of information must be combined to create the best possible model and this will be examined in the experiments performed.

Finding ways to successfully convey a message to the driver is difficult. The types of warning that are the most time critical, for example warning of someone braking hard ahead or an imminent collision, are uncommon and therefore the driver will not be used to reacting to them. Explanatory visual or auditory messages, for example in the case of someone braking hard a voice output of "hard braking ahead", may be too slow. Shorter warnings, such as an auditory alarm or physical feedback, lack any context and drivers may become confused and react incorrectly. Exactly how to interact with the driver is another difficult problem that needs to be looked at but is outside the scope of this thesis.

Finally, the simulation of VANETs and VBDA is a complex task. Accurate simulation is key to producing realistic results. In order to complete the proposed experiments a simulation environment that can simulate realistic vehicle movement, wireless communication and vision is required. The use of open source projects along with the necessary modifications and extensions completed will be explored throughout the thesis.

1.5 Use Cases

The goal of VANETs and VBDA is to provide the driver with extra information not normally available. VBDA can monitor the surrounding environment. VANETs facilitate V2V communication with vehicles well out of sight to learn their location and trajectory. Similarly, V2I communication can be used provide warning of accidents that are kilometers away or to download traffic information from which to plan the best route. However, this thesis will focus on three specific use cases, with VANETs and VBDA providing a Cooperative Collision Warning System (CCWS), an Emergency Warning System (EWS) and a Hazard Warning System (HWS). Both technologies will provide information on vehicle positions and trajectories that will be combined into a unified model of the world.

The use case for the CCWS works as follows. We refer to a single vehicle as the Subject Vehicle (SV) and all other vehicles in reference to the SV as Neighbour Vehicles (NVs). The OBU in each vehicle will periodically transmit the position, trajectory, transmission time and potentially other parameters about the vehicle provided by GPS and other sensors on the vehicle. The OBU in the SV will hopefully receive these transmission from all nearby NVs and use the input to help build a model of the world. As such, the OBU in our SV will know its own current location and trajectory along with a good estimate of the position and trajectory of the vehicles surrounding it. From this model the SV will also be able to predict the location of each NV at a future point in time. By looking for any intersections in the SVs path with those of NVs during the next two to five seconds collisions can be predicted ahead of time and the driver can be warned. A simple example of a CCWS in operation can be found in Figure 1.1 where a vehicle is provided with advanced warning it might not have based purely on visual information.

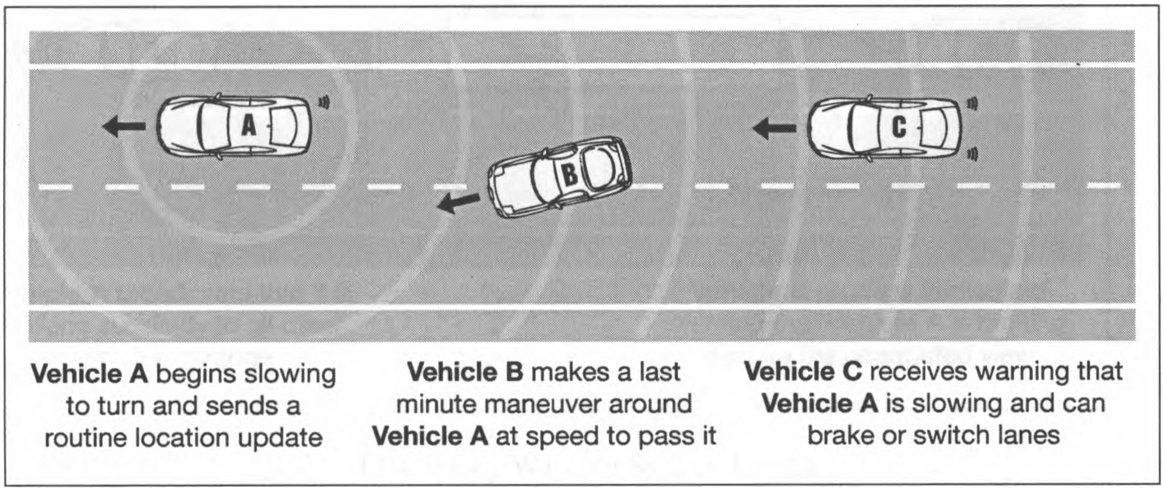

Figure 1.1: Cooperative Collision System Use Case

VBDA will provide additional information for the CCWS. By locating in real time the surrounding vehicles their position and trajectory can be calculated. This will provide a more up to date estimate of the location of the vehicles immediately surrounding the SV than VANET technology alone. This is important because the SV is most likely to be involved in a collision with the surrounding vehicles due to their proximity. Furthermore, in case a vehicle is not equipped with VANET technology and is not broadcasting its position the SV will still be able to detect it. The data attained from VBDA and VANET will be fused into a unified model for the CCWS to use.

The use case for an EWS is similar to a CCWS however instead of periodic location updates it will consist of rare but very important emergency warnings. Examples of such would be hard braking or involvement in a collision. While visually it is rare to be able to see more than a few vehicles ahead we can use VANETs to broadcast an emergency warning nearly instantaneously over a few hundred meters. An example of an EWS warning of a vehicle braking hard and warning a vehicle following behind that has obstructed vision can be found in Figure 1.2.

When reacting to an emergency situation normally each driver would first see the visual stimulus in front of them, for example a collision, process the information and move their foot to the brake pedal activating their brake lights. The next driver behind would then see the visual stimulus, the brake lights, and react as well. With the reaction time of drivers averaging at over one second this introduces significant delay when there are numerous vehicles follow one another closely [8]. With this delay if one driver fails to pay attention and brake in time, causing a collision, the vehicles behind him may no longer have time to react even in the best case and will end up as part of the collision as well. By broadcasting these warning to all vehicles the effect of human error can be reduced and multi-vehicle collisions can hopefully be avoided.

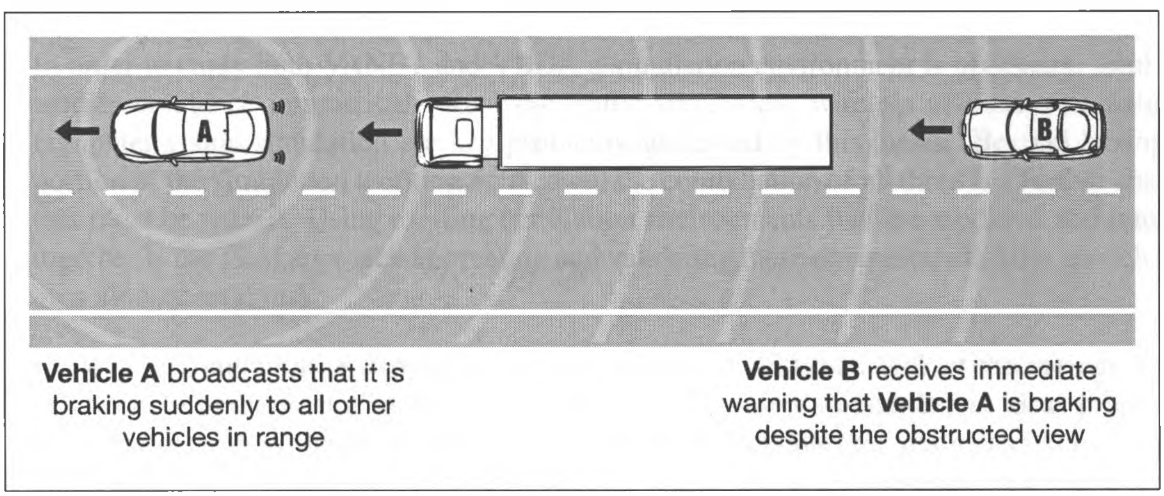

Figure 1.2: Emergency Warning System Use Case

The final use case for a HWS is similar to an EWS except it is less urgent. Instead of being a situation that requires immediate action it is something to take note of. The method of notifying the driver could be more robust as there is more time before any action is necessary. Examples would be a broken down vehicle stopped in a lane, lane closures for construction or weather conditions such as ice causing low traction. By broadcasting this warning to approaching vehicles the drivers can react before they see they reach the hazard. In Figure 1.3 we have an example of a HWS where a vehicle loses traction on ice and warns the vehicles behind.

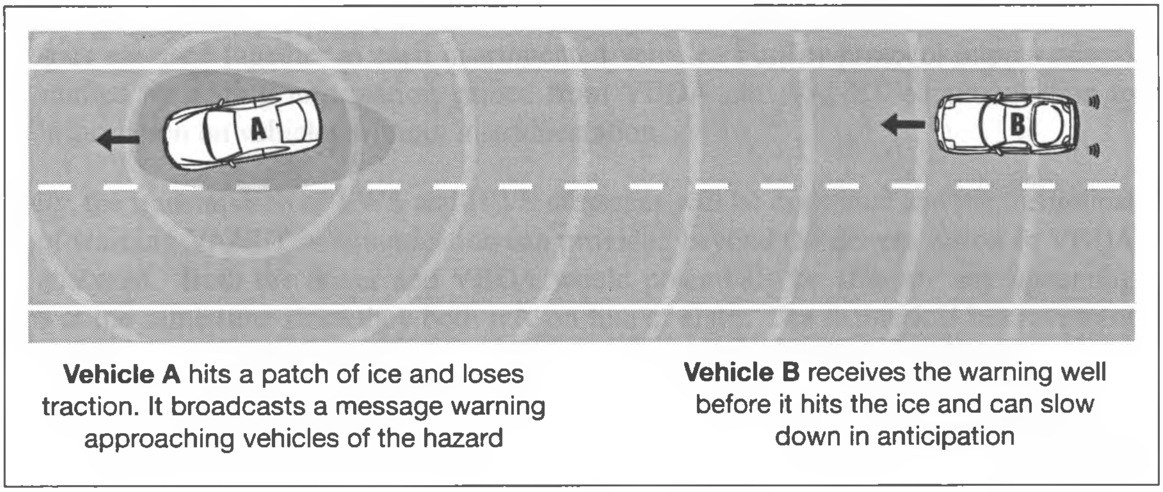

Figure 1.3: Hazard Warning System Use Case

1.6 Problems Addressed

In order to study both VANET and VBDA a simulation environment is necessary. Real world studies would be impractical. Realistic traffic simulation, wireless network simulation and computer vision simulation are key problems addressed by this thesis. Beyond having each portion of the simulation working on its own, the combination of all three is a further challenge that must be solved. Using existing simulation environments that are modified and integrated together is the ideal approach as creating and validating these environments from scratch would be a monumental task.

With a simulation environment in place this thesis will then look at the various network protocols to implement a successful VANET using industry standards and ideas proposed by other researchers in the field. Basing the VANET off technologies and ideas likely to be used in the real world makes this research much more relevant to technologies that will eventually be used in production vehicles. The various networking protocols will be either implemented from standards if they exist or be based on previous research and their operation described in detail. Various parameters can be adjusted and their effects can be studied on the overall performance of the VANET for CCWS, EWS and HWS.

With VANET and VBDA simulation in place the data collected from both systems can be recorded and analyzed for accuracy in comparison to the world model. The strengths and weakness of both systems can be examined and various strategies for combining the two data sources can be explored to find the best approach. The two data sources can then be fused into a unified model to base future decisions from.

With both systems in place and a unified model of the world we can then examine how adoption rates will effect system performance. It is assumed that both VANET and VBDA technologies are used together in each instrumented vehicle. Further protocol improvements will be studied by adding information gained from VBDA into VANET communication to provide information on vehicles without instrumentation.

Finally, the transmission of EWS and HWS messages will be examined and the theoretical amount of warning VANET communication can provided beyond the drivers vision or VBDA will be analyzed. Both the driver and VBDA would potentially be able see any upcoming problems at the same time since they both rely on line of sight. The theoretical improvement in warning time will provide a measure for just how effective these systems could be.

The end goal of the experiments performed in this thesis is to establish a base line measure for how effective VBDA and VANET technologies can be together. By leveraging the strengths of both, a system that out performs even the best implementation of single one will hopefully be possible. This is a research area that has yet to be explored.

1.7 Overview of the Thesis

Chapter 1 of this thesis has hopefully provided a good introduction to both VANETs and VBDA, outlined the motivations and challenges in implementing these systems, provided use cases for the various systems and explained what problems will be addressed. With this base the general area of research should be clear.

Chapter 2 will examine related work in both VANETs and VBDA. By seeing work done by other researchers and institutions we will be able to see where the technologies have been heading up to this point.

Chapter 3 will expand in detail the operation of VANETs. As VANETs are the primary focus their operation is crucial to understand. The various network layers will be examined in relation to the Open System Interconnection (OSI) networking model. Parameters and equations used in VANET simulation will be outlined in this chapter.

Chapter 4 will provide an outline of the simulator requirements, the specific simulators used and the modifications that were required for the experiments in this thesis. Additionally, the integration of the three simulators will be outlined to show how they work together.

Chapter 5 will include the remaining details for the various experiments that were performed. The experiments will be staged with later stages using the best strategies from previous stages. Any remaining parameters not touched on in previous chapters will be provided.

As well, the specific data to be collected in each experiment and reason for collecting that data will be provided.

Chapter 6 will provide the results of the experiments and their analysis broken down by stage and simulation. A discussion interpreting the results will follow each experiment to gain meaningful insight.

Finally, Chapter 7 will provide closing remarks and future directions for VANETs, VBDA and the integration of these technologies.

Chapter 2

Related Work

Over the past decade a wealth of research has gone into vehicular networks of all types. Standards have been implemented to guide how to use the limited wireless spectrum available. We will first look at a number of papers and standards that provide the basis for VANETs. We will also examine some other ideas that have been proposed but do not adhere to current standards to get an idea of potential issues and improvements.

Computer vision and associated applications for VBDA is another area that has seen considerable research. We will examine one project in specific, the RoadLab project at the University of Western Ontario, and the approach it takes to VBDA. The implementation and structure of the RoadLab project will be explained to get a general sense for its operation.

Finally, research that has been done on combining multiple sources of information in a vehicular environment will be looked at. The research done in this thesis will then be compared to previous work on VANETs, VBDA and their integration to get an idea of where it stands in comparison to previous work.

2.1 VANET System Configuration

Numerous papers have proposed a layout for VANET technology. The OBU will consist of a computer and wireless radio unit. The computer can be located anywhere on the vehicle. Given that power and weight are not restrictions there should be no problem housing a significantly powerful computer to run VANET applications. The radio transmitter and receiver are mounted either on the dash or on the roof of the car and broadcast in all directions [7]. A simple VANET configuration can be seen in Figure 2.1.

The computer will interface with all the vehicles sensors, GPS, and any additional sensors deemed necessary for the system to have all of the information needed on the vehicles current state. The OBU will also be connected with the dashboard interface to communicate with the

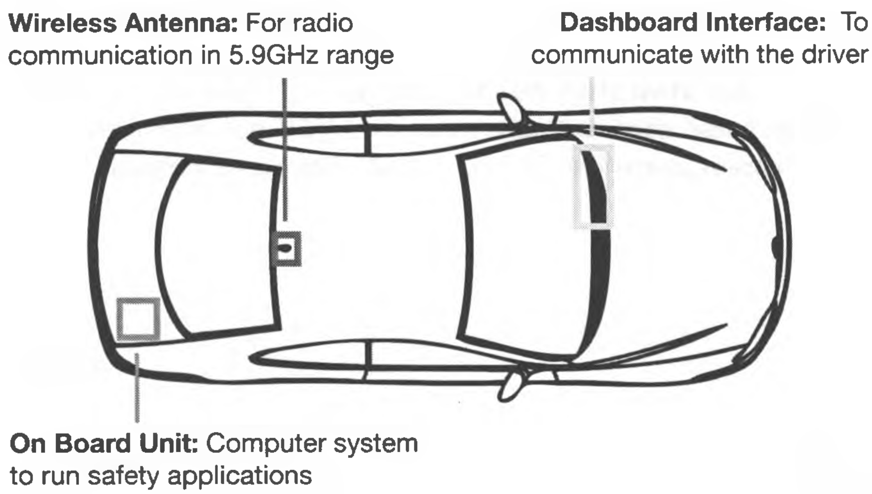

Figure 2.1: VANET System Configuration

driver. The exact method of communication is to be determined however we can speculate that it will usually take the form of simple non-visual cues to avoid interfering with the driver's field of vision. In the case of both VANET and VBDA technologies they will be tightly coupled and share a single computer system. In essence the wireless radio and cameras are both just sensors feeding information to the OBU.

2.2 VANET Communication

2.2.1 Physical Layer

In relation to the OSI physical layer the physical transmission characteristics of VANETs were dictated mainly by the US Federal Communication Commission (FCC). In 1999 they allocated 75MHz of Dedicated Short Range Communication (DSRC) spectrum near 5.9GHz. Given the size of the automobile market in the US it makes sense to standardize the frequencies used worldwide. As such, we have seen similar frequency allocations in Europe and Asia [9]. Canadian frequency allocations are likely to be similar to those found in the US. While there is some variation in the exact frequencies assigned in different regions, they are close enough that the same antenna, transmitter and receiver will work.

The IEEE also played a major role in dictating the design of the physical layer. They have developed the IEEE 802.11p standard for Wireless Access for Vehicular Environments (WAVE). It is based heavily on the IEEE 802.11a standard which was ratified in 1999. Both standards use Orthogonal Frequency Division Multiplexing (OFDM) along with a variety of modulation schemes to transmit data in the 5.9GHz spectrum. While IEEE 802.11p was only ratified on July 15th, 2010 it has existed in draft form for the better part of a decade.

Since the physical layer for VANETs was set relatively early there was little research into alternatives. As such most work on V2V communication has been based on the 802.11 set of standards with early research using 802.11b or 802.11a and newer research implementing 802.11p.

2.2.2 MAC Layer

IEEE 802.11p also defines the Media Access Control (MAC) layer for VANETs. The MAC protocol is based heavily on the standard 802.11 Carrier Sensing Multiple Access with Collision Avoidance (CSMA/CA) scheme. As VANETs are highly time critical the random access nature of CSMA/CA can pose a problem. There is no guarantee of timely or successful transmission. The random backoff time associated with 802.11 CSMA/CA MAC provides no upper bound on transmission latency and the unreliable nature of wireless communication provides no guarantee of delivery. Coupled with the fact that 802.11p was only recently ratified there has been research into alternative MAC protocols.

One type of alternative MAC protocol is a Time Division Multiple Access (TDMA) protocol. In a paper by F. Yu and S. Biswas a distributed TDMA MAC protocol is proposed [10]. Each node transmits in a set time slot that is not shared by any of its one or two hop neighbours. By including a bitmap of all available time slots with each transmission, vehicles self organize to achieve this. Whenever new two hop neighbours are discovered vehicles reorganize what time slot they broadcast in automatically to ensure no packet collisions, although they are still possible before reorganization. This allows vehicles to avoid packet collisions and interference under most circumstances and puts a small upper bound on maximum time before transmission. This is a desirable behaviour since the transmission of safety messages is guaranteed in a timely fashion. This guarantee of course comes at the cost of higher overhead and a more complicated MAC protocol.

Simpler modifications to the 802.11p MAC protocol have also been proposed. While maintaining the same general MAC protocol T. Taleb, K. Ooi and K. Hashimoto provide a risk aware backoff procedure instead of the standard 802.11 random backoff window [11]. Instead of using the standard 802.11 backoff procedure a risk level is calculated. Depending on the risk level the backoff time is modified to ensure that any emergency situations have priority in transmission. This again comes at the cost of higher overhead and requires vehicles to self organize into clusters for computing the risk level.

With the ratification of 802.11p a more basic approach has been taken maintaining the majority of the 802.11 CSMA/CA MAC protocol and reducing the amount of overhead. While this does introduce some new problems it generally provides excellent performance and no connection setup time.

The 802.11p protocol is complemented by the IEEE 1609 for upper layer operation. The 1609.4 protocol defines multi-channel operation using 802.11p. A more complete description follows in Section 3.3 but there is a total of seven channels with one Control Channel (CCH) and six Service Channels (SCHs). This allows safety messages to take priority in the CCH while nodes can periodically switch off to SCHs for non-safety related applications.

The performance of the ratified 802.11p protocol has been evaluated by numerous researchers [2] [12] [13]. In the work by S. Grafling, P. Mahonen and J. Riihijarvi they looked at the throughput and delay for packet transmission under a variety of scenarios using 802.11p and 1609.4 [2]. They found that delay in the control channel for safety messages was well below 100ms, a commonly agreed upon maximum delay for effective safety applications, until over 1000 messages per second were being broadcast on the channel. Well designed applications should hopefully avoid this kind of network congestion and allow safety messages to be transmitted in a timely fashion. Overall the protocols performed well under almost all conditions without incurring extra overhead associated with other proposed MAC protocols.

2.2.3 Network and Transport Layers

The network layer for safety applications has not been a popular research topic. Due to the broadcast nature of safety applications there is little management that needs to be done at these layers and as such they are not particularly important in implementation. IEEE 1609.3 defines the network layer services. Non-safety applications may use IPv6 as a network layer in SCHs and safety applications are to use the WAVE Short Message Protocol (WSMP).

In terms of safety applications the transport layer is generally not needed. For non-safety applications either Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) can be used on top of IPv6. This allows existing internet services to be adapted to VANETs easily.

2.2.4 Application Layer

A great deal of research has been done in terms of the various applications that can be built on top of a VANET. One large area of research is distributed traffic management [14] [15]. While better traffic management certainly has widespread appeal and worthwhile benefits it is outside the scope of this thesis. The other large area of research in terms of VANET applications has been either in CCWS or EWS type applications.

In work by H. Tan and J. Huang the feasibility of implementing a CCWS is considered [16]. Three main requirements are put forth in order to create a CCWS. First, accurate GPS data is required to properly place a vehicle inside of a lane. Since lanes are typically three to four meters wide an error of less than a meter is outlined as required. Second, the CCWS must have low enough latency that vehicles can keep an up to date model of the world. This means that the latency and bandwidth of the channel must be adequate for each vehicle to transmit its location many times per second. Finally, GPS data should be available at all times. Urban environments often interfere with GPS signals so alternatives must be put in place to ensure positioning data is available at all times.

They find that all of these criteria are possible. Differential GPS (DGPS), which uses ground based stations to provide more accurate GPS estimates, is available and can easily reach sub meter accuracy. Furthermore, by adding in additional sensors, position information can remain accurate even when GPS signals are temporarily blocked. The paper suggests a motion sensor suite which could rely on a gyroscope, accelerometer and speed sensor to ensure both position accuracy and availability. The paper leaves the latency of the channel to the reader but we have already seen that sub 100ms latency and high message rates are attainable under a wide variety of conditions [2].

From here the paper presents a proposed CCWS system and some promising experimental results. Overall it shows that a CCWS is feasible and produces real world results to back that assertion up. Finally, it also touches on the interplay between a regular Collision Warning System (CWS), using on-board sensors such as in our VBDA system, and CCWS and suggests that they might be useful in combination due to the more regular update intervals of the CWS.

In work by S. Rezaei et al. they propose an advanced CCWS and a number of schemes for how often to broadcast location estimates [6]. The most basic scheme is broadcasting location updates on a set interval. A more complex scheme is based on error thresholds for longitudinal and lateral position. The SV has a Self Position Estimator (SPE) and Remote Position Estimator (RPE). The SPE is based on all available GPS and vehicle sensor data to provide the most accurate position estimation for the current time. The RPE is where the vehicle is projected to be based on the last set of broadcast location information. If the difference in position between the SPE and the RPE crosses a set error threshold then a new position message is created using the up to date information from the SPE and is broadcast as soon as the channel is free. The error is tracked in both lateral and longitudinal directions as lateral error has more of an effect since it is possible a small lateral error could place the vehicle in an incorrect lane.

As there is no guarantee of message delivery to all surrounding vehicles a number of issues are introduced with this scheme. First, if a NV fails to receive a location update message from the SV then the difference between the Neighbouring Vehicle Estimator (NVE) of the NV and the SPE of the SV may be greater than the set error threshold. As this is unavoidable applications must be error tolerant. One proposed solution is for the SV to rebroadcast the message a second time within 50ms of the initial broadcast in order to increase the likelihood of all NVs receiving the message. While this can help it cannot eliminate the risk that a NV will fail to receive both packets. A second problem lies with communication delay. Once the difference between the SPE and RPE of the SV crosses the set error threshold and a new location message is generated there can still be a delay before the channel is idle. During this time the difference between the NVE and the SPE will be greater than the error threshold. However, the paper shows that even with a relatively long transmission delay of 50ms the effect of the additional delay will be negligible.

In order to maintain frequent contact with nearby vehicles another idea is proposed by C. Huang et al. [17]. As the WSMP provides the ability to vary the power level of transmissions, the application layer can dynamically change the transmission power. The greatest risk of collision is of course with the vehicles immediately surrounding the SV therefore it is most important for nearby vehicles to remain aware of each other's position even if this means vehicles further away go out of range. By dynamically raising or lowering the transmission power of CCWS messages we can reduce network congestion without reducing frequency of location updates.

Finally, another proposed safety application in the paper by F. Yu and S. Biswas is an Urban Intersection Crash Warning System (UICWS) [10]. The system detects when a vehicle is going to run a red light and cross into oncoming traffic. This is a specific case of an EWS proposed earlier. The system aims to provide drivers who are at risk of colliding with the vehicle running the red light advanced warning of the situation. While it may not be possible to stop all vehicles before a collision occurs the hope is to avoid multi-vehicle collisions. Again, this paper performs simulations and shows promising results for a reduction in the number of vehicles involved in an accident.

2.2.5 Security

Security is very important for VANETs and must be considered in depth before they are implemented. As we have seen with other computer networks malicious users are common and often cause widespread problems. With the potential for fatal consequences in a vehicular environment the motivation for high security is obvious. To make security in VANETs more difficult there is no way to centrally authenticate a node in the network. Messages will often need to be dealt with in milliseconds and messages must be authenticated independently.

There are three primary security issues to be considered when dealing with VANETs. First, we do not wish for vehicles to be tracked easily using the information they broadcast. Second, we do not want drivers to be able to alter their VANET transmissions for their benefit. Finally, there is the possibility of malicious users sending out messages with intent to harm others or cause problems. All of these problems could seriously harm the image of VANETs and hamper their adoption by the public.

Numerous strategies have been outlined by various researchers for how to secure VANETs however one predominant set of ideas regarding security for safety applications using Public Key Infrastructure (PKI) stands out. The core of this idea has been implemented in IEEE 1609.3: Security Services for Applications and Management Messages. In a paper by M. Raya et al. they outline this scheme in general [18]. Every vehicle that has been authorized to take part in the VANET receives a number of public and private key pairs for PKI encryption. These key pairs are provided by a Central Authority (CA) who is most likely a government agency or automobile manufacturer. All key pairs have a corresponding certificate signed by the CA. Safety application messages are signed with a Hash-based Message Authentication

Code (HMAC) using the vehicle's private key. A signed certificate from the CA that corresponds to the vehicle's private key is included along with the message as well. The increased overheard from this scheme is limited to the size of the message signature and certificate. Using PKI encryption any vehicle that receives the message can authenticate that the message was indeed signed by the private key corresponding to the public key provided and that public key was issued by the CA. With this information it can be determined a vehicle has authorization to take part in the VANET.

If the Coordinated Universal Time (UTC) is included along with the message replay attacks are defeated since the time stamp will indicate the message is no longer relevant. By securing all keys in a Tamper Proof Device (TPD) extracting the keys is made as difficult as possible. This prevents the large majority of possible attacks. If a malicious user does get a hold of key pairs, measures can be implemented to detect malicious messages by looking for inconsistencies. If any malicious messages are detected the public key is reported to the CA automatically via V2I communication. The credentials deemed compromised are then added to the Certificate Revocation List (CRL) which is distributed back to vehicles via V2I and any messages using the compromised credentials are ignored. This solves the vast majority of security issues and allows vehicles to authenticate messages without querying the CA.

By providing multiple sets of credentials for each vehicle they can randomly cycle throughout the sets. This prevents easy tracking of the vehicle based on its public key. Choosing the best time to switch credentials, preferably at the same time as a switch in trajectory so the vehicle position cannot be linked to its predicted position, enhances security in this regard. Denial of Service attacks by jamming the channel are one final attack that is difficult to prevent. However, with VBDA we can still provide driver assistance in the absence of VANET communication. Overall this scheme solves the main security issues with minimal overhead and is likely to be widely adopted.

2.3 Vision-Based Driver Assistance

The RoadLab project run by Professor Steven Beauchemin at the University of Western Ontario focuses on VBDA. While the use of cameras to monitor the world surrounding a vehicle and computer vision algorithms to process these images into usable data to create a model is not the focus of this thesis the general idea behind it is important to understand. A simple VBDA configuration can be seen in Figure 2.2.

The vehicle is instrumented with ten cameras to give it a complete three hundred and sixty degree view of the world. The cameras are all placed in pairs in order to provide stereo images from which 3D location data can be extracted. There are two pairs of cameras facing forwards, one wide angle and one telephoto, to cover both nearby and objects far ahead. Two pairs of cameras are required as the wide angle cameras can only sense a limited distance ahead. To cover both the full range of vision and to detect vehicles far ahead a single camera system can't produce adequate performance. Facing left, right and backwards is only a pair of wide angle

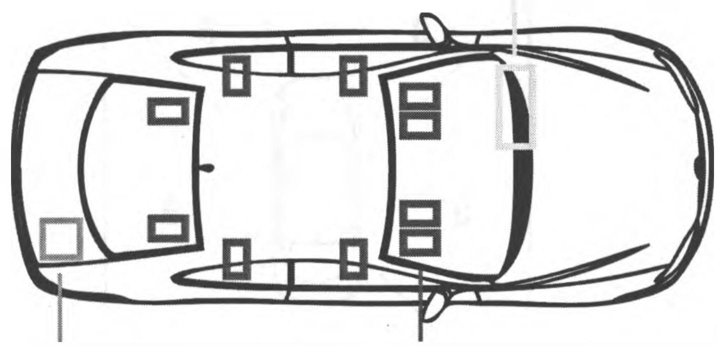

Figure 2.2: VBDA System Configuration

cameras. Being able to detect objects far away in these directions is less important. The camera systems and their layout is illustrated in Figure 2.3.

In addition to cameras monitoring the world around the vehicle, information from any other sensors in the vehicle is made available. Furthermore, there is another camera system monitoring the driver's gaze position inside of the vehicle. By knowing where the driver is currently looking and by association where the driver is not looking problems outside of the driver's gaze can be highlighted to ensure they are noticed before a collision could occur. Together all of these inputs are made available to the the OBU for processing and decision making.

The RoadLab project takes a layered approach in its architecture as illustrated in Figure 2.4 [1]. The first layer is Instrumentation. This includes the cameras, vehicle sensors, location information from GPS and even VANET communications. The second is Device Level Data Processing. Images provided by the cameras need to be analyzed and raw vehicle information needs to be transformed into a usable state. The third is Data Fusion and Integration. Information from multiple sources, multiple sets of cameras and information on the vehicle itself, needs to be combined and fused into a four dimensional space consisting of $\{x, y, z, t\}$. Finally, the fourth is Predictive Behaviour Model which uses the information from layer three to make decisions. The layered approach allows additional information sources to be plugged in easily on layers one and two and fused into world model in layer three without modifying layer four.

To make all of this happen the RoadLab project has put together a heavy duty real-time data processing and fusion computer [1]. It features sixteen 3.0GHz cores, 16GB of memory and solid state data storage. Processing images in real time requires a high level of computational power and we could easily run additional VANET applications on the same system.

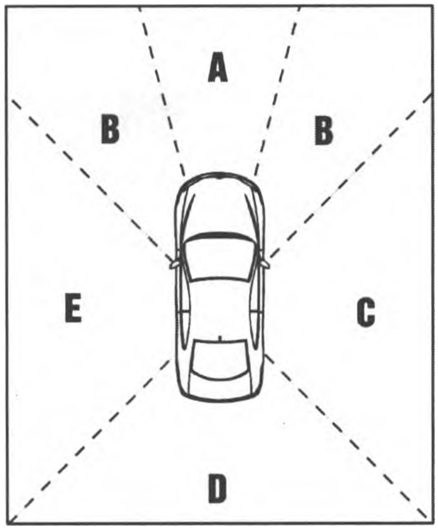

Figure 2.3: Vision zones diagram

The sets of cameras can identify surrounding vehicles and draw a 2D bounding around them in 3D space. The distance estimate to surrounding vehicles is highly accurate even at the furthest reaches of its range providing an excellent estimate for their position. Using subsequent position estimates in time the speed and trajectory of neighbouring vehicles can be estimated as well.

2.4 VANET and VBDA Interaction

The combination of both technologies is a field that has yet to be explored in great depth. The goal using both technologies is to create the most realistic model of the world. However, how to resolve conflicts in data has yet to be examined. Some work has been done combining vision or other on-board sensor based information with VANETs however it has been limited in scope so far.

In work by A.R. Girard et al. they describe a fusion of ACC, Cooperative Adaptive Cruise Control (CACC) and Forward Collision Warning System (FCWS) [19]. The instrumented vehicles have forward facing radar to sense the distance to a vehicle ahead. When using ACC the vehicle will cruise at the set speed if there is no vehicle ahead. When the SV approaches another vehicle ahead, as sensed by radar, it will slow down and begin following at the same speed. If both vehicles are equipped for CACC they will communicate their desired speed and characteristics, such as max braking deceleration, to set a safe following distance. In the case of a change in speed or beginning to brake they will communicate these things as well.

This is a good but simple example of fusing two sources of information, one gained from

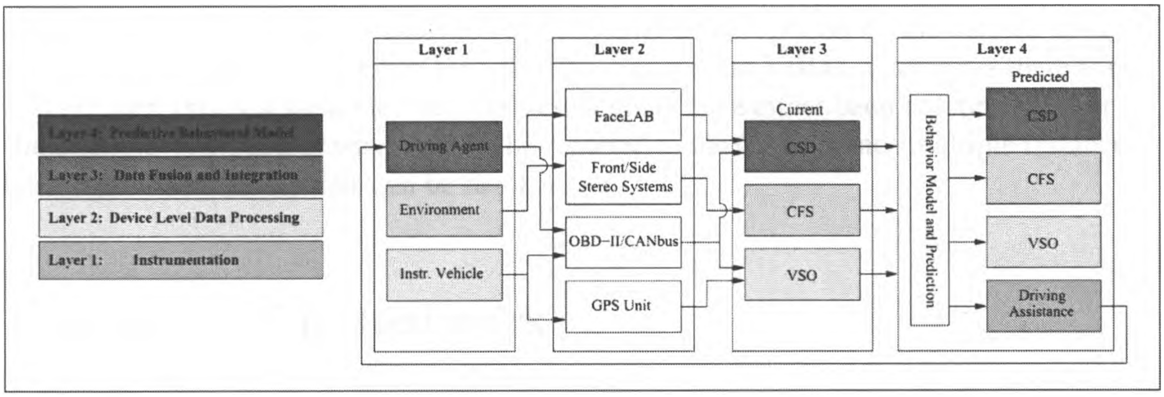

Figure 2.4: RoadLab architecture showing both the layers and interaction between layers [1]

sensors monitoring the environment around the vehicle and the other from wireless communication. The article provides further insight into how such a system might work, controlling the throttle and brakes automatically under the majority of circumstances. However, when there is a conflict the response is not elegant. If the forward sensing RADAR detects the vehicle ahead is not where it should be based on CACC, the cruise control system will automatically disengage and return control to the driver. While this may be the ideal behaviour in this simplified case in our fusion of VANET and VBDA technologies this will not be possible. As the OBU will not be making any decisions, only providing warnings to the driver, it must resolve conflicts in data on its own.

Another approach that merges VANETs and vision is proposed in papers by two sets of researchers [20] [21]. Both of these papers use VANETs to share video feeds between vehicles. In both papers a video stream from one or more surrounding vehicles is compiled and displayed to the driver in real time. Both papers deal with a situation where the driver of a vehicle has obstructed vision. One while overtaking another vehicle and the other while entering an intersection. This additional video feed allows the driver to make informed decisions about how to proceed.

While streaming video between vehicles is certainly an interesting idea there are two main drawbacks. First, the driver must take his eyes off the road to view the video feed. This presents a distraction and could cause a collision in itself. Second, the sharing of video is bandwidth intensive while simultaneously requiring low latency. There is limited bandwidth available for VANETs with potentially a large number of vehicles sharing it so high bandwidth applications such as this may be unrealistic.

A third application that is discussed in the literature is the use of cameras to help with security in VANETs. As discussed previously a likely security scheme will involve a PKI based scheme where each authorized to be part of the VANET will be issued credentials to sign messages with. Any malicious nodes will need to have their credentials placed on the CRL so they can no longer participate. Before this happens though the malicious node must be detected and reported to the CA. Any discrepancies between VANET communication and data obtained from VBDA can be analyzed and automatically reported by the vehicle to help aid in this. For example, if a vehicle is reporting its location and VBDA determines there is no vehicle there then this is a signal the vehicle's credentials have either been compromised or it is malfunctioning [22]. Either way, this can be reported to the CA and once multiple instances have been reported its credentials can be revoked.

2.5 Summary of related work

A wide variety of work has been done in both VANETs and VBDA. All of the pieces to implement an effective VANET exist currently. Researchers have examined the proposed technologies, added refinements and performed testing to ensure they will be able to handle real world circumstances. VBDA and related technologies have been looked at in depth both in the RoadLab project and automobile manufacturers for production vehicles. It is clear that both of these technologies will one day be reality on consumer vehicles. Looking at how the two mature technologies could interact has gone unexamined however.

By using existing work on VANETs that adheres to the existing standards or likely will, simulations can be performed on a prototype VANET setup. Again, using the existing RoadLab vehicle setup, results that VBDA will likely produce in the real world can be simulated as well. By simulating both of these technologies side by side the benefits of both can be quantitatively categorized. The information from both sources will then need to be combined. This thesis will look at the simulation of both technologies, how they complement one another and how to combine both sources of information. The combination of these technologies has yet to be examined in depth.

Chapter 3

VANET Fundamentals

3.1 Physical Layer

The physical layer properties and protocols for vehicular networks are well defined as a result of work by the FCC and the IEEE. The FCC has allocated a 75MHz band near 5.9GHz in the DSRC band for use by vehicular networks. The 75MHz band is divided into seven 10MHz wide channels as shown in Figure 3.1. There is one dedicated CCH reserved primarily for safety messages and six SCHs that are either open to all services or reserved for a special purpose. The CCH is used for broadcasting short safety frames and advertisements for services offered on SCHs. The IEEE 802.11p standard defines how messages should be broadcast within these channels. An OFDM transmission scheme is used for the physical radio transmission.

Channel Number	172	174	176	178	180	182	184
Channel Type	SCH	SCH	SCH	CCH	SCH	SCH	SCH
Centre Frequency	5.86	5.87	5.88	5.89	5.9	5.91	5.92
Max TX Power	33	33	33	44.8	23	23	40
Radio Range	V2V	Medium	Medium	All	Short	Short	Intersection
Application	Non-safety	Non-safety	Traffic Efficiency	Critical Safety	Critical Safety	Traffic Efficiency	Traffic Efficiency

Figure 3.1: WAVE DSRC Channels [2]

OFDM transmission incorporates a number of complex steps. It entails mapping a wide band signal to a number of narrow band signals and broadcasting them simultaneously on numerous subcarriers. The orthogonal nature of OFDM allows the subcarriers to overlap one another increasing throughput. First, bit order in the packet to be broadcast is scrambled to avoid long runs of ones and zero. Next, the packet has a convolution code applied where extra bits are added to it for Forward Error Correction (FEC). Bits are then interleaved to change their order and reduce the effect of burst errors during transmission. The encoded and interleaved bit pattern is then mapped to symbols, according to the modulation scheme chosen, to be broadcast simultaneously on all subcarriers. The completed signal is transformed to an analogue signal using the inverse fast Fourier transform and broadcast. This is not a complete overview of OFDM but should help in the understanding of it. Received OFDM signals are decoded in a similar fashion.

IEEE 802.11a already communicates at 5.9GHz and was adapted for use in vehicular networks. Both the 802.11a and 802.11p protocols uses OFDM to encode messages for transmission. However the channel width in 802.11p is half of that of 802.11a, 10MHz and 20MHz respectively. To accommodate the reduction in bandwidth all of the timing parameters were doubled. The symbol duration was increase from $4\mu s$ to $8\mu s$ in turn halving the data rate. Included in the duration of each symbol is a $1.6\mu s$ guard period during which no data is transmitted. One advantage of the increased symbol and guard period duration is that it provides additional protection against multipath fading and the resulting inter-symbol interference. It has been found that timing parameters used in 802.11a might not be suitable for the complex multipath environment found in vehicular networks and that using 10MHz channels at half the data rate was the best solution [23]. Another benefit of the reduced channel bandwidth is a reduction in the thermal noise present. Thermal noise is the base level of noise always present. It has been demonstrated that a -3dBm reduction in thermal noise over 802.11a is expected, with an expected overall level of -98dBm [24].

There are four possible modulation schemes, Binary Phase-Shift Keying (BPSK), Quadrature Phase-Shift Keying (QPSK), 16-Quadrature Amplitude Modulation (QAM) and 64-QAM, each available at two different coding rates providing 8 data rates ranging from 3Mbps to 27Mbps. There is a total of 52 orthogonal sub carriers with 48 subcarriers for data and 4 subcarriers serving as phase reference. For the purposes of our simulations a data rate of 6Mbps was chosen. To achieve 6Mbps the QPSK modulation scheme is used with a coding rate of $\frac{1}{2}$, that is for every one bit input there are two bits output.

The physical layer header has a total duration of $40\mu s$ during which a number of training symbols, symbols and finally a BPSK modulated header, containing the modulation scheme to be used for the rest of the transmission, are broadcast.

The ideal method to simulate the physical layer is in the bit domain [25]. That is each individual bit is simulated in transmission and has attenuation effects applied to it. There are a variety of ways outlined by K. Wehrle to do this [25]. While bit domain simulations produces accurate results the computation complexity is staggering. One study which simulated OFDM in the bit domain had run times of 50 seconds per 100 OFDM symbols [26]. While computing power has progressed significantly in the past seven years this figure still puts the computational complexity into context. We wish to simulate thousands of symbols per simulation second making this technique inappropriate for our desired experiments.

Instead we will simulate the physical layer in the packet domain with Packet Error Rate (PER) model [25]. Each individual packet is simulated, has attenuation applied to it and an overall minimum Signal-to-Interference plus Noise Ratio (SINR) is calculated for it. The SINR is represented by the receiving signal power over the thermal noise plus the signal power of all interfering transmissions. Based on the minimum SINR for the packet a theoretical Bit Error Rate (BER) is calculated. Packet errors are then predicted in a stochastic fashion. An existing BER and PER model for OFDM signals is used in order to ensure validity [27].

Modeling the physical layer takes all of the parameters mentioned into account during the transmission in order to produce accurate results without enduring the overhead of actually simulating the full encoding and decoding process of OFDM. This allows large scale simulations that would not otherwise be possible.

3.2 Channel Modeling

After a packet is transmitted by the physical layer as a radio signal it has various attenuation effects applied to it by the environment it passes through. These attenuation effects along with the transmitted power determine the received power. Three types of attenuation effects are of interest to us, path loss, shadowing and fading.

Path loss represents the average attenuation of the signal over a particular propagation environment. While the resulting receiving signal strength will vary widely from millisecond to millisecond and second to second, path loss represents the bulk of the signal attenuation applied in our model. So much so that the path loss parameters are used to determine the maximum transmission distance. The path loss is modeled on the distance between two nodes, the wave length $\lambda = \frac{c}{f_c}$ with f_c being the frequency in our case 5.89GHz and a path loss coefficient. We use a path loss coefficient of $\beta = 3.0$, a value within the range that roughly models an urban environment [24] [25]. A path loss model included in the simulation environment is used.

The exact path loss coefficient is relatively unimportant as its attenuation effect is applied to all packets equally. The maximum transmission power of 35.4dBm was decided based on a target transmission range of 300m. With a maximum transmission power of 44.3dBm on the CCH we would simply increase or decrease the transmission power to account for the effect of different path loss models.

Shadowing is another attenuation effect to consider. It accounts for second to second variations in signal strength based on a changing environment. As such, it is a stochastic measure. These second to second effects are caused by the movement of objects in the propagation environment. In a vehicular environment this will obviously be very pronounced. Again, an included log normal shadowing model is used with a mean of 0 and standard deviation of 4dBm to match existing models [25].

The final attenuation effect considered is fading. Fading represents short term variations in signal strength, measured in milliseconds, that are the result of multiple copies of the signal being reflected and received slightly offset in time at the receiver. It is based on both Line of Sight (LOS) and Non-Line of Sight (NLOS) copies of the signal. No fading model is used in our simulations are an accurate model is unavailable for a vehicular environment.

3.3 MAC Layer

IEEE 802.11p uses the same Distributed Coordinated Function (DCF) MAC scheme, which follows a CSMA/CA strategy, as with the other 802.11 protocols to control shared access to the channel. All network layer packets are encapsulated in a 34 byte MAC header. The MAC header contains control information, packet duration, addressing, sequence control, Quality of Service (QoS) and a Cyclic Redundancy Check (CRC) checksum. Figure 3.2 shows the complete 802.1pp MAC packet structure. The complete MAC packet is then based down to the physical layer for transmission.

2	2	6	6	6	2	2	6	?	4
Frame Control	Duration /ID	Address 1	Address 2	Address 3	Sequence Control	QoS	Address 4	Data	FCS

* length of fields in bytes

Figure 3.2: IEEE 802.11p Packet Structure

The CSMA/CA scheme works as follows. If a node has a message to send and the channel is free then the node first monitors the channel for the amount of time defined by the DCF Interframe Space (DIFS). If the channel remains free for the duration of the DIFS then the node immediately transmits the packet. If the channel is busy or becomes busy at any point during the DIFS then a random backoff window will be chosen. The random backoff window is a number of time slots between $[1, CW]$, with CW being a number in the range $[CW_{min}, CW_{max}]$ starting at CW_{min}. If packet transmission fails then CW is doubled each retry until it reaches CW_{max}. After the channel becomes free again the node will wait for the duration of the DIFS then begin counting down the duration of the random backoff window. After counting down the random backoff window the node then immediately transmits the packet. If instead another node begins transmitting while counting down the random backoff window the node will wait until the channel is free again, wait the duration of the DIFS and continues counting down. Using a Carrier Sensing Multiple Access with Collision Detection (CSMA/CD) scheme is not possible since the radios used for 802.11 are half-duplex. It is not feasible to monitor the channel while broadcasting so collision cannot be detected.

IEEE 802.11e Enhanced Distributed Channel Access (EDCA) is implemented as well. It adds packet prioritization creating different categories of traffic and adjusting the timing parameters for each. It defines four Access Categorys (ACs) for packets and for each replaces the DIFS with an Arbitrary Interframe Space (AIFS) [7]. The AIFS acts in exactly the same way as the DIFS. The random backoff window is also altered for each AC. Table 3.1 lists the four AC available in IEEE 802.11p with EDCA. The MAC layer is responsible for maintaining separate queues for each AC and choosing the packet to be sent via an internal contention mechanism [2]. Our implementation simply always sends the highest priority packet first. The other relevant parameters to IEEE 802.11p MAC are the slot time of 13μs, the Short Interframe Space (SIFS) of 32μs and the seven channels defined in Figure 3.1.

	CW_{min}	CW_{min}	**AIFSN**	**AIFS (μs)**	**Backoff (μs)**
A3	3	7	2	58	39
A2	3	7	3	71	39
A1	7	225	6	110	91
A0	15	1023	9	149	195

Table 3.1: 802.11p MAC EDCA Parameters

To find out when the channel is free the MAC layer also handles channel sensing. The current status of the channel can be determined in two ways. The first is provided by the physical layer in the form of Clear Channel Assessments (CCAs). If the physical layer is current receiving a valid frame or if the received energy levels are above a set threshold then the channel is determined to be busy. The second is a virtual carrier sense provided by the Network Allocation Vector (NAV). The MAC header contains a NAV which indicates the length of the transmission. Upon receiving the start of a frame the NAV is set and then decreased until it reaches zero. If either physical or virtual carrier sensing indicates the channel is busy then it is considered busy, otherwise it is considered free.

Addressing in 802.11p is slightly different than in other 802.11 standards. RSUs have a fixed 48 bit MAC address. However, OBUs generate a random 48 bit MAC address on start up. If a collision occurs where two OBUs have the same MAC address they both automatically generate a new one.

Communication between nodes is different slightly in 802.11p than other 802.11 standards. Due to the time sensitive nature of VANET safety applications overhead before communication must be reduced. Normally 802.11 requires nodes to join a Basic Service Set (BSS) before communicating. The BSS then has a Basic Service Set ID (BSSID) that is the MAC address of the AP in infrastructure mode or randomly generated MAC in independent (ad-hoc) mode. Nodes join a BSS in a multi-step authentication process and then include the BSSID with all communications. The BSSID is then used to filter out communication intended for other BSSs. This multi-step process is too time consuming for VANETs [28].

802.11p introduces the wildcard BSSID , all bits set to 1, which allows instantaneous communication between nodes. All nodes can send and receive messages addressed to the wildcard BSSID as long as they are on the same channel. 802.11p also introduces the WAVE Basic Service Set (WBSS) which is similar to the BSS in wireless computer networks but with less overhead to join. A WAVE Service Announcement (WSA) is first sent out by the node offering a WBSS in the CCH. The WBSS can then be joined by any node in a single step. However, WBSSs are only formed on one of the SCHs. While WBSSs are unimportant for safety applications running on the CCH it should be noted that nodes in a WBSS still receive frames addressed to the wildcard BSSID as long as they are currently monitoring the same channel. This allows vehicles to communicate safety messages instantaneously without any setup to all other nodes in the vicinity.

Also of note is that both Acknowledgement Packet (ACK) frames and Request to Send Packet (RTS)/Clear to Send Packet (CTS) frames are generally not used on the CCH. Most safety messages are broadcast in nature and there is no ACK specified in IEEE 802.11 for broadcast packets. A single broadcast packet could generate hundreds of ACK packets in response. This reduces overhead and helps ensure timely communication of safety messages. As there is no ACK in response for most safety packets there is no transmission confirmation. However, since the network topology is constantly changing it would be impractical and of limited use to implement features that rely on transmission confirmation, such as retransmission of failed packets.

Since there is no central AP controlling wildcard BSSID packets there is no way to implement RTS/CTS packets. This is desired as it reduces overhead however it introduces a second problem. The hidden node problem, as demonstrated in Figure 3.3, it is when Node A and Node C are outside each other's transmission range. They both think the channel is free and both broadcast a message during the same time period. Node B, who is inside the transmission range of both, receives the message from both vehicles at the same time and due to the interference can't successfully receive either message. The hidden node problem is a trade off of the reduced overhead required for quick communication between nodes in a VANET.

Another drawback caused by the lack of ACKs with broadcast packets is that the typical auto bit rate selection schemes cannot be implemented. Normally multiple unsuccessful transmissions in a row will result in the bit rate being lowered and multiple successful transmissions will result in the bit rate being raised. Without transmission confirmation it is unknown if the transmission was successful. This is not possible therefore it is left up to the application layer to choose the correct bit rate under most circumstances.

Multi-channel operation is governed by the IEEE 1609.4 standard. Each node has access to the UTC from the attached GPS unit. Each second is split into ten sync intervals that alternate between CCH intervals and SCH intervals. All nodes must monitor the CCH during the CCH interval. Nodes may switch to the SCH during SCH intervals to join WBSS or for safety applications choosing to use a SCH, otherwise they continue monitoring the CCH. As GPS relies on having a highly accurate estimate of the UTC, coordination between nodes is simple. There is however a short guard interval between CCH and SCH intervals to ensure small errors in the UTC do not cause any problems [29].

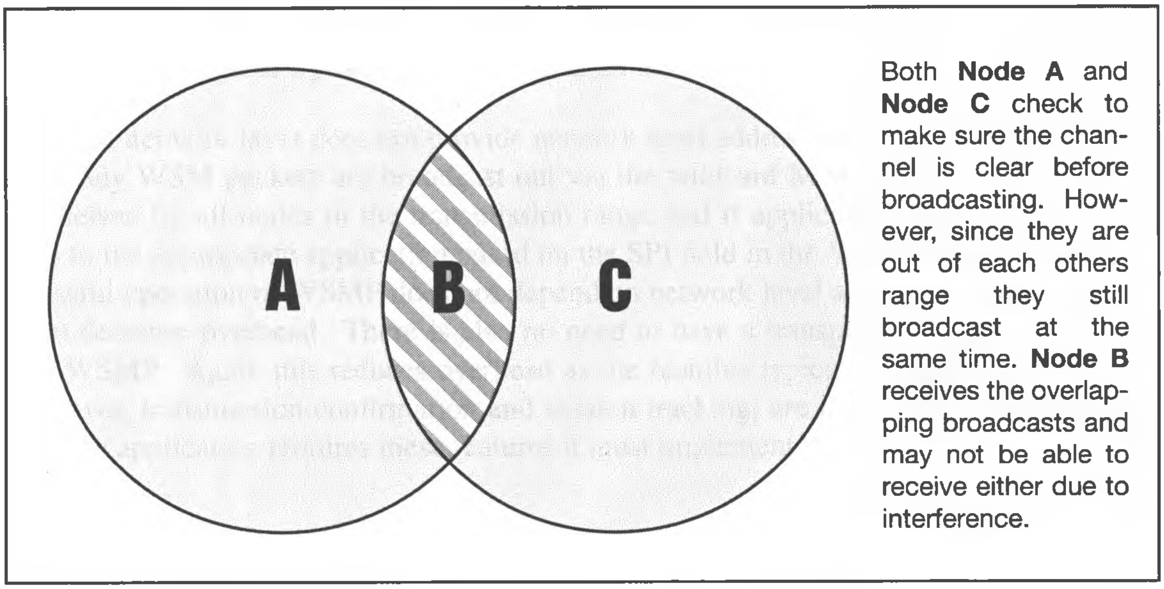

Figure 3.3: Hidden Node Problem

3.4 Network Layer

The network layer for vehicular networks is defined in the IEEE 1609.3 standard. There is support for running both IPv6 or WSMP on top off 802.11p. IPv6 is used for non-safety applications on SCHs only. As such, IPv6 as a network layer is not important to this thesis and WSMP will be focused on. WSMP is lightweight and designed for safety applications that can be used on both the CCH or a SCH. It provides a small header, comprising 11 bytes, that identifies the service, the channel to be broadcast on, the data rate, transmission power level and a payload. The service is identified by the Service Provider Identifier (SPI) which allows the network layer to correctly pass incoming packets to the correct application layer service. Figure 3.4 shows the complete WAVE Short Message (WSM) packet structure.

1	1	1	1	1	4	2	?
WSM Version	Security Type	Channel Number	Data Rate	Transmit Power	Provider Service ID	Length	Data

* length of fields in bytes

Figure 3.4: WSM Packet Structure

The inclusion of the data rate and power level in a WSM packet allows the application to make decisions on how the physical transmission occurs. Given that the application layer will have the most information available to it regarding vehicle densities and positions it can make the most informed decisions. It also could potentially allow poorly designed applications to interfere with wireless communication. As such care must be taken to avoid transmitting too frequently or at too high of a power by the application design.

The WSM network layer does not provide network level addressing or handle address resolution. Many WSM packets are broadcast out via the wildcard MAC address via 802.11p. This is received by all nodes in the transmission range and if applicable the data payload is passed up to the appropriate application based on the SPI field in the WSM packet. Therefore the successful operation of WSMP does not depend on network level addressing and it can be omitted to decrease overhead. There is also no need to have a transport layer protocol built on top of WSMP. Again this reduces overhead as the features typically implemented on the transport layer, transmission confirmation and session tracking, are not needed for broadcast packets. If an application requires these features it must implement them.

3.5 Application Layer

While there are existing standards defining the physical, link and network layers in a definitive fashion there is not yet any set standards for vehicular network applications. The safety applications to be examined, a CCWS, EWS and HWS will be described in greater detail now.

3.5.1 CCWS

The goal of a CCWS is for each vehicle to remain aware of the location of the vehicles surrounding it. This is accomplished by knowing the location and trajectory of both the SV and any NVs. From this information any intersections in their paths can be predicted. In the case of two vehicles being on a collision course the drivers can be warned and the collision can hopefully be avoided.

A CCWS requires each vehicle to periodically broadcast some information about itself. At the very minimum this would include the vehicle's current location, speed, direction of travel, size and time of broadcast. This information would be attained from GPS system and onboard sensors. If available more specific information, such as lateral and longitudinal acceleration or wheel position, can be included as well. This can help increase the accuracy of the future location estimates when a vehicle is accelerating, braking or turning. Figure 3.5 contains a theoretical CCWS packet 242 bytes in length that will be used in our simulations. The AC for CCWS packets will be A1 as they are very frequent and can tolerate longer delays before sending than messages requiring immediate attention.

The time, location and trajectory fields all have a reasonable level of precious to allow float or double values. The vehicle information field provides room for additional vehicle information such length, width and performance characteristics. There is room for additional information here as well. Finally, the signature and certificate fields provided a hash of the

8	8	8	4	4	4	8	16	54	128
UTC	Longitude	Latitude	Elevation	Speed	Acceleration	Direction	Vehicle Info	Signature	Certificate

* length of fields in bytes

Figure 3.5: CCWS Packet Structure

message up to that point signed with the private key of the vehicle and the matching public key signed by CA as outlined in Section 2.2.5. The size of the security fields is based off of estimates found by other research [30]. Given the combined size of the application, network and MAC packets any variation in actual application layer packet sizes should have little effect on the simulation results and the theoretical CCWS packet provides a good base for estimating network performance.

How often the vehicle should broadcast this information is important. Very rarely do vehicles follow a set path without any acceleration, braking or wheel movement. In short time frames the difference in actual vehicle position should not differ significantly from the predicted vehicle position based on the last broadcast. For example, if a vehicle traveling 33.3m/s (120km/h) were to begin braking at 1g[1] immediately after sending out a location update, after half a second it would only be 1.3m behind its estimated position from the last broadcast. This is a near worst case scenario as only high performance sports cars vehicles can approach a 1G deceleration rate and location information can be broadcast more frequently than every 0.5s.

However, the error found between position updates compounds itself when predicting vehicle location further into the future. The location estimate that was only 1.3m off after half a second would be off by 44.1m, assuming constant deceleration, if we were to instead predict the location of the vehicle 3 seconds later. Therefore we must broadcast location estimates frequently enough to minimize errors. On the other hand broadcasting too frequently with high vehicle density is also an issue. Long contention periods can cause high latency and transmission interference can cause packets to be lost.

As such a balance must be struck between how often transmission of location updates take place. Broadcasting at a constant interval is one suggestion with proposals for both 100ms and 500ms [31]. These simple proposals can work under most circumstances. However, in the case of a vehicle rapidly changing its trajectory 500ms may be too long of an interval. Likewise, under high traffic densities 100ms may be too frequent of an interval resulting in high latency and congestion. For these reasons an adaptive CCWS is preferable.

The variable broadcast scheme proposed sends location updates after the difference between the vehicles actual location and predicted location varies by more than a set threshold

[1] 1g is the equivalent amount of force that earths gravitation field exerts on objects under its influence. It is equal to approximately $9.81 m/s^2$

[6]. The proposed scheme tracks error in both the lateral and longitudinal directions. However, the nature of the traffic simulator we have selected provides little variation in the longitudinal direction unless the vehicle is actively changing lanes in which case it instantly violates the threshold. Instead of tracking the position error independently a single position error size is tracked for both directions combined. Each time a location update is sent out the vehicles RPE is updated. The SPE provides the most up to date estimate of the vehicles position. At short intervals the error between the RPE and SPE is checked.

In the same paper the authors present a model of error in an advanced DGPS derived from measurements. We implement the same model for our location coordinates. Gaussian noise is added to the x, y, v, H fields according to an autoregressive model. The model, w, is defined by the as [6]:

$$w(k+1) = \alpha w(k) + \beta z(k)$$

$$\alpha = 0.9, \beta = 0.436, z(k) = N(0, \sigma_w)$$

$N(0, \sigma_w)$ is defined as white Gaussian noise with a mean of zero and standard deviation of 0.2m, 0.2m, 0.2m/s and 0.017 radians for x, y, v, H respectively. While the SPE tracks the vehicles actual location for statistical purposes, location and trajectory information used to update the RPE or sent out in CCWS messages has GPS error added to it.

The exact variable CCWS broadcast scheme used for our simulations is defined as the following. Let $P_e(k)$ be the 2D vehicles position vector (x, y) stored by the RPE when sending the last position update at time k, $P_c(k)$ be the vehicles real current position, H be the 2D unit vehicle direction vector, v be the current vehicle speed, a be the vehicles current rate of acceleration and t be the time interval since the RPE was last updated. The error between the predicted position and the actual position is represented as ε and and error threshold be represented by ρ.

$$P_e(k+t) = P_e(k) + H * (v\Delta t + \frac{1}{2}at^2)$$

$$\varepsilon = |\overrightarrow{P_c(k+t)P_e(k+t)}|$$

If $\varepsilon > \rho$ then a new location update is broadcast and the RPE is updated. If not another check is scheduled for 100ms in the future. Once $\varepsilon > \frac{1}{2}\rho$ the error checks are reduced to every 10ms to ensure a location update is sent out soon after the threshold is crossed. The rebroadcast scheme proposed is also implemented. Within the interval of 50ms after the first location update is broadcast, it is rebroadcast a second time. A maximum update interval is also defined for this CCWS scheme. If more than 1 second passes without sending a location update a new update will be generated without crossing the error threshold. This is to ensure nodes keep in contact while stopped.

The second portion of the CCWS scheme is dynamically adjusting the transmission power. The scheme proposed has each vehicle keep track of the average channel occupancy $U_j(t)$, a real number between 0 and 1, that represents how much of the past second the channel has been free for [17]. This number is calculated by monitoring CCAs from the physical layer. A minimum and maximum channel occupancy is chosen, U_{min} and U_{max}, as well as a minimum and maximum transmission power, L_{min} and L_{max}. If the average channel occupancy is higher than U_{max} than L_{min} is used. Conversely, if the average channel occupancy is lower than U_{min} than L_{max} is used. Otherwise the transmission power is chosen by the following equation:

$$L_j(t) = L_{min} + \frac{U_{max} - U_j(t)}{U_{max} - U_{min}} \times (L_{max} - L_{min})$$

In the case of low channel occupancy broadcast frequency and power can be maintained. In the case of high channel occupancy broadcast frequency can be maintained while broadcast power is lowered. For these simulations L_{min} is set to a transmission power that provides reliable transmission over approximately 100m and L_{max} is set to to a transmission power that provides reliable transmission up to 300m or the maximum channel power. The lower bound on channel occupancy L_{min} is set to 0.25 and the upper bound L_{max} is set to 0.75.

3.5.2 Emergency Warning System

A EWS is quite similar in most ways to the CCWS. In the case of an emergency a vehicle will send out a message detailing its current location and trajectory along with the nature of the emergency. Unlike the CCWS instead of frequent periodic updates a EWS message would be a rare occurrence. These messages are also much more important.

For the purposes of this paper there are a few emergency situations we define that will create an EWS message. The first situation, hard braking, is essentially when a driver has to slam on the brakes. It could be defined as a threshold of maximum brake pedal pressure, for example when the brake pedal pressure is at ninety percent or more of the maximum brake pedal pressure. Alternatively it could be defined as when the deceleration rate goes above a set threshold, for example 4m/s. In this case we want to warn drivers behind the vehicle that they will have to brake as well. A second situation would be in the event of a collision. There are a number of indicators that a vehicle has been in a collision, for example airbags deploying or deceleration forces on a vehicle reaching above some set threshold, and many production vehicles today are already designed to sense when a collision occurs. The final situation would be a major loss of traction event. This could be caused by black ice or some other weather condition. Given that ABS, traction control and individual wheel speed sensors are standard equipment on most new vehicles in North America it is possible to detect a loss of traction. Again a threshold would be set for a dangerously low level of traction and if a vehicle encountered it an EWS message would be broadcast.

These emergency situations would most likely be visible to the vehicle immediately following. However, depending on the road layout, vehicle size and neighbouring vehicle positions, other vehicles immediately preceding would have to wait for a cascade effect before they are informed. Without an EWS message the vehicle immediately following would brake, informing the vehicle behind him, and this would continue along the line of vehicles with a time delay between each vehicle being informed. Instead, all vehicles within communication range could be informed at the same time providing additional reaction time for their drivers.

The EWS message would include the same information as a CCWS message in order to provide location data for where the event occurred and some idea of where the vehicle that broadcast it is heading, along with some event specific information. For example, rate of deceleration could be included in the hard braking warning if it is not already part of the CCWS, a measure of accident severity could be included along with the accident warning or an estimated coefficient of friction could be included along with the major loss of traction warning. The overall packet structure would be very similar to Figure 3.5 with a few extra bytes for added information. The AC used for EWS messages would be A3 as they are of the utmost importance.

A concern is avoiding network congestion when disseminating emergency warning messages. The system described so far is single hop only so there is no risk of too many nodes forwarding a message and creating network congestion. However, due to the emergency nature of these situations many of the vehicles following will generate their own EWS message. When one vehicle brakes hard the following vehicle must brake hard as well. In the case of a collision the vehicles following would have to be brake hard or could perhaps be involved in the collision themselves. It is easily apparent how the situation can spread from one or two vehicles to many vehicles.

As there is the potential for each EWS message to set off a string of other EWS messages from the vehicles reacting further refitment is warranted. Any EWS messages sent out in reaction to a previous EWS message will have the AC of A2 instead of A3. If another important emergency event were to happen the non-reactionary EWS message regarding it would then take priority at AC A3. For example, if a vehicle were to be involved in a collision and sent out a EWS message it would be classed A3. The vehicles behind the collision braking would send out EWS messages with class A2. If one of the braking vehicles were to then be involved in a second collision it could broadcast out a EWS message at class A3 and hopefully have lower latency than the emergency braking messages. All EWS messages will still take priority over routine traffic while allowing prioritization of more important EWS messages.

Since these messages are so important successful transmission should be strived for. Rebroadcasting EWS messages a second time could again help with this however we want to avoid network congestion. One possible scheme is for only the first vehicle to rebroadcast a second time and all others only broadcast once. Once warning are received of multiple vehicles slowing down ahead a driver should already have been warned to slow down and successful reception of all EWS message is no longer crucial. However, only a single transmission will be examined in the simulations.

3.5.3 Hazard Warning System

The concept of a HWS is very similar to the EWS except the situations may not require the same level of immediacy. Examples of an HWS message could include lane closures warnings, disabled vehicle warnings or minor loss of traction warnings. Potentially if the situation were to worsen, for example a vehicle first losing traction then completely losing control, first a HWS message would be sent and then a EWS message would be sent after. HWS messages would be sent out with AC A1. If network latency were to not allow the HWS message to be sent out and an EWS message was then generated it would be sent out first due to the MAC prioritization scheme informing vehicles of the more important situation first. The packet structure would be nearly identical to the packet structure described for EWS packets just with different information detailing the nature of the problem.

If the problem that caused the HWS message persists the message will be resent periodically. A simple scheme is sending the message out every 1s instead of sending any CCWS messages. This will be used in the simulations. There are a number of potential improvements possible regarding how often to send out these messages and for vehicles to forward HWS messages using multi-hop communication however they are outside of the scope of this thesis.

Chapter 4

Simulation

Simulating vehicular traffic, wireless network communication or computer vision on its own is no trivial task. Simulating all three alongside one another only compounds this problem. The basic theory behind traffic, network and vision simulation will be described throughout the next chapter. To make the task more manageable existing open source projects were used where ever possible. The choice of projects used and any modifications that were required for them will be explained. Finally, an overview of how the various components all fit together will be provided.

4.1 Traffic Simulator

Creating a realistic mobility model for the simulation of VANETs and VBDA is important. Vehicular traffic typically moves in relatively predictable ways along a set path. These movements are governed by how the road network is laid out. The placement of lanes and traffic control features, such as traffic lights, turning lanes or traffic signs, combined with both a source, destination and other vehicles decides how a vehicle will move in the real world. To get accurate results for how VANET and VBDA technologies will work it is important to model these movements with a high degree of accuracy.

Initial work on MANETs often used random node movements. In essence nodes would choose random directions to move in and periodically change direction. This practice was initially carried over into VANET research. Of course it is nothing like vehicle traffic in the real world. Studies have shown that random node movements are a poor substitute for a mobility model and should not be used [32].

A second approach that was taken for a mobility model was the use of real world mobility traces. Obtained by tracking the location of real world vehicles using GPS or other technologies they mimic the real world exactly. Nodes within the simulation are then moved according to these traces exactly. While they do an excellent job of simulating mobility as it occurs in the real world they are of limited flexibility. Changing parameters, such as traffic density, isn't feasible for large scale simulation.

A better approach is the use of a dedicated traffic simulator. There are a wealth of traffic simulators available and there are three fundamental approaches to their operation. Macroscopic traffic simulators model large scale vehicle movements much like the flow of liquid. Mesoscopic traffic simulators model the movements of clusters of vehicles as this is how vehicles tend to move in the real world. Finally, microscopic traffic simulators model the behaviour of each individual vehicle. Potentially they could even model the various subsystems of each vehicle in detailed granularity. As VANET simulations require exact node locations to model wireless communication only a microscopic traffic simulator will suffice [15].

Microscopic traffic simulators are based on a microsimulation model of vehicle behaviour. The model must realisticly mimic how a human driver would react to the world surrounding them. There are a wide variety of published models, as discussed by C. Sommer et al. including the car following model by Stefan Krau [15] [33]. These various models were compared and were found to all perform equally well in relation to network simulation [34].

Simulation of Urban Mobility (SUMO) is an open source microscopic traffic simulator that has been used in a wide variety of VANET projects [15] [35]. The road network, vehicle types and vehicle routes are all highly configurable and allow for customized simulations. Furthermore, Traffic Control Interface (TraCI) allows SUMO to communicate bi-directionally with any network simulator implementing TraCI. This allows the results of network simulation to influence the traffic simulation and vice versa. By default, SUMO uses the Stefan Krau car following model to realistically model the acceleration and deceleration of each vehicle [33].

SUMO road networks are defined by a network file. In the network file lanes are defined as edges in a directed graph with vertices taking the form of connections between lanes. Individual lanes have attributes such as speed limits or turning restrictions. Connections between lanes can simply indicate a change in direction or can be complex multilane intersections with traffic lights or priority traffic direction.

While quite complex there is a suit of included tools for generating SUMO road networks. Simple geometric road networks can be generated using the NETGEN utility. To model real life road networks map data from a variety of sources can be imported using the NETCONVERT utility. One such source is the Open Street Map (OSM) project. It provides a Google Maps like interface to viewing community generated map data. It is also possible to download the underlying map data to convert using NETCONVERT. These tools help to provide a way to generate realistic road networks.

Vehicle traffic is defined by a route file. Again there is a suite of tools to generate routes. The simplest approach is random routes. By picking a random start and end point on the road network and finding the shortest route between the two points a random route is generated. The amount of traffic can be controlled by generating more or less routes in a given time period. Unfortunately these routes are not entirely realistic. Further enhancements such as prioritizing roads by number of lanes, where the larger arterial roads get more traffic than smaller side streets, are possible to increase the realism of random routes.

SUMO also provides tools for demand modeling. Municipalities often measure traffic flow over specific roads. By providing these measurements for sections of road, routes can be generated to mimic the same level of traffic flow over the measured locations. By using real road networks and generating routes based on measured data using the DFROUTER utility we can create realistic traffic scenarios with maximum flexibility.

Individual vehicle parameters are again entirely configurable. Vehicle size, performance and driver reaction time can be tuned to create a realistic model of a specific vehicle. A weighted distribution of various vehicles can be used to populate the simulation at runtime in order to generate a realistic population of cars, trucks and buses can be simulated.

Overall, a highly realistic mobility model is possible using SUMO. TraCI will discussed further in subsequent sections but it allows SUMO to be connected to other simulators without any modifications to the code. As such it is an excellent choice for traffic simulator. Only the road network and route files need to be generated for each simulation configuration.

4.2 Network Simulator

Wireless network simulation is a complicated and resource intensive task. The entire network stack, as discussed in Chapter 3, must be simulated along with node mobility according to the mobility model generated by SUMO in order to gather accurate results. There are a wide variety of network simulators that have been used by researchers to simulate VANETs including NS2 [35], OMNeT++ [15], OPNET [17] [6] and NCTUns [36]. All of these simulators are suitable for running network simulations however OMNeT++ was chosen for this thesis because of its integration with SUMO.

4.2.1 Description

OMNeT++ is an open source discrete event simulator. It provides the core modules required for discrete event simulation and statistic collection along with a powerful Integrated Development Environment (IDE) based on Eclipse. It also allows simulation with arbitrary time precision. By default it uses picosecond precision which is more than sufficient for wireless communication simulation. This provides enough granularity to accurately simulate wireless communication parameters on the microsecond level and still allows for simulations reaching over 100 days in length [37]. However, it is not a network simulator in its own right. Instead there are a number of projects that provide network simulation frameworks for OMNeT++, for example INET and MiXiM.

For our VANET simulations we have chosen the MiXiM framework as it has strong support for physical, MAC and mobility layers [38]. Furthermore, it already has support for simulating OFDM type communications on the physical level as required for 802.11p simulations. The network layer and above are left relatively unimplemented however given the simple nature of the WSMP and the custom application layers being implemented this isn't a major drawback.

MiXiM is roughly organized according to the OSI network layers. It takes a modular approach implementing a module for the physical, MAC, network and application layers. The physical and MAC layers are further grouped into a compound Network Interface Card (NIC) module. Communication takes place along predefined gates at each level of the network stack. Each module has gates for both data and control information going both up and down. There is also a mobility module unrelated to the communication stack that is responsible for updating the position of each node that interfaces with SUMO.

The physical layer models radio communication. It keeps track of the radio state and allows the node to transmit packets. It has two main components that it must model. First, it models attenuation effects on the transmitted signal. Any number of attenuation effects can be modeled to account for path loss, shadowing and fading. The attenuation effects take into account the transmitted power, distance between nodes, radio frequency and transmission duration along with any parameters they are set up with. Second, the physical layer calculates the SINR, determines the probability of bit errors and decides whether the packet was successfully received. If the packet is successfully received it is passed up to the MAC layer.

The MAC layer mediates between the physical layer, allows for channel sensing and manages channel access. Any incoming packets from the physical layer are decapsulated and passed to the network layer if they are intended for the node. It also handles packets passed down from the network layer. Incoming packets are encapsulated with a MAC header and added to the queue for transmission based on the specific MAC protocol. It also handles channel sensing in order to facilitate MAC protocols that require it.

Finally, the network and application layers are left undefined by MiXiM. In general, the network layer encapsulates and decapsulates messages between the application and MAC layer. The application layer generates packets as it requires and handles incoming packets addressed to it.

4.2.2 Implementation

To implement the physical, MAC, network and application layers described in Chapter 3 custom OMNeT++/MiXiM modules were required. This section will describe the various components that were created to implement VANET communication. Modules at the various networking layers, helper classes and message classes were all required. All modules and classes mentioned were developed from scratch specifically unless otherwise noted.

The application layer is implemented as the WAVEApplLayer module. It interacts with helper CCWS, EWS and HWS classes to implement the specific behaviours described in Section 3.5. The WAVEApplLayer provides the node with location, speed and direction data modeling the GPS. It passes outgoing packets down to the WSMNetwLayer and passes incoming packets up to the appropriate application.

The CCWSApplLayer implements a CCWS as described earlier. It receives and processes incoming location updates and sends outgoing location estimates when required. It also can receive input from vision simulation with additional vehicle position information. It tracks all vehicles it has information on and discards old information once it is deemed no longer relevant. There are also EWSApplLayer and HWSApplLayer classes, however they are not as complete as the CCWSApplyLayer. They only have the ability to send EWS or HWS messages at a specified time and record when they handle incoming messages.

A PositionEstimator helper class was created to implement the SPE, RPE and NVEs required. The SPE provides the most up to date estimate of vehicle position based on input from SUMO. Since location updates only occur every 100ms this enables a higher level of accuracy in positioning by calculating the intermediate locations based on speed, heading and acceleration. The SPE tracks the vehicle's actual location before GPS error is added on. Whenever vehicle location information is used, GPS error is first added onto it using the GPS error model. The RPE provides the estimated position of the vehicle based on its last position update. It is updated every time a location update is sent out and the difference between the SPE and RPE model is what is used for the error threshold broadcast scheme. The NVEs provide an estimate of NV locations based on received location updates.

The WSMNetwLayer module is a simple implementation of the WSMP. It encapsulates packets from above, sets the relevant header fields for channel, bit rate, transmission power and SPI and sends the packet down to the Mac80211pLayer. It also decapsulates packets from below and passes the application layer up along with the channel, bit rate, transmission power and SPI from the WSM header.

The Mac80211p module implements a basic IEEE 802.11p protocol based on the parameters described in Section 3.3. It is based heavily on the existing MiXiM Mac80211 module. Timing parameters were modified accordingly and only broadcast transmission is supported. It handles the contention protocol and sends out broadcast with the bit rate and transmission power specified by the WSM header.

The Decider80211p module implements a PER model, as described in Section 3.1, to decide if a packet is properly received. The BER model itself is described for 802.11g. 802.11g and 802.11p differ only in frequency and channel bandwidth [27]. Furthermore, it is already used for 802.11a in the INETMANET OMNeT++ package as well [39]. The code from the INETMANET 802.11a radio model to calculate BER replaces the existing 802.11b BER model in the Decider80211 module. The frequency and channel bandwidth used by the model are modified so it produces accurate results for 802.11p. Using the BER the PER is calculated stochastically.

Mobility is controlled by the TraCIMobility module from the Vehicles in Network Simulation

(VEINS) project with small modifications to allow for vision simulation as well. It interfaces with SUMO over a TCP socket for traffic simulation. Finally, a number of other included MiXiM utilities are used for convenience without a modification.

4.3 Vision Simulator

Computer vision algorithms involve converting a series of images into a three dimensional mathematical model of the world. Creating an accurate model from images is a difficult and computationally expensive task. However, when running VANET and traffic simulations we already have access to a complete mathematical model of the world without images available. As such we are presented with the opposite problem. We must determine what images would be visible based on the mathematical model and therefore what vehicles would theoretically be visible. We have access to a list of all vehicle positions, trajectories and sizes as well as the position and size of any other objects we wish to specify, such as buildings. Calculating what is visible must take into account which pair of cameras would have the object in their view, if the object is close enough to identify and if the object is occluded by any other objects. This section will explain the algorithm and modifications to the simulation environment used to identify which vehicles would be visible.

4.3.1 Description

The vision simulator is based on a fairly simple algorithm. Since the algorithm is run for each vehicle in the simulation multiple times per second it must be efficient. In theory we need to compare each vehicle to every other vehicle in the simulation so the algorithm could potentially be $O(n^2)$. In order to reduce the computation complexity the algorithm should be made as efficient as possible. In Table 4.1 we have a list of angles covered by each camera system and the associated maximum distance over which they can accurately detect and position objects.

Zone	Angles	Distance
A	-15 to +15	150m
B_1	+15 to +45	20m
B_2	+315 to +345	20m
C	+45 to +135	15m
D	+135 to +225	15m
E	+225 to +315	15m

Table 4.1: Zone angle and distance table

Note that the maximum distance a vehicle is visible by any camera system is 150m. The algorithm is described for a single vehicle as the SV and the n other NVs in the simulation.

When vehicle positions are updated each vehicle would in turn be considered as the SV and have the algorithm applied to it. All vehicles are modeled as rectangles. To simplify calculations 8 points are defined on the rectangle, the four corners and the center of the four line segments. The closest of the 8 points is what overall distance to the vehicle is measured as.

1. For the SV, known as S, and each NV, known as N_i, construct the set $D = \{\{S, N_0\}, \{S, N_1\}, ..., \{S, N_{n-1}\}\}$.

2. For each pair $\{S, N_i\}$ in D calculate the distance from the center point of S to the center point of N_i. Let t_i be defined as the maximum range of any vision system, in this case 150m, plus the distance from the center point to a corner of N_i. If $|\overrightarrow{SN_i}| < t_i$ then N_i is considered potentially visible. Otherwise the pair is removed from D. To make this step more efficient the world is divided into grid segments 200m by 200m in size. The vehicles that are in each grid segment are known. This way distance comparisons only need to be performed between vehicles within the same or adjacent grid segments. Any vehicles that are separated by one or more grid segments are known to be out of range. Also, by using grid segments slightly larger than the maximum vision distance we can ensure large objects that cover more than one grid segment are considered.

3. For each pair $\{S, N_i\}$ remaining in D the angle between the direction vector of S and the vector $\overrightarrow{SN_i}$ is compared to find the angle between the two vehicles relative to the direction of travel of S. Using this angle the camera system responsible is chosen based on Table 4.1. The angles are illustrated in Figure 2.3. If the distance between S and the closest point of N_i is less than maximum distance for the camera system then N_i is added to a list L sorted based on distance.

4. Iterating through the list L starting at the closest object the pair of minimum and maximum angles between S, based on its direction of travel, and the two outermost corners of N_i, based on its direction of travel, length and width, are calculated. If more than 50% of the range of angles has already been covered the vehicle is considered not visible, as more than 50% of the vehicle would be occluded. Otherwise the vehicle is considered visible. The range of angles covered by N_i is then added to the list of angles covered. Figure 4.1 illustrates this step.

5. For the set of vehicles considered visible to S, or D_v, the distance $|\overrightarrow{SN_i}|$, trajectory, minimum and maximum angles to each vehicle N_i is returned for use in the application layer.

This approach is of course somewhat simplified. It does not take into account the z axis at all. In real life it is sometimes possible to see over other vehicles. However, given that most roads are relatively flat being able to identify vehicles because of changes in elevation would be relatively rare. Due to SUMO only operating in 2D this is unfortunately not possible regardless. Furthermore, computing trajectory information would normally require at least two intervals in which vision detection normally occurs. However, since the vision algorithm updates less frequently, every 100ms versus at least every 33ms, this is taken into account by

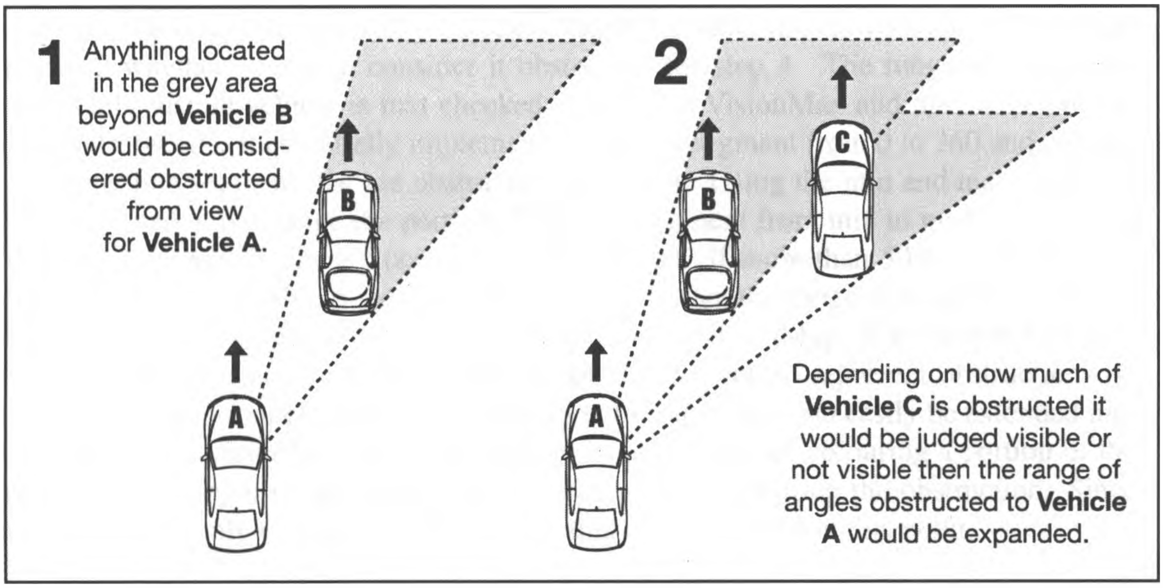

Figure 4.1: Vision obstructions by other vehicles explained

the longer update intervals. Given the complexity of VBDA a conservative estimate of the vehicles visible is a safe approach regardless.

4.3.2 Implementation

The vision simulation is implemented as part of the network simulation environment as it has both access to all of the location information needed and tools to simplify the process. Since there has been little research in the integration of vision and network simulation there were no existing components which served this specific purpose. Using existing network simulation tools and custom components a new OMNeT++ module was created.

The VisionManager module is tasked with keeping track of what vehicles are visible to other vehicles. It is based heavily on the ConnectionManager module. The ConnectionManager already implements a grid system and will find all entities that are within a maximum distance of one another. The ConnectionManager is typically used to determine what nodes are within wireless communication distance of one another. Each vehicle is registered with the VisionManager upon creation. The updated vehicle position is sent to the VisionManager every time it is updated by SUMO. With some modification the VisionManager implements step 1 and 2 of the vision algorithm finding all vehicles within 200m. Step 3 is implemented by a simple check of the angle and distance between the two vehicles within VisionManager. Step 4 is implemented with two new classes VisionEntity and VisionMap.

The VisionEntity class represents each vehicle and keeps track of the important information such as vehicle positions, directions, size and the list of potentially visible vehicles from step

3. The VisionMap class implements the check of angles to determine how much of the vehicle is visible and whether or not to consider it obstructed for step 4. The range of angles for each potentially visible vehicle is first checked against the VisionMap and then added to the VisionMap afterwards. It essentially implements a 1D line segment from 0 to 360 and colours portions of the line segment that are obstructed from view. Using the min and max angle for each NV from step 4 as a range the portion of the line segment from min to max is examined and the percentage that is already coloured is determined. If more than 50% of the range is coloured we decide the vehicle would not be visible. After the range is checked then it is coloured on the line segment to add the obstruction to the VisionMap. If environmental data is available these items can be added to the VisionMap as well to represent obstructions by buildings or other non-vehicle objects. Furthermore, this approach can easily be extended into 3D by modeling the VisionMap as a 2D plane segment. Instead of colouring a portion of the line segment, a rectangle in the plane would be coloured representing the obstruction. Since the traffic simulator only operates in 2D however this is not required at this point.

The list of visible NVs with their distance, trajectory and minimum and maximum angles are then available to the application layer of the SV taking care of step 5. The list of visible vehicles is updated every time vehicle positions are updated by SUMO. This interval can be chosen to be as large or as small as desired. For our purposes an interval of 100ms is used.

4.4 Simulator Integration

The individual pieces of each simulator are fairly simple but the combination of pieces quickly becomes quite complicated. Both SUMO and OMNeT++ were chosen as they were supported by the VEINS project, put together by researchers as the University of Erlangen in Germany. TraCI is the outward facing interface for SUMO to communicate with another application via TCP. The VEINS project implemented a number of OMNeT++ modules inside the MiXiM environment to interface with TraCI and control node mobility. As such, it is an excellent starting point for VANET simulation. The application layer as described previously is already aware of vehicle location and makes an excellent interface point for the vision simulator. The VisionManager interfaces with the application layer to provide the list and information on visible vehicles.

Figure 4.2 shows how all of the pieces fit together graphically. Arrows represent the flow of information between various modules. The directionality of the arrows represents either one or two flows of information. As indicated there are quite a few interconnected parts.

As a discrete event simulator OMNeT++ coordinates all of the timing. Events are scheduled for a specific time and all events are executed in chronological order. OMNeT++ has existing timer utilities available to implement all of the network layer components as required. SUMO is a step based traffic simulator. A discrete step of time is executed and results are provided at the end of each step. As previously mentioned we simulate 100ms in each SUMO time step. In order to coordinate between OMNeT++ and SUMO the OMNeT++ TraciMobili-

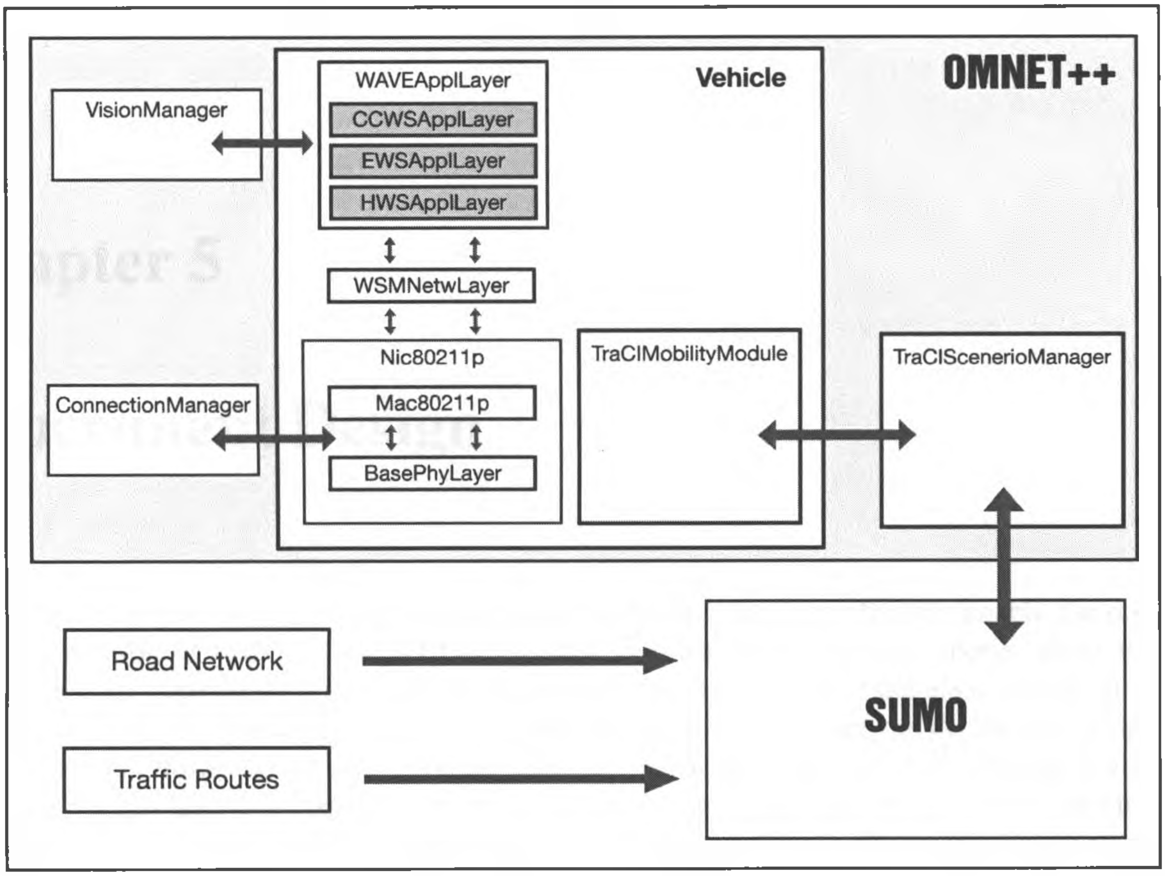

Figure 4.2: Simulator Component Interaction

tyManager module, provided by the VEINS project, schedules an event every 100ms. Between these events simulation occurs inside of OMNeT++. It is then paused and SUMO then runs through one step of traffic simulation and updates nodes positions over TraCI. Control is then switched back to OMNeT++ and this cycle carries on until the simulation has ended.

OMNeT++ also handles a number of other important tasks. It has utilities for generating random numbers and random statistical distributions for use in the stochastic modeling of wireless communication. It also has strong statistical collection and analysis tools that will be discussed in Section 5.3. The last step in the simulation environment is compiling the final simulator environment. OMNeT++ handles the compilation of our VANET model. The model is compiled, referenced with MiXiM and OMNeT++ libraries and a stand alone executable is created.

Chapter 5

Experiment Design

The technologies, protocols and simulators described in the previous chapters enable the realistic simulation of VANETs and VBDA together. The end goal of these experiments is to determine what parameters provide the most accurate information about the environment surrounding a vehicle. Generally the more information we have the better the decision making process can be. However, old and incorrect information can often be worse than having none at all since it can lead to incorrect assumptions. This is especially relevant in a vehicular environment when vehicle positions and trajectories can change drastically in the course of a second.

By simulating in detail both VANETs and VBDA side by side we can measure just how effective each is at monitoring the surrounding environment. We have access to both the actual vehicle locations as well as all location estimates from VANET or VBDA. This allows us to compare how accurate the estimates are to actual vehicle locations. Three stages of experiments will be conducted to study three different areas, first the effective implementation of VANETs and VBDA, second the effect of adoption rate on the effectiveness of these technologies and finally a look at EWS and HWS technologies. The following chapter will describe the simulations that will be run in detail. How they are setup, what parameters are being used and what is being measured will be included. The end goal is to quantitatively measure how effective these technologies can be.

5.1 Road Networks

One of the most important pieces of each simulation is the road network. The road network describes all of the roads and traffic features that vehicles must follow while in the simulation. Each vehicle is given a route to follow along the road networks. The traffic simulator moves each vehicle along its route and this provides the node locations for VANET and VBDA simulation. Therefore, realistic road networks are key to realistic overall simulations. The road network itself is defined in a *.net.xml file. Individual lanes, connections between them and traffic control features are all defined in this file. Vehicle routes are defined in *.rou.xml files. Vehicle types, routes and departure times are defined in this file. Vehicle types and routes can either be explicitly defined for each vehicle that enters the simulation or they can be picked from a set distribution at runtime. In Appendix D the basic layout of both these file types is included.

Three different road networks are used in the following simulations. The first is a simple manhattan grid type road network, with roads running both horizontally and vertically in a evenly spaced grid pattern, approximately 4 square kilometers in size. The second is a city road network based on downtown London, Ontario that is again approximately 4 square kilometers in size. Finally, the third is a road network of highway approximately 6 kilometers in length based on Highway 401. Each of these road networks has two route files defined for it with one representing light traffic and one representing heavy traffic. For all three road networks the heavy traffic route file has approximately double the number of vehicles present during the simulation.

5.1.1 Manhattan Grid

The manhattan grid road network is the most basic road network of the three. It is not based on any real set of roads however it does resemble a New York City like urban center. The road network consists of four lane arterial streets and two lane residential streets. The arterial streets are located every 600m. In between each set of arterial streets are two residential streets spaced 200m apart. There are a total of four arterial streets running both horizontally and vertically with two secondary streets in between each set. Figure 5.1 shows a map of the road network. Short streets on the outside of the map area are included to manage traffic entering and exiting the road network. Intersections between two arterial streets are controlled by traffic lights and all other intersections are priority controlled. In the case of a arterial and secondary street meeting, the arterial street has priority over the residential. In the case of two residential streets meeting they have equal priority. The speed limit on all streets in the road network is 50km/h in keeping with a typical urban center.

Traffic for this road network is generated randomly as it is only theoretical. Random start and end points inside of the map are chosen using a python script. These start and end points are then fed to the DUAROUTER utility which chooses the best path. Arterial roads are weighted as a more likely choice for both choosing start and end points as well as in choosing the best path. Typically traffic is much heavier on arterial streets inside of urban areas as they are better suited for traveling long distances.

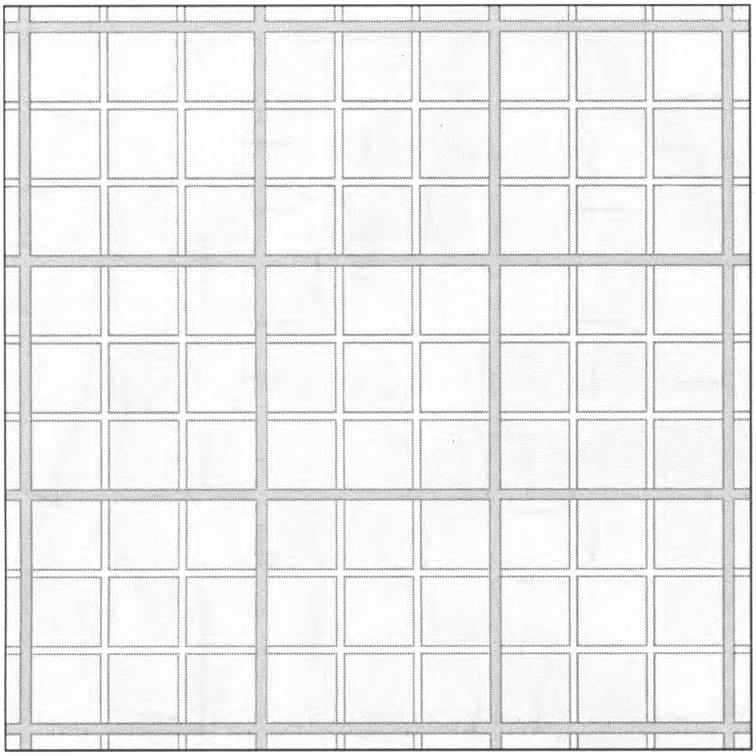

Figure 5.1: Manhattan grid road network

5.1.2 City

The city road network is based on downtown London, Ontario. The area is roughly bordered at the north by Oxford Street, at the south by Horton Street, at the west by Richmond Street and at the east by Adelaide Street. A small number of roads outside of this boundary are included to help manage traffic entering and exiting the downtown area. Figure 5.2 shows the area in question. Map data from the OSM project of the area was used to generate this map. The raw map data was refined with the Java Open Street Map Editor (JOSM) package to be as true to life as possible. Using Google Street Maps traffic light locations were mapped out. All other intersections are priority controlled based on the number of lanes. The speed limit is again 50km/h for all streets in the road network as this is the case in the real world. Using the NETCONVERT package the OSM file was converted to a road network file.

Traffic routes for this network were generated based on traffic survey data available from the City of London [40]. The survey provides the average number of vehicles per day that cross a segment of road. These numbers were input for certain road segments. On Richmond Street, Adelaide Street, Oxford Street, York Street and Horton Street traffic information was input for every other block. For other smaller streets traffic information was input for one or two segments. Using the average number of vehicles per day that cross each segment a custom Python script was used to create a set of routes that provides approximately the same

Figure 5.2: City road network based on London, Ontario

distribution of traffic. The script generates 10 000 random routes, rates the routes based on how well they maintain the desired distribution, then picks the best route removing it from the list of random routes. This process is repeated, where each time all of the remaining routes are ranked again, and the new best route is selected.

There were multiple reasons for choosing to base a city road network on London, Ontario. First, it is familiar so any major inconsistencies with the real world could easily be recognized and corrected. This would not be feasible if we were to use map data from another city. Second, the downtown core of London sees both heavy traffic during peak hours and almost no traffic during light hours so simulation of both heavy and light traffic on it will produce realistic scenarios. Finally, the availability of traffic statistics makes it easy to produce realistic traffic routes. This makes it an ideal candidate for a simulated road network.

5.1.3 Highway

The final road network is a simulated highway based on Highway 401. An approximately 6 kilometer section of the highway in Milton, Ontario was used. The length of highway is roughly bounded by the sets of interchanges at James Snow Parkway and Highway 25. Each interchange has one off-ramp and two on-ramps. Again, map data from the OSM project was used as the basis. The highway in question is three lanes wide in each direction with a speed limit of 100km/h. Figure 5.3 shows a map of the area. Only the highway as marked in blue, minus the interchanges, is included in the road network. The road network was created in the same fashion as the city road network using the JOSM and NETCONVERT tools to generate the network file.

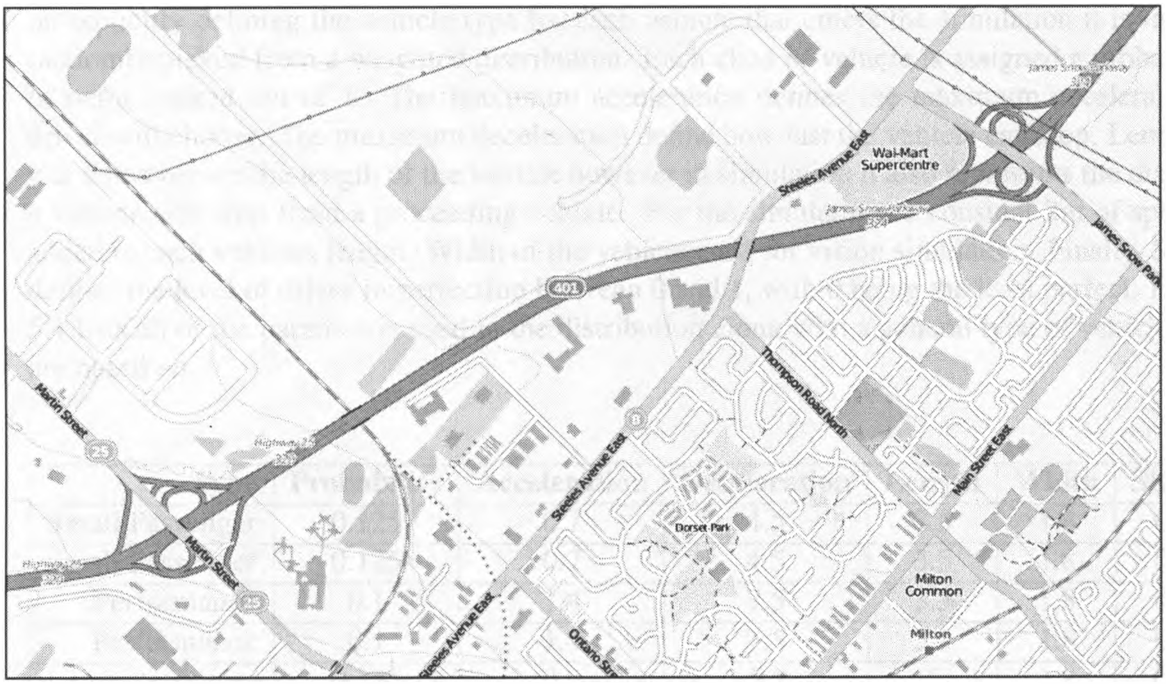

Figure 5.3: Highway network based on Highway 401 near Milton, Ontario

Traffic routes for this road network are very simple as most vehicles simply drive down the length of the highway going either east or west. To produce traffic provided by the on-ramps and off-ramps, some vehicles enter and exit the road network at these locations. The ramps themselves were not modeled due to a bug SUMO. At the off-ramps 8%, or 2% per ramp, of traffic is assigned to exit the network having started at the beginning of the highway. At the on-ramps 8%, or 1% per ramp, of the traffic enters the network and proceeds to the end of highway. No vehicles enter at an on-ramp and exit at an off-ramp for simplicity. All other traffic starts at one end and drives to the far end of the highway. All vehicles enter and exit the network attempting to go 100km/h. To model high and low traffic volume more or less vehicles are simply added along the same routes.

This area was again chosen for a number of reasons. It is a familiar highway so we can ensure no major errors occur in its layout. The interchanges are well spaced out and the highway has some light curves to provide realistic vehicle paths. The section of highway also frequently sees high traffic volume during rush hour while being relatively empty at night so it does see both high and low traffic volumes in the real world. Again this makes it an excellent candidate.

5.1.4 Vehicle Types

The vehicle type decides certain characteristics about how the vehicle will perform. Instead of explicitly defining the vehicle type for each vehicle that enters the simulation it is instead randomly picked from a weighted distribution. Each class of vehicle is assigned a probability of being picked out of 1. The maximum acceleration defines the maximum acceleration a driver will choose. The maximum deceleration define how fast the vehicle can stop. Length as it is listed here is the length of the vehicle however in simulation it also represents the distance a vehicle will stop from a proceeding vehicle. For the simulations a constant 2m of space is added to each vehicles length. Width of the vehicle used for vision simulation. Finally, Sigma defines the level of driver imperfection between 0 and 1, with 0 being the least perfect. Figure 5.1 lists all of the parameters used in the distribution along with a general type of vehicle they are based on.

Type	Probability	Acceleration	Deceleration	Length	Width	Sigma
Small Passenger	0.125	0.7	4.5	3.5	1.8	0.8
Small Passenger	0.125	0.7	4.5	3.5	1.8	0.4
Performance	0.1	1.4	7.5	3.5	1.8	0.7
Performance	0.1	1.4	7.5	3.5	1.8	0.3
Large Passenger	0.125	0.9	5.5	5	1.8	0.8
Large Passenger	0.125	0.9	5.5	5	1.8	0.4
Van/Truck	0.1	0.7	4.5	5	2	0.8
Van/Truck	0.1	0.7	4.5	5	2	0.4
Transport Truck	0.1	0.6	4	19	2.5	0.8
Bus	0.1	0.6	4	14	2.5	0.8

Table 5.1: Vehicle Type Distribution

These numbers are based on an average vehicle of the same type that has specifications listed for it. None of the values are extreme and should produce a realistic set of vehicles for simulation. The only difference in vehicle types between the different road networks is that the manhattan grid and city network include a city bus where the highway includes a transport truck, with these vehicles varying only in length.

5.2 Simulations

The goal of the experiments is to compare both VANET and VBDA technologies. During simulation statistics are collected and analyzed afterwards to produce results. The simulation environment provides a number of utilities for the collection of scalar, vector and histogram results. The output is written to text files in an efficient fashion by OMNeT++. Analysis is done in part with provided utilities in OMNeT++ and with custom Python scripts where the provided utilities do not provide enough flexibility.

The experiments will take place in three distinct stages. Each stage will test a number of related parameters and examine how they effect the results. The world model, essential all information about every vehicle, will also be available for comparison. In general, each set of parameters will be run on each road network under both light and heavy traffic. The simulation will be repeated twice with different seeds for the random number generators. Since vehicle types and exact routes are chosen at runtime, performing the simulation two times with different seeds will provide slightly different traffic and vehicle patterns. It will also provide different seeds to statistical distributions used for the simulation of wireless communication. In total 12 executions of the simulation, 4 on each network, will be run for each change in parameters.

The length of time simulated and recorded for each scenario will be 120 seconds unless otherwise noted. This provides a wealth of statistics while keeping runtime low. There will be a lead-in time of 240 seconds for the manhattan grid and city network and 260 seconds for the highway network. During this lead-in time there will be no VBDA or VANET simulation and no results will be recorded to allow the network to fully populate with vehicles in a realistic fashion. This is required to maintain realistic traffic patterns given the limitations of SUMO. Through experimentation the runtime of 120 seconds was selected as it provides a wealth of data to analyze while keeping the runtime and results generated to a acceptable level. The number of statistics recorded can reach hundreds of millions per simulation minute and the runtime can exceed one hour per simulation minute. The number of vehicles in the simulation for the 120 seconds during which statistics are recorded is listed in Figure 5.4.

All parameters other than those mentioned will be fixed. Relevant parameters are listed in Appendix C.

5.2.1 Stage One

The goal of stage one is to ensure that VANET communication is working and effective and to then determine how VANET and VBDA compare to one another. The parameters and protocols used for VANET simulation need to be effective in order to draw useful results from it. As discussed in previous chapters the VANET simulation is based on research and standards that have been tested. Ensuring that our implementation of them works effectively first is important however. A summary of all simulations performed in stage one is available in Table 5.2.

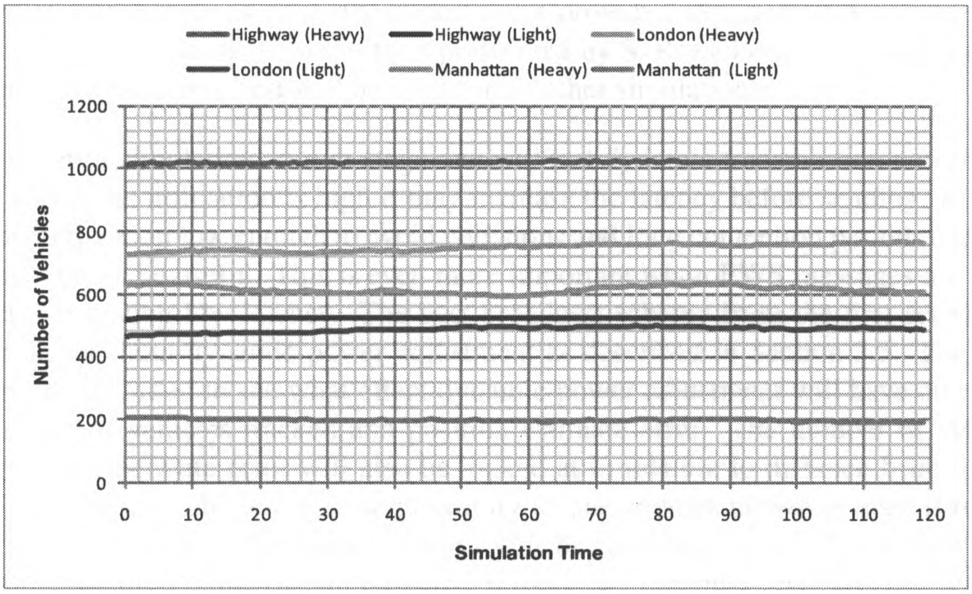

Figure 5.4: Number of vehicles in simulation

Simulation	Parameters	Runs	Description
A	4	48	Testing broadcast update intervals
B	2	24	Testing dynamic power updates
C	4	12	Testing time before discarding CCWS data
D	1	12	Testing CCWS with both VANET and VBDA
		96	**Total**

Table 5.2: Stage One Experiments

The first set of simulation of stage one models a simple CCWS to find the best broadcast scheme. As discussed in Section 3.5.1 four possible broadcast schemes are proposed. The first two schemes are a fixed broadcast interval of either 500ms or 100ms. The last two schemes are variable broadcast based on an error threshold of 0.75m, one without rebroadcasting and one rebroadcasting once within a 50ms interval. The set of simulations will be run using these four schemes without simulating VBDA. The time between transmissions, packet loss rate and transmission latency will be recorded for all vehicles. This provides information on how well the VANET performs and gives some indication of for how well a CCWS system will perform.

However, this does not provide a complete picture for the performance of the CCWS. The number of vehicles tracked and accuracy of NVE position estimates provides a better metric of CCWS performance. To accomplish this, every 100ms the number of vehicles tracked and the distance error between the estimated and actual vehicle position will be recorded. The more vehicles we can successfully track, the better information we have available. The distance error provides us with a measure of accuracy to ensure we do not compromise quality of information.

While breadth of information is certainly important, it provides little help if it is inaccurate. These results will be compared to similar tests performed by S. Rezaei et al. to ensure validity [6]. The scheme that performs best will be used for all other simulations.

The second set of simulations run in stage one will test dynamic transmission power level adjustments. If channel utilization is high it may increase the latency before sending. In order to keep channel utilization low enough to avoid collisions and long contention periods dynamic power adjustment may be useful. This is even more important when EWS messages are introduced that rely on having low latency. The best broadcast scheme from the first set will be run both with and without dynamic power adjustment as described in Section 3.5. The same parameters will be recorded to see what effect dynamic power adjustment will have on packet loss rates, number of vehicles tracked and position estimate error. The number of vehicles tracked and position estimate error will also be looked at in relation to distance from the SV to see if performance for vehicles in close proximity to one another improves even if overall system performance does not. Separate statistics will be recorded for vehicles within 20m, 50m and 100m in addition to the overall measure. If dynamic transmission power adjustments can reduce latency noticeably it will be used.

Vehicles will be constantly entering and exiting transmission range for VANET communication so eventually position estimates must be discarded if no further updates are received. The amount of time an old record is kept before discarding the record is the NVE timeout period. A set of simulations will be run that examines the number of vehicles tracked and position error if data is discarded with a timeout period of 1, 2, 3 and 5 seconds. Position estimates will not be discarded until after 5 seconds however separate statistics will be recorded as if each time period was in place. This allows the comparison to be done with identical simulation results. The timeout period that produces the best tradeoff between breadth and accuracy will be used.

The final set of simulations in stage one will examine a CCWS that uses both VBDA and VANET technology. Using the best VANET scheme and using the vision simulator, as described in Section 4.3, the simulation will be run. As with previous sets every 100ms a number of statistics will be recorded. The number of vehicles tracked by VANET and VBDA will be recorded. Statistics on error in positioning will be recorded as well. Finally, a unified model will be created where vehicles visible and matched correctly will be assumed to have zero position error. The position error of VANET estimates alone and the positioning error of the unified model will eventually be recorded. This allows us to compare the accuracy and range of both technologies again with identical simulation results for the most accurate contrast.

Overall, the goal of stage one is to show that both VBDA and VANET technologies can be used to monitor the environment around the vehicle. In theory, VBDA should do a much better job of tracking nearby vehicles accurately while VANET communication will do a much better job of tracking vehicles overall. By measuring how well the model provided by VANET communication and VBDA can be combined into a unified model we can demonstrate the strength of both technologies used together. In an ideal situation without any error creating a unified model would be simple. Unfortunately error will exist and a further decision making process will be required. With access to the world model in addition to estimates the accuracy of our estimates, from both sources and our unified model, can be validated quantitatively.

5.2.2 Stage Two

After establishing the effectiveness of both technologies in stage one we will then examine how varying adoption rates affect VANETs and VBDA in stage two. There will be a long period of time before consumer adoption of driver assistance technologies reaches anywhere near 100%. Up to that point they still need to provide a benefit. A summary of all simulations performed in stage two is available in Table 5.3.

Simulation	**Parameters**	**Runs**	**Description**
A	5	60	Testing impact of adoption rate
B	5	60	Testing impact of adoption rate on ECCWS protocol
		120	**Total**

Table 5.3: Stage Two Experiments

The first set of simulations will use the best scheme from stage one for VANET communication combined with VBDA to produce a unified model. It is assumed that if a vehicle is equipped with one technology it will be equipped with the other due to the similar requirements in instrumentation and computing power. The set of simulations will be run with vehicles randomly being either equipped or not equipped at 10%, 25%, 50%, 75% and 90% adoption rates. Again, the number of vehicles successfully tracked and average position error will be recorded. This will give us a baseline data set for how effective these technologies will be under various adoption rates.

A second set of simulations will be run with a slightly modified CCWS protocol. Vehicles equipped with driver assistance technologies will append extra position information for up to four other vehicles to its own CCWS position updates. Only position estimates for vehicles detected by VBDA and not broadcasting CCWS packets will be appended. Up to the four closest neighbouring vehicles that satisfy this criteria will be included in the CCWS message and the message size will increase by 40 bytes for each extra vehicle to accommodate the extra information. Since vision is very accurate and no error model exists for the proposed VBDA system the actual location and trajectory of the neighbouring vehicle with the GPS error of the sending vehicle is broadcast. By providing information that other vehicles might not be able to detect from vision alone hopefully better system performance can be attained without requiring a high rate of adoption by drivers.

For both sets of simulations essentially the same statistics will be recorded. Every 100ms the number of vehicles tracked by the unified model, the average position error and the number of incorrectly identified vehicles will be recorded. For the second set of simulations when each position update is broadcast the number of additional vehicles appended to each CCWS message will also be recorded.

5.2.3 Stage Three

Stage three will be based on CCWS from stage one as well. It will examine the performance of EWS and HWS compared to what is possible with VBDA. The same CCWS scheme decided upon in stage one will be run to ensure a realistic VANET environment with other wireless network traffic. The EWS or HWS will run as well and its performance will be measured. A summary of all simulations performed in stage three is available in Table 5.4.

Simulation	Parameters	Runs	Description
A	1	24	Testing EWS performance against VBDA
B	1	12	Testing HWS performance against VBDA
		36	**Total**

Table 5.4: Stage Three Experiments

The first set of simulations will involve sending out an emergency warning message. Since these events are very short lived instead of performing two runs at 120 seconds each we will instead perform four runs at 30 seconds in length on each road network. The event in question will occur within the first 5 seconds and 25 seconds will provide ample time to measure how the vehicles surrounding it react. A vehicle surrounded by heavy traffic will be picked and it will proceed to brake at its maximum deceleration rate after 5 seconds. At the same time it will send out an EWS message, as described in Section 3.5.2, with its rate of deceleration. Both the time that a vehicle receives the EWS message and the time at which the vehicle reacts, either by braking themselves or locating the vehicle in question with VBDA, will be recorded and compared. Sender latency will also be recorded to see if the higher priority AC of EWS makes any difference.

Finally, to test a HWS a set of simulation will be run. A vehicle within the first 30 seconds will be chosen to break down. It will slowly decelerate and stay stopped. At this point it will send out a HWS message to let other vehicles know there is a disabled vehicle blocking a lane. It will periodically resend this message as described in Section 3.5.3. These simulations will run for the full 120 seconds. For vehicles approaching the broken down vehicle the amount of time between receiving the first HWS message and identifying the vehicle with VBDA will be recorded. These results will show how much of a time advantage VANET communication can provide in relation to a HWS.

5.3 Analysis

Analysis of results will be done after the simulation has been run. Scalar results are recorded by OMNeT++ as the end of the simulation in a text file with one entry for each node and result. Vector results are recorded periodically throughout simulation by OMNeT++, again grouped by node and result. While the OMNeT++ environment provides some basic tools for result analysis and visualization, it is not robust enough to handle the number of results generated in large simulation runs and can not generate all of the summary results desired. After each set of simulations a custom Python script is used to analyze both the scalar and vector files.

For the scalar results a spreadsheet is generated with one row for each vehicle and one column for each result. The mean, standard deviation, minimum and maximum value for each result is calculated as well. Since there are a small number of scalar results this is a quick process.

For the vector files the entire file is analyzed sequentially. First, the entire file is read and the vectors put in memory grouped by node and result. The mean, standard deviation, minimum and maximum values are calculated for each node and result. Then all values of each result from all nodes are combined to calculate an accurate overall mean, standard deviation, minimum and maximum. Due to the large number of results the NumPy package is used. It can quickly and efficiently calculate statistics on millions of values so it is an ideal choice. These results are again exported to a spreadsheet.

Since there are two runs made for each parameter, road network and traffic level combination we have two mean, standard deviation, minimum and maximum values for each. Since the simulations should be nearly identical and the population size they were taken from very large the results should be almost identical. Any major discrepancies between two values will be noted if they occur and the mean of the two values will be used. Finally, all values for any particular statistic in both simulation runs will be included in the final mean and standard deviation calculations.

Chapter 6

Results and Analysis

The experiments described in Chapter 5 were all carried in the simulation environment discussed in Chapter 4. All simulation parameters used are either mentioned in the following discussion or listed in Appendix C. The parameters that are tested in later simulations are based on those found in related research or best guesses from initial experimentation. Unless otherwise noted, one run of the simulation entails running each scenario, road network and traffic level combination, twice with the specific parameters, for a total of 12 executions.

Simulations were executed on the SHARCNET visualization computer system to take advantage of the greater computer power available. The visualization systems provided an ideal system allowing multiple independent process to communicate easily via TCP, unlike the main SHARCNET cluster, while still having the power to keep simulation runtimes manageable. All simulations were run via command line outputting vector and scalar results for analysis.

6.1 Stage One

6.1.1 Part A

The first set of simulations in stage one seeks to test a number of broadcast schemes for a CCWS to find the ideal candidate. In the paper by S. Rezaei et al. they perform a similar series of experiments testing a large number of both variable and periodic transmission schemes [6]. Four schemes have been selected to test in our simulation environment. This serves two purposes. The first is to find a suitable broadcast scheme itself. Second, by validating our results against those attained by other researchers we can ensure that no significant issues exist with our implementation of the IEEE 802.11p physical and MAC layers, WSMP network layer and our CCWS application layer.

The variable broadcast schemes are set to broadcast after the SPE and RPE difference in

Broadcast Scheme	ACT	Loss Rate	Estimate Error (m) Mean	Estimate Error (m) STD	Vehicles Tracked Mean	Vehicles Tracked STD	Sender Latency (s) Mean	Sender Latency (s) STD
Highway Heavy								
Periodic 500ms	500ms	7.3%	0.338	0.348	78.30	14.07	0.000181	0.000194
Periodic 100ms	100ms	22.0%	0.310	0.323	79.00	14.02	0.000426	0.000409
Variable	975ms	6.2%	0.378	0.361	74.75	13.49	0.000154	0.000151
Variable w/ 50ms Repeat	493ms	7.3%	0.356	0.312	79.76	14.40	0.000191	0.000214
Highway Light								
Periodic 500ms	500ms	6.2%	0.335	0.345	40.56	7.45	0.000149	0.000136
Periodic 100ms	100ms	12.2%	0.295	0.293	41.37	7.51	0.000278	0.000303
Variable	973ms	5.8%	0.374	0.354	38.24	7.04	0.000140	0.000124
Variable w/ 50ms Repeat	495ms	6.1%	0.353	0.304	40.77	7.46	0.000158	0.000163

Figure 6.1: Various broadcast schemes for the highway road network

position crosses the 0.75m threshold. The rebroadcast happens within a randomly selected time slot from [0, 50] milliseconds after the first broadcast. Upon receiving a location update from a NV the SV creates a NVE with the information enclosed. This information is used to predict the vehicle's location in the future. If no further updates are received after 2 seconds the NVE is deleted.

The simulation was run under all four broadcast schemes. ACT represents the average communication time or the interval between sending location updates. For the periodic broadcast schemes you can note that the ACT is fixed. The loss rate is the number of packets received with errors over the total number of packets received. Estimate error represents the difference between a vehicle's actual location and where a NV estimates it should be based on the last received location update. The number of vehicles tracked is a count of all vehicles tracked in NVEs for each node. Both the estimate errors and number of vehicles tracked was recorded every 100ms throughout the simulation. The total of all values from both simulation runs for each statistic were collected to calculate the mean and standard deviation. Finally, the sender latency is the time between generating the location update and the time it was sent by the physical layer and is recorded every time a location update is sent out. The results are listed by road network in Figures 6.1, Figure 6.2 and Figure 6.3.

The first statistic of note is the high loss rate and mean latency under the 100ms periodic broadcast scheme. While loss rate does not measure the effectiveness of the scheme in regards to a CCWS, it does provide some insight into how the CCWS will perform in a real world scenario. With other VANET applications running or higher traffic densities a high loss rate could degrade system performance. Higher latency can also cause congestion and interfere with higher priority broadcasts being sent in time. In terms of overall tracking accuracy the 100ms scheme does performs best with the consistent lowest mean estimate error.

The variable broadcasting with rebroadcast has the second best performance in terms of mean estimate error in both the manhattan grid and city road networks. Since the highway

Broadcast Scheme	ACT	Loss Rate	Estimate Error (m) Mean	Estimate Error (m) STD	Vehicles Tracked Mean	Vehicles Tracked STD	Sender Latency (s) Mean	Sender Latency (s) STD
London Heavy								
Periodic 500ms	500ms	8.1%	0.311	0.425	62.64	25.87	0.000162	0.000157
Periodic 100ms	100ms	20.0%	0.281	0.329	62.72	25.14	0.000356	0.000385
Variable	948ms	7.2%	0.329	0.425	59.80	25.72	0.000144	0.000127
Variable w/ 50ms Repeat	484ms	8.4%	0.309	0.299	63.87	27.05	0.000181	0.000202
London Light								
Periodic 500ms	500ms	8.7%	0.326	0.519	40.63	17.50	0.000151	0.000144
Periodic 100ms	100ms	16.5%	0.282	0.344	41.10	17.31	0.000281	0.000321
Variable	933ms	8.0%	0.342	0.471	38.27	16.93	0.000140	0.000115
Variable w/ 50ms Repeat	477ms	8.6%	0.318	0.396	41.36	18.26	0.000162	0.000165

Figure 6.2: Various broadcast schemes for the city road network

road network involves few changes in direction the fixed periodic broadcast scheme performs quite well even at 500ms. However, the overall performance of the variable broadcast with rebroadcast makes it a better second choice in terms of mean estimate error. It has low latency, low mean estimate error and often beats the fixed 100ms broadcast scheme in terms of number of vehicles tracked.

The variable broadcast without rebroadcast scheme does perform quite well; however the rebroadcast scheme performs better with no significant penalties in loss rate or mean sender latency. As such, there is no compelling reason to choose the former. Finally, the 500ms periodic scheme would pose an issue in any situation where vehicles are moving unpredictably. For this reason alone it can be discounted. Therefore the final decision on which broadcast scheme to choose comes down to the 100ms period and variable broadcast with rebroadcast scheme. Mean estimate error performance is very close and the variable scheme performs better in all other accounts it so it will be used in further simulations.

Broadcast Scheme	ACT	Loss Rate	Estimate Error (m) Mean	Estimate Error (m) STD	Vehicles Tracked Mean	Vehicles Tracked STD	Sender Latency (s) Mean	Sender Latency (s) STD
Manhattan Heavy								
Periodic 500ms	500ms	7.7%	0.312	0.417	60.89	29.03	0.000165	0.000172
Periodic 100ms	100ms	19.9%	0.284	0.315	60.80	28.17	0.000363	0.000379
Variable	943ms	7.0%	0.332	0.389	58.28	28.46	0.000143	0.000130
Variable w/ 50ms Repeat	481ms	8.0%	0.310	0.296	62.05	29.89	0.000177	0.000196
Manhattan Light								
Periodic 500ms	500ms	6.4%	0.327	0.450	19.24	10.07	0.000127	0.000081
Periodic 100ms	100ms	9.0%	0.276	0.289	19.56	9.86	0.000197	0.000218
Variable	978ms	6.2%	0.342	0.374	18.19	9.74	0.000127	0.000074
Variable w/ 50ms Repeat	475ms	6.5%	0.316	0.288	19.25	10.08	0.000141	0.000125

Figure 6.3: Various broadcast schemes for the manhattan grid road network

These results mimic the results by S. Rezaei et al. [6] The four schemes performed in roughly the same order during their tests in regards to estimate error and loss rate. With confirmation from a second study our confidence in this approach is quite high moving onto further simulations.

6.1.2 Part B

The second set of simulations in stage one seeks to test the adaptive transmission power adjustment scheme proposed by the authors of [17]. It is paired with a similar variable broadcast scheme to the one we have selected. When the wireless channel is under high occupancy, nodes lower their transmission power. This reduces the communication radius allowing better communication with nearby vehicles by reducing the number of vehicles in communication with one another. Each node keeps track of the channel occupancy over the past one second via CCAs from the physical layer. At a set threshold the power is reduced linearly until we reach the minimum transmission power. However, given the the low latency we attained in the previous simulations attaining high channel occupancy in our simulations is unlikely.

We set our lower threshold at 25% and upper threshold at 75% channel occupancy. It should be noted that attaining 100% channel occupancy is essentially impossible in a real world scenario and approaching it would cause problems for the low latency requirements of VANET safety applications. The minimum transmission power level is set to 23dBm or 200mW and the maximum transmission power level is set to 34.3dBm or 3500mW, the same transmission power level used in the previous set of simulations. This provides a reliable communication range of 100m at the low end and 300m at the high end. One run was made with dynamic transmission power adjustment and one without. Statistics are calculated in the same way as the previous simulation. The channel occupancy was recorded each time a node broadcast a packet under the dynamic transmission power adjustment scheme.

Broadcast Scheme	Mean Channel Occupancy	Loss Rate	Estimate Error (m)		Vehicles Tracked		Sender Latency (s)	
			Mean	STD	Mean	STD	Mean	STD
Highway Heavy								
Without DTPA	NA	7.4%	0.352	0.298	79.87	14.44	0.000193	0.000218
With DTPA	11.9%	7.3%	0.353	0.299	79.89	14.42	0.000194	0.000219
London Heavy								
Without DTPA	NA	8.5%	0.302	0.304	63.86	27.04	0.000183	0.000205
With DTPA	9.5%	8.5%	0.301	0.303	63.89	27.06	0.000182	0.000202
Manhattan Heavy								
Without DTPA	NA	8.1%	0.305	0.291	62.04	29.87	0.000182	0.000203
With DTPA	9.5%	8.1%	0.305	0.314	62.06	29.87	0.000181	0.000203

Figure 6.4: Dynamic power adjustment for VANET

It is not possible for nodes in the simulation to constantly monitor the wireless channel so they are assigned 100 polling intervals each second. The total number of intervals during which the NAV is set or physical channel sensing decides the channel is occupied is tracked with a rolling array. The mean channel occupancy was approximately 10% for the Manhattan Grid and London road networks and 12% for the Highway road network with heavy traffic as we can see in 6.4. While it occasionally crossed over the 25% threshold with heavy traffic, lowering the transmission power, the effect it had on packet loss rate and latency was negligible. For road networks under light traffic with dynamic transmission power adjustment had no effect as the 25% threshold was never crossed.

As this result was expected the simulations were used to examine two other factors. First, the error threshold for the variable broadcast scheme was lowered from 0.75m to 0.5m. As both sets of simulations used this new error threshold it should have no effect on the conclusions drawn about dynamic transmission power adjustment. It had a negligible effect on both the loss rate, latency and a slight decrease in mean estimate error. This can be attributed to the tendency for SUMO to make large changes in trajectory in one time step instead gradual changes. These changes often violate the error threshold at either size. Since the new threshold provides a slight benefit it is used in later simulations.

	Tracked < 20m		Tracked < 50m		Tracked < 100m		Tracked Total	
Road Network	Tracked	Error (m)	Tracked	Error (m)	Tracked	Error (m)	Tracked	Error (m)
Highway Heavy	1.87	0.316	11.44	0.316	25.19	0.352	79.87	0.352
Highway Light	0.29	0.320	5.03	0.317	12.34	0.350	40.83	0.350
London Heavy	4.22	0.275	10.56	0.276	18.68	0.302	63.86	0.302
London Light	3.04	0.278	6.85	0.278	10.96	0.311	41.37	0.311
Manhattan Heavy	3.48	0.280	9.56	0.280	18.36	0.305	62.04	0.305
Manhattan Light	1.74	0.456	3.76	0.285	6.04	0.313	19.29	0.313

Figure 6.5: The effect of distance on estimation error

The second factor examined was the effect of the distance of NVs from the SV on estimate error. The results, listed in Figure 6.4, show that estimate error is fairly evenly distributed throughout vehicles in the various ranges. Two interesting results exist however. One is the high estimate error for the manhattan grid network under light traffic under 20m, which can be explained by the low number of vehicles tracked and therefore low population. The other is the very small number of vehicles tracked for the highway scenario under 20m, which can be explained by the large vehicle gaps required when traveling at a high rate of speed. While there does appear to be a slight increase in error going from 50m to 100m it is relatively small.

While the dynamic power adjustment scheme may not be necessary under the level of traffic present in these simulations, it should not be written off outright as it could prove very useful. The success of VANET for safety applications will depend on how well the limited wireless spectrum is used.

6.1.3 Part C

The third set of simulations examine the timeout interval for NVEs. When no further updates are received from a node after this interval the NVE is deleted. Too low of an interval and the number of vehicles tracked will be reduced without reducing estimation error. Too high of an interval will result in higher estimate error and inaccurate results. The previous timeout period of 2 seconds is tested alongside 1, 3 and 5 seconds. Since it is possible to maintain four sets of NVE statistics this set of simulations only requires one run and all four timeout intervals are examined at once.

	NVE Timeout 1s		NVE Timeout 2s		NVE Timeout 3s		NVE Timeout 5s	
Road Network	Tracked	Error (m)	Tracked	Error (m)	Tracked	Error (m)	Tracked	Error (m)
Highway Heavy	73.30	0.320	79.87	0.352	83.80	0.402	90.58	0.593
Highway Light	37.68	0.320	40.83	0.350	42.80	0.396	46.26	0.574
London Heavy	58.95	0.284	63.86	0.302	65.48	0.322	67.10	0.388
London Light	38.06	0.289	41.37	0.311	42.46	0.336	43.62	0.424
Manhattan Heavy	57.67	0.287	62.04	0.305	63.54	0.326	65.38	0.400
Manhattan Light	18.05	0.294	19.29	0.313	19.77	0.335	20.48	0.423

Figure 6.6: The effect of NVE timeout interval on estimation error

The results listed in Figure 6.6 from this set of simulations are not surprising. A timeout interval of 1 second does produce the lowest estimation error with a reduction in the number of vehicles tracked. Each subsequent increase in the timeout interval increases estimation error. The one result of interest is the large increase going from 2 to 3 seconds on the highway road network. Due to the high velocities, estimation error has the potential to increase very quickly. From this observation 3 and 5 seconds are both inadequate for real world traffic scenarios.

With the choice between 1 and 2 seconds the effect of the maximum update interval must be considered. Vehicles retransmit their location after at most 1s to ensure constant contact. With latency this can often be slightly over one second meaning the NVE would be deleted needlessly. As such, a NVE timeout interval between 1 and 2 seconds is ideal. As 2 seconds works well it will continue to be used in later stages.

6.1.4 Part D

The finally set of simulations in stage one introduce VBDA and the required vision simulation. At the same time VANET communications continues for a CCWS. Vision takes into account the position of all vehicles, the heading of the SV, to decide which camera system is used, and the heading of other vehicles along with their size to determine the orientation of their rectangular representation. Each 100ms when SUMO updates vehicle positions the vision simulation is rerun to produce a list of visible vehicles. This list is comprised of the distance to closest point on the vehicle as well as the minimum and maximum angles that would represent the 2D bounding box drawn over the vehicle in 3D space. We also know which node the visible vehicle corresponds to for verification purposes at this point. Of course in reality this would be unknown.

Road Network	Tracked VANET	Visible			Occluded
		All	VANET	No VANET	
Highway Heavy	79.92	12.30	12.19	0.02	1.66
Highway Light	40.83	6.07	6.02	0.01	0.45
London Heavy	63.91	8.23	8.20	0.01	3.89
London Light	41.37	5.65	5.63	0.01	2.03
Manhattan Heavy	62.09	7.88	7.85	0.01	3.39
Manhattan Light	19.27	2.91	2.89	0.01	0.83

Figure 6.7: Preliminary vision results

A single run was performed that measured how well VBDA can track vehicles and compared the position information gained from VBDA to the estimated positions from VANETs. In Figure 6.7 we have some basic statistics on how the number of vehicles tracked via VBDA. We have the mean number of vehicles tracked via VANET communication compared with the mean number of vehicles visible. We also have results for how many vehicles are visible and tracked via VANET and visible but not tracked. It is apparent that a CCWS does an excellent job of tracking nearby vehicles. Finally we have the mean number of vehicles occluded. This is the number of vehicles that are within range for the appropriate camera system but are blocked from view by another object. A comparison between the number of vehicles tracked via VANET and VBDA is presented in Figure 6.8

While VANET communication allows us to track a larger number of vehicles than VBDA this is to be expected due to its much larger range. VBDA provides us with a realtime feed of the most relevant information, that is the environment immediately around our vehicle. In Figure 6.8 we can see the difference in vehicles tracked.

Next, in Figure 6.9 we have a list of the standard deviations of the error between information provided by VBDA and the estimate of the same provided by our NVE location estimate. Since the error is relatively randomly distributed the mean is near 0 for the standard deviations presented. For a NV to be considered here it had to be both visible and tracked by an NVE in the SV. Using the location estimate provided by the NVE, the same vision calculations done in our vision simulator are perform again to get the estimated distance and angles to the NVE estimates. The difference is then recorded for each. Three different measures of this error were taken depending on the time since the last NVE updated. Initially it was thought that a longer period since the NVE was last updated meant that there would be more error. While this held true for distance the error for minimum and maximum angles was actually lower the longer it has been since the last NVE update.

This can be explained through basic probability. Since changes in vehicle trajectory result in an increase in the number of location updates sent we can theorize that the vehicles that

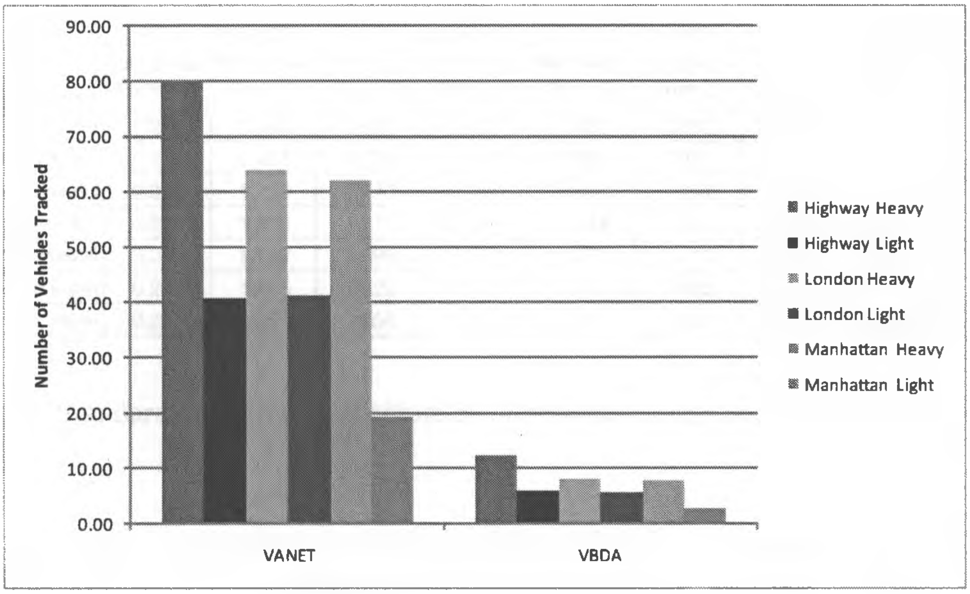

Figure 6.8: Number of vehicles tracked by VANET communication and by VBDA

are tracked and have not been updated 1.33 to 2 second interval have transmitted a minimum number of messages. If the vehicle were to deviate from its trajectory it would generate more location updates reducing the probability of the SV losing all of packets.

Overall, both VANET based CCWS and VBDA performed well. The performance of the VANET closely match results from other researchers. The only large variation was in PER. This can be explained by the variety of different PER models available. Detailed PER models for 802.11p have yet to be published at this point. With the results of these simulations further studies into the effectiveness of both technologies in more complex situations can be examined.

6.2 Stage Two

With the data from last section of stage one we aim to create a unified model from both VANET and VBDA. NVEs provide us a center position, heading and size for vehicles. VBDA provides us a distance to the vehicle and bounding box. It is not possible to estimate the exact center point of a vehicle via the information provided by VBDA. However the opposite is true, we can estimate a distance to and bounding box equivalent from our NVE location estimate.

Figure 6.9 provides us with the standard deviations for the error between VANET and VBDA results. The standard deviation of distance is relatively small suggesting that error in the predicted distance is relatively constant. However, the standard deviations of the minimum and maximum angles are much larger in some cases. The change in angle, via change in overall position from our up to date vision information in relation to our NVE estimate, is correlated

Road Network	0s < Last Update < 0.66s Distance Error (m)	Min Angle Error (°)	Max Angle Error (°)	0.66s < Last Update <1.33s Distance Error (m)	Min Angle Error (°)	Max Angle Error (°)	1.33s < Last Update < 2s Distance Error (m)	Min Angle Error (°)	Max Angle Error (°)
Highway Heavy	0.247	0.686	0.643	0.381	0.939	0.862	0.765	0.251	0.287
Highway Light	0.249	0.453	0.325	0.383	0.588	0.375	0.759	0.224	0.254
London Heavy	0.230	1.230	1.140	0.264	1.338	1.192	0.458	0.318	0.334
London Light	0.236	1.323	1.211	0.268	1.434	1.247	0.729	0.419	0.246
Manhattan Heavy	0.240	1.156	1.090	0.271	1.267	1.111	0.536	0.640	1.522
Manhattan Light	0.247	1.485	1.329	0.275	1.668	1.405	0.541	0.128	0.146
Mean	0.241	1.055	0.956	0.307	1.206	1.032	0.631	0.330	0.465

Figure 6.9: Standard deviation of absolute error between vision and NVE estimates

to the distance between the two objects. We can speculate the large standard deviation in angle is a result of this. Therefore, our unified model should take this into account.

The standard deviation of the distance error is well below 1m in all cases. Therefore we will use 1m of error, in either distance or in relation to angle, as our cutoff for matching an NVE and vision entry. To allow for a total of 1m of movement, the maximum error for angle in either direction will be 1m, $\arctan(\frac{1}{d})$ which can be approximated by $\frac{57}{d}$. If we define the distance between the vehicles as d, the absolute distance error as d_e, absolute minimum angle error as a_{min} and absolute maximum error as a_{max} between NVE position estimate P_i and visible vehicle V_j the unified model is as follows:

$$d_e < 1m$$

$$a_{min} < \frac{57}{d}$$

$$a_{max} < \frac{57}{d}$$

If the above above holds true then it is assumed $P_i = V_j$. Using this model we will simulate a CCWS supplemented by VBDA. We wish to examine three things through these simulations. The first is the effectiveness of our unified model in selecting the correct tracked vehicle and in turn reducing estimation error. We will record the number of corresponding NVE and visible vehicle correctly matched together, incorrectly matched together and incorrectly not matched to anything despite the possibility of being matched correctly. Second, we wish to examine the effect of various adoption rates on a VANET based CCWS in relation to the number of vehicles it can track. Vehicles will either be equipped with VANET and VBDA technology or not according to the adoption rate being tested. Finally, we wish to examine the effect of adding information obtained from VBDA, on vehicles who are not participating, back into VANET. We will be appending position estimates on up to four vehicles visible but not tracked by an NVE to position updates.

Adoption	ECCWS	Tracking Error (m) NVE	Tracking Error (m) Unified	Tracking Error (m) NVE	Tracking Error (m) Unified	Tracking Error (m) NVE	Tracking Error (m) Unified
10%	Yes	0.364	0.303	0.323	0.278	0.322	0.277
10%	No	0.352	0.293	0.303	0.263	0.314	0.271
25%	Yes	0.375	0.314	0.317	0.274	0.322	0.280
25%	No	0.358	0.305	0.302	0.264	0.308	0.270
50%	Yes	0.371	0.311	0.321	0.277	0.322	0.276
50%	No	0.355	0.301	0.306	0.267	0.306	0.265
75%	Yes	0.372	0.312	0.319	0.275	0.319	0.276
75%	No	0.360	0.305	0.309	0.268	0.312	0.272
90%	Yes	0.363	0.306	0.316	0.276	0.315	0.274
90%	No	0.359	0.304	0.312	0.274	0.312	0.273
100%	Yes	0.361	0.306	0.310	0.270	0.314	0.274
100%	No	0.359	0.304	0.310	0.270	0.312	0.273
		Highway		London		Manhattan	

Figure 6.10: Mean estimation error for both strictly VANET CCWS vs unified model

In Appendix B we have the results from all three experiments combined by the road network they were performed on. For the sake of brevity only the results pertaining the heavy traffic flows are included. VO is vehicles visible but not tracked by VANET communication. VT represents vehicles visible and tracked. VM represents the number of vehicles that could be linked between VBDA and VANET but were not in error over the total number of vehicles tracked minus the number of visible only vehicles. VE represents the number of incorrect matches between VBDA and VANET results over the total number of visible and tracked veicles. EWCCS is the number of vehicles tracked via position estimates from our enhanced CCWS protocol. Finally, the number of vehicles tracked in the unified model is the number of unique vehicles tracked by VANET, ECCWS if applicable and VBDA combined.

The unified tracking model itself performs well in terms of error rates. The number of missed matches between VANET and VBDA results is quite low. For all road networks except the highway road network it is under 0.5% and slightly over 0.5% on the highway road network. This increase in missed matches on the highway road network can be attributed to the higher rate of speed present and could be addressed with a more advanced model. The number of tracking errors, matches made between VANET and VBDA results where they are actually different vehicles, is consistently below 0.5% for all road networks. While a more advanced model is likely to be used in practice, our unified model provides excellent performance and allows us to measure its potential.

In regards to the NVE and unified tracking error it is calculated as follows. NVE tracking error is calculate as usual with the tracking error for all NVE estimates being recorded every 100ms. For the unified tracking error, if the NVE is linked to a visible vehicle the error is assumed to be 0. While vision is not entirely error free having a realtime estimate of the position provides a wealth of information unavailable with just a position estimate. Furthermore, as no vision error model exists trying to assign an error value to it would be counter productive.

The effect of adoption rate on VANET based CCWS technology is very predictable. The

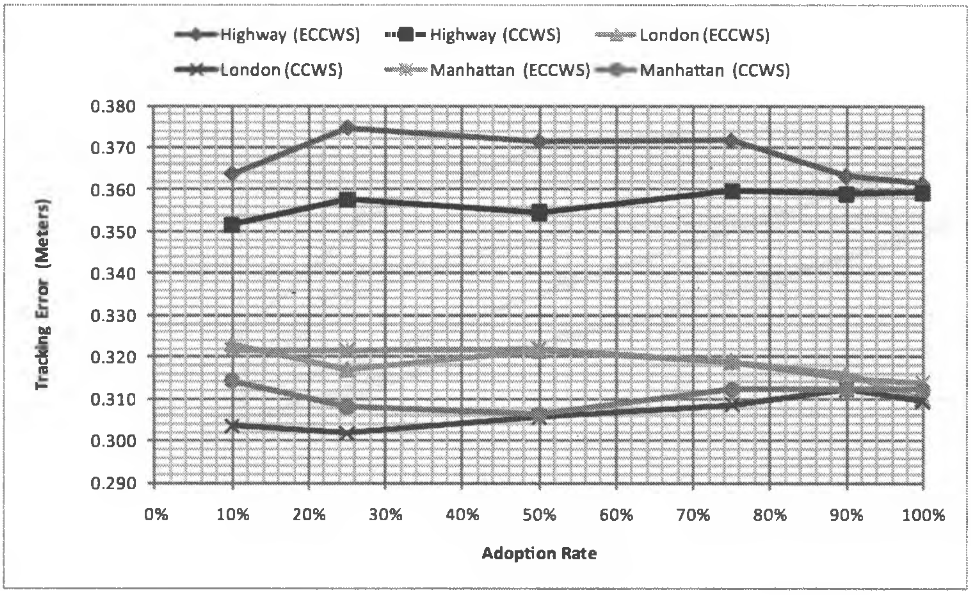

Figure 6.11: Tracking error of NVEs at various adoption rates

number of vehicles tracked with VANET alone scales linearly with the adoption rate as we can see in Figure 6.12. Mean tracking error remains constant through all adoption rates as expected since we have a low packet loss rate and latency to begin with. With our unified model it is clear even that under low adoption rates we can track a useful number of vehicles.

Finally, our enhanced CCWS protocol is what really stands out as an exceptional result. It allows vehicles equipped with both VANET and VBDA to track a large number of vehicles at a low adoption rate. In fact it is so effective it reaches its peak at about 50% adoption, as we can see in Figure 6.12. As the extra information is only appended to position updates for vehicles assumed to not be part of the VANET, it should put no additional load on the wireless channel. When adoption is low a large number of records will be appended making each CCWS packet larger, however since less vehicles will be participating there will be less overall channel utilization. Once most vehicles are equipped for VANET communication, very few extra records will need to be appended and the packet size will return to normal.

Finally, the effect on PER from our enhanced CCWS protocol is minimal. As we can see in Figure 6.13 the PER for the enhanced CCWS protocol at low adoption is higher than that of the CCWS protocol. This is due to the larger packet size. The performance of the enhanced CCWS is still adequation as we can see in Figure 6.11 with the estimation error only increasingly slightly at low adoption rates. Overall, an enhanced CCWS shows great promise as a way to significantly improve system performance with low adoption rates.

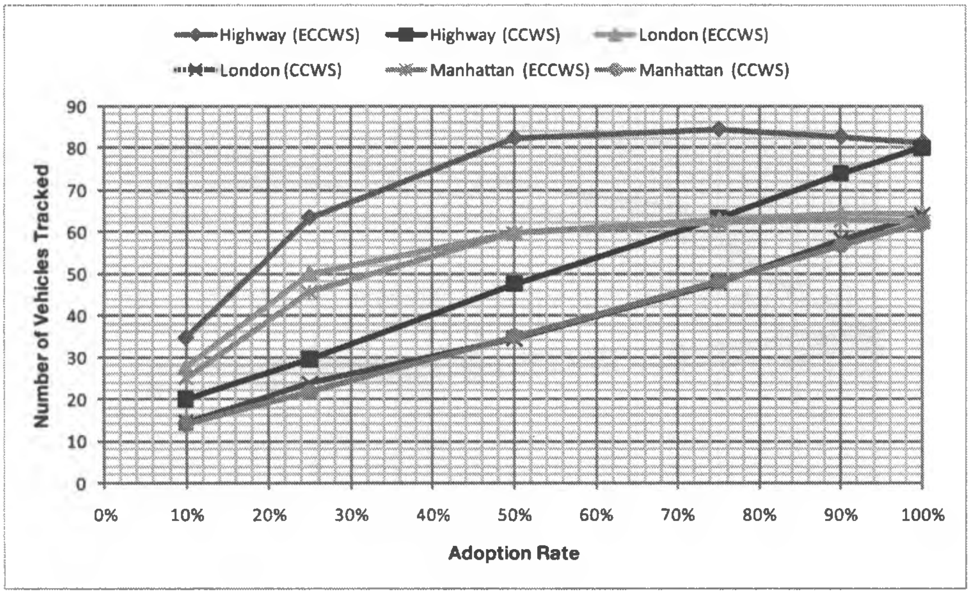

Figure 6.12: Number of vehicles tracked by CCWS and enhanced CCWS protocols at various adoption rates

6.3 Stage Three

The simulations performed in stage three look to examine the length of time between receiving a VANET message, either a HWS or EWS message, and visually identifying the vehicle that sent it. In an emergency situation having advanced notice gives the driver more time to react. Both services are simulated alongside CCWS network traffic but in theory the lower AC of EWS packets will allow them to be broadcast almost immediately. Two simulation runs as described in Chapter 5 were executed with short 30 second runs of for EWS simulation and full length runs of 120 seconds for HWS simulation.

One vehicle per run was chosen to slow and stop for the remainder of the simulation. For the EWS a fast stop at maximum deceleration mimicking an accident was made. For the HWS a slow stop representing vehicle troubles was performed. The time all other nodes received either a EWS or HWS message was recorded. If the vehicle subsequently picked up the sender of the EWS or HWS message with VBDA the length of time between receiving the message and sight was recorded. One modification to the EWS protocol was added. After broadcasting an EWS packet about a situation where the vehicle will be stationary afterwards, such as with an accident, it makes sense to switch over and start broadcasting HWS packets afterwards. After broadcasting a EWS packet the node enters HWS mode and broadcasts every 1 second as described in Section 3.5.3.

Unfortunately, the results of the runs produced no meaningful data. The mean time for detection for both sets of simulations varied widely based on the location of the vehicle as a

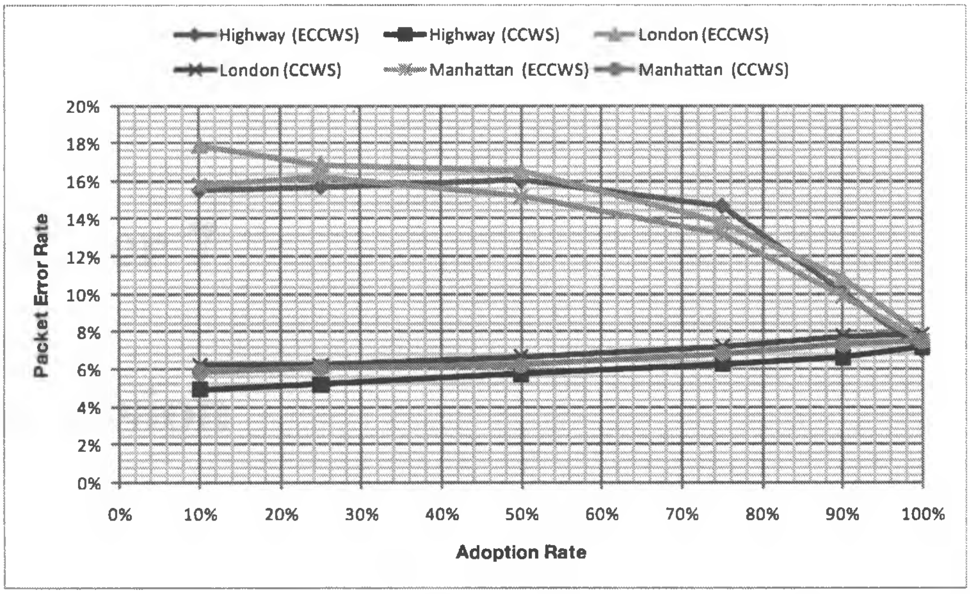

Figure 6.13: Packet error rates of CCWS and enhanced CCWS protocols at various adoption rates

function of the random number generator seed. A more specific approach to scenario creation is required to study the effect of emergency situations. The effect of AC was also negligible since network utilization is already quite low. Finally, compounding the problem SUMO is poorly suited to modeling emergency events. Overall, a new approach is require for simulating these types of network traffic.

Chapter 7

Discussion

As the previous chapters have demonstrated both VANET and VBDA have strong potential. They both have the capabilities to create a very complete model of the environment surrounding a vehicle. Together they perform extremely well even under less than ideal conditions, such as low adoption rates. While these experiments have established a promising baseline measure numerous improvements on the proposed protocols and models are possible. Furthermore, the simulation environment itself has the potential to become quite robust with the addition of particular features. The final chapter will outline some of these improvements and areas for future study.

The baseline provided by the results in this thesis will hopefully illuminate a number of areas where further study can be performed. While the results show that VANET based CCWS, HWS and EWS are feasible this is not a new realization. The effectiveness of tracking neighbouring vehicles under a variety of conditions over a VANET has been reinforced however. Additionally, the logistics of an EWS operating at the same time as a CCWS have been introduced as well. Most research models one system in isolation and lack the competing traffic a CCWS presents. Unfortunately it will need further study to draw meaningful conclusions from.

More importantly than these VANET oriented results are the possibilities presented by the combination of VANET and VBDA. With no existing work on simulating computer vision, network and traffic alongside one another, a wealth of new data can be examined. The approach to vision simulation is both quite simple and able to produce robust results with minimal overhead. With the simulation environment in place future experimentation in this regard will should be simplified.

7.0.1 Experiments and Results

The results attained from the three stages of experiments serve to highlight the potential for both technologies. All of approaches taken are basic in nature. More accurate models could be implemented in regards to vehicle movement and vehicle tracking, both VANET and VBDA based, and this would only serve to improve their performance.

The results of stage one show that highly accurate tracking of vehicles is possible in a VANET environment. Even under very high traffic level, using a low bit rate, network congestion is kept to a minimum when using an effective broadcast scheme. In real world scenarios network utilization may be even higher however there is ample capacity and the option of using a higher bit rate exists. Higher network utilization could be the result of either more dense traffic or more VANET applications running simultaneously. Simulating higher traffic levels is certainly one possible improvement that could be made however it is a difficult problem because the computational complexity grows exponentially. For every node in communication range full SINR, BER and PER calculations must be done for every attempt at communication. Smaller scale simulations maintaining the same number of nodes more densely packed is one possible solution to this. Simulating more VANET applications is another approach but without a specific purpose in mind generating random network traffic may not produce realistic results.

In relation to channel utilization the dynamic power adjustment scheme proposed could very well be useful in a mature VANET environment. While in the experiments performed the channel utilization was too low for it to have any noticeable effect it will have to be tested further with more network traffic. The NVE timeout interval could also be reexamined in the form of a weighted model where NVE estimates are kept longer but with less confidence in old results. Overall, 2 seconds does provide a simple solution with a balance between estimate error and the number of vehicles tracked.

The integration of VANET and VBDA results again could be examined in great detail to find an ideal unified model. As the simulations were run on a single computer, computation considerations for efficiency had to be made. Having hundreds of nodes performing expensive calculations 10 times per simulation second would drastically increase runtime. A more robust search for the most likely link between results from both systems would certainly be possible to implement without these considerations.

The promising results shown under low adoption rates with both technologies working together using an enhanced CCWS protocol shows this approach is worth examining in detail. The implementation of the positioning in the two systems is done very differently, with one system providing a point in space and the other providing the equivalent of a 2D bounding box, and the best way to integrate the two coordinate systems deserves more thought.

All in all the results have shown that both these technologies can work well together. The logistics of implementing them are sound. Vision, RADAR or LIDAR based driver assistance is already a reality. Once these systems become widespread the addition of vehicular networking is an inexpensive way to increase their reach significantly.

7.0.2 Issues and Improvements

While there are certainly a number of areas to improve upon in regards to the experiments and results themselves, there are a number of other issues that can be touched upon in regards to future improvements in other areas. The simulation environment performed well however it can be improved in a number of ways.

SUMO was an obvious choice for traffic simulation because of its open source nature and integrations with OMNeT++ via VEINS. While it does an excellent job, a more advanced traffic simulator could be useful for a number of reasons. Vehicle movement in regards to lane changes is not realistic which introduces a small source of error into results. Furthermore, a traffic simulator that can model in 3D would allow for 3D vision and wireless simulation as all of the other components either already support 3D simulation or could easily be extended to support it.

The addition of obstruction information, mainly the position of buildings, to the road networks is another improvement that could be made. The obstruction information could be used for both simulating radio shadowing, using the model described by C. Sommer et al. and implemented in OMNeT++, and to model visual obstructions in regard to vision simulation [41]. Both of these tasks would be simple to implement in the simulation given the existing tools to do it. Unfortunately, acquiring realistic obstruction information is a difficult task.

Further study into the type and nature of error from computer vision, and how these errors will affect VBDA, could help to produce better results from vision simulation. While there is no doubt that it provides an excellent source of up to date information better comparisons can be made with VANET data if an error model did exist. The current vision model is entirely deterministic and may not mimic real results. A stochastic vision error model could improve accuracy.

The statistic collection utilities in OMNeT++ could also be modified to better suit the large simulations performed for our experiments. Using binary files instead of text files could greatly reduce disk space usage and decrease the runtime for both simulation and statistical analysis. Furthermore, better linking of related results together could enable further statistical analysis. While it would certainly be possible to implement this due to the open source nature of OMNeT++, statistics collection is a core functionality of the environment and modifying it would not be a trivial task. Furthermore, despite the short comings of the existing statistical analysis facilities they do perform extremely well even with hundreds of millions of records being generated.

Finally, real world trails of the technologies will help to verify the results of simulations as correct. We have seen a number of researchers using custom built 802.11p radio units or prototypes supplied by industry [41] [13] [42] [16]. With access to the RoadLab project we have an excellent platform to begin our own real world trials to verify the accuracy of traffic, network and vision simulation.

7.0.3 Closing Thoughts

The possibilities provided through driver assistance technologies for improved safety are certainly promising. We are seeing today a wealth of driver assistance technology being installed in production vehicles. At the same time traditional internet based wireless communication is being installed as well. Having VBDA and VANETs in every vehicle does not seem too far off in the future. Consumers have shown a willingness to adopt new vehicular technologies and given proliferation of technology throughout every aspect of our lives the adoption of advanced driver assistance systems is likely.

When advanced driver assistant systems become a reality the possibilities for future automation, and improved safety, in a vehicular environment is even more incredible. Perhaps one day driving, one of the worlds most deadly activities, can be reformed.

Bibliography

[1] S. Beauchemin, M. Bauer, D. Laurendeau, T. Kowsari, J. Cho, M. Hunter, and O. McCarthy, "Roadlab: An in-vehicle laboratory for developing cognitive cars," 2010.

[2] S. Grafling, P. Mahonen, and J. Riihijarvi, "Performance evaluation of IEEE 1609 wave and IEEE 802.11p for vehicular communications," in *Ubiquitous and Future Networks (ICUFN), 2010 Second International Conference on*, pp. 344 –348, June 2010.

[3] Transport Canada, "Canadian motor vehicle traffic collision statstics." Online, 2008.

[4] World Health Organization, "Global road safety crisis." Online, 2003.

[5] Transport Canada, "Transport canada releases first, systematic analysis of cost of urban traffic congestion in canada." Press Release, 2006.

[6] S. Rezaei, R. Sengupta, H. Krishnan, X. Guan, and R. Bhatia, "Tracking the position of neighboring vehicles using wireless communications," *Transportation Research Part C*, 2009.

[7] T. Strang and M. Rockl, "Vehiclular networks lecture," 2008.

[8] D. McGehee, E. Mazzae, and S. Baldwin, "Driver reaction time in crash avoidance research : Validation of a driving simulator study on a test track," *International Ergonomics Association. Congress (14th : 2000 : San Diego, Calif.). Ergonomics for the new millennium*, 2000.

[9] EU Telecommunication Commission, "Cars that talk: Commission earmarks single radio frequency for road safety and traffic management." Press Release, 2006.

[10] F. Yu and S. Biswas, "Impacts of Radio Access Protocols on Cooperative Vehicle Collision Avoidance in Urban Traffic Intersections," *JOURNAL OF COMMUNICATIONS*, vol. 3, no. 4, p. 41, 2008.

[11] T. Taleb, K. Ooi, and K. Hashimoto, "An Efficient Collision Avoidance Strategy for ITS systems," in *IEEE Wireless Communications and Networking Conference, 2008. WCNC 2008*, pp. 2212–2217, 2008.

[12] Y. Wang, A. Ahmed, B. Krishnamachari, and K. Psounis, "IEEE 802.11p performance evaluation and protocol enhancement," in *Vehicular Electronics and Safety, 2008. ICVES 2008. IEEE International Conference on*, pp. 317 –322, 2008.

[13] P. Fuxjäger, A. Costantini, D. Valerio, P. Castiglione, T. Zemen, and F. Ricciato, "IEEE 802.11p Transmission Using GNURadio," in *Proceedings of the 6th Karlsruhe Workshop on Software Radios (WSR10)*, 2010.

[14] T. Fujiki, M. Kirimura, T. Umedu, and T. Higashino, "Efficient Acquisition of Local Traffic Information using Inter-Vehicle Communication with Queries," in *IEEE Intelligent Transportation Systems Conference, 2007. ITSC 2007*, pp. 241–246, 2007.

[15] C. Sommer, R. German, and F. Dressler, "Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis," *IEEE Transactions on Mobile Computing*, vol. 10, pp. 3–15, January 2011.

[16] H. Tan and J. Huang, "DGPS-based vehicle-to-vehicle cooperative collision warning: Engineering feasibility viewpoints," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 4, pp. 415–428, 2006.

[17] C.-L. Huang, Y. Fallah, R. Sengupta, and H. Krishnan, "Adaptive intervehicle communication control for cooperative safety systems," *Network, IEEE*, vol. 24, no. 1, pp. 6 –13, 2010.

[18] M. Raya, P. Papadimitratos, J. Hubaux, and E. de Lausanne, "Securing vehicular communications," *IEEE Wireless Communications*, vol. 13, no. 5, pp. 8–15, 2006.

[19] A. Girard, J. de Sousa, J. Misener, and J. Hedrick, "A control architecture for integrated cooperative cruise control and collision warning systems," in *Decision and Control, 2001. Proceedings of the 40th IEEE Conference on*, vol. 2, pp. 1491 –1496 vol.2, 2001.

[20] C. Olaverri-Monreal, P. Gomes, R. Fernandes, F. Vieira, and M. Ferreira, "The see-through system: A VANET-enabled assistant for overtaking maneuvers," in *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pp. 123 –128, 2010.

[21] K. Kotani, W. Sun, T. Kitani, N. Shibata, K. Yasumoto, and M. Ito, "Inter-Vehicle Communication Protocol for Cooperatively Capturing and Sharing Intersection Video," in *IEEE CCNC, 2010. Proceedings*, 2010.

[22] G. Yan, S. Olariu, and M. Weigle, "Providing VANET security through active position detection," *Computer Communications*, vol. 31, no. 12, pp. 2883 – 2897, 2008.

[23] L. Cheng, B. Henty, R. Cooper, D. Stancil, and F. Bai, "A measurement study of time-scaled 802.11a waveforms over the mobile-to-mobile vehicular channel at 5.9 ghz," *Communications Magazine, IEEE*, vol. 46, no. 5, pp. 84 – 91, 2008.

[24] E. Minack, "Evaluation of the influence of channel conditions on Car2X communication," Master's thesis, Chemnitz University of Technology, 2005.

[25] K. Wehrle, M. Gunes, and J. Gross, *Modeling and Tools for Network Simulation*. Springer, 1st ed., 2010.

[26] G. Yeung, M. Takai, R. Bagrodia, A. Mehrnia, and B. Daneshrad, "Detailed OFDM modeling in network simulation of mobile ad hoc networks," in *Proceedings of the eighteenth workshop on Parallel and distributed simulation*, PADS '04, pp. 26–34, 2004.

[27] S. Cocorada, "An IEEE 802.11g simulation model with extended debug capabilities," 2008.

[28] D. Jiang and L. Delgrossi, "IEEE 802.11p: Towards an international standard for wireless access in vehicular environments," in *Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE*, pp. 2036 –2040, May 2008.

[29] *IEEE Standard for Wireless Access in Vehicular Environments (WAVE)–Multi-channel Operation*, 2011.

[30] T. Chen, W. Jin, and A. Regan, "Multi-Hop Broadcasting in Vehicular Ad Hoc Networks with Shockwave Traffic," in *IEEE CCNC, 2010. Proceedings*, 2010.

[31] "Vehicle safety communications project: Task 3 final report - identify intelligent vehicle safety applications enabled by DSRC," tech. rep., National Highway Traffic Safety Administration, 2005.

[32] J. Yoon, M. Liu, and B. Noble, "Random waypoint considered harmful," *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, 2003.

[33] S. Krau, "Microscopic traffic simulation: Robustness of a simple approach," Tech. Rep. 306, Deutsches Zentrum für Luft- und Raumfahrt, 1997.

[34] E. Brockfeld and P. Wagner, "Testing and benchmarking of microscopic traffic flow simulation models," 2004.

[35] M. Piorkowski, M. Raya, A. L. Lugo, P. Papadimitratos, M. Grossglauser, and J.-P. Hubaux, "TraNS: Realistic joint traffic and network simulator for VANETs," *ACM SIGMOBILE Mobile Computing and Communications Review*, 2007.

[36] S. Wang, C. Chou, Y. Chiu, Y. Tzeng, M. Hsu, Y. Cheng, W. Liu, and T. Ho, "NCTUns 4.0: An integrated simulation platform for vehicular traffic, communication, and network researches," in *Vehicular Technology Conference, 2007. VTC-2007 Fall. 2007 IEEE 66th*, pp. 2081 –2085, 2007.

[37] OpenSim Ltd, *OMNeT++ User Manual Version 4.1*, 2010.

[38] K. Wessel, M. Swigulski, A. Kpke, and D. Willkomm, "Mixim the physical layer an architecture overview," *2nd International ICST Workshop on OMNeT++*, May 2009.

[39] "INETMANET." Online, 2011.

[40] City of London, "City of london traffic." Online, 2006.

[41] C. Sommer, D. Eckhoff, R. German, and F. Dressler, "A Computationally Inexpensive Empirical Model of IEEE 802.11p Radio Shadowing in Urban Environments," Technical Report CS-2010-06, University of Erlangen, Dept. of Computer Science, September 2010.

[42] A. Costantini, "Implementation of an IEEE 802.11p transmitter in open-source software defined radio," Master's thesis, Universit del Salento, 2009.

Appendix A

Definitions

ABS	Anti-Lock Braking System
AC	Access Category
ACC	Adaptive Cruise Control
ACK	Acknowledgement Packet
AIFS	Arbitrary Interframe Space
AP	Access Point
BER	Bit Error Rate
BPSK	Binary Phase-Shift Keying
BSS	Basic Service Set
BSSID	Basic Service Set ID
CA	Central Authority
CACC	Cooperative Adaptive Cruise Control
CCA	Clear Channel Assessment
CCH	Control Channel
CCWS	Cooperative Collision Warning System
CRC	Cyclic Redundancy Check
CRL	Certificate Revocation List
CSMA/CA	Carrier Sensing Multiple Access with Collision Avoidance
CSMA/CD	Carrier Sensing Multiple Access with Collision Detection
CTS	Clear to Send Packet
CWS	Collision Warning System
DARPA	Defense Advanced Research Projects Agency
DCF	Distributed Coordinated Function
DGPS	Differential GPS
DIFS	DCF Interframe Space

DSRC	Dedicated Short Range Communication
FEC	Forward Error Correction
FCC	Federal Communication Commission
FCWS	Forward Collision Warning System
EDCA	Enhanced Distributed Channel Access
EWS	Emergency Warning System
GPS	Global Positioning System
HMAC	Hash-based Message Authentication Code
HWS	Hazard Warning System
IDE	Integrated Development Environment
IEEE	Institute of Electrical and Electronics Engineers
JOSM	Java Open Street Map Editor
LAN	Local Area Network
LOS	Line of Sight
MAC	Media Access Control
MANET	Mobile Ad-Hoc Network
NIC	Network Interface Card
NAV	Network Allocation Vector
NLOS	Non-Line of Sight
NV	Neighbour Vehicle
NVE	Neighbouring Vehicle Estimator
OBU	On Board Unit
OFDM	Orthogonal Frequency Division Multiplexing
OSM	Open Street Map
OSI	Open System Interconnection
PER	Packet Error Rate
PKI	Public Key Infrastructure
QoS	Quality of Service
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase-Shift Keying
RPE	Remote Position Estimator
RSU	Road Side Unit
RTS	Request to Send Packet
SCH	Service Channel
SPE	Self Position Estimator
SIFS	Short Interframe Space
SINR	Signal-to-Interference plus Noise Ratio

SPI	Service Provider Identifier
SV	Subject Vehicle
SUMO	Simulation of Urban Mobility
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access
TPD	Tamper Proof Device
TraCI	Traffic Control Interface
UDP	User Datagram Protocol
UICWS	Urban Intersection Crash Warning System
UTC	Coordinated Universal Time
V2V	Vehicle-to-Vehicle
V2I	Vehicle-to-Infrastructure
VANET	Vehicular Ad-Hoc Network
VBDA	Vision-Based Driver Assistance
VEINS	Vehicles in Network Simulation
WAVE	Wireless Access for Vehicular Environments
WBSS	WAVE Basic Service Set
WHO	World Health Organization
WSA	WAVE Service Announcement
WSM	WAVE Short Message
WSMP	WAVE Short Message Protocol

Appendix B

Stage Two Results

Adoption	ECCWS	Vision						Vehicles Tracked		
		Occluded	Visible	VO	VT	VM	VE	VANET	ECCWS	Unified
10%	Yes	1.73	12.40	10.98	1.42	0.52%	0.04%	8.44	26.15	34.59
10%	No	1.73	12.31	10.75	1.55	0.53%	0.04%	9.25	NA	20.00
25%	Yes	1.69	12.22	9.09	3.10	0.67%	0.42%	19.25	44.06	63.32
25%	No	1.69	12.26	9.26	2.98	0.62%	0.39%	20.22	NA	29.48
50%	Yes	1.62	12.35	6.01	6.30	0.68%	0.30%	38.68	43.51	82.19
50%	No	1.63	12.34	6.02	6.28	0.60%	0.34%	41.45	NA	47.47
75%	Yes	1.68	12.30	3.04	9.19	0.69%	0.28%	57.12	27.11	84.22
75%	No	1.69	12.31	3.09	9.16	0.63%	0.29%	60.11	NA	63.20
90%	Yes	1.66	12.29	1.11	11.11	0.65%	0.22%	71.16	11.38	82.54
90%	No	1.66	12.29	1.08	11.13	0.66%	0.22%	72.73	NA	73.81
100%	Yes	1.66	12.30	0.02	12.19	0.69%	0.21%	79.87	1.32	81.19
100%	No	1.66	12.30	0.02	12.19	0.65%	0.22%	79.92	NA	79.94

Figure B.1: Stage Two results for highway road network under heavy traffic

Adoption	ECCWS	Vision						Vehicles Tracked		
		Occluded	Visible	VO	VT	VM	VE	VANET	ECCWS	Unified
10%	Yes	4.04	8.17	7.19	0.98	0.19%	0.74%	7.07	20.59	27.66
10%	No	3.94	8.04	7.07	0.97	0.17%	0.78%	7.30	NA	14.37
25%	Yes	4.51	8.73	6.49	2.24	0.25%	0.86%	16.64	33.35	49.98
25%	No	4.40	8.66	6.49	2.17	0.23%	0.91%	17.24	NA	23.73
50%	Yes	4.04	8.34	4.57	3.76	0.21%	1.18%	27.52	32.15	59.68
50%	No	4.08	8.37	4.56	3.81	0.20%	1.19%	29.80	NA	34.35
75%	Yes	3.93	8.25	2.30	5.94	0.21%	0.62%	43.77	18.90	62.68
75%	No	3.91	8.31	2.30	6.00	0.20%	0.65%	45.43	NA	47.73
90%	Yes	3.87	8.14	1.08	7.05	0.24%	0.54%	55.87	8.36	64.23
90%	No	3.87	8.13	1.08	7.04	0.23%	0.54%	56.94	NA	58.02
100%	Yes	3.89	8.23	0.01	8.20	0.21%	0.45%	63.89	0.23	64.12
100%	No	3.89	8.23	0.01	8.20	0.22%	0.44%	63.91	NA	63.92

Figure B.2: Stage Two results for London road network under heavy traffic

Adoption	ECCWS	Vision						Vehicles Tracked		
		Occluded	Visible	VO	VT	VM	VE	VANET	ECCWS	Unified
10%	Yes	3.85	8.08	7.20	0.88	0.37%	0.35%	6.43	18.61	25.04
10%	No	3.93	8.28	7.36	0.91	0.44%	0.33%	6.70	NA	14.06
25%	Yes	3.53	8.04	6.20	1.84	0.32%	1.19%	14.22	31.31	45.53
25%	No	3.59	8.12	6.19	1.92	0.38%	1.27%	15.44	NA	21.63
50%	Yes	3.67	8.34	4.21	4.12	0.30%	0.82%	29.10	30.45	59.55
50%	No	3.78	8.48	4.32	4.15	0.31%	0.79%	30.54	NA	34.86
75%	Yes	3.48	8.12	2.18	5.92	0.32%	0.66%	44.13	18.03	62.16
75%	No	3.51	8.16	2.20	5.95	0.31%	0.64%	45.82	NA	48.02
90%	Yes	3.37	7.87	0.84	7.00	0.34%	0.60%	54.63	8.02	62.65
90%	No	3.36	7.83	0.83	6.99	0.32%	0.60%	55.80	NA	56.62
100%	Yes	3.39	7.88	0.01	7.85	0.34%	0.52%	62.06	0.34	62.40
100%	No	3.39	7.88	0.01	7.85	0.33%	0.52%	62.09	NA	62.10

Figure B.3: Stage Two results for Manhattan grid road network under heavy traffic

Appendix C

Parameters

Parameter	Value	Notes
Physical Layer		
Maximum Transmit Power	35.3dBm	Default transmission power to use and maximum level when using DTPA
Minimum Transmit Power	23dBm	Minimum transmit power to use for DTPA
Receiver Sensitivity	-87dBm	Minimum signal level required to be detected
Thermal Noise	-98dBm	Base level of noise present in receiver
Header Duration	40μs	
Center Frequency	9.89GHz	
Channel Bandwidth	10MHz	
Modulation Scheme	QPSK	
Coding Rate	1/2	
Bit Rate	6Mbps	
Channel Model		
Path Loss Coefficient	3.0	
Shadowing Mean	0dB	Mean signal attenuation from shadowing
Shadowing STD	4dB	Standard deviation of signal attenuation from shadowing
MAC Layer		
Header Length	272bits	
Channel	178	
Network Layer		
Header Length	88bits	
Application Layer		
Header Length	1936bits	
CCWS Error Threshold	0.5m	
CCWS Retransmit Time	50ms	
Maximum Update Interval	1s	Maximum time before a CCWS location update is broadcast even if error threshold has not been crossed
Vision		
Vision Cutoff	50%	Percentage of vehicle required to be visible to be detected by VBDA

Appendix D

File Formats

D.0.4 Road Network XML File Format

```
<net>
   <edge id=":A0_0" function="internal">
      <lanes>
         <lane id=":A0_0_0" maxspeed="13.90" length="18.55" ... />
      </lanes>
      <lanes>
         <lane id=":A1_0_0" maxspeed="13.90" length="18.55" ... />
      </lanes>
   </edge>

   ...

   <junction id="A0" type="priority" x="0.00" y="0.00" ... />
   <junction id="A1" type="priority" x="0.00" y="150.00" ... />

   ...

   <succ edge="A0A1" lane="A0A1_0" junction="A1">
      <succlane lane="A1B1_0" ... />
      <succlane lane="A1A2_0" ... "/>
   </succ>

   ...

</net>
```

D.0.5 Route XML File Format

```
<routes>
    <vtypeDistribution id="vehicles">
```

```
        <vtype id="p1" accel="0.7" decel="4.5" sigma="0.8" probability="0.125" />

        ...

    </vtypeDistribution>

    <vehicle id="t0" depart="0.00">
        <route edges="C1C2 C2B2 B2A2 A2A1 A1A0 A0B0" type="vehicles" />
    </vehicle>
    <vehicle id="t1" depart="2.00">
        <route edges="C0B0" type="vehicles" />
    </vehicle>

    ...

</routes>
```