A Review of VANET Data Dissemination and Intelligent Traffic Signal Control Strategies

Dave McKenney and Dr. Tony White

November 1st, 2012

Contents

1 Introduction

In recent years, mobile computing and communication technology has become more widespread and affordable. A large amount of research has now emerged which aims to use these technologies to address a common problem: traffic. Throughout the world, millions (if not billions) of people move through traffic networks every day. If this traffic is not controlled efficiently and effectively, a wide range of economic, environment and social problems may arise. Intelligent, autonomous control of traffic, then, is an important research area which can help alleviate traffic congestion. Generally, proposed intelligent traffic control systems assume that specific information is available, such as traffic densities and average vehicle speeds. Using current road-based sensor technology (i.e., induction loop sensors), however, does not allow for most of this information to be accurately observed in real-time. By sharing information through the use of a Vehicular Ad Hoc Network (VANET), this information can be made available, enabling effective intelligent control to become a reality. Beyond the intelligent control of traffic, several safety-related applications based on VANET technology have also been proposed. Combining these VANET-based systems can lead to safer and more efficient traffic networks, resulting in decreased emissions, lower travel times, and less traffic-related incidents. Also, since VANETs largely rely on communication and sensor devices located within vehicles, a VANET-based control system may lead to increased savings for municipalities, who may no longer have to maintain sensor and communication devices (i.e., induction loop sensors and intersection communication devices). Currently, spectra have been allocated in the United States, Europe and Japan for vehicle-specific communication (see Section 2.3 for details), signalling that governmental authorities are beginning to acknowledge the possible power VANETs may hold. With the required technology available, research now needs to focus on designing scalable, self-organizing, intelligent control solutions that harness the power of VANETs. This paper will discuss existing algorithms/literature relating to data dissemination in VANETs and intelligent control of traffic. The paper will also discuss how these algorithms may be validated experimentally within realistic simulations. Finally, this paper will outline future research directions in the area of intelligent traffic control using VANET technology.

1.1 Scope

This paper deals with the research areas of data dissemination and traffic control within VANETs. The focus of this paper will be on the development and investigation of VANET systems within realistic traffic simulations. Furthermore, this work will largely deal with the algorithmic/software aspects of VANET systems, and will only briefly touch on hardware specifications. From an algorithmic point of view, this work will focus on approaches which are scalable, self-organizing and realistic. Several other approaches (i.e., coding-based data dissemination) are not included within this work due to the unrealistic assumptions required.

1.2 Objectives

This paper has a number of overall objectives. First, the paper aims to introduce VANETs (architecture, protocols, defining characteristics) and identify some of the main concerns of VANET system design. The second goal of the paper is to discuss the state of the art in VANET data dissemination algorithms. Along with this discussion, the strengths and weaknesses of the various approaches to data dissemination will be identified. Furthermore, the paper will detail various types of intelligent traffic control applications, including approaches that do not rely on VANET technology and those that use VANET technology to some degree. The applications discussed will involve intelligent traffic signal management, safety applications and dynamic vehicle route assignments. This paper will also introduce several popular network simulators that are currently available and identify considerations that must be made when designing a realistic VANET simulation. Finally, the paper will conclude by outlining the future research vision for VANET technology and VANET traffic systems.

1.3 Document Outline

The remainder of this document is divided into a number of sections. This subsection will outline the structure of the document and identify the main topics covered in each of the remaining sections.

Section 2 acts as an introduction to VANETs and will, therefore, be of most use to readers who have little background knowledge in the area of VANET research. Section 2.1 begins by identifying a number of defining VANET characteristics and comparing VANETs to the more general mobile ad hoc networks (MANETs). Section 2.2 then introduces a typical VANET architecture, including various types of sensors that may be used to generate data (Section 2.2.1), and the role that additional roadside infrastructure may play in a VANET system (Section 2.2.2). The section concludes with Section 2.3, an introduction to the DSRC/802.11p protocol, which is the most widely used communication protocol for VANET communications.

Readers interested in the movement of data through a network will most likely find Section 3 to be the most interesting. This section discusses several approaches and algorithms used for data dissemination within a VANET. The section begins by introducing dissemination and data broadcasting, while also defining other types of dissemination goals (i.e., multicasting, unicasting and geocasting). Algorithms are then divided into those that operate using multi-hop (Section 3.1) and single-hop (Section 3.2.1) forwarding. Within these sections, an effort is made to identify the relative strengths and weaknesses of each discussed algorithm. These strengths and weaknesses are used within the section's summary (Section 3.3) to outline desirable characteristics of a dissemination algorithm. The summary of the section also includes a table comparing the estimated performance of several of the proposed algorithms over a number of these desired characteristics (see Table 1 in Section 3.3).

Section 4 discusses communications simulators that may be used to model and investigate VANET systems. This section will be of particular interest to readers who have some background in the simulation or modelling of traffic, but are relatively new to the idea of including wireless communications within these simulations. The section includes discussion of mobility models, which are used to determine vehicle locations over time within a simulation (Section 4.1), and wireless propagation models, which determine the abilities of two vehicles to communication with each other (Section 4.2). The section concludes with an introduction to several existing wireless communication simulators (Section 4.3), as well as a summary of desirable simulation environment properties that will be considered when selecting (or designing) a simulation environment for use within this project (Section 4.4).

Section 5 discusses a number of different methods for controlling traffic. The first three topics covered in this section (platooning, dynamic speed limits and collision avoidance) are generally proposed as safety-related control methods. Some of these control methods, however, can also affect the overall traffic flow within a network. Section 5.4 discusses a more commonly used control method for improving traffic flow: traffic signal control and optimization. This section introduces a wide range of intelligent approaches that have been put forward in the area of traffic signal control, such as market-based systems, virtual traffic lights, neural networks and fuzzy logic. The section will be very useful to any readers who are interested in the control of traffic signals using intelligent systems. Section 5.5 covers vehicle routing within traffic networks, which is an important, albeit much less researched, control method. The section on traffic control concludes with the identification of a number of desirable characteristics that should be possessed by a system used to control traffic on a city-wide scale.

Section 6 briefly introduces the concept of trust within a VANET system. As VANET systems are a form of social system, trust can play an important role in the decision making process. An in-depth discussion of trust is not presented within this work; however, references within Section 6 will point interested readers towards a more complete discussion.

Section 7 outlines future research directions in the areas of data dissemination and traffic control using VANETs. Most of these proposed research directions are works that should be undertaken within a relatively short time period (next few years) to improve the functionality of VANET systems and help move VANET control to the mainstream.

Finally, the document concludes with Section 8. This section outlines the important points covered within the document and discusses future work to be carried out as part of this project.

2 VANETs

A VANET is a self-organized information system composed of vehicles (and possibly additional infrastructure) capable of short-range communication. There is a wide range of possible application areas of VANETs, including warning systems, collision avoidance/notification, autonomous vehicles, and traffic optimization. This section begins by introducing MANETs (Mobile Ad Hoc Networks) and identifying important differences between MANETs and VANETs (Section 2.1). Section 2.2 will explain the basic VANET architecture, including typical sensors and the use of infrastructure to augment the VANETs functionality. This section concludes with Section 2.3, which introduces the DSRC/802.11p specification, which is the most commonly used specification for communication within VANETs.

2.1 VANET vs. MANET

A MANET is a wireless network formed by mobile agents without central control or fixed infrastructure. This is different from cell networks, for example, as nodes communicate amongst themselves without the use of fixed access points (see Figure 1). Originally, the application of MANET technology was limited to military uses, but with the decreased cost and increased power of technology, the range of applications has expended significantly. A list of MANET applications, presented by Hoebeke et al. (2004), is included in Figure 2. MANETs are defined as having a number of characteristics different from those of regular networks, which require special considerations when designing MANET algorithms systems. Stojmenovic (2002) outlines a number of these characteristics, which are explained below:

- Dynamic Topology: Since nodes move throughout operation and communicate via wireless broadcasts of limited range, the topology of the

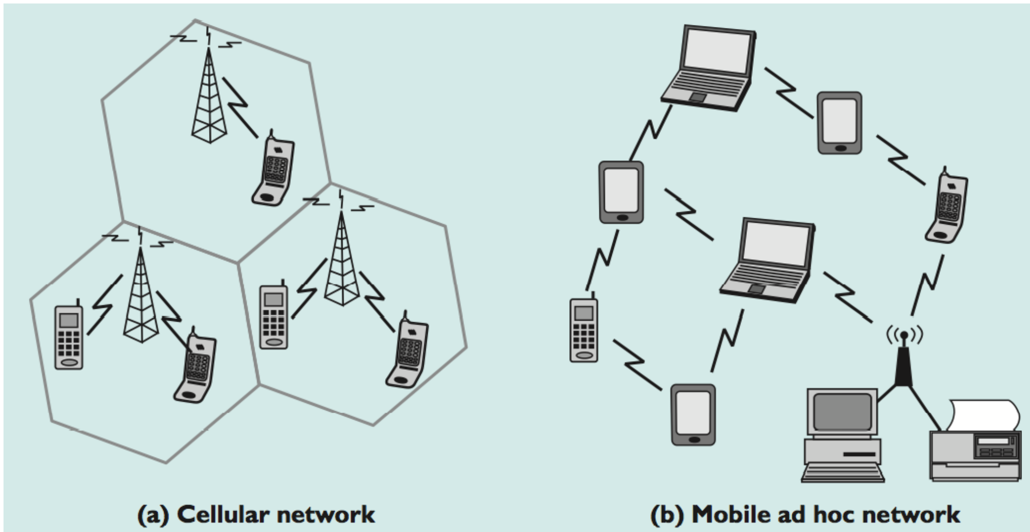


Figure 1: Cellular vs. Mobile Ad Hoc Network Architecture (Hoebeke et al., 2004)

network changes over time. Further to this, the changes in the topology can be difficult to predict, depending on the mobility patterns of the nodes. These topology changes must be considered when developing algorithms (i.e., for routing), as nodes can no longer assume another node is at a static position.

- Limited Capacity Links: As the nodes in MANETs communicate using wireless devices, the bandwidth capacity of each node is of much lower capacity than typical wired networks. In MANETs, then, decreasing the amount of data that must be sent between nodes is an important task. One of the main approaches to decreasing the data transferred is data aggregation, which uses statistical/mathematical methods to combine multiple data/observations into a single piece of data.

Application	Possible scenarios/services
Tactical networks	• Military communication and operations • Automated battlefields
Emergency services	• Search and rescue operations • Disaster recovery • Replacement of fixed infrastructure in case of environmental disasters • Policing and fire fighting • Supporting doctors and nurses in hospitals
Commercial and civilian environments	• E-commerce: electronic payments anytime and anywhere • Business: dynamic database access, mobile offices • Vehicular services: road or accident guidance, transmission of road and weather conditions, taxi cab network, inter-vehicle networks • Sports stadiums, trade fairs, shopping malls • Networks of visitors at airports
Home and enterprise networking	• Home/office wireless networking • Conferences, meeting rooms • Personal area networks (PAN), Personal networks (PN) • Networks at construction sites
Education	• Universities and campus settings • Virtual classrooms • Ad hoc communications during meetings or lectures
Entertainment	• Multi-user games • Wireless P2P networking • Outdoor Internet access • Robotic pets • Theme parks
Sensor networks	• Home applications: smart sensors and actuators embedded in consumer electronics • Body area networks (BAN) • Data tracking of environmental conditions, animal movements, chemical/biological detection
Context aware services	• Follow-on services: call-forwarding, mobile workspace • Information services: location specific services, time dependent services • Infotainment: touristic information
Coverage extension	• Extending cellular network access • Linking up with the Internet, intranets, etc.

Figure 2: Example MANET applications (Hoebeke et al., 2004)

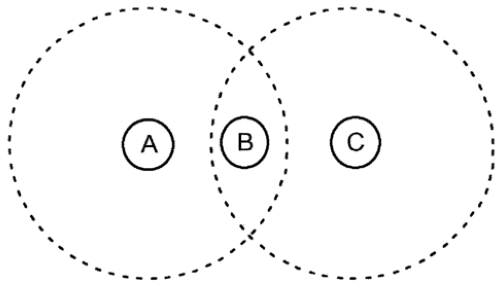


Figure 3: The hidden node problem: Nodes A and C are unaware of each other and my communicate with B at the same time (Blum et al., 2004)

- Loss Rates and Delay: MANETS experience comparatively large delays and levels of data loss. This is due in part to the use of wireless communication, but also due to the decentralized and mobile nature of the nodes. As nodes do not generally have global topology knowledge, a node A can be sending data to a neighbour B at the same time as another node C (unknown to A), with a collision resulting in data loss at node B. This problem is known as the hidden node problem, demonstrated in Figure 3. This hidden node problem can be solved using additional communication and coordination, but these efforts imply additional delay.

- Limited Power: Many MANETs use unsupervised nodes with limited battery life. For this reason, energy consumption is an important factor in the performance/lifetime of a MANET. As mentioned above, data aggregation can be an important step in decreasing the amount of energy used to send data to other nodes (although the extra computation used in aggregation also requires power).

- Physical Security: As data is transmitted using wireless communication, physical security in MANETs is severely limited. The fact that any data sent from one node to another can be intercepted by any other node within range must be taken into account when developing MANET systems.

- Infrastructure: Many MANETs are designed to be free of infrastructure. In many cases (i.e., hostile environment monitoring), building additional infrastructure is infeasible. Further to this, more complex infrastructure units may require additional power sources, which may not be available in more remote locations.

While VANETs are, in fact, a type of MANET, they possess several characteristics that must be considered when developing solutions specifically for VANETs. Singh et al. (2011) outline a number of important differences between MANETs and VANETs (many of these differences are investigated via simulation and discussed in Blum et al., 2004). Below, these differences and the effects they may have are outlined:

- Topology: Due to the high velocity of vehicles and the variability of routes, the topology in VANETs can change very quickly. Certain algorithms that are developed for MANETs, in which nodes typically move much slower, are ineffective in VANETs due to the frequency of topology change. For example, topology-based algorithms such as

AODV (Perkins and Royer, 1999) can become ineffective in VANETs as they require additional communication to re-establish routes every time the topology changes.

- Power Constraints: While MANETs are often affected by power limitations, VANETs do not suffer from the same problem. It is assumed that sensing and communication devices in a VANET are connected to a vehicle's power supply and, thus, have a sustainable power souce.

- Network Disconnection: It is expected that, within a VANET, network disconnections will occur regularly. This is especially true when considering an initial adoption phase, in which the vast majority of vehicles do not have communication abilities. VANET solutions, then, must address the possibility of disconnections and handle them in an effective way. Also, it is important when designing VANET systems to consider what level of market penetration is required to ensure satisfactory performance (something that is often not considered in VANET research works).

- Varying Network Size: MANETs are generally designed for a specific purpose, leaving the number/density of nodes to be decided by the designers. VANETs, on the other hand, can have widely varying structures. On a rural highway, for example, there may be very few cars spread out over an extremely large geographic area. In an urban setting, the number and density of vehicles will generally be much higher.

As an example, the City of Los Angeles had nearly 2.5 million vehicles registered in 2009 (The City of Los Angeles Transportation Profile, 2012); if even 10% of these vehicles were to participate in a VANET, there would be almost 250,000 nodes within the network. VANET solutions, then, must be designed with this variability in mind.

- Infrastructure: Many MANETs are designed for use in unsupervised and possibly hostile regions. For this reason, they are often designed to be completely infrastructure-free. VANETs developed for urban scenarios, however, could benefit from the use of additional infrastructure, as will be discussed in more detail in Section 2.2.2.

- Limited Mobility: Vehicles are (generally) constrained to operating on the road network, which is known and does not frequently change. For this reason, VANET solutions may take advantage of this additional knowledge to improve performance. Further to this, vehicles within a road network usually have a specified origin, destination and route which can also be used by a VANET system.

2.2 Architecture

In its simplest form, a VANET requires a set of vehicles equipped with some form of communication device. This simple form of vehicle-to-vehicle (V2V) communication, however, would not offer many benefits to the users of the system, other than the ability to communicate with nearby drivers. This section will discuss a number of augmentations to this basic architecture which could enable much more useful systems.

2.2.1 Sensors

In-vehicle sensors allow a vehicle to monitor the status of the vehicle over time. A number of sensor devices, several of which are included in the system proposed by Ozbay et al. (2007), and possible use cases are explained below.

Accelerometer

Accelerometers are capable of measuring the acceleration of an object. Within a vehicle safety application, accelerometers could be used to detect immediately detect accidents, allowing nearby vehicles and/or emergency services to be notified.

Tire Pressure Sensor

Many vehicles are now equipped with tire pressure monitors that are capable of sensing under-inflated tires. Similar to accelerometers, these sensors could be used to detect tire failure in vehicles, allowing nearby at-risk vehicles to be alerted.

Cameras

On-board cameras have become more common in recent years. These cameras could be used to detect imminent collisions or extremely local traffic state (i.e., whether another vehicle is in front of or behind the vehicle with a camera).

Airbag Deployment Sensors

Similar to accelerometers, airbag deployment sensors could be used to detect collisions.

Vehicle Speed Sensors

These sensors can provide an accurate measure of the vehicles current speed. Combined with knowledge of nearby vehicles, these sensors could predict possible collisions before they occur. Combined with an on-board computing device, these sensors could be used to record vehicle speeds over specific time intervals.

Wheel Speed Sensors

Ozbay et al. (2007) proposes the use of wheel speed sensors, which are used in anti-lock brake systems, to predict slippery/dangerous road conditions (i.e., rain or ice).

Global Positioning Systems (GPS)

Many vehicles now come equipped with GPS devices, and off-the-shelf GPS devices are readily available to people who wish to have GPS functionality in their vehicle. GPS devices are capable of identifying the vehicles current position, speed and direction. Analysis of this data as a vehicle progresses through the road network allows for estimation of travel time for specified road segments, which can be used to inform other drivers in the network. Also, the sharing of positional data between vehicles can assist in some broadcasting algorithms (i.e., the delay-based propagation protocols discussed in Section 3.1.1).

2.2.2 Infrastructure

Introducing infrastructure with communication capabilities into a VANET system allows for vehicle-to-infrastructure (V2I or V2X) communication and can add additional enhancements to the system. Unlike MANETs, the addition of infrastructure in an urban VANET is feasible and in some cases available infrastructure already exists. For example, many traffic light controllers are equipped with some form of computation equipment or connected directly to a control center. Fixed infrastructure nodes within the system can act as 'super-nodes', as they can be equipped with more powerful hardware. Some of the possible uses for these fixed infrastructure nodes within a VANET system are outlined below:

Storage

While data storage in vehicles may be very limited, infrastructure nodes could maintain a much larger database of information. With a higher storage capacity, fixed nodes could store more information for a longer period of time than a vehicle. This could allow for more advanced data analysis than is possible with a vehicle-only architecture. Furthermore, the data stored and infrastructure nodes could be used off-line for capacity planning and other analysis.

Data Aggregation

Along with increased storage, infrastructure nodes could have much higher levels of computational power. This computational power could be used to aggregate the large amount of data stored at the node. Along with aggregation, the computational power could also be used for performing traffic predictions or intelligent traffic control.

Communication

Wireless V2V communication can be extremely limiting when considering the amount of data and transmission range. Transmission range can also become a significant problem in an urban environment, where many large buildings may greatly affect the 'line-of-sight' of vehicles. Fixed nodes could be equipped with more powerful wireless devices to increase the transmission range/bandwidth, and can also be elevated to avoid interference from street-level objects. Further to this, infrastructure nodes placed at intersections generally have a clear view of each incoming road. This allows nodes placed in these locations to broadcast data more efficiently in these directions, as is done in Korkmaz et al. (2004). Finally, these nodes could be connected via a wired network, allowing for a significant improvement in transmission speed, bandwidth and transmission distance.

Device Control

Infrastructure nodes could be used to control traffic devices (i.e., traffic signals, road signs) using the data they receive from vehicles and other

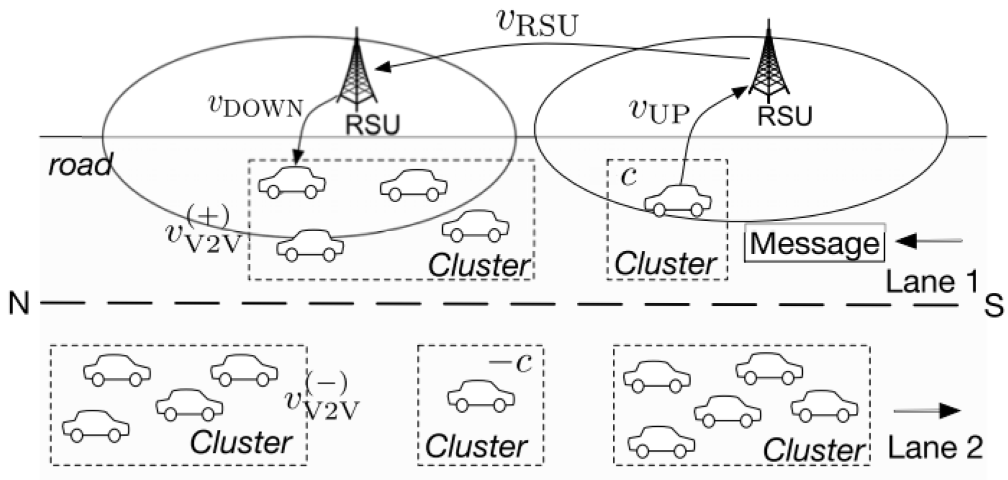


Figure 4: A hierarchical architecture with inter-infrastructure communication (Vegni and Little, 2010)

infrastructure nodes.

The use of architecture in VANETs has been proposed in a number of research works. Vegni and Little (2010) and Berlin and Anand (2012) propose hierarchical architectures, in which vehicles communicate information to road-side infrastructure units, which in turn communicate information to other infrastructure units. Figure 4 shows an example of this type of system, in which the inter-infrastructure communication is used to bridge vehicle gaps that would otherwise result in a disconnected network. This architecture could be extremely important when considering the initial adoption phase of a VANET system, as they improve the likelihood that data will be available to vehicles when required.

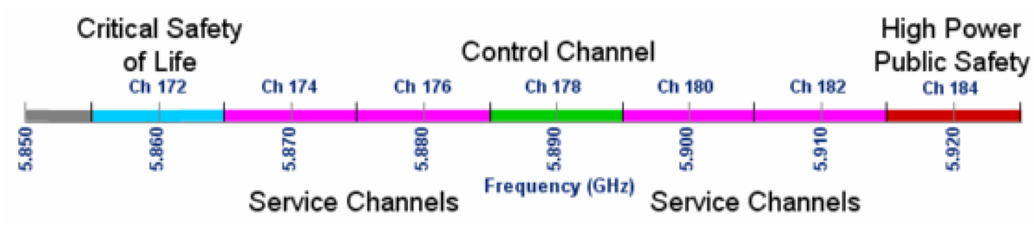


Figure 5: The division of the 7 10 MHz DSRC channels with their required usage (Jiang and Delgrossi, 2008)

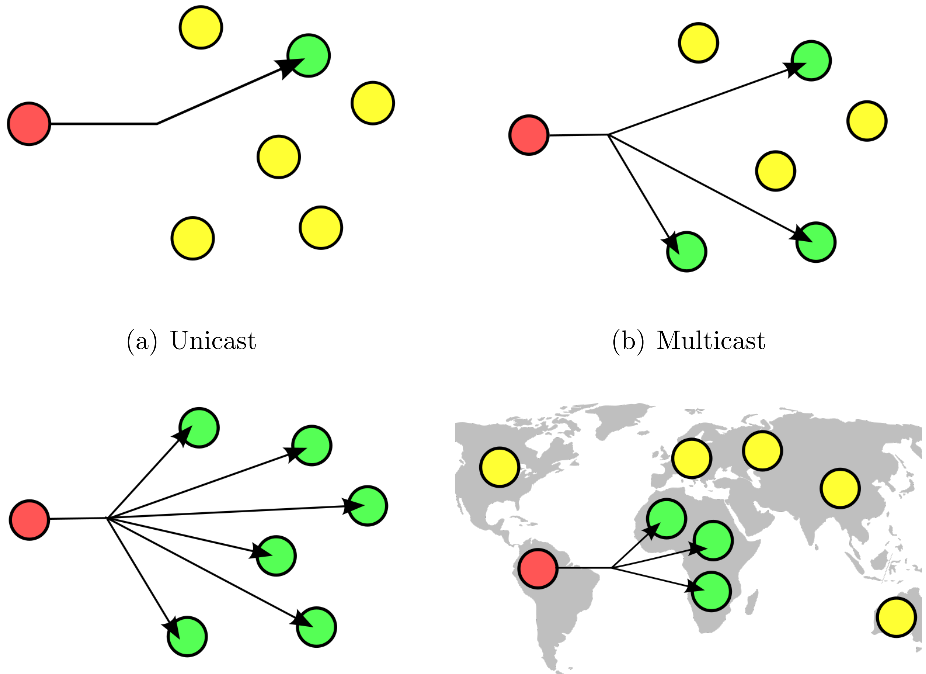
2.3 DSRC/802.11p

The vast majority of communication protocols proposed in V2I/V2X research use the Dedicated Short Range Communications (DSRC) spectrum and the 802.11p specification. Within the USA, the DSRC spectrum (allocated by the United States Federal Communication Commission) is a free but licensed spectrum, which is divided into 7 channels of 10MHz each, as shown in Figure 5 (Jiang and Delgrossi, 2008). Zeadally et al. (2010) state that the DSRC standard in the U.S. operates with a data transmission rate of 3-27Mbits/second and a maximum communication range of 1000 meters. Zeadally et al. (2010) also state that spectra have been allocated for DSRC use in Europe (20MHz bandwidth, 0.5Mbits/second transmission rate) and Japan (80Mhz bandwidth, 1-4Mbits/set transmission rate). While the DSRC spectrum is free to use, the fact that it is also a licensed spectrum requires devices to operate in a specific way and conform to the specified standards. For example, comfort applications (i.e., video streaming) may only use the four channels specified as service channels, while only emergency applications may use the critical safety of life channel. The original DSRC specifications was absorbed by an IEEE task group in 2004 and work began on an amendment to the 802.11 specification (Uzcategui and Acosta-Marum, 2009). This amendment was called IEEE 802.11p and eventually led to the development of the wireless access in vehicular environments (WAVE) standard, which aims to support the VANET-specific requirements outlined by Uzcategui and Acosta-Marum (2009), shown below.

- Longer range of operation (up to 1000m)
- High vehicle speeds
- High multipath environment
- Multiple overlapping ad hoc networks
- Special beacon frame

3 Data Dissemination

The goal of data dissemination in a VANET is to spread data that may be useful to other vehicles in the network. Important characteristics to consider when analysing a VANET dissemination protocol are how much data is sent, how fast does the data travel through the network and what is the likelihood that a vehicle receives the data. There are four broad approaches to message dissemination that can be applied in a VANET, depending on



(a) Unicast (b) Multicast

(c) Broadcast (d) Geocast

Figure 6: Examples of four main dissemination approaches (Wikipedia - Routing, 2012)

the requirements of the system: unicast, multicast, broadcast and geocast. Figure 6 shows an example of these four dissemination approaches, while a brief explanation of each is given below:

Broadcast

Broadcasting attempts to propagate information to all vehicles within the network. This paper focuses specifically on broadcasting, as it is the most common dissemination technique used in VANET algorithms. Broadcasting is important for most of the VANET applications discussed here, as data regarding any part of the network may be useful to vehicles at any other point in the network.

Geocast

Geocasting attempts to send information to all vehicles within a specific geographic region. Within a VANET, geocasting can be useful when data needs to be sent to only vehicles within a specific region. For example, if a vehicle detects that a small section of road is covered in ice, only vehicles nearby that may cross this section of road need to be notified. A geocast, then, could be used to send this information to vehicles on the same road that are currently 500m or less from the patch of ice. A review of geocasting algorithms is available in Allal and Boudjit (2012).

Unicast

Unicasting involves sending a message to a single node in a network. For most VANET purposes, aside from personal/comfort applications, unicast is typically not a useful tool. For this reason, unicast algorithms are not discussed here. A detailed review of unicast algorithms, however, is available in Bernsen and Manivannan (2009).

Multicast

Multicasting attempts to send data to a subset of all nodes within a network. As with unicast, a broadcast (and possibly geocast) approach generally makes more sense than a multicast approach for use in a

VANET. Junhai et al. (2009) presents a review of available multicast algorithms.

The remainder of this section uses the taxonomy presented by Panichpapiboon and Pattara-atikom (2012) to present a number of algorithms used for broadcasting in VANETs. Section 3.1 presents multi-hop algorithms, which move data geographically by transferring the data between vehicles. Multi-hop dissemination can be further divided into delay-based (Section 3.1.1) and probabilistic-based (Section 3.1.2) algorithms. While Panichpapiboon and Pattara-atikom also includes network coding approaches to dissemination, these approaches are not covered here as these typically rely on neighbourhood knowledge that is difficult to maintain VANETs due to frequent topology changes. Section 3.2 presents single-hop dissemination algorithms (also known as store-and-forward), which rely on the movement of vehicles throughout the network to disperse packets geographically. Single-hop dissemination can be further subdivided into fixed interval (Section 3.2.1) and adaptive interval (Section 3.2.2) approaches. The section concludes with Section 3.3, which presents a summary of the key characteristics required of a successful dissemination algorithm.

3.1 Multi-Hop

This section discusses two classes of multi-hop data dissemination algorithms: delay-based and probabilistic. As mentioned briefly in the previous section, multi-hop algorithms transport data throughout the network by sending the data to multiple vehicles in the direction of propagation. In a dense network, this allows the data to move very quickly (faster than the speed of traffic) as it can be sent to the next hop (which may be hundreds of meters away) in a very short time period. In a sparse network, however, the use of a simple multi-hop dissemination algorithm can result in a large amount of data loss or delay. This is because multi-hop algorithms rely on nearby vehicles in the direction of propagation to forward the data packet. If no nearby vehicles are capable of forwarding the packet, it may be lost. The first class of multi-hop algorithm covered in this section is delay-based algorithms, which generally attempt to choose a relay node to maximize the distance between the source and the relay. The second class of multi-hop algorithms discussed choose relay nodes based on probabilities. As with delay-based algorithms, these probabilities are usually assigned such that nodes further from the source have the highest probability of becoming a relay node.

3.1.1 Delay-Based

The Urban Multi-hop Broadcast (UMB) protocol (Korkmaz et al., 2004) is one of the earliest and most frequently referenced protocols for delay-based broadcasting within VANETs. UMB divides the space in the direction of propagation into a number of slots, and attempts to determine the furthest vehicle away from the source of the broadcast source to select as a relay node. Figure 7 demonstrates the basic functionality of the UMB algorithm.

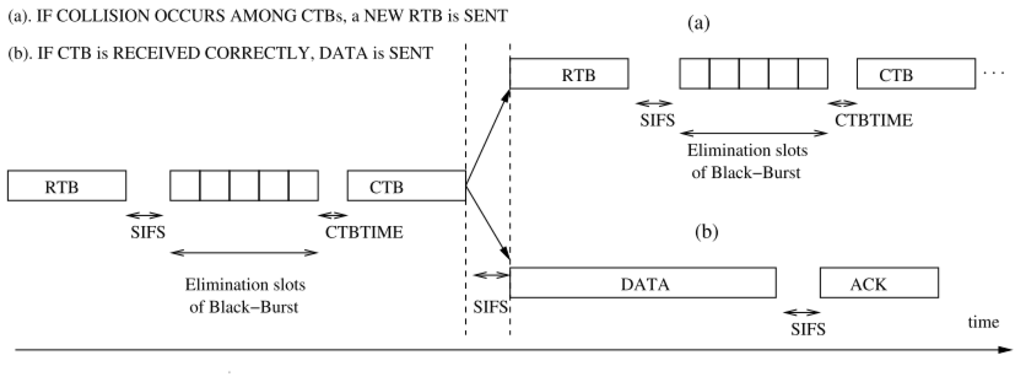


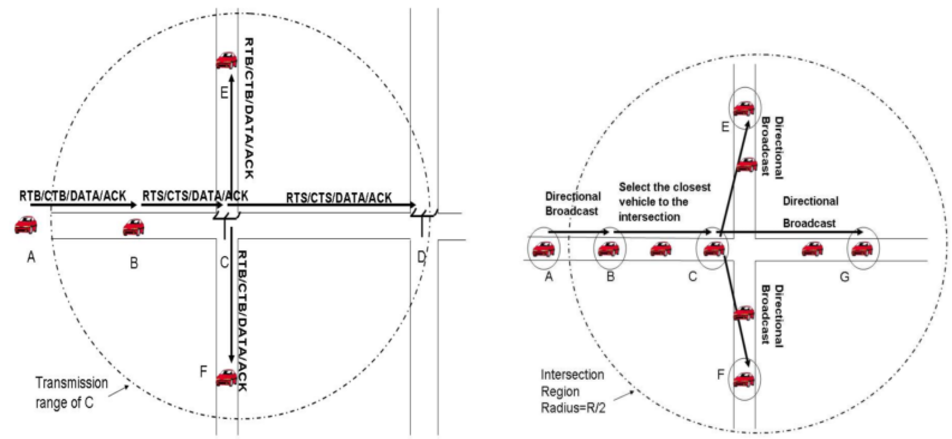
Figure 7: Broadcasting sequence in UMB Korkmaz et al. (2004)

The process is started by some source node, which broadcasts a request-to-broadcast (RTB) packet that is received by all neighbouring nodes within the wireless broadcast range. Each node that receives the RTB packet determines its distance, d, from the source node and can then determine the slot that it should belong to using Equation 1, where r is the broadcast range and N_{max} is the total number of slots.

$$\left\lfloor \frac{d}{r} \times N_{max} \right\rfloor \tag{1}$$

By multiplying the slot number determined using Equation 1 by a time interval, each node that received the RTB packet can determine how long of a 'black-burst' signal (a channel jamming signal) they should send, with nodes in further slots sending the black-burst for a longer time interval. Each node then broadcasts a black-burst signal for the calculated length of time, and upon completion, checks to see if the channel is still busy (i.e., another node is still sending a black-burst). If the channel is clear upon completion of the black-burst, the node can identify itself as being in the furthest slot and will reply with a clear-to-broadcast (CTB) packet. However, since each node calculates its black-burst length based only on the slot it is in, if two or more nodes are in the furthest slot, they will broadcast the CTB packet at the same time and a collision will result. The source node detects this scenario, as the CTB packet it receives will be corrupted by the collision, and begins another round that uses the same process but divides the furthest slot from the previous round into a number of smaller slots (this is the situation shown in Figure 7(a)). Eventually, only a single node will be present in the furthest receiving slot, at which point the source selects this node as the relay node and begins to send the data to that node (Figure 7(b)). UMB also uses ACK (acknowledgement) packets, which are sent from a relay node to the source to inform the source that the data was successfully received. This allows the source to re-send the data packet in the event that the relay node did not receive it.

To further improve broadcasting in urban networks, where obstructions may prevent wireless signals from travelling effectively to cross-roads, the UMB protocol also relies on intersection repeaters as shown in Figure 8(a). Since a clear line of sight is available to all connecting roads at an intersection, these repeaters are capable of sending packets in directions that might otherwise be difficult to reach due to obstructions. As can be seen in Figure 8(a), the intersection repeater C is capable of forwarding data to nodes E



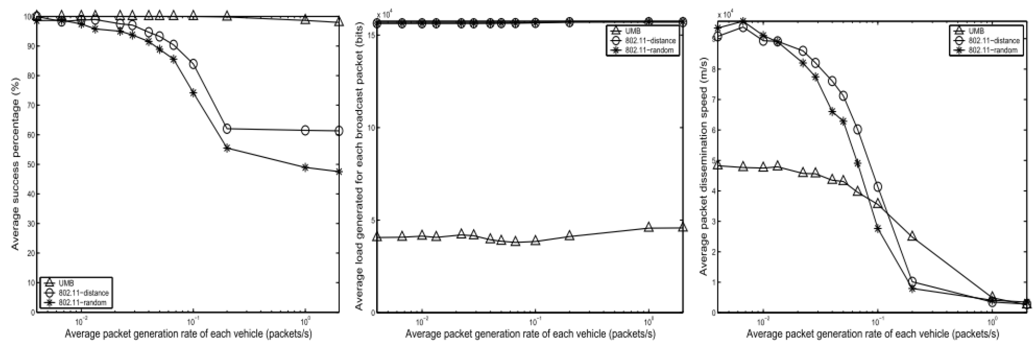
(a) Using intersection repeater

(b) Selecting best vehicle for intersection re-broadcast

Figure 8: Elimination of infrastructure requirements in UMB Korkmaz et al. (2004)

and F, which would not be able to receive data from node B itself. Also, it is assumed that these intersection repeater devices could be equipped with more powerful communication devices, which in this case allows C to also communicate data directly to the next intersection D. This allows data to move quickly forward (or backward) through the network, even if vehicles aren't available to act as relay nodes. In Korkmaz et al. (2006), the use of intersection repeaters is eliminated and, instead, the relay node is selected in a way that both maximizes its distance from the source and minimizes its distance from an intersection, which should allow it to effectively broadcast information to nodes on cross-roads (see Figure 8(b) for an example).

To investigate the ability of the UMB algorithm to successfully broad-



(a) Success percentage

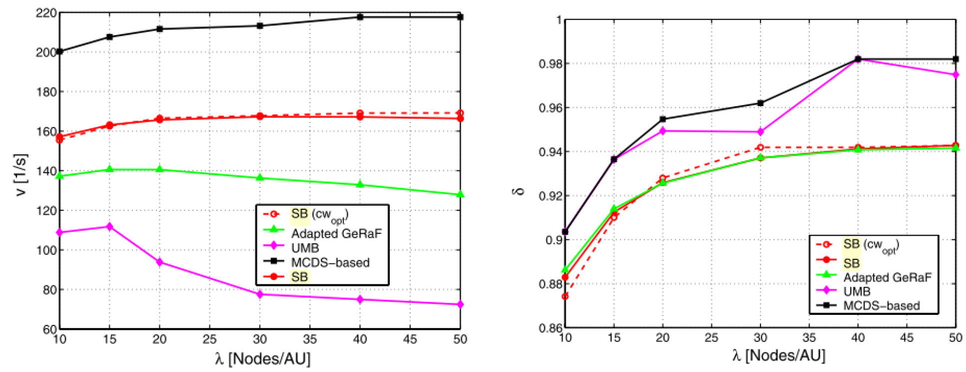
(b) Normalized average load per broadcast packet

(c) Packet dissemination speed

Figure 9: Performance of UMB, 802.11-random and 802.11-distance (Korkmaz et al., 2004)

cast information through a network, the authors compared the protocol to two simple protocols: 802.11-distance, where nodes wait a period of time inversely proportional to their distance from the source before re-broadcasting, and 802.11-random, where nodes wait a random time before re-broadcasting. In both of these protocols, a node does not re-broadcast if it receives a packet a second time (indicating that another node has relayed the packet already). Figure 9 shows the comparison of UMB to these two simple protocols using three metrics on a 4x4 grid network. The first metric (Figure 9(a)), success percentage, represents the average percentage of vehicles that receive a broadcast packet within the simulation. Within Figure 9(a), the UMB protocol maintains a near-perfect success percentage for all packet generation rates, while the basic protocols experience a deteriorating success percentage as the packet generation rate increases. This can be attributed to the fact that the UMB protocol greatly reduces packet loss due to collision through the selection of a single relay node and the use of ACK packets. Figure 9(b) shows the normalized load per broadcast, which is the amount of data that is sent within the network to broadcast a data packet to all nodes. UMB obviously maintains a much lower packet load than the other two protocols, which indicates a much more efficient use of the limited channel bandwidth. As with the success percentage, this efficiency is realized because UMB uses a relatively small amount of data to determine a relay node that covers the largest distance, resulting in the large data packet being broadcast fewer times. Figure 9(c) demonstrates one of the main problems with the UMB protocol: speed of dissemination. For most packet generation rates in this scenario, data broadcast via UMB travels at approximately half the velocity of data broadcast via the simple protocols. This decrease in data velocity when using UMB is due to the prolonged black-burst phase which increases in length as the quality of the node as a relay (the distance from the source) increases.

This problem is addressed in the Smart Broadcast (SB) algorithm of Fasolo et al. (2006), where the assignment of delay is reversed in relation to UMB's. As with UMB, space is divided into segments/slots, with each segment being assigned a time interval. Each node that receives an RTB packet determines which segment it belongs to and selects a random time within that time interval to respond with a CTB packet. This CTB packet signals to other nodes, who are still in the delay phase, that another node will be the relay candidate. This approach offers another advantage over UMB, in



(a) (b)

Figure 10: Performance comparison of SB with other protocols (Fasolo et al., 2006)

that nodes within the same segment randomly select a delay amount from within the time interval. In this way, the likelihood that more than one node will reply with a CTB packet at the same time is decreased. The difference between UMB and SB is clear from the results presented by Fasolo et al. (2006), included here in Figure 10. Figure 10(a) compares the propagation speeds of UMB, SB, Geographic Random Forwarding (GeRaF) and Minimum Connected Dominating Set (MCDS). SB maintains a significantly higher propagation speed when compared to UMB, even though UMB generally covers a significantly higher distance per hop, as is evident from Figure 10(b). Again, this is because the SB protocol aims to quickly identify a node that is *one of* the furthest away from the source, while the UMB protocol spends significantly more time determining the *exact* furthest node. The GeRaF and MCDS algorithms are designed for use in stationary networks and make assumptions about network topology that do not apply to VANETs, so their performance is not discussed here.

Several other works have proposed solutions similar to UMB and SB. Osafune et al. (2006) proposed the Multi-Hop Vehicular Broadcast (MHVB) algorithm, which uses the same principle for selecting relay nodes. A node using MHVB, however, dynamically adjusts how often it attempts to broadcast its data based on local vehicle density estimates (higher traffic density results in less broadcasting), which are achieved using two separate approaches. First, periodic 'Hello' messages are broadcast, which require neighbours to respond so the node can estimate the number of neighbours it has. Second, a node can also estimate the surrounding traffic density based on the current travel speed (i.e., slower speeds are caused by high traffic volumes). Results presented in Osafune et al. (2006) show a slight increase in the delivery success ratio when broadcast rate is set inversely proportional to the estimated traffic volume. Li et al. (2007) proposed Efficient Directional Broadcast (EDB), an algorithm similar to SB, where each node selects a waiting time proportional to $1 - \frac{D}{R}$ (where D is the distance from the source and R is the transmission range) before replying with a CTB packet. To evaluate the performance of the EDB protocol, Li et al. applied the protocol to a system based on real-world vehicle traces generated from taxis in the city of Shanghai. Similar results to those found in Fasolo et al. (2006) were realized in this realistic scenario. In each work mentioned previously, it is assumed that each vehicle in the network has an identical broadcast range and an equally sensitive receiver. Amoroso et al. (2011) acknowledge that this is unlikely to be true in a real-world implementation and propose a solution that aims to identify the node which has the farthest span (the node which can disseminate the packet the furthest distance), and not just the farthest distance from the source. It was shown that using this protocol may result in increased dissemination speed when compared to a protocol similar to those discussed previously.

While the delay-based methods proposed thus far have shown to be effective at propagating data through the network, they rely on the assumption that the network is connected. When a disconnection occurs (i.e., due to an empty space of road larger than the broadcast range), any data being disseminated through the above methods will not be able to traverse the gap. This problem is highlighted in Figure 11, which shows that under lower vehicle densities, the distance a message can travel using a simple multi-hop broadcasting protocol is quite short. To address this problem, several hybrid approaches have been proposed that combine multi-hop delay-based broadcasting with a store-and-forward approach (see Section 3.2 for details on store-and-forward dissemination). In essence, these approaches attempt to broadcast data using a standard multi-hop approach, but they also rely on vehicles travelling in the opposite direction to receive the data and carry it over distances that cannot be covered by the transmission range of a single vehicle. Schwartz et al. (2010) developed the Simple and Robust Dissemination (SRD) protocol to operate in this way and compared the performance under several scenarios to that of a simple multi-hop protocol based on the

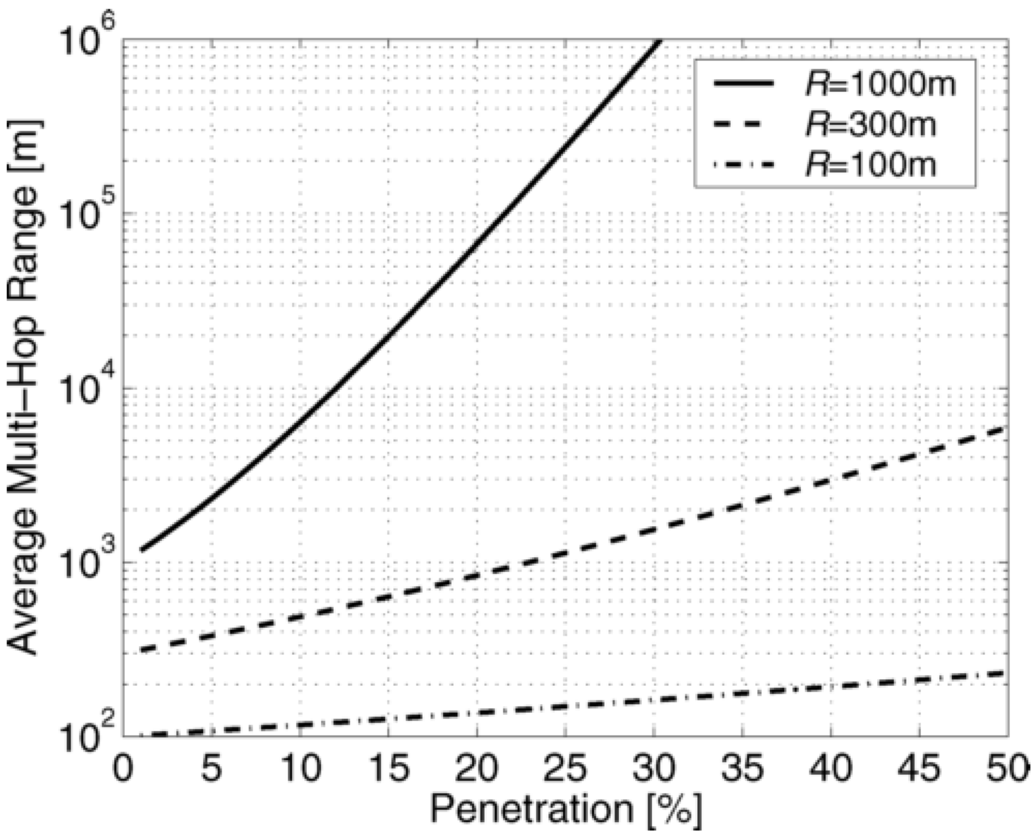


Figure 11: Expected value of multi-hop range for given penetration rates and transmission ranges (Wischhof et al., 2005)

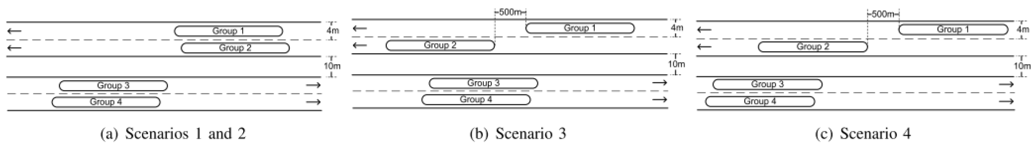


Figure 12: Scenarios investigated by Schwartz et al. (2010)

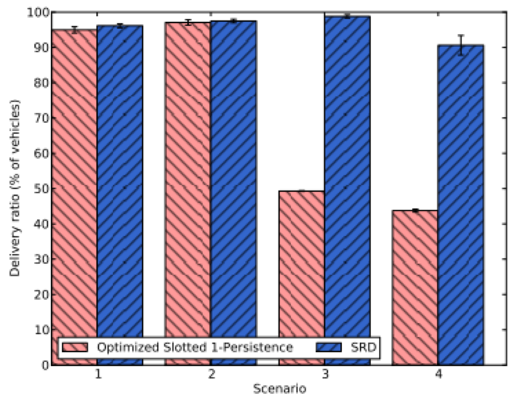


Figure 13: Data delivery ratios of SRD vs. Slotted 1-Persistence over four scenarios (Schwartz et al., 2010)

Slotted 1-Persistence protocol (Wisitpongphan et al., 2007). Figure 12 shows the example scenarios used by Schwartz et al., with Scenarios 1 and 2 representing cases in which the network is connected and Scenarios 3 and 4 representing situations in which the gap between two groups of vehicles is too large to be covered by a single wireless transmission. A comparison of the ratio of successfully delivered messages using the two protocols over the four scenarios is included in Figure 13. These results show that in the connected scenario, the SRD and Slotted 1-Persistence protocols result in nearly identical performance. In the disconnected network scenarios, however, the store-and-forward approach included within the SRD protocol results in a significantly higher percentage of messages being delivered successfully when compared to the Slotted 1-Persistence protocol. Li et al. (2011) developed a similar protocol, capable of switching between multi-hop and store-and-forward modes, which produced similar results when compared to simple multi-hop broadcasting.

3.1.2 Probabilistic-Based

Wisitpongphan et al. (2007) developed several broadcasting protocols for VANETs, one of which was the probabilistic Weighted p-Persistence algorithm. Within the Weighted p-Persistence algorithm, a node j receiving a packet from a source node i will rebroadcast the packet with a probability $p_{ij} = \frac{D_{ij}}{R}$, where D_{ij} is the distance between i and j, and R is the transmission range. In this way, vehicles that are located further away from the source have an increased probability of forwarding the packet, as shown in Figure 14. This protocol works similar to a delay-based algorithm, but it avoids the additional delay associated with determining which nodes are furthest from the source. Wisitpongphan et al. compared the performance of the Weighted p-Persistence algorithm to a number of other approaches, including a delay-based approach called Slotted 1-Persistence. Figure 15(a) compares the packet loss ratio using these protocols over various traffic densities. This figure shows that the Weighted p-Persistence approach results in a significantly higher packet loss ratio at high node densities, when compared to the

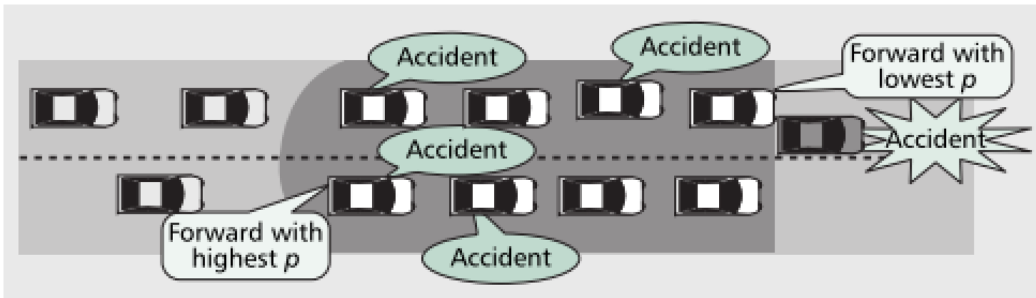


Figure 14: Example of probability assignment using Weighted p-Persistence algorithm (Wisitpongphan et al., 2007)

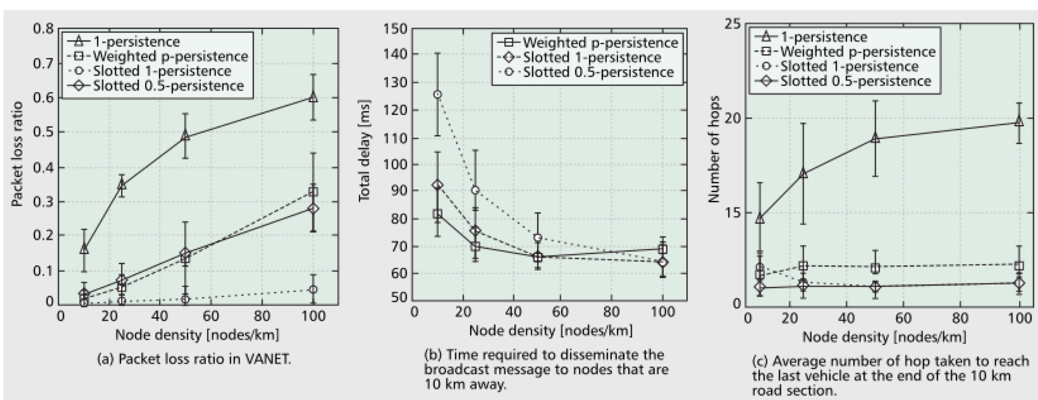


Figure 15: Comparison of Weighted p-Persistence to the delay-based Slotted 1-persistence (Wisitpongphan et al., 2007)

Slotted 1-Persistence protocol. As the difference in packet loss ratio increases with node density, the problem is most likely caused by an inability of the Weighted p-Persistence scheme to effectively limit contention within the network. Figures 15(b)/(c) compare packet delay and average number of hops, respectively, and show that both the probabilistic and delay-based approach achieve similar performance over these two metrics. A similar probabilistic broadcasting approach, with mathematical analysis based on percolation theory, is presented in Slavik and Mahgoub (2010). Another simple probabilistic approach, in which the probability of broadcasting is determined by the number of 2-hop neighbours that can be reached, is proposed by Alshaer and Horlait (2005). It should be noted, however, that maintaining 2-hop neighbourhood information can be difficult/expensive within networks with rapidly changing topologies (i.e., VANETs).

Irresponsible Forwarding (IF) is a probabilistic approach that varies the rebroadcast probability based on traffic density, proposed by Panichpapiboon and Ferrari (2008) and Busanelli et al. (2009). Using IF, the probability p that a vehicle will rebroadcast a packet it receives is calculated using Equation 2, where d is the distance from the source, z is the source's transmission range, p_s is the estimated traffic density (in vehicles/meter), and c is a parameter used to 'shape' the distribution.

$$p = e^{-\frac{p_s(z-d)}{c}} \tag{2}$$

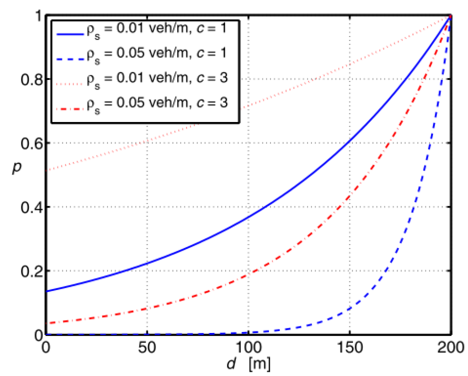


Figure 16: Examble probability distributions used for IF (Panichpapiboon and Ferrari, 2008)

Figure 16 demonstrates how the probability of rebroadcast increases with distance from the source and decreases as traffic density increases. This allows the system to adapt to changes in vehicle density and/or transmission range, avoiding the increased network contention seen in the simple probabilistic method from Wisitpongphan et al. (2007). Panichpapiboon and Ferrari investigated the behaviour of the proposed algorithm through simulation over a range of vehicle densities and transmission ranges. Figure 17 plots the expected number of rebroadcast packets ($E[M]$) at different hops for varying products of vehicle density (p_s) and transmission range (z). This figure demonstrates how the IF algorithm adapts to maintain a low number of rebroadcasts, even with widely varying vehicle densities/broadcast ranges. Furthermore, the expected number of rebroadcasts decreases as the number of hops increases, which can be beneficial as data becomes less relevant as it travels further away from its initial source.

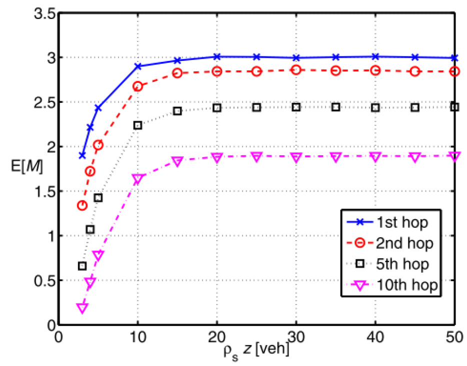


Figure 17: Expected number of rebroadcasts relative to vehicle density ($p_s z$) (Panichpapiboon and Ferrari, 2008)

Like delay-based broadcasting, probabilistic broadcasting itself is incapable of addressing network disconnections. Wegener et al. (2007) propose a novel method for dealing with network disconnection within their Autocast broadcasting algorithm. Newly observed data is disseminated within Autocast using a probabilistic method. To address network disconnection, nodes store received data packets in memory and periodically broadcast a list of packet IDs contained in their memory. Neighbouring nodes, which receive these lists, can then identify which packets their neighbour has missed and send these packets in a periodic broadcast. The time interval between the broadcast of these extra packets is adapted based on estimated traffic density, which is an example of an adaptive single-hop broadcasting approach (see Section 3.2.2 for details). Nekovee and Bogason (2007) use a different method for addressing the problem of intermittently connected networks. A node that receives a packet waits a random amount of time (bounded by a small time interval) before deciding whether to rebroadcast or not. When this time interval has expired, the node calculates its probability to rebroadcast the packet using Equation 3, where N_F is the number of detected rebroadcasts in the direction of propagation and N_B is the number of detected broadcasts in the opposite direction.

$$P = \begin{cases} 1 & \text{if } N_F = 0, \\ 1 - exp\left(-\alpha \frac{N_F}{N_F + N_B}\right) & otherwise \end{cases} \tag{3}$$

In this way, the node will be guaranteed to rebroadcast a packet if no node ahead in the direction of propagation has rebroadcast, while it will have a low probability of broadcasting if it has received a proportionally high number of rebroadcasts from the forward direction.

3.2 Single-Hop

Using a single-hop dissemination algorithm (also known as store-and-forward), a vehicle uses itself to transport data geographically (store) and intermittently broadcasts a subset of its data to nearby vehicles (forward). This technique allows vehicles to easily deal with intermittent connectivity, as the data is stored in memory and rebroadcast at certain time intervals. Two classes of single-hop dissemination algorithms are presented here: fixed interval, in which vehicles broadcast a subset of their data at specific time intervals, and adaptive interval, which involve varying the rate of broadcast to achieve better system performance.

3.2.1 Fixed Interval

TrafficInfo, developed by Zhong et al. (2008), is an example of a single-hop fixed interval broadcasting algorithm. Vehicles using TrafficInfo have a fixed time interval which specifies how often they should broadcast their data to neighbouring vehicles. If each vehicle attempted to broadcast its entire database, however, an infeasible amount of data would have to be exchanged between vehicles in the network. For this reason, TrafficInfo uses a data selection policy that attempts to send only the k most relevant reports from a vehicle's database. The k most relevant reports are selected based on 'supply' and 'demand', as defined below.

Demand

The demand of a report R at location p and time t represents the relevance of that report to a vehicle currently at location p and time t. Demand is calculated as $\frac{1}{c+g}$, where c is the age of the report and g is the shortest path from p to the mid-point of the segment the report pertains to. In this way, reports that are newly generated and pertain to nearby road sections will be given the highest demand.

Supply

The supply of a report R is an estimate of the percentage of vehicles in the network that have received the report. While this is a difficult value to estimate, Zhong et al. proposed a machine learning approach that uses 'supply indicators' (i.e., age) to predict the supply of a report. When a new report is received, the machine learning algorithm uses the report as a new training example (the supply indicators and whether the report is new to the vehicle are known). In this way, the system within each vehicle is constantly learning how to estimate the supply of reports stored in the database.

With supply and demand defined for a report, the algorithm assigns a rank to each report using the equation $rank = demand \times (1 - supply)$. The highest ranked reports can be identified based on this calculation and intermittently broadcast to neighbouring vehicles. To evaluate the performance of TrafficInfo, Zhong et al. created a new metric named Difference in Knowledge (DIK). If all road segments within the network are labelled $s_1, s_2, ..., s_n$, a vehicle O's DIK can be calculated using Equation 4, where g is the free-flow travel time along the shortest path to s_k from O's current position, $T(s_k)$ is the actual travel time of s_k, and $T(O, s_k)$ is the estimated travel time of s_k stored in O's database.

$$DIK(O) = \sum_{k=1}^{n} \left(\frac{1}{g} \left| T(s_k) - T(O, s_k) \right| \right) \tag{4}$$

TrafficInfo was compared through simulation to a system in which no information is passed between vehicles (NonInfo), as well as to a system using Grassroots (a dissemination approach that does not use store-and-forward).

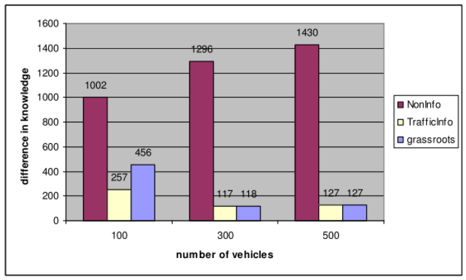


Figure 18: DIK measures for three algorithms over varying number of vehicles (Zhong et al., 2008)

Figure 18 shows the difference in the DIK metric for the three approaches. Both Grassroots and TrafficInfo have significantly less DIK, compared to the NonInfo case, over all three vehicle densities. However, at low density, TrafficInfo manages to perform much better than Grassroots, demonstrating the importance of the store-and-forward technique in VANETs that may suffer from intermittent connectivity.

Algorithms similar to TrafficInfo have been proposed by other researchers. Nadeem et al. (2004) proposed TrafficView, which like TrafficInfo, aims to decrease the amount of data that must be transferred between vehicles. Instead of decreasing the number of reports, however, TrafficView uses data aggregation to merge similar reports (i.e., similar locations/observations). Kitani et al. (2008) present another fixed interval single-hop broadcasting system, which uses city buses as storage points for data. An advantage to this approach is that the buses can store a large amount of data and move in predictable patterns. A disadvantage of using buses is that they tend

Examples for Provocations

Event	Intention
Reception of information being out-of-date	Transmitting vehicle needs updated information
Reception of packet with significantly different new information	Favor propagation of changes
Reception of information from vehicle with large distance	Favor large hops in propagation
Indication (e.g. by lower layers) of excessive bandwidth	Decrease delay of information propagation

Examples for Mollifications

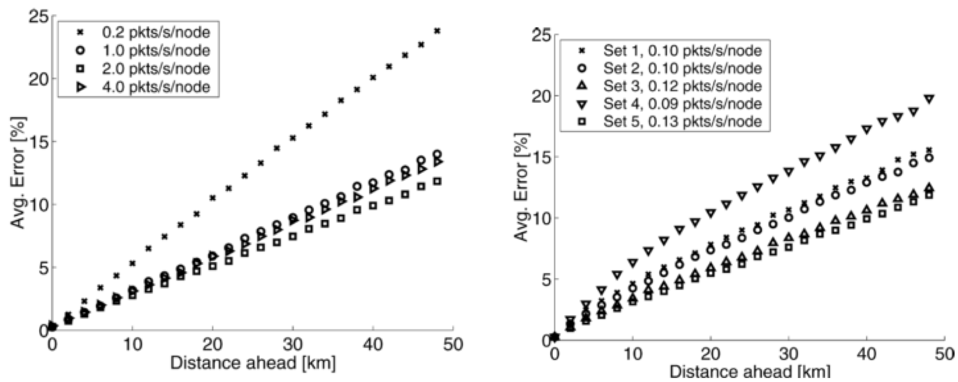
Event	Intention
Reception of similar/more up-to-date information from nearby	Avoid redundant transmissions
Indication that number of received reports exceeds threshold	Limit maximum used bandwidth

Figure 19: Examples of mollification and provocation events (Wischhof et al., 2005)

to travel slower than most other traffic, which slows the progress of data through the network.

3.2.2 Adaptive Interval

SODAD (Wischhof et al., 2005) was one of the first adaptive interval single-hop dissemination algorithms proposed. Using SODAD, a vehicle begins with a default transmission interval, which is small enough to ensure a passing vehicle would receive a message while it was in the transmission range (based on an assumed maximum relative velocity). The transmission interval is then extended via mollification events or decreased via provocation events (examples of mollification/provocation events are presented in Figure 19). For example, when a vehicle receives a new piece of data that differs significantly from the data stored in its database, its transmission interval will decrease, resulting in faster dissemination of the new data. Conversely,



(a) Fixed Interval

(b) Adaptive Interval

Figure 20: Information error rates using (a) fixed interval and (b) adaptive interval broadcasting (Wischhof et al., 2005)

if a vehicle receives several pieces of similar data, the transmission interval will increase, as this data does not need to be propagated through the network. To investigate the performance of the proposed algorithm, Wischhof et al. compared the error rate of available information using a fixed interval dissemination scheme (Figure 20(a)) and the SODAD system (Figure 20(b)). Figure 20(a) shows that when broadcasting 0.2 packets per second per node (pkts/s/node), an error in information of nearly 25% is present 50km away from the source, while using a fixed interval of 2.0pkts/s/node decreases this error rate to just over 10%. However, increasing the interval further, to 4.0pkts/s/node causes the error to increase once again, demonstrating that the collisions under higher broadcasting rates are detrimental to the overall system performance. The adaptive interval tests (using five different parameter sets) shown in Figure 20(b) show a similar error level to the best performing fixed interval tests. However, the overall pkts/s/node that are sent using the adaptive approach are 10-20 times less than that of the fixed approach, demonstrating a much more efficient use of network bandwidth. SODAD is used as the data dissemination protocol within the proposed SOTIS (Wischhof et al., 2004) system. Resulting dissemination speeds from within simulation of the SODAD/SOTIS system showed data dissemination speeds of 100km/h were achievable with small adoption rates (2%-10%).

He et al. (2010) also presented a protocol in which vehicles begin with a default time interval and modify that interval based on feedback received from the system. The method for adjusting time intervals is based on the additive increase/multiplicative decrease (AIMD) algorithm, which is most popular for its use in TCP (Transmission Control Protocol) congestion control. In this case, the rate at which data is broadcast is increased until a 'congestion event' is detected, at which time the broadcast rate is halved. The Collision Ratio Control Protocol (CRCP), proposed by Fujiki et al. (2007), uses a similar approach, in which vehicles modify their broadcast interval to maintain a specified collision ratio (it is shown that an optimal collision ratio is approximately 60%). Using CRCP, a node decreases its broadcast interval by one second until it estimates the local collision ratio to be higher than a threshold, at which point it doubles the interval length. Another method for interval modification, proposed by Xu and Barth (2006), uses vehicle speed as an estimate of traffic congestion. Slower-moving vehicles, then, are deemed to be in an area with high traffic volumes, resulting in an increased broadcast interval. However, this can be misleading as vehicles may travel slow for various reasons, and vehicles in areas of medium or high congestion may maintain high velocities (i.e., in a highway scenario).

Eichler et al. (2006) and Adler et al. (2006) propose three types of context that may be defined for a message in a traffic information system and used to adapt the data broadcast interval.

Message Context

This includes message-related information such as age, time since last broadcast, and time since last reception.

Vehicle Context

This includes information such as route choice, route flexibility, neighbourhood information, and vehicle speed.

Information Context

This includes information such as distance from source, type of information, and rate of change information.

By quantifying this information, an overall estimate of the message's utility to other vehicles in the network can be produced. This utility estimate can then be used to prioritize different information and modify the rates at which they are broadcast to neighbours. Eichler et al. suggests three ways in which data can be selected for broadcasting:

1. Data with a utility above a specified threshold will be broadcast. Broadcasting of data with utility below this threshold will be suppressed until

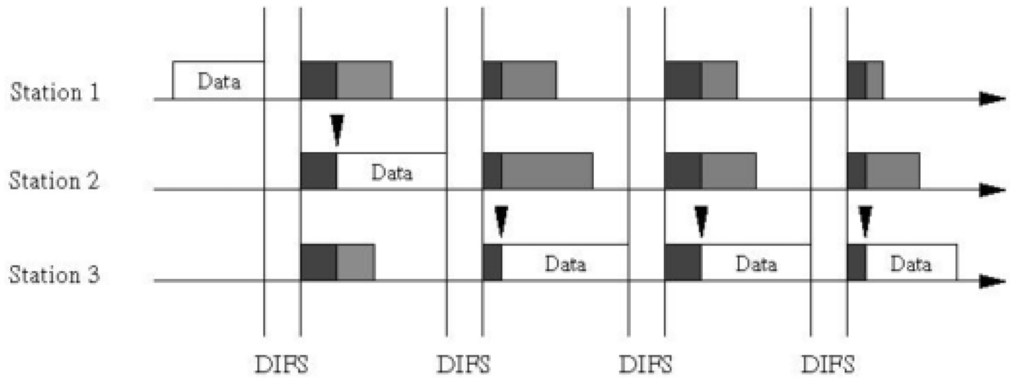


Figure 21: Contention between three network nodes (stations) (Adler et al., 2006)

the utility changes.

2. A vehicle can select a subset of all data to broadcast based on the set of utility values (similar to the method used in Zhong et al. (2008)).

3. Vehicles may contend for the shared wireless medium, with vehicles possessing data of highest utility being prioritized over vehicles with less useful data.

An example of how vehicles may compete for limited broadcasting bandwidth is included in Figure 21. Similar to a delay-based propagation approach (where the distance from the source affects delay), vehicles may decrease their broadcasting interval based on the estimated utility of the data they possess. Vehicles with data judged to be the most important, then, will request access to the channel before vehicles with less important data. The effect that this data contextualization may have on message transmissions is

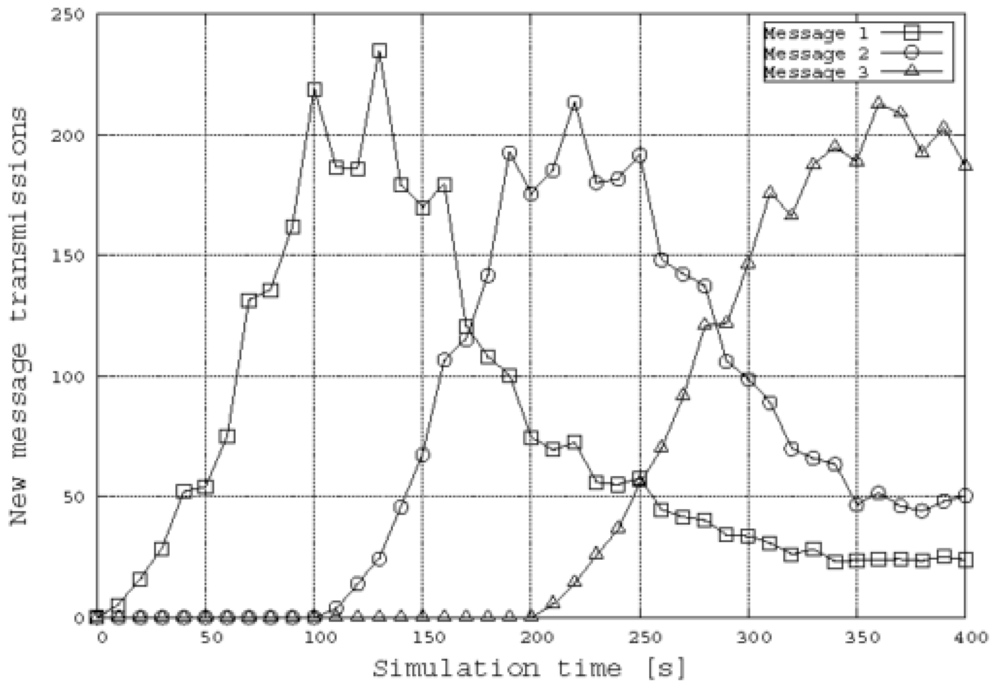


Figure 22: New message transmissions for three hazards over time (Adler et al., 2006)

shown in Figure 22. Within a simulation, a new hazard is introduced into the traffic network at simulation times of 0, 100, and 200 seconds. When the initial hazard is introduced, message transmissions pertaining to this hazard begin to increase, as it is the only hazard that has been observed within the system. When a second, more recent, hazard is introduced after 100 simulation seconds, the number of message transmissions regarding the first hazard begins to decrease. This is because the contextual information (i.e., age) indicated that messages pertaining to the second hazard are more useful at this point in time. The same effect can be seen when the third hazard is introduced after 200 simulation seconds. A similar approach to rate selection using contextual data is presented by Shankar et al. (2008). Shankar et al., however, uses only environmental context indicators (e.g., vehicle speed, location, weather), which may affect the overall deliverability of a packet.

3.3 Summary

Based on the discussion of the available dissemination algorithms, a number of desired algorithm properties can be identified. These algorithm characteristics are outlined below:

Sensitive to Volume

A dissemination algorithm should be sensitive to the current traffic volume and capable of adapting to fit the current environment. This may involve switching between a multi-hop (in dense traffic) algorithm and a store-and-forward approach (in sparse traffic). This may also involve adapting the amount of data that is shared by a vehicle in the network.

Selecting/Aggregating Data

The algorithm should be capable of decreasing the total amount of data that needs to be broadcast using data aggregation or selection. This can involve fusing a number of reports into a single report, based on various available indicators (i.e., time, location), as well as selecting the most relevant data to share with nearby vehicles.

Localized

The algorithm should be capable of operating without complex neighbourhood knowledge, as this is difficult to maintain in a highly dynamic VANET environment.

Dissemination Rate

As observed traffic information tends to be extremely time-sensitive, the algorithm should be capable of spreading information through the network at a high rate of speed.

Unreliable Data

The algorithm should be capable of identifying and eliminating unreliable data within the network. This may include data maliciously inserted into the network, data produced by malfunctioning sensors, or data that is no longer relevant. The system may achieve this goal through outlier analysis or some type of trust model (Section 6).

Table 1 lists several of the important algorithms discussed in this section, and summarizes general properties of each algorithm.

4 Simulators

Evaluating a VANET solution in the real-world requires a large number of vehicles equipped with communication devices, and possibly additional infras-

Algorithm	VS	ND	DA	L	DR	UD	OH
Multi-Hop Algorithms							
UMB (Korkmaz et al., 2004)	No	No	No	Yes	Slow	No	High
SB (Fasolo et al., 2006)	No	No	No	Yes	Fast	No	Med
MHVB (Osafune et al., 2006)	Yes	No	No	Yes	Med	No	High
SRD(Schwartz et al., 2010)	Yes	Yes	No	Yes	Med	Some	Med
Weighted p-Persistence (Wisitpongphan et al., 2007)	No	No	No	Yes	Fast	No	Low
IF (Panichpapiboon and Ferrari, 2008)	No	No	No	Yes	Fast	No	Low
Autocast (Wegener et al., 2007)	No	Yes	No	Yes	Fast	No	Med
Single-Hop Algorithms							
TrafficInfo (Zhong et al., 2008)	No	Yes	Yes	Yes	Med	Some	Low
TrafficView (Nadeem et al., 2004)	No	Yes	Yes	Yes	Med	No	Low
SODAD (Wischhof et al., 2005)	Yes	Yes	Yes	Yes	Varies	No	Low
AIMD (He et al., 2010)	Yes	Yes	No	Yes	Med	No	Low
Context-aware (Eichler et al., 2006)	Yes	Yes	Yes	Yes	Varies	Some	Low

Table 1: Summary of important algorithms presented in this section

VS: Sensitive to varying traffic volume
ND: Handles network disconnections
DA: Uses data aggregation/selection
L: Uses localized information
DR: Qualitative dissemination rate estimate
UD: Capable of identifying unreliable data
OH: Qualitative estimate of overhead

tructure. For this reason, the vast majority of VANET research is performed using simulations, where realistic scenarios can be modelled and investigated at a low cost. To demonstrate the feasibility of proposed solutions, they must be analysed within a realistic model. When developing realistic traffic and communication models, there are two important trade-offs that must be considered:

Available Information

The amount of information available (or obtainable) can effect the overall realism of a simulation environment. For example, real-world vehicle traces are capable of producing the most realistic mobility models; however, these traces are also the least available, as they require GPS data from trips made by a large number of drivers. Artificially generated vehicle traces, on the other hand, can be generated automatically by a computer program. While these artificial traces may not be as accurate as real-world recordings, it is much easier to produce a large set of scenarios.

Computation Cost

Increasing the realism of a simulation generally requires an increase in computation time, as more factors must be considered/evaluated. For example, the free-space wireless propagation model requires a small amount of computation, as it only takes the relative location of possible receiving nodes into account. The two-ray ground reflection model, however, takes additional signal reflection into account and can provide more accurate results. This increase in accuracy, though, comes at an increased computation cost. Other proposed models (i.e., Martinez et al., 2010) consider buildings which may affect signal propagation.

This section outlines three important components in any VANET simulation: a mobility model, a radio propagation model, and a simulator. Section 4.1 outlines different mobility models that may be used within a VANET simulation, including the random waypoint model (Section 4.1.1), real-world vehicle traces (Section 4.1.2), artificially generated traces (Section 4.1.3) and bidirectionally coupled simulation (Section 4.1.4). Section 4.2 will briefly describe various available radio propagation models, while Section 4.3 will discuss some popular wireless network simulators including ns-2, JiST/SWANS and GloMoSim.

4.1 Mobility Models

The role of a mobility model within a network simulation is to specify how, where, and when agents/nodes move through the network. Sommer and Dressler (2008) defined four classes of mobility models: random movement, real-world traces, artificial mobility traces, and bidirectionally coupled simulators. Each of these classes will be discussed in more detail below. A summary of the benefits, drawbacks, and network simulator availability for each class is shown in Figure 23.

Mobility Model Class	Integrated Framework Support	Benefits	Drawbacks
Random Movement	Virtually All	⊕ Straightforward, intuitive ⊕ Readily available	⊖ Imprecise
Real-World Traces	GloMoSim, QualNet, OPNET, ns-2, Shawn, JiST/SWANS, OMNeT++/INET Framework	⊕ Most realistic node movement ⊕ Re-usable traces	⊖ Costly and time consuming ⊖ No free parameterization
Artificial Mobility Traces	GloMoSim, QualNet, OPNET, ns-2, Shawn, JiST/SWANS, OMNeT++/INET Framework	⊕ Realistic node movement ⊕ Free parameterization ⊕ Re-usable traces	⊖ No feedback on driver behavior
Bidirectionally Coupled Simulators	Ongoing efforts for: ns-2, Shawn, JiST/SWANS, OMNeT++/INET Framework	⊕ Realistic node movement ⊕ Free parameterization ⊕ Feedback on driver behavior	⊖ No re-usable traces

Figure 23: Summary of benefits/drawbacks of four classes of mobility models (Sommer and Dressler, 2008)

4.1.1 Random Waypoint

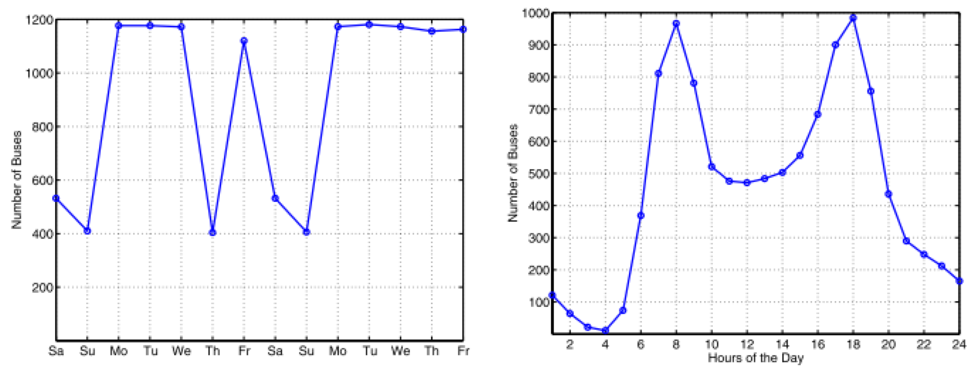
Using a random waypoint model (Bettstetter et al., 2004), start and end points for each agent are selected randomly within the defined simulation area. Agents then move from their start point to their end point in a completely unconstrained manner. For vehicular simulation, this approach is obviously poor as it does not enforce standard vehicular movement constraints (i.e., staying on roads). The main advantage to random waypoint modelling is the simplicity with which scenarios can be created. However, the lack of constraint on movement makes them unsuitable for use in VANET research.

A slight improvement on the completely unconstrained approach is the Manhattan Grid model. The Manhattan Grid model defines roads, which can accommodate one-way or two-way traffic and form a grid network. Generally vehicles are inserted at random places within the network and make probabilistic decisions on which way to proceed at each intersection (i.e., 50% straight, 25% left, 25% right). While this is a slightly more realistic approach, it still fails to model the dynamics of traffic, such as increased volumes of vehicles departing from specific areas.

4.1.2 Real-World Traces

Real-world vehicle traces are developed from data recorded from a vehicle moving through a real traffic network (typically, GPS data). Real-world traces offer the most realistic specification of movement, as they are developed from actual routes taken by real vehicles. However, it is also difficult to collect a large number of traces, which results in them being used less frequently than other methods.

As mentioned above, the main difficulty in applying real-world vehicle traces is the effort involved in obtaining a large number and diversity of traces (i.e., different days, times, origins, destinations). Jetcheva et al. (2003) used the movements of buses within the Seattle, Washington area King County Metro bus system to generate real-world vehicle traces. The King County bus system makes real-time bus locations available online, which allowed Jetcheva et al. to mine bus traces over time. This approach allows a large number of vehicle traces to be created over time. As can be seen in Figure 24(a), hundreds of buses were monitored everyday. In addition to this, many buses run routes repeatedly, so each bus would generate a number of traces. Another advantage offered by tracking buses is that they can be representative of overall travel demands. For example, Figure 24(b) shows peak number



(a) Daily Measurement (b) Hourly Measurement

Figure 24: Number of buses generating reports by (a) day and (b) hour (Jetcheva et al., 2003)

of buses during the morning and evening rush hours, which is representative of the overall traffic state during those time periods. Furthermore, locations of high demand (i.e., the downtown area in the morning) are typically serviced by a larger number of buses. It may be possible, then, to extrapolate overall traffic volumes from this bus data (this is not done by Jetcheva et al., however).

To evaluate the performance of a proposed broadcasting protocol, Li et al. (2007) generated realistic vehicle traces from data generated by taxis within Shanghai. The 4200 taxis reported their position, speed, direction and vehicle ID approximately every 40 seconds. This allowed the system receiving the data to produce snapshots of the current traffic state, as shown in Figure 25. Since the data is spaced in approximately 40 second intervals, the route information in between each report had to be inferred algorithmically. To

Figure 25: A snapshot generated by the system used in Li et al. (2007)

do this, Li et al. used a simple algorithm in which a vehicle chooses, at each intersection, the direction in which the distance to the next reported position is minimized.

4.1.3 Artificial Traces

The lack of realism present in random waypoint mobility and the challenge of collecting real-world vehicle traces has led to researchers creating artificial means of generating realistic vehicle traces.

Fiore et al. (2007) proposed VanetMobiSim, an open source vehicle trace generator, which was designed to work with communications simulators. VanetMobiSim is capable of producing example road networks from user-defined graphs, Geographic Data Files (GDFs), TIGER (Tiger Products -

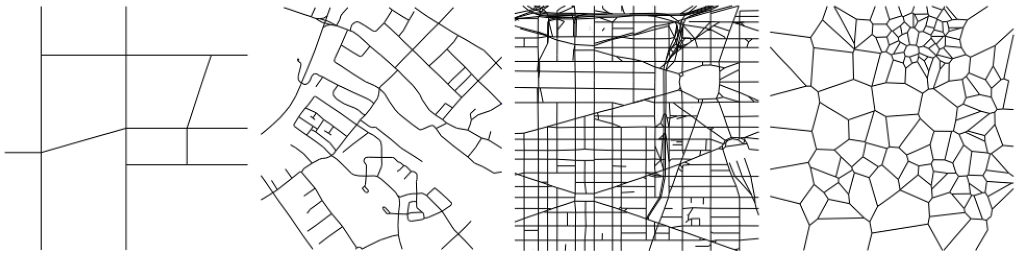

(a) User-defined topology (b) GDF map topology (c) TIGER map topology (d) Clustered Voronoi

Figure 26: Four sample networks generated by VanetMobiSim (Fiore et al., 2007)

Geopgraphy - US Censur Bureau, 2012) maps, or randomly using a clustered Voronoi graph. Examples of generated networks using each method are included in Figure 26. To deal with the non-randomness of traffic patterns, Fiore et al. apply a trip generating module, used to identify points of interest within the network, and a path computation model, used to compute routes for vehicles moving between two points of interest. After defining routes for vehicles, the vehicles actual behaviour (accelerations, decelerations, lane changes, etc.) is generated. Once routes and behaviour have been established, vehicle traces can be computed and used within a communications simulator.

Karnadi et al. (2007) developed a tool called MOVE (MObility model generator for VEhicular networks), which works alongside the SUMO traffic simulator. As with the work of Fiore et al. (2007), MOVE offers several methods for automatically/easily generating maps from available sources. MOVE also offers a route editor, which allows the user to define vehicle flows

	Area 1	Area 2	Area 3	Area 4	Area 5
Area 1	0	15	3	9	9
Area 2	18	0	27	5	8
Area 3	1	21	0	6	10
Area 4	9	7	7	0	13
Area 5	7	6	11	15	0

Table 2: An example origin-destination matrix for 5 areas within a network

(the number of vehicles introduced to the network) over sections of roads, as well as turning probabilities for intersections within the network. Once flows and routes are defined, the SUMO traffic simulator is capable of generating a set of routes for vehicles and creating vehicle trace information based on a simulation of vehicles using these routes.

Similar to MOVE, VERGILIUS (Giordano et al., 2010) is another set of tools designed to allow users to efficiently produce realistic vehicle traces. Figure 27 shows the general information flow used by VERGILIUS to move from TIGER database files to network simulation. As with the previous methods, VERGILIUS first requires a map of the desired area to be created, which is done using the TIGER database. To define routes, VERGILIUS can use traffic flows and turning probabilities or an origin-destination (OD) matrix. Origin-destination matrices are used to define the number of vehicles which leave one point/area (the matrix row) and travel to another point/area (the matrix column). Table 2 shows an example OD matrix from McKenney (2011). As with MOVE, a traffic simulator is then used to simulate vehicles

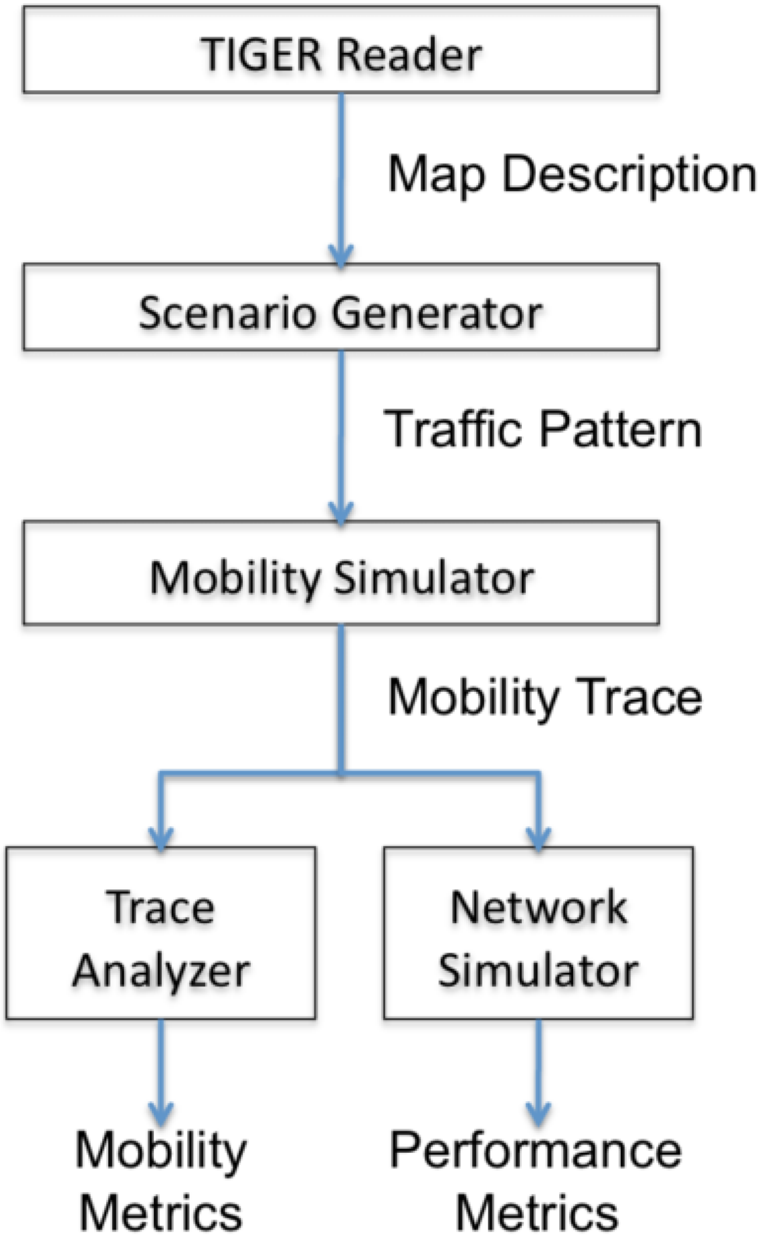


Figure 27: Information flow within the VERGILIUS system (Giordano et al., 2010)

using the specific set of routes and generate vehicle traces for use in a network simulator.

4.1.4 Bidirectionally Coupled Simulations

While artificially created realistic traffic scenarios are useful, they fail to capture the dynamic driver behaviour that may be necessary when investigating VANET solutions (i.e., dynamic routing based on traffic congestion). The goal of bidirectionally coupled simulation is to share information from the traffic simulator (i.e., vehicle traces) with the network simulator, and vice versa (i.e., information to modify behaviour). The TraNS project (TraNS - Traffic and Network Simulation Environment, 2012) integrated the SUMO traffic simulator and the ns-2 network simulator, allowing realistic VANET scenarios to be created/evaluated. This project, however, is no longer in development and, therefore, does not support the latest versions of SUMO or ns-2. As mentioned in Sommer and Dressler (2008), there is continuing efforts to couple common traffic and network simulators to enable realistic VANET scenarios to be easily created and investigated.

4.2 Wireless Propagation Models

Within a network simulator, the wireless propagation model determines how (or if) signals sent from one device are received by another device. The choice of radio propagation model, then, is an important step when attempting to build a realistic VANET simulation. The free-space model, for example, assumes there is no reflection, diffraction or scattering present in the system. This is an easy model to implement/evaluate; however, it is also one of the least realistic models available. The two-ray ground reflection propagation model considers the effect of reflection caused by the ground within a simulation. This offers a more realistic solution at the increased cost of computation. Some network simulators (i.e., ns-2) include other considerations, such as 'shadowing', which includes noise to account for environmental influences such as buildings (Martinez et al., 2009a). This can be useful when simulating a VANET system, as it allows for more realism when considering vehicles operating in an urban setting.

4.3 Available Simulators

When selecting a network simulator, there are a number of important questions that must be answered, such as:

- What radio propagation models are supported?
- What communication protocols are supported?
- What mobility models are available?
- Can the simulator be coupled with a traffic simulator?
- What support/documentation is available?
- What specifications are included (i.e., 802.11p)?

This section discusses three of the most popular network simulators (ns-2, JiTS/SWANS, GloMoSim), with a focus on answering these important questions.

4.3.1 ns-2

One of the most widely used network simulators is ns-2 (ns-2, 2012). ns-2 is a multi-platform simulator which was originally developed for the simulation of wired networks, but was extended by the Monarch research group at Carnegie Mellon University to include node mobility and wireless radio propagation models (Martinez et al., 2009b). ns-2 includes several radio propagation models, including the free space, two-ray ground reflection, and shadowing models. Due to the number of people using ns-2, a number of tools have been created to generate node traces for use within the simulator (e.g., the MOVE project discussed in Section 4.1.3, as well as many traffic simulators), which allows for realistic mobility models to be used. Further to this, the TraNS project (TraNS - Traffic and Network Simulation Environment, 2012) was developed to couple the functionality of ns-2 with the SUMO traffic simulator. As 802.11p is the main communication protocol used in VANET communications research, it is important for any network simulator being used to support simulation of 802.11p. Fortunately, previous works have used ns-2 for simulation of 802.11p networks (Gukhool and Cherkaoui, 2008), and scripts are available online for generating 802.11p parameters for use in ns-2 (NS2 Simulation with 802.11p, 2012). Finally, ns-2 includes detailed online

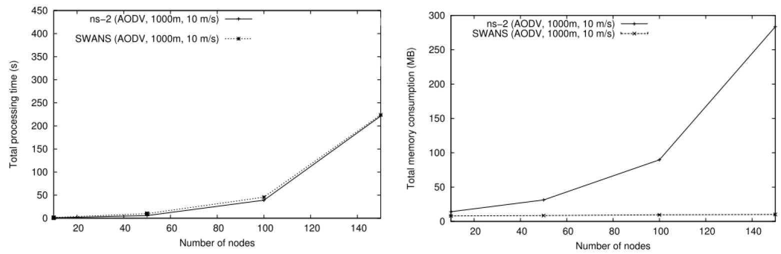


Figure 28: Comparison of (a) processing time and (b) memory required in ns-2 and SWANS simulations for varying network sizes (Kargl and Schoch, 2007)

documentation and example scripts to assist new developers, along with a mailing list for soliciting help from the community of ns-2 developers/users.

One of the main criticisms of ns-2 is a lack of scalability in relation to the number of networked nodes within a simulation (i.e., Perkins et al., 2001, found a network of 100 nodes was the largest network that could be simulated in a feasible amount of time). However, Naoumov and Gross (2003) proposed a method which greatly increased the speed with which ns-2 could simulate large networks. Figure 28, from Kargl and Schoch (2007), compares the total processing time and memory consumption using ns-2 and JiST/SWANS for networks of varying size. While very little difference is seen in processing time, the memory requirements for ns-2 are significantly higher than those of JiST-SWANS. For modern computers, 300 megabytes of memory may not be of concern; however, the rate at which the memory requirements of ns-2 are increasing may indicate an inability to scale to large networks.

4.3.2 JiST/SWANS

JiST (Java in Simulation Time) is a discrete event simulation engine, which runs on a Java virtual machine (JiST/SWANS, 2012). SWANS is a wireless network simulator built on top of JiST, which was designed to provide a scalable and fast network simulation engine. While ns-2 is still the most popular network simulator, several recent works have used the JiST/SWANS framework for network simulation (i.e., Zhong et al., 2008). The signal propagation algorithms used within SWANS offer an improvement over the 'naive' approaches previously implemented within ns-2, which can lead to improved simulation times (JiST/SWANS, 2012). As with ns-2, SWANS offers several radio propagation models (free-space, two-ray), as well as signal fading algorithms. As SWANS is not yet as popular as ns-2, there are fewer methods for generating vehicle traces for use in network simulation. This may make it difficult to include realistic vehicle traces within the network simulation. Also, as SWANS has not been around as long as ns-2, the amount of documentation, support and examples available for SWANS is relatively small when compared to that of ns-2. This lack of initial information may make it difficult for a new user to learn to use SWANS effectively in a short period of time.

4.3.3 GloMoSim

GloMoSim (GloMoSim, 2012) is a wired and wireless network simulator built upon the simulation language Parsec (UCLA Parsec Programming Language, 2012). Of the three simulators discussed in this section, GloMoSim seems to be the least used and least maintained project. GloMoSim offers free-space and two-ray propagation models, along with random waypoint, random drunken, and trace-based mobility models. Documentation and tutorials on the use of GloMoSim are limited, and it is also recommended that users have some knowledge of the Parsec language before using GloMoSim.

4.4 Summary

From the information provided in the preceding section, a number of important characteristics can be identified, which should be considered when selecting or designing a VANET simulation environment. These important characteristics are described below, and will be considered when selecting (or designing) a simulation environment for use within this project.

Mobility Model

Ideally, the simulation environment should include tools capable of automatically generating realistic mobility models for vehicles. This includes using realistic maps, varying traffic volumes/flows, and complex traffic movements (i.e., lane changes). If such a tool is not present in a simulation environment, the software should at least provide an interface to allow for existing tools to be used for creation of mobility models.

Radio Model

When considering an urban environment, the effect of buildings and other physical structures should be considered by the radio model. Exact calculations taking these effects into account, however, may carry extremely high computational costs. For this reason, a simulation environment capable of providing accurate approximations (i.e., through probabilistic analysis) of these effects is desirable.

Simulator Integration

As discussed in Section 4.1.4, the integration of traffic and communication simulations allows for the dynamic effects of communication on traffic flows (and vice versa) to be investigated. The overall system dynamics within a VANET solution are by far the most useful, and for this reason, it is important that any VANET simulation environment used for this project is capable of coupling the simulation of both traffic and communication. In the case that a satisfactory coupling solution does not exist, methods for integrating a communication simulation with a popular traffic simulator may have to be investigated.

Learning Curve

When considering the adoption of any new software environment, it is important to consider the time and effort that may be required to learn the required skills. For this reason, any simulation environment used within this project should have detailed documentation, available support resources, and example implementations for general problems.

5 Traffic Control

One of the main applications proposed for VANET technology is within the area of traffic control. Currently, without VANET systems, traffic control relies on inaccurate volume measurements such as those produced by induction sensors or infrequent surveying of traffic volumes. Using VANETs, control systems will have real-time access to accurate and useful data such as travelling speeds, vehicle locations, vehicle counts, and collision information. Access to this information will extend traffic control beyond the simple optimization of phase and cycle lengths at intersections, to real-time management of traffic on a local and network-wide scale. This section will discuss a number of proposed VANET traffic control applications, beginning with localized vehicle control applications (platoons, variable speed limits, and collision avoidance/detection) and ending with network-wide control strategies (traffic signals and vehicle routing).

5.1 Platoons

A vehicle platoon can be considered a 'train' of vehicles that are attached via virtual links (as opposed to physical links). Each platoon consists of a leader (the car in front) and a number of followers. The main challenge in a platoon is controlling the follower vehicles in a way that maintains a close following distance while avoiding collisions. One of the obvious advantages to platooning is the fuel savings that can be realized by following vehicles

	Fuel consumption [%]	Average Velocity [km/h]
Lead Truck	100	69.89
Time Gap 1	92.3	69.90
Time Gap 3	93.6	69.90
Time Gap 5	95.3	69.89

Figure 29: Simulated fuel savings when using a platoon (Alam et al., 2010)

due to the aerodynamic benefits of travelling closely behind another vehicle. For example, Alam et al. (2010) predicted a fuel-savings of almost 8% when considering platooning among heavy-duty vehicles, as shown in Figure 29. Another advantage of platooning is that, when properly controlled, platoons can also reduce the probability of collisions between vehicles. Finally, platoons have also been proposed for improving flow at intersections, as they can eliminate/reduce the 'rubber-band' effect observed when a queue of vehicles begins to move.

Several research works have considered vehicle platoons from a number of perspectives. El-Zaher et al. (2011) propose a physics-inspired model of platoons, in which a following vehicle is virtually connected to the vehicle in front by two springs, as in Figure 30. Using this model, calculations can be made to determine the forces on both springs, which in turn can be transformed into the required longitudinal (speed) and lateral (left/right) changes. This type of approach, in which each following vehicle calculates its own adjustments, is known as local platooning. In contrast to local platooning, Kaku et al. (2012) presents a centralized approach to platoon adjustment, in

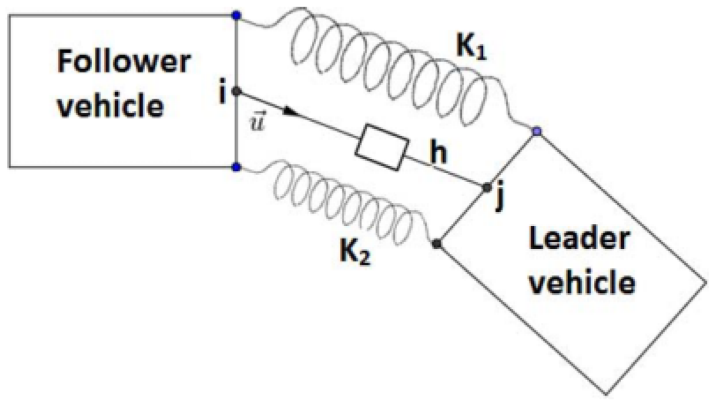
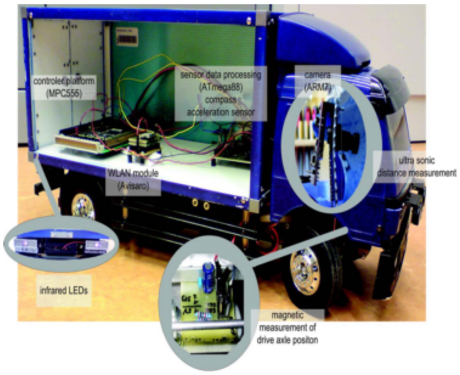


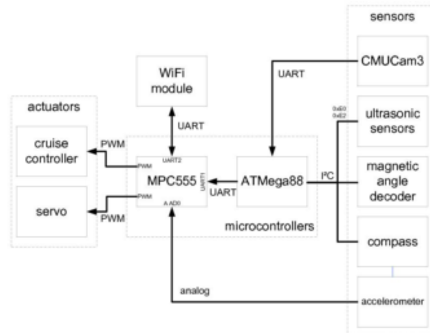
Figure 30: Physics-based platoon example, during a turn (El-Zaher et al., 2011)

which a central entity (i.e., the lead vehicle) computes the optimal behaviour for the entire platoon. Using a centralized approach, however, requires communication between the central controller (lead vehicle) and all other vehicles in the platoon. In the case of a large platoon, this communication distance may be too large and in other cases the communication required may be prohibitive.

To investigate the real-world implementation of vehicle platoons, Diab et al. (2010) used 1:14 scale trucks to create a platoon of four vehicles. Each of the four vehicles were equipped with several sensors, wireless communication devices, small computation devices, and actuators for controlling speed/ wheel direction (see Figure 31(a)). As can be seen in Figure 31(b), the sensor measurements are passed to the microcontrollers, which can then calculate adjustments to be made using the actuators. The microcontrollers can also



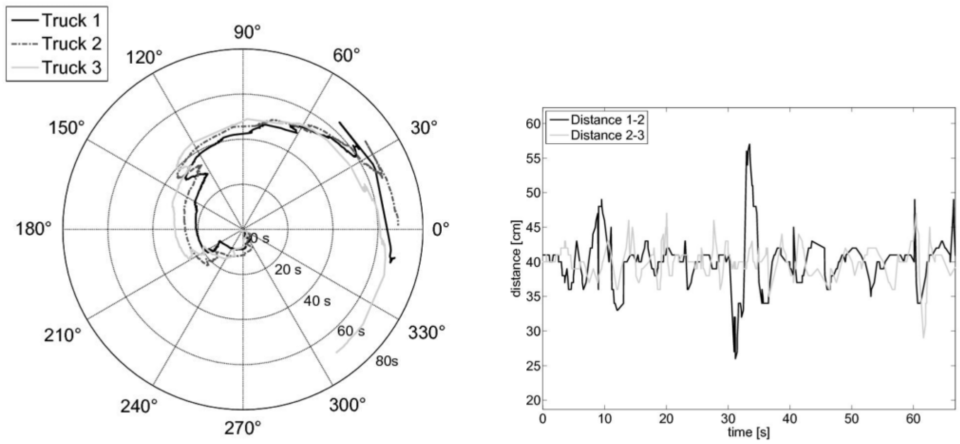
(a) 1:14 Scale Truck



(b) Vehicle Architecture

Figure 31: The model vehicle and architecture used by Diab et al. (2010)

communicate to nearby vehicles using the wireless device. One of the most important contributions of this work is the proposal and implementation of sensors for achieving successful vehicle platooning. The ultrasonic sensors, for example, are used for computing distance between vehicles for longitudinal adjustment. To perform lateral adjustment, a small camera is used to capture an image to be analysed by the microcontrollers. The performance of the proposed system was evaluated by driving the platoon around a circular track. Figures 32(a) and 32(b) show the angular direction of the trucks over time and the distance between the trucks over time respectively. These figures show that the platoon control system was capable of maintaining a safe but satisfactory following distance between the vehicles.



(a) Truck Angular Direction (b) Distances Between Trucks

Figure 32: Results of the 1:14 scale platoon on a circular track (Diab et al., 2010)

5.2 Dynamic Speed Limits

Variable speed limits are a traffic control approach that have been proposed to both decrease the frequency/severity of collisions and improve traffic flow in congested areas. Carsten and Tate (2005) investigated the effect that a number of possible systems may have on injuries due to collisions. Within this work, three types of variable speed limit systems were proposed:

Advisory

The speed limit is displayed to drivers and the drivers are reminded of changes in the speed limit.

Voluntary

The driver can enable/disable automatic control of the vehicle's maximum speed.

Mandatory

The maximum speed of all vehicles is limited at all times.

Along with these three systems, Carsten and Tate also outline different ways in which speed limits can be defined:

Fixed

Speed limits throughout the network are fixed.

Variable

Different areas within the network may have different speed limits (i.e., school zones, dangerous road sections).

Dynamic

Speed limits within an area can vary over time based on traffic and weather conditions.

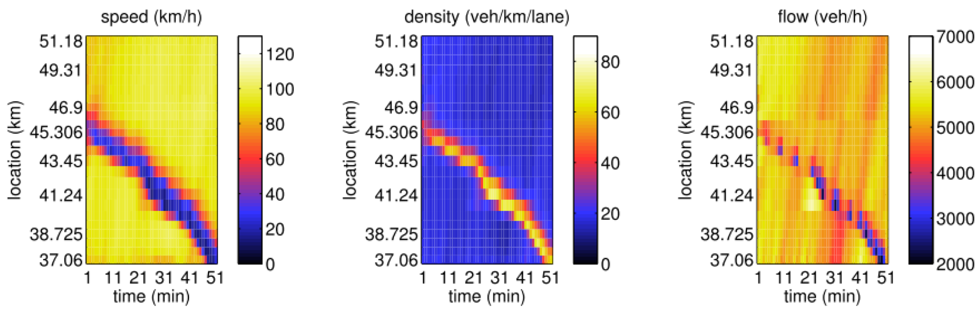
Using several models, which relate changes in speed to the probability and severity of collisions (see references within), Carsten and Tate predicted the reduction in injury due to collision when implementing different permutations of the proposed systems. The results, shown in Figure 33, found that injuries due to collision could be reduces by up to 50% when using a mandatory system with dynamic speed limits. While this estimate was the high estimate based on the data, the low estimate was a reduction in injury of 19%. Figure 33 also shows that the dynamic speed limit approach generally results in

System type	Speed limit type	Predicted injury accident reduction		
		Low estimate (%)	Best estimate (%)	High estimate (%)
Advisory	Fixed	2	10	21
	Variable	2	10	22
	Dynamic	3	13	27
Voluntary	Fixed	5	10	21[a]
	Variable	6	11	22[a]
	Dynamic	10	18	27[a]
Mandatory	Fixed	11	20	31
	Variable	12	22	33
	Dynamic	19	36	50

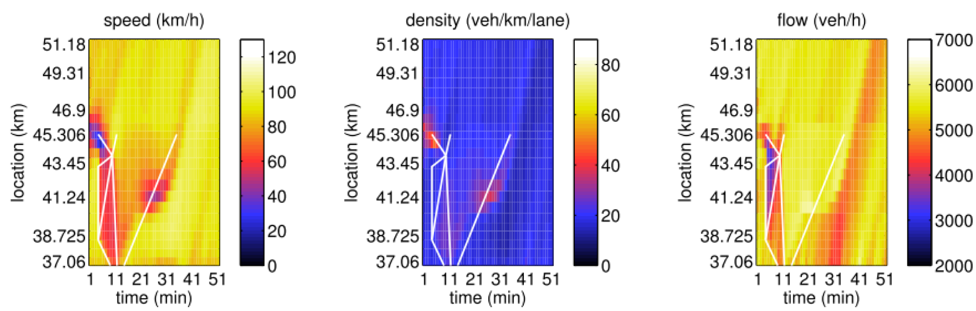
Figure 33: Injury reduction using variable speed limit systems (Carsten and Tate, 2005)

significant reduction, when compared to the other two approaches. The same is true when considering the mandatory system, compared to the voluntary and advisory systems.

While the previously discussed work considered only injury prevention/ reduction, several works have also proposed variable speed limits as a method for improving traffic flow in congested scenarios. The main concept behind these approaches is to increase the speed limit in front of the traffic congestion (i.e., at the front of stop-and-go traffic), while decreasing the speed limit behind the traffic congestion. The increase in speed limit ahead of the congestion allows vehicles to clear space at the front quicker than they would be able to with a fixed speed limit. Decreasing the speed limit behind the congestion decreases the rate at which vehicles arrive at the point of congestion.



(a) Uncontrolled



(b) Controlled

Figure 34: Results of the 1:14 scale platoon on a circular track (Hegyi et al., 2008)

If more vehicles are leaving than are arriving, the mass of vehicles should dissipate over time, allowing traffic to flow freely again. For example, the system proposed in Hegyi et al. (2008), based on shock-wave theory, is capable of identifying and alleviating traffic congestion. Figures 34(a) and 34(b) show the results of simulation in an uncontrolled and controlled situation respectively. The simulation was designed to model a section of the Dutch A12 freeway. In the uncontrolled case, it is evident that the congested situation persists over space and time, while in the controlled case, the congestion

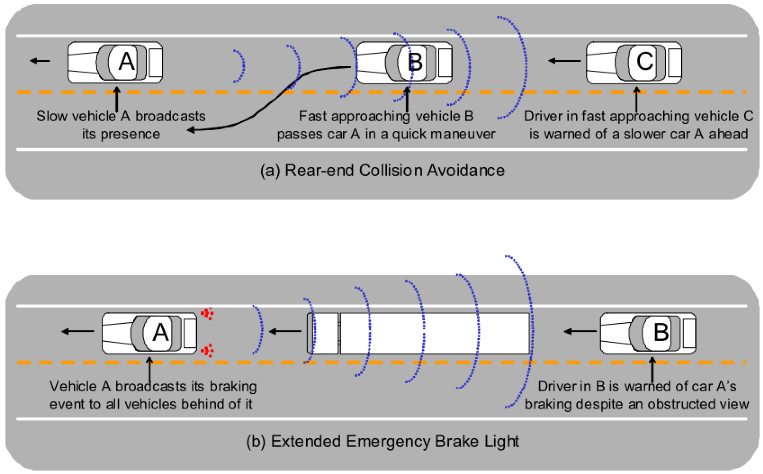


Figure 35: Example safety applications using V2V communication (Jiang and Delgrossi, 2008)

dissipates over the first 10 minutes of simulation.

5.3 Collision Avoidance/Prevention

One of the main focus to date of VANET research has been on applications related to advanced emergency warnings and collision avoidance (Yousefi et al., 2006), such as those applications demonstrated in Figure 35. Figure 35(a) shows how V2V communication could be used for rear-end collision avoidance. Within the figure, vehicle B must make a quick maneuver to avoid hitting vehicle A. An on-board computer system in vehicle B could identify this maneuver (i.e., using accelerometers and a heuristic algorithm) and notify vehicle C that a slow moving or stopped vehicle is ahead. Vehicle C, then, is given more time to react than it would have otherwise. Figure

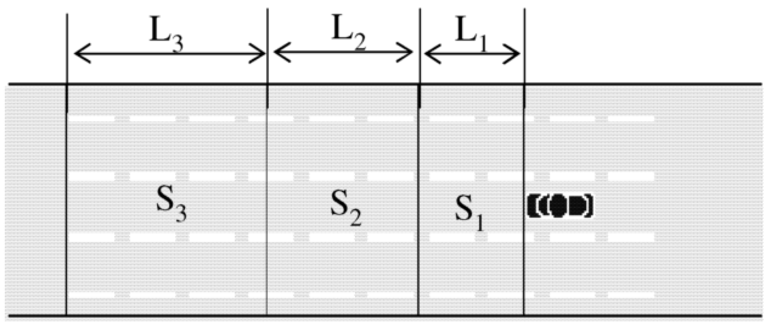


Figure 36: Example segmentation of space around a vehicle (Taha and Hasan, 2007)

35(b) shows a system in which vehicle A broadcasts an emergency braking message to nearby vehicles. This message is passed through the middle vehicle to vehicle B, which is now aware that A is braking urgently, even though its view of vehicle A is obstructed.

Many of the dissemination/broadcast algorithms presented in Section 3 were designed with emergency notification as the target application (i.e., Fasolo et al., 2006; Osafune et al., 2006). Most of these solutions, however, were designed to propagate information a far distance in a relatively short amount of time. Taha and Hasan (2007) propose a protocol that aims to identify the 'vehicle most in danger' and guarantee that the identified vehicle receives the message in a timely manner. Figure 36 can be used as an example to demonstrate the goals of the proposed protocol. The majority of broadcasting algorithms would dedicate time to discovering a node in section S_3, as this would provide the most efficient dissemination of the data. However, in an emergency situation (i.e., collision avoidance), vehicles in section S_1

require notification as soon as possible, while vehicles in section S_3, which may be hundreds of meters away, do not need to take immediate action.

5.4 Traffic Signals

The optimization of traffic signals is one of the most studied areas of traffic control. Typically, the proposed systems assume that some level of information is made available to intelligent controllers places at each intersection or within a centralized control center. Some of the most commonly used information is summarized below:

Queue Lengths

The current (or expected) queue length on incoming edges of the intersection.

Vehicle Counts

The number of vehicles on a section of road entering or leaving the intersection.

Network Information

Information regarding the properties of the nearby road network, such as road lengths, speed limits, and lane counts.

Travel Speeds

The average speed of vehicles travelling on road segments connected to the intersection.

Turning Probabilities

The probability of a vehicle taking each possible exit of the intersection available to it. These are often assumed constant or estimated based on short term traffic flow measurements, as in the work of McKenney (2011).

Offsets

The cycle offset values of the intersection and its neighbouring intersections.

Neighbour State

Measurements of the neighbour state, which may consist of any combination of the information already mentioned.

As the majority of the works presented within this section only deal with traffic signal optimization, and not VANET communication, certain pieces of this data are simply assumed to be available. Without the use of VANETs, however, much of this information is difficult to accurately predict. For example, McKenney (2011) predicts the number of vehicles on a road segment based on the number of vehicles that have entered and exited that segment. While these measurements could be made available using simply induction sensors, they do not account for the fact that vehicles enter/leave the segment along its length, or for the inaccuracy in induction loop measurements. Travel speeds, turning probabilities and neighbour state information are also difficult or impossible to calculate without some form of communication within the system. Using VANETs, however, this information is easily attained from vehicles travelling within the network, allowing for much more advanced intelligent traffic control to become a reality.

The remainder of this section discusses a number of broad approaches to traffic signal optimization, including decision support systems (Section 5.4.1), fuzzy logic (Section 5.4.2), reservation and economic-based systems (Sections 5.4.3 and 5.4.4), self-organizing systems (Section 5.4.5) and machine learning (Section 5.4.6).

5.4.1 Decision Support Systems

The main goal of a decision support system is to propose possible effective signal plans to human operators, making the human operator more efficient and informed. While this is not exactly intelligent traffic control, simply bypassing the human operator and implementing the best plan would allow these systems to control a traffic network (it is for this reason that these types of systems are included here). Typically, decision support systems rely on a pre-created list of traffic scenarios and possible control actions (along with the estimated effectiveness of these control actions). These lists are created using human traffic experts and historical traffic data, which can be very time consuming. Also, relying on this type of list means the system can be incapable of addressing unexpected problems that are not captured within the database. Performance may degrade significantly, then, if the system cannot make proper decisions due to the limitations of the scenarios list. Also, these types of systems can suffer in the same way that fixed traffic plans do as the traffic dynamics change and the original assumptions become invalid.

One of the first researchers to propose a traffic decision support system was Cuena (1995). This work divided the entire network into subnetworks of a manageable size, with a single agent used to control each subnetwork. These agents were each aware of some key information: the network design, possible conflicts and possible control actions.

Similar to the above mentioned approach, Hegyi et al. (2001) and De Schutter et al. (2003) implemented a decision support system which uses fuzzy logic to classify traffic states. Once again, the entire network is broken into smaller subnetworks, each of which has a specific casebase that is generated offline. Each case within these casebases represents a specific traffic situation, control measure and predicted effect. In the event of an unrecognized traffic state, fuzzy logic is used to compare the observed state to the states described in the casebase. Based on the level of similarity the observed state has with various cases within the casebase, the effectiveness of the control actions outlined by the similar cases can be predicted. The action with the highest predicted effectiveness is then implemented by the control devices within the subnetwork. In the later work, prediction of traffic flows in and out of the subnetwork over a period of time is also included. This prediction allows for the effect of other intersections' decisions to be considered automatically, eliminating the need for a coordinating agent. The performance of the system, however, could severely degrade if the actual state is not close to any of the states specified within the casebase, as fuzzy logic will not be able to select a similar case.

A fuzzy neural network based decision support system was described by Almejalli et al. (2007a, 2008). A flowchart representing the operation of this system is included in Figure 37. As with the work completed by Hegyi et al. (2001) and De Schutter et al. (2003), the most important piece of information in this system is the set of possible control actions available for each traffic situation. From the diagram, it can be seen that the set of states and control actions are generated from experiential human knowledge coupled with historical traffic data and available traffic control devices, which can be tedious and problematic in the long term. Not only do the list of states and control actions need to be modified as traffic parameters change over time, but the list also would not apply if the topology of the network were to change (e.g., a long term construction project closes several lanes). The fuzzy neural network (which has a structure similar to that shown in Figure 41) takes both traffic conditions (available from sensor data) and control actions as inputs. The first step involves converting these inputs into fuzzy membership values (e.g., traffic density is high, medium or low). From these values, there are a number of neurons which each represent a single fuzzy rule (the fuzzy rules themselves are identified using a GA, as proposed in Almejalli et al. (2007b)). Each of these fuzzy rules considers a set of possible traffic conditions and a possible control action, outputting expected

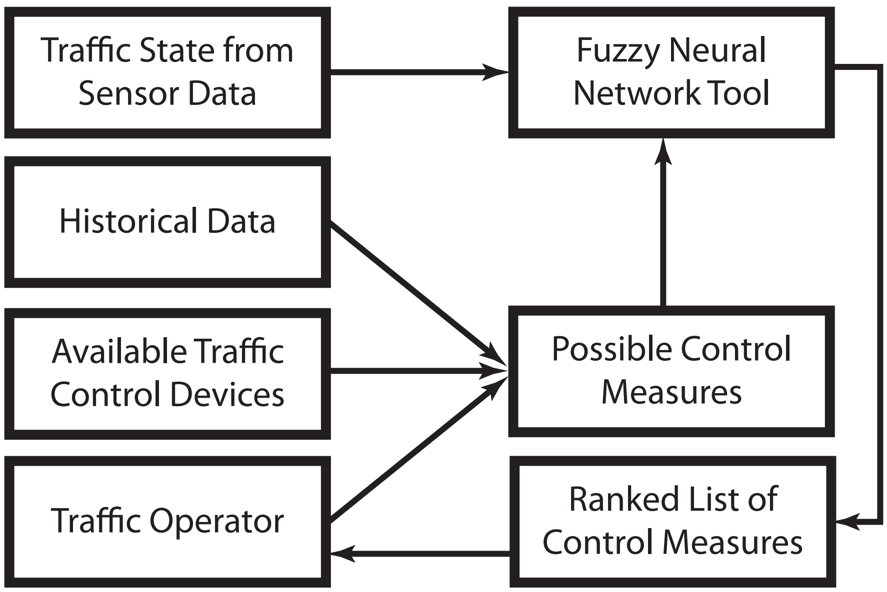


Figure 37: Flowchart for the system developed by Almejalli et al. (2007a, 2008)

traffic measurements (such as waiting time will be low). These fuzzy outputs are then defuzzified into exact fitness numbers, which allow each possible control action to be ranked against the other actions. The effectiveness of this system to predict a traffic situation after applying a control measure is shown by comparing the expected results to those found from a simulation. The results found that the predicted traffic state very closely resembled those found through simulation.

In Almejalli et al. (2009), the aforementioned support system is extended into a hierarchical framework (shown in Figure 38), with a coordinator agent (CA) being used to decide globally effective sets of actions based on the suggestions of a number of agents, each of which implements the fuzzy neural network previously mentioned. For each subnetwork, the coordinator agent maintains a table (the CA table) which details the possible effects any control action implemented within that subnetwork may have on all other subnetworks. Agents controlling each subnetwork then use the decision support system developed by Almejalli et al. (2007a, 2008) to propose a ranked list of control actions. Using the CA table, the coordinator can predict the effectiveness of a global set of control actions using the ranked lists of actions provided by the lower-level agents. Through a case study consisting of 3 subnetworks within a realistic model of a section of Riyadh, Saudi Arabia, the authors showed that the coordinator agent was able to consider the effects of the actions on the other subnetworks and determine which action each subnetwork should take.

5.4.2 Fuzzy Logic

The first application of fuzzy logic in the traffic signal control domain is accredited to Pappis and Mamdani (1977). This approach consisted of 3 input variables: time (e.g., very short, medium), recent arrivals at the green phase (e.g., few, many) and queue length at the red phase (e.g., none, small, larger). The rulebase, which was developed by trial-and-error in this case, produces a single output from these inputs: the extension time for the current phase. As is seen with the decision support systems described in Section 5.4.1, this rulebase remains static and does not adapt along with changing traffic parameters, which can lead to degraded performance over time as the

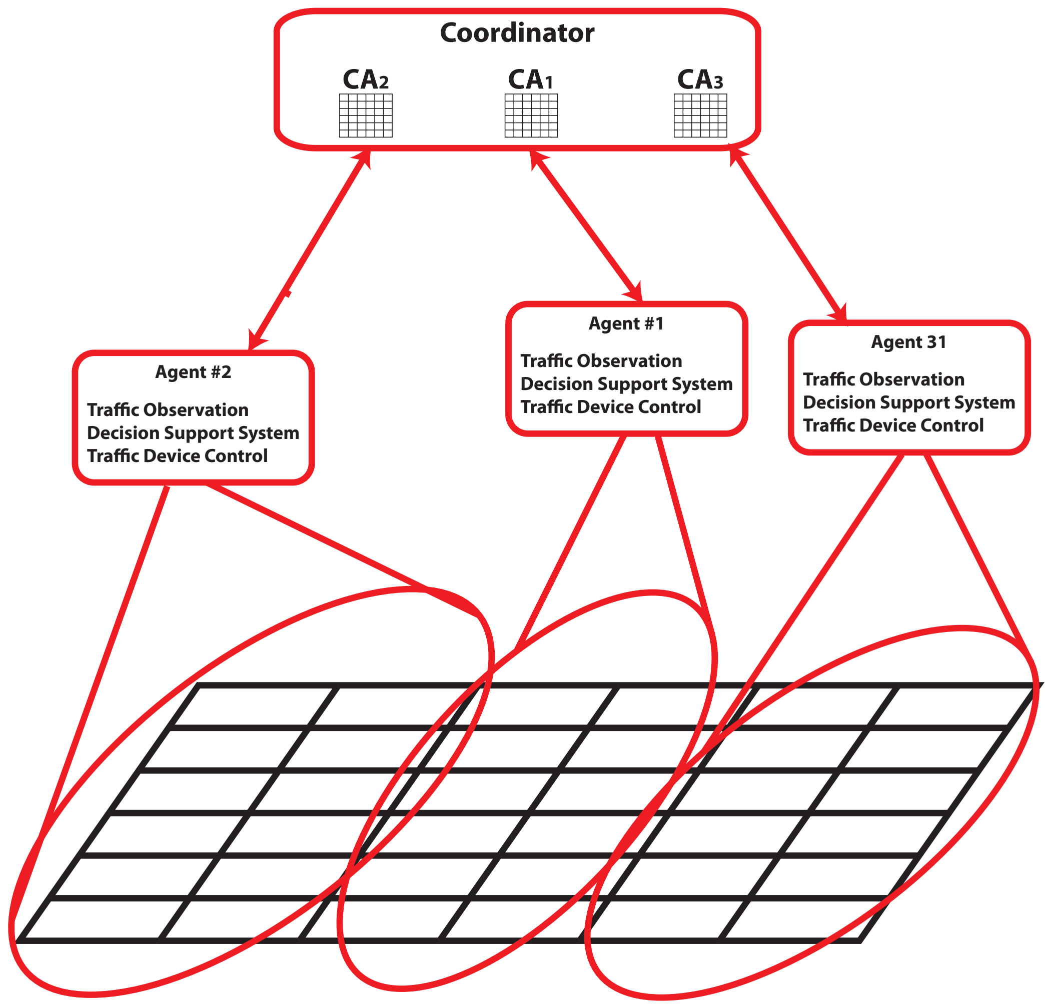


Figure 38: Architecture of the hierarchical decision support system implemented by Almejalli et al. (2009)

system does not generate a predictive model of traffic. After running for a specific amount of time, the fuzzy logic controller takes the input values and checks the rulebase to see how long the current phase should be extended. This process is repeated until it is decided that no extension should be given. The fuzzy controller approach presented was compared to a vehicle-actuated controller on a simple single intersection network for various fixed traffic volumes. Using the fuzzy logic controller resulted in a 10-21% decrease in overall delay. Chou and Teng (2002) also proposed an extension-based fuzzy controller, using only the queue lengths of the 4 incoming edges as input variables. The controller was shown, through simulation, to perform much better than a fixed-timed approach to signal control.

As opposed to using the extension principle, Chiu and Chand (1993a) and Chiu and Chand (1993b) designed a controller which adjusts phase split, offset and cycle time (using a different rulebase for each of these calculations) based on the saturation levels of the network's edges. After each cycle, an adjustment is made to both the phase splits and cycle length based on the saturation levels of the E/W traffic flow and the N/S traffic flow. This is done by calculating a fraction of the current cycle time for each flow based on the inputs. The rulebase for modifying offset values relies on the volume difference between nodes, as well as the estimated change required to produce a 'green wave' on the edge with the highest saturation. The effects of this controller were investigated in a simulation on a 3x3 grid network with vehicle insertion rates varying for each incoming edge. Initially, the signals in the network would be controlled using a fixed signal plan with equal time in each direction. After 30 simulated minutes, 3 intersections would be selected for adaptation, resulting in approximately 66% as much delay as the fixed plan. After another 30 simulated minutes, all intersections would be allowed to use the fuzzy controller, resulting in around half the delay of the fixed plan.

Hoyer and Jumar (1994) developed a more complex fuzzy controller which can deal with a varying number of phases. The fuzzy controller also included phase selection as an output, allowing dynamic phase ordering based on real-time traffic data. Inputs used in the fuzzy controller included current traffic volumes on the incoming lanes, as well as the elapsed time since the last phase change. The controller was compared to four other controllers (including a fixed time controller, two vehicle actuated approaches, as well as a simple fuzzy logic controller with 2 phases and a fixed cycle time) on a 3 intersection network with different traffic volumes and turning rates. With low volume of vehicles, all but the fixed controller maintained a similar level of travel time. At higher volumes though, the proposed approach seems to adapt better than the simple fuzzy controller and the vehicle actuated approaches. Similar to this, Lee and Lee-Kwang (1999) created a controller which optimized phase length and order, but also considered the state at neighbouring junctions. This was accomplished by adding input variables to describe the state of congestion at the neighbouring junctions, as well as the congestion between the intersection and its neighbours. Lee and Lee-Kwang (1999) compared the developed controller to a vehicle actuated controller on

3 grid-like (with minor mutations) networks, with both fixed and steadily increasing traffic levels. Improvements of 3.5-8.4% were found with fixed traffic volumes, while 4.3-13.5% was found with the increasing flows.

Similar to several of the previous approaches outlined, the FUSICO project (Niittymaki and Pursula, 2000) developed a fuzzy controller that included both a fuzzy extender for determining the time to switch phases, as well as a fuzzy selector for selecting phases. The controller was shown to outperform a simple extension-based control method for volumes of 400-1500vph (vehicles per hour), while the extension controller performed better on volumes less than 400vph. When speaking about the rulebase of FUSICO, the authors state that rulebase creation became complicated when the number of possible phases increased from 3 to 4. This highlights a key concern of fuzzy control design: how to design an effective set of rules. If 4 phases can cause difficulty, the problem will be further compounded when working with a complex intersection with twice this number of phases. A similar approach was presented in Murat and Gedizlioglu (2005) and compared to those presented in Niittymaki and Pursula (2000) and Pappis and Mamdani (1977). The new approach, however, does not improve upon the two previous solutions.

While the fuzzy control approach was much the same as seen in other proposals, Wei et al. (2001) proposed using a genetic algorithm to optimize parameters of the fuzzy controller to improve functionality; however, little detail was given on the implementation or effectiveness of this idea. A genetic algorithm was effectively used in Heung et al. (2005) for rule generation in a fuzzy traffic controller. A set of fuzzy rules was maintained for offline evaluation, in which the effectiveness of different rule combinations was analyzed. If an appropriate performance increase was found regularly while using a specific rule, that rule would be promoted to the knowledge-base and used within the fuzzy controller. Using the optimized rulebase, the authors were able to realize small improvement gains over a non-optimized set of rules.

5.4.3 Reservation-based Control

Dresner and Stone (2004) moved away from a traditional view of traffic light operation, proposing an architecture in which intersections reserve time and space for vehicles to cross the junction. Using this approach, the space inside an intersection is divided into an NxN grid of 'reservation tiles', each of which may contain at most one vehicle at any time (to prevent collisions). Driver agents then communicate with the intersection to reserve the space they will need to cross the intersection, passing several important pieces of information such as vehicle attributes, arrival velocity and arrival time. Using this information, the intersection agent will check to ensure that the space required for the requesting driver agent is available at the time requested before approving the reservation. If any of the necessary space is already reserved at the required time, the reservation will be rejected. In the case of a reservation rejection, the driver agent must attempt to make a new reservation until one is successfully granted. This method of intersection control can perform extraordinarily well because it eliminates the lost time from phase switching found in typical intersection control and can also allow vehicles to cross the intersection at their arrival velocity without having to stop. This approach, however, also relies on advanced and highly reliable communication networks, as well as sophisticated automated vehicle control. Reliance on these modules, however, can cause safety concerns, as will be discussed below. In this initial work, the reservation system was evaluated on a very small network, consisting of a single intersection with East/West and North/South traffic flow and a number of lanes ranging from 1 to 6. In each case, the system resulted in an average delay of nearly 0 with the best case being found using 1 lane per road (0.016 step average delay) and the worst being found with 5 lanes per road (0.031 step average delay). There are, however, several key requirements that make this approach infeasible. First, vehicles must be in very specific places at specific times to avoid any collisions. While this may be possible, it requires a very precise and automated vehicle controller. Furthermore, it this type of control were possible, it would be extremely difficult when considering all of the factors found in the real environment, such as differing road conditions. Also, traffic dynamics may result in missed reservations. For example, a vehicle may have an earlier reservation than that of the vehicle it is following, meaning it will undoubtedly miss its own reservation.

A key improvement to the original work is presented by the same authors in Dresner and Stone (2005). The original intersection reservation agent assigned reservations to vehicles based on a constant velocity. The reservation agents in the improved system, however, consider the possibility of vehicle acceleration within the intersection boundaries, which allows for a larger chance that a driver agent's reservation request will be approved. While including possible acceleration within the model allows for an increase in reservation approval, it also adds yet another variable which must be precisely controlled to avoid infringing on other vehicle's space within the intersection. This precise control of vehicle's can become even more difficult if sensor noise is included (it was not included within this work, however). This work also formalizes the communication protocol used within the system, something which was not specified in the original work.

Dresner (2006) and Dresner and Stone (2006) propose several further improvements to the prior work. First, it is suggested that the reservation agent be modified such that it can delay a response to a vehicle's reservation request. Previously, with immediate responses, a single vehicle may have a reservation approved shortly before two other vehicles request reservations that conflict with the first vehicle (though not conflicting with each other), resulting in two rejected reservations and one accepted. With the ability to delay the confirmation/rejection of a request, the reservation agent instead could approve the two non-conflicting reservation requests, while only denying the single conflicting request. Also, within the improved system, different types of vehicles can be assigned priorities (e.g., emergency vehicles are a high priority) which determine which reservations are rejected/accepted. The third improvement proposed is to treat the intersection as a market where vehicle agents place bids for time/space, with the highest bidder receiving the reservation. While no implementation details or results are presented regarding market based control (Wellman, 1993), this approach is discussed further in Section 5.4.4.

A much more detailed version of these works, as well as new simulation results, is presented in Dresner and Stone (2008). One of the main contributions of this new work is the consideration of human drivers. The authors acknowledge that the system would need to be implemented in stages and that it may never be fully adopted. For this reason, they consider changes that must be made to accommodate human drivers, such as ensuring that lights are green/red according to the current use of the intersection. While using a reservation-based system for intersection control has seemed very promising so far, the results from the experiments including human driver within the system show a dramatic decay in performance. With 1% of drivers being human, the average delay of vehicles is much higher than with no human drivers. In fact, around the median range of vehicle rates tested, 1% of drivers being human caused average delay to increase nearly 500%. These results signify a reliance on an extremely advanced intelligent driver model which is used to control all vehicles within the network, which make this approach infeasible assuming the system is not adopted by 100% of vehicles within the network (which is almost guaranteed, as noted by the authors). At the highest vehicle volumes, the average delay was found to be nearly 10 times as high; these numbers were even worse when considering higher percentages of human drivers. This work also investigates another ability that an intersection controlling agent must have in a real world scenario - accident detection. It is found that if the controlling agent has no means of detecting incidents within the intersection, then a large number of incidents will occur in the time following the original incident, as vehicles who assume the intersection will be clear collide with the stopped vehicles. After enabling the intersection to instantaneously detect an incident and assuming that all vehicles can be made aware of this incident, the number of resulting crashes in the 60 seconds after the initial crash was reduced from approximately 90 to 1.77 (using 6 lane roads). It is interesting to note, that with 5% human drivers included in the simulation, the incident rate was further lowered to 1.50 crashes over the 60 second interval as the human drivers do not assume the intersection will be clear of other vehicles. It is also shown that a delay in detecting the incident of only 5 seconds can increase the number of crashed vehicles during the interval to over 3.5. Furthermore, this approach relies on information such as maximum acceleration/deceleration of vehicles to calculate space/time requirements of vehicles. While this is easily done in a simulation environment, in the real world these values can vary based on a large number of difficult to determine factors such as weather conditions, tire conditions/choice and brake quality. Thus, it would seem that while this reservation approach may be extremely effective at controlling traffic flows in real time, there needs to be much more work done in the area of intelligent vehicle control before a real-world implementation could be considered safe or effective.

Similar to the reservation-based control approaches described previously, which eliminate traffic lights completely, Ferreira et al. (2010) proposes a 'virtual traffic light' system, in which traffic lights appear in vehicles as they approach the intersection. Within this system, communication is used to determine if there are any vehicles approaching an intersection in a conflicting direction. When multiple vehicles are approaching an intersection, one is elected the leader and has the responsibility of assigning priorities to each conflicting traffic flow. Details on the election and priority assignment algorithms are not presented by Ferreira et al., however market-based control (Section 5.4.4) may be well suited for this application. This system should be easier to implement and result in safer operation than the reservation-based systems discussed above; however, the current system assumes that vehicles are aware of the location of every nearby vehicle, which would require a large amount of data to be constantly broadcast. This problem could be at least partially alleviated through the use of infrastructure at the intersection or prediction of arrival times by the vehicles themselves. A similar virtual traffic light system is also presented by Wu et al. (2009).

5.4.4 Market-based Control

Market-based control (Wellman, 1993) is an approach which views a system as a virtual marketplace, with economic agents interacting amongst each other. While Dresner (2006) and Dresner and Stone (2006) proposed a market-based extension to the reservation controlled intersection architecture, one of the first real work on market-based intersection control was completed by Balan and Luke (2006). Intersection control agents within this work make decisions based on the number of 'credits' that the waiting drivers hold. Each driver agent begins with a fixed number of credits and also receives credits while waiting at a red light. When a vehicle crosses through the intersection, it must pay a fixed number of credits. The combination of this credit gaining/losing system, coupled with the fact that intersection control agents make decisions based on the number of credits each waiting vehicle possesses, has several beneficial properties:

- The intersection will generally favour incoming edges with more vehicles, as more vehicles results in larger credit totals.
- There is a level of fairness involved, as vehicles who have waited a long time previously (gaining credits) will be more likely to travel through the intersection.
- Emergency and other special vehicles can be given priority by augmenting the number of credits they have.
- It allows for a further extension in which drivers who are in a hurry can sacrifice credits they have stored to get through the network faster.

Experiments were carried out on a small traffic network (4x4 grid) where the method of assigning values to vehicles based on credits was compared to the following methods:

Counting Cars

One point per vehicle in the queue.

In-Range Time

One point per vehicle for each second that vehicle is within the sensor range of the intersection.

Mean Waiting Time

A vehicle's points are equal to the average time spent waiting at all intersections during the current trip (including the current intersection).

Previous Mean Waiting Time

A vehicle's points are equal to the average time spent waiting at all intersections during the current trip (not including the current intersection).

Initially, the main goal of the research was to develop a control scheme which was fair, by decreasing the variance of delay experienced by vehicles. While this goal was realized (the credit system resulted in lower variance than the other methods), the proposed method was also capable of controlling intersections more efficiently than the other methods. The credit system, however, was not compared to any other traffic control methods, thus it is difficult to comment on the method's overall effectiveness.

Vasirani and Ossowski (2009) combined both reservation and market-based control within the traffic domain. Unlike the work by Balan and Luke (2006), the intersection agents attempt to maximize profit by setting the cost of a reservation based on the level of demand. It is assumed that there is communication enabled between intersections and that all driver agents are aware of the current price of a reservation at all intersections. This work also assumes that real currency is used to pay for intersection use (an idea many may find undesirable) and that a secure payment method is available for the driver agents. The proposed system could, however, easily be applied using a virtual currency as is done in Balan and Luke (2006). A driver agent can request a reservation from an intersection and in the event it is approved, must transfer part of the cost in advance to secure the reservation. The driver agent then pays the remainder of the cost when it meets the reservation, or loses the initial fee if it is late or cancels the reservation. If a vehicle happens to arrive at an intersection without a reservation (either because it failed to make one, or was late), it must wait for a free reservation. The intersection manager will provide priority to any paying driver agents, but must grant a reservation to a driver who has been waiting a specified amount of time. Driver decision making, as far as routing is concerned, is based on a weighted sum of travel time and cost. This allows each driver to find a balance between how long it will take them to complete their trip and how much that trip will cost. An emergent feature of this strategy is dynamic traffic volume balancing caused by variable intersection costs. Figure 39 provides an example network with two available routes between points A and B. Intersections along the most direct route between the two points have, due to high demand, been assigned a cost of \$10 each (for a total trip cost of \$50). The slightly longer route, which has experienced less demand, has a cost of $5 per intersection for a total cost of $35. Drivers will be motivated by these price differences to select the slightly longer route, as the time lost begins to be outweighed by the money gained. One aspect of these systems that remains unclear is that of vehicle queueing. There is no discussion included regarding the obvious problem of a vehicle wishing to pay and proceed through the intersection being physically behind a vehicle that does not wish to pay at the moment. The paying vehicle must (in theory) travel through the intersection during its reservation, but may not be able to do so due to the vehicle preceding it. This situation also violates the properties of a market, in which the highest bidder should travel through the intersection first. To investigate the properties of this model, the authors used a mesoscopic simulation of an area of Madrid, although the traffic volumes were theoretical. It was found that varying the cost of reservations at the different intersections can help balance traffic flow and decrease overall travel time. As previously mentioned, this balancing of traffic volumes is realized because intersections which are not busy will decrease the cost of a reservation, making it more likely a vehicle will travel that route to save money.

5.4.5 Self-organizing Systems

Gershenson (2005) developed a simple, reactive self-organizing system which was capable of controlling traffic signals within a network. The developed control system was evaluated in the NetLogo (Wilensky, 1999) modelling

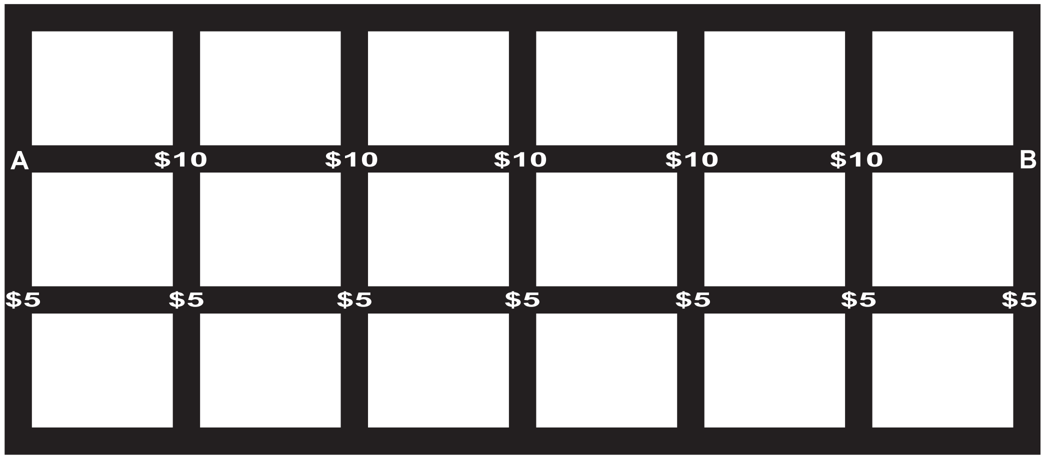


Figure 39: An example network with different reservation costs at different intersections

environment using the included Traffic Grid (Wilensky, 2003) model. This simple model of a grid network allows the user to define the volume of cars (the total number of vehicles in the network) and uses simple rules to control the behaviour of the vehicles within the model (accelerating if they can, stopping if they must). The initial method outlined in this work, named Sotl-Request, involved a single agent maintaining a counter at each intersection. At every discrete timestep, this counter is incremented by the number of cars approaching the red light of the intersection. Once the counter reaches a specific threshold value (determined by trial-and-error in this case, as were all system parameters), the light at the intersection is switched. Using this method, the signals at the intersection are completely self-controlled and manage to switch to allow incoming vehicles (or those waiting) to pass through. The counter at each intersection increases as vehicles drive toward the intersection, which allows this approach to automatically coordinate and offset lights throughout the network. The main problem with the counter-based approach is that a large number of vehicles in the network can result in rapid switching of the lights, which leads to high levels of inefficiency. For this reason, the second approach outlined (Sotl-Phase) added a minimum amount of time that must pass before switching lights at an intersection. This allows the system to be controlled effectively, even at higher volumes. While the Sotl-Phase control method was capable of controlling the signals well, the third method (Sotl-Platoon) added further restrictions to the system in an attempt to keep groups of vehicles travelling together. In Sotl-Platoon, when an agent wishes to switch the lights at its intersection, it will not do so if there is an approaching vehicle within a certain distance of the intersection (this rule is ignored if there is a number of cars greater than another parameter value, as these vehicles will still remain as a platoon). Cools et al. (2008) extended this work in three ways. First, the authors implemented the algorithm in a more realistic traffic simulator (MoreVTS, 2011), as the NetLogo Traffic Grid model provides only very basic traffic simulation. Second, a 12 intersection stretch of main road with incoming side roads was modelled to duplicate an area of Brussels. Finally, the algorithm proposed was evaluated against the plans used by the traffic authority for the area. These results show the proposed algorithm, on average, resulted in 50% less trip waiting time than the plans currently being used to control the network.

A much more analytical approach to self-adaptive traffic control was taken by Lämmer and Helbing (2008). Agents at each intersection in this approach use either of two possible strategies (depending on the local traffic situation). The first, which is used when traffic flow volumes are below saturation, determines optimal times to switch the lights at an intersection. This is achieved by taking into account the arrival and departure rates at the intersection, which allows waiting times to be predicted for each flow in the event they are stopped. Using these waiting time estimates and taking the time to switch lights into consideration, the decision to either switch lights or leave the lights can be made at any point. If the length of an incoming queue becomes longer than a specified value, the agent switches to a stabilization strategy. While using the stabilization strategy, an intersection maintains a list of incoming queues which are longer than should be allowed, changing the lights to clear these queues one after another. Simulations were performed, on a single intersection and an irregular 9x9 grid network, to compare the combined strategy approach to a fixed cycle approach, as well as both the optimization and stabilization strategies by themselves. The simulation results showed that the combined strategy was much more successful at keeping small queue lengths. Also, it was shown that neither of the two strategies worked effectively on their own.

5.4.6 Machine Learning

A number of machine learning approaches have been applied to intelligent traffic control. Three of these approaches will be discussed here: evolutionary computation, neural networks, and reinforcement learning.

Montana and Czerwinski (1996) presented one of the first works using an evolutionary approach for intelligent traffic signal control. Two strategies were developed and compared in this work: a strongly-typed genetic programming (Montana, 1995) approach to generating a traffic controller and a genetic algorithm (Holland, 1975) approach to optimizing fixed time signal plans. A description of each approach is provided below:

Genetic Programming

> A genetic programming parse tree is evaluated at every second, with the Boolean value of this tree resulting in either phase change (true) or no change (false). The parse tree uses typical Boolean functions (AND, OR, NOT, >) as well as a number of terminals such as number of vehicles approaching a light, whether vehicles are backed up to a sensor downstream and how long the current light has been in operation.

Genetic Algorithm

> Each individual consists of three real-value entries for each intersection in the network, with the first 2 numbers representing the lengths of each phase and the third number representing the signal offset value.

For both approaches, the fitness function was based on total delay within a simulation, with lower delay resulting in higher fitness. The performance of both the genetic programming and genetic algorithm strategy were compared for three different small networks with constant traffic flow rates. The

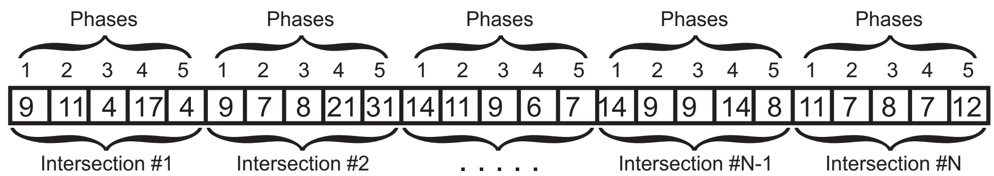


Figure 40: Example of a chromosome encoding phase lengths for a traffic network

performance of the genetic programming approach was better than that of the genetic algorithm for all three cases that were investigated. The individuals which performed best in the training scenarios for both approaches were also compared with different random number seeds to investigate the generalizability of the solutions. Once again, the genetic programming individuals performed the best, with results closely comparable to those found in training. The genetic algorithm had much wider variability, showing that the fixed signal time approach did not generalize as well.

A genetic algorithm (GA) was used once again by Sánchez et al. (2004) to optimize the signal timing of a set of intersections. In this work, each individual encodes the length of each phase within the cycle for each intersection. An example of what each chromosome may look like is shown in Figure 40. Within the figure, it can be seen that the chromosome has an integer value for each phase within each intersection. Although the figure has 5 phases for each intersection, this number can vary based on the control logic required for each intersection. A traffic scenario is simulated for each individual and the population evolves in an attempt to minimize overall travel times. Using a small traffic network, the GA approach was compared to results found using a random search technique, a fixed control scheme in which every traffic light maintains the same green period as the others and a scheme in which the traffic lights switch to allow the flow with most cars present to proceed. The GA approach presented performed well in comparison to the random control (83.82% decrease in travel time) and the fixed control scheme (56.74% decrease in travel time). It failed, however, in generating plans which performed better than the third approach which maintained travel times of approximately 60% of the times found using the GA. This work was expanded further by Sánchez et al. (2008), with the GA optimization being applied to a real world traffic network and compared to signal plans supplied by the traffic authority of that area. The modelled network consisted of 20 intersections located in the city of Santa Cruz de Tenerife and the traffic flows were probabilistically created using an origin-destination matrix inferred from the information available from the traffic department. After evolving a population of 200 individuals for 250 generations, average improvements ranged from approximately 7% to over 20% when compared to 9 signal plans supplied by the traffic department. This same GA optimization approach was further investigated by Sánchez et al. (2010), this time modelling the 'La Almozara' area in Saragossa, Spain (it should be noted that the traffic volumes used within these evaluations did not fluctuate over time). While analyzing the performance of the system, the authors noted that the traffic volumes within the modelled area were simply too low, resulting in little opportunity for optimization and thus unconvincing and inconclusive results. While this approach improved over other plans, which is certainly promising, it has still been evaluated using rather small networks. As the networks increase in size (as they would when controlling an entire city network), the search space involved in finding effective signal plans will increase significantly. This may decrease the utility of this approach if there is no further work completed to divide the problem into manageable sub-problems.

Neural networks (NN) were used by Spall and Chin (1994) to search for optimal signal plans. An entire day is divided into a number of intervals, with a single NN determining signal plans for a specific interval. An assumption is made by the authors that traffic situations in the same interval on varying days will be similar. When looking at long time intervals (e.g., the 3 hour weekday morning rush period), traffic levels are most likely quite similar from day to day, as a similar number of people travel to the same place at the same time. This assumption may not apply as well, however, outside of these typical rush-hour times and also may not apply within an interval itself. On a short-term basis (e.g., 5 minutes), interactions and dynamics can result in traffic volumes which vary from the average for the entire interval. This assumption then, may hold true in some cases, but may also be false in many others. The NN then, trains on each day over the interval it controls. On a simple 3x2 grid network, using fixed insertion rates for the 2 main roads (high average volume) and 3 crossing roads (low average volume), it was shown that the NN control approach could decrease the total system waiting time by nearly 35 percent over 50 training days. This approach is investigated further in Spall and Chin (1997), using a 3x3 grid network. The volumes within the network are designed (using data from Rathi (1988)) to match those of the Manhattan area the network is based on. The neural network control was used for a 4 hour period over the evening rush hour, with one test consisting of constant volumes throughout the 90 day simulation period and one test increasing the volumes on all roads after 10 days. In the fixed volume case, the NN approach to control was shown to decrease waiting time by approximately 10% over a fixed timing scheme. It is also shown, however, that the NN approach takes many days (approximately 20 to return to the initial value) to adapt to the increased demand from the 10th day. This is because the neural network learns how to handle a single traffic distribution in a highly effective manner. When the traffic volumes change, the neural network must re-learn how to effectively control the traffic lights, which takes time and results in a period of low performance. This is a common problem that can be seen in machine learning approaches, as the system becomes over-specified for controlling traffic with specific parameters.

Wei and Zhang (2002) combined both a fuzzy logic and neural network approach to decide whether to extend a phase or not. An example of this type of control architecture can be found in Figure 41. In this approach, measures of traffic volume are initially used as input to a fuzzy neural network (layer #1 within Figure 41). This input is then converted into fuzzy

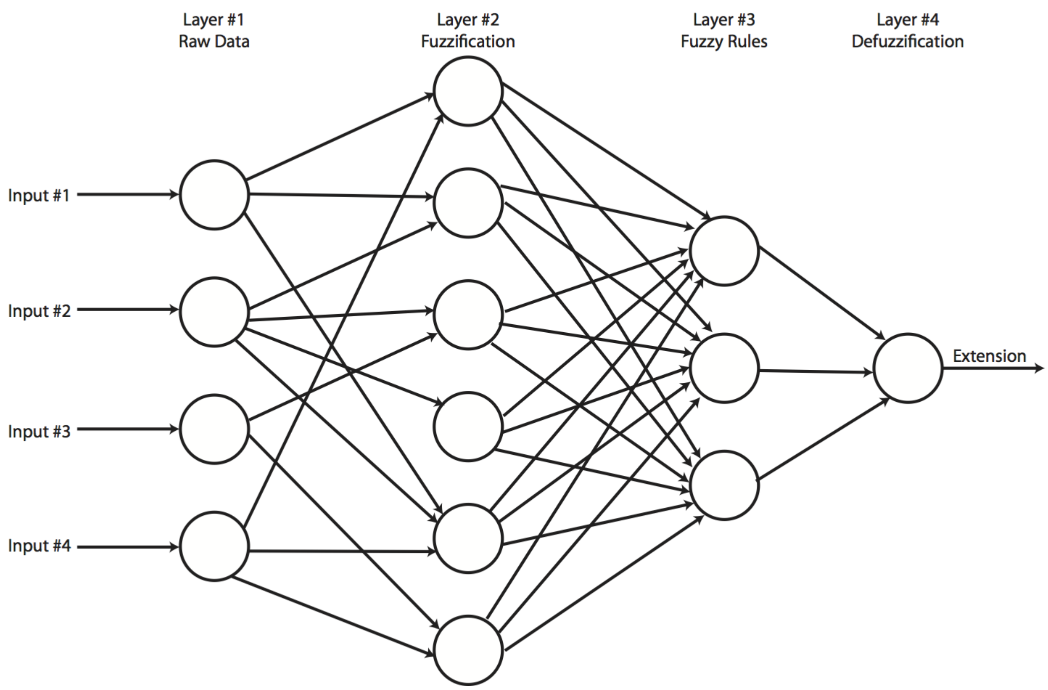


Figure 41: Example fuzzy neural network taking 4 inputs and determining phase extension

values (layer #2) so the fuzzy rule base (layer #3) can be used. The output of the fuzzy rule layer is then defuzzified (layer #4) which determines whether the light should be extended or switched. After a minimum amount of time, the decision process is carried out every second until the light is either switched or a maximum amount of green time has occurred (at which point the lights change automatically). The performance of the fuzzy neural approach was compared to that of a simple extension-based approach for a single intersection and multiple volumes (with volumes remaining fixed for each simulation). Through simulation, it was shown that the fuzzy neural control results in 15-25% less stops as well as 15-30% less average delay.

The power of fuzzy neural networks were combined with a hierarchical architecture by Choy et al. (2003a,b). The hierarchical architecture divided the entire network into smaller subnetworks, each of which is managed by an intersection controller agent. These intersection controller agents observe traffic state and pass measurements to a higher-level zone controller agent. The zone controller agents use a fuzzy neural network (similar in structure to that used in the decision support system developed by Almejalli et al. (2009, 2008, 2007a)) to calculate a zone signal policy and a zone cooperation factor (which determines the level of cooperation required amongst agents). Once zone policies are proposed, a conflict resolution process takes place to ensure all policies that will be implemented are compatible. Online reinforcement learning is also employed to constantly check the effectiveness of the fuzzy rule base. This learning process is responsible for determining which neurons are connected within the inner layers of the neural network (thereby determining the possible fuzzy rule base). To evaluate the effectiveness of this system, a 25 intersection network was implemented using data provided from the Land Transport Authority (LTA) of Singapore and the results of the proposed algorithm were compared to those used by the LTA. Two scenarios were simulated using both control methods: one consisting of 3 hours of simulation with a single volume peak and one consisting of 6 hours with 2 volume peaks. From these simulations, it was noted that the overall mean vehicle delay was decreased by approximately 15% for the single peak scenario and nearly 30% for the double peak scenario. A further traffic scenario consisting of 24 hours of simulation with a total of 8 traffic peaks is presented in Srinivasan et al. (2006). The proposed controller maintained lower mean delays over all peak periods and also performed much more effectively on later peaks, where the LTA control scheme began to fail.

Inspired by behavioural psychology, reinforcement learning (Kaelbling et al., 1996) is a machine learning approach which allows agents to interact with the environment, attempting to learn the optimal behaviour based on the feedback received from interactions. This typically involves breaking the environment into states, from which each agent can select a possible action. The reward gained from taking an action within a state determines the level of reinforcement, which in turn affects the likelihood that the agent will select that action when it is next in that state. As agents in the network monitor the traffic situation, they can identify problems as they occur and select possible control actions to alleviate these problems. Since the actions of each agent can affect the state in other subnetworks, a coordinator agent is included which is capable of determining the compatibility of the differing control actions suggested by the agents. This coordinator also has a notion of agent priority, so it is capable of deciding which control actions (from those proposed) should be implemented by the various agents.

Wiering (2000) and Wiering et al. (2003, 2004) presented one of the first reinforcement learning approaches to traffic signal control. These works assume that the traffic environment can be represented as a Markov Decision Process (Puterman, 1994), which can be defined as $M =< S, A, P, R >$, where S is a set of states, A is a set of actions, P is a transition function which determines the probability of moving to the next state when choosing a specific action from a specific state and R is a reward function which assigns a reinforcement value to each state/action/future state pairing. While the P and R functions are not known a priori, they are inferred from the experience of an agent as it interacts with the environment. It is unclear whether it is valid to assume that a Markov Decision Process models traffic correctly. While it is possible to predict a resulting traffic state from a state/action pair, this can certainly be difficult in a large network with a large number of interactions. Also, the number of possible states within a large network can be extremely large, which may make implementation of an MDP traffic model difficult. Within an MDP environment, however, each agent interacts with the environment, receiving rewards based on the actions they choose in various states. These rewards are used to induce a model which represents the expected gain when choosing an action in a certain state. Eventually, these values converge (if the assumptions hold true), at which point the maximum expected reward can be chosen by the agent each time. The traffic domain, however, involves constantly changing traffic distributions, which hinder the performance of reinforcement learning approaches designed to work in stationary environments. Two possibilities for learning are identified by the authors here: traffic light learning (which looks at the number of cars in each direction) and car-based learning (which involves summing the estimated rewards for each waiting vehicle). The authors chose the car-based learning approach (which makes traffic light decisions based on the estimated rewards for each vehicle in the queues) noting that it can be difficult to learn all possible situations a traffic node can experience. Through simulation, the learning approach outlined was compared to several simple traffic control mechanisms on a small 4x4 grid-like network and a small 'city-like' infrastructure. It was shown that the learning algorithms performed consistently better than the others on both network architectures, with an algorithm in which vehicles simultaneously learn better route choices as they progress through the network achieving the highest level of performance.

Steingröver et al. (2005) extend an approach similar to that described above by adding neighbouring junction congestion levels into the intersection state description. This addition can allow the intersections to learn how to cooperate with each other, but also increases the size of the state space. This new method was compared in simulation to the work shown in Wiering (2000) and Wiering et al. (2003, 2004) with both fixed and varying traffic volumes. With fixed volumes, the new method cut average trip waiting time approximately in half. The results were even more drastic when varying traffic volumes were included, with improvements ranging from around 50-75% less.

One problem with the reinforcement learning processes described so far is that they fail to adapt quickly in a non-stationary environment (where parameter values describing the traffic state change quickly). Since the traffic domain could certainly be considered non-stationary, de Oliveira et al. (2006)

attempted to use reinforcement learning with context detection (da Silva et al., 2006) to address this problem (in this case, a context represents a specific traffic distribution). With this approach, multiple models of traffic are maintained, each of which has a learning system associated with it. Each learning system then, is responsible for learning how to optimally control traffic that matches (or at least closely resembles) its corresponding model. An error value can be calculated for each model, allowing the system to decide which model matches the current traffic state, with the best matches being used to control the signals within the network. If no model exists with an error below a specified threshold, a new model will be created and the learning process will begin. The system begins with only a single model, adding new partial models as required by changing traffic parameters. Through simulation, it is shown that the context detection and multiple models allow this approach to perform better than a standard reinforcement learning approach when there is noise in the traffic volumes, as the traditional approach takes a period of time to relearn everything each time the volumes fluctuate.

5.5 Vehicle Routing

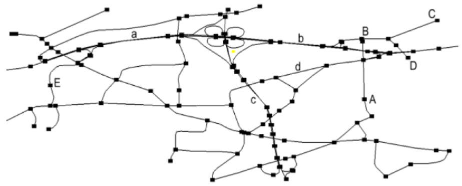
While effective traffic signal optimization can decrease trip times within a traffic network, the routes chosen by drivers can also have a significant effect. Two main applications of dynamic routing are problem-area avoidance and global route optimization, which are described in more detail below.

Problem-Area Avoidance

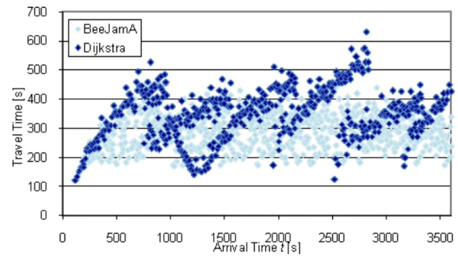
A problem-area within a traffic network is any area experiencing extended travel times. Typical causes of such a scenario are accidents, construction/lane closure and inefficient traffic signals; however, it has also been shown that traffic jams can form spontaneously when the number of vehicles in a road segment becomes too high (Sugiyama et al., 2008). A vehicle using dynamic routing could identify these problem-areas and create alternative routes. Re-routing vehicles to avoid these problem carries two benefits. First, the vehicles that are re-routed avoid the problem area and experience a shorter trip duration. Second, if these vehicles were not re-routed, they would enter the problem area and most likely exacerbate the problem.

Global Route Optimization

Current GPS routing devices typically use a basic shortest-path algorithm, which chooses a route based on the shortest geographic distance possible. A problem may arise, however, if a large number of vehicles going from point A to point B choose the exact same route based on distance. The increased volume along this route may increase the travel time an amount that would make selecting an alternative, slightly longer route a better choice for some vehicles. Global route optimization aims to incorporate some form of coordination among the vehicles, allowing them to be balanced across all possible routes.



(a) Road Network

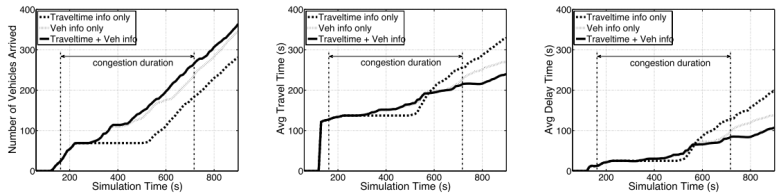


(b) Travel times throughout simulation

Figure 42: Simulated network and results from Wedde et al. (2007)

Wedde et al. (2007) proposed the application of a previously developed dynamic routing algorithm (Wedde et al., 2004) to traffic control. The routing algorithm was inspired by the foraging behaviour of bees, which is a sub-area of the general research area of swarm intelligence (Bonabeau et al., 1999). In Wedde et al. (2007), the initial algorithm, which was developed for use in computer networks, is modified for use in a traffic network. The algorithm was compared to a fastest-path finding algorithm, which works similar to a shortest-path (distance-based) algorithm, but uses the travel times of road segments instead of physical length. The traffic information used by the fastest-path algorithm is updated every 10 minutes (noted by the authors as an optimistic time interval). Figure 42(a) shows the network used to evaluate the algorithms, in which vehicles began at points A, B, C or D, and travelled to point E. Figure 42(b) shows average trip travel times throughout the one hour simulation of the road network. The figure shows that that the trip times of vehicles using the bee-based routing algorithm experience much less fluctuation over time than vehicles using the fastest-path approach. Also, it can be seen that the fluctuations of trip times using the fastest-path algorithm tend to oscillate in 10 minute intervals. As this is the same interval used to update the information, the oscillations are most likely due to the fact that the observed traffic volumes become out-dated as time passes. This further motivates the use of real-time optimization techniques, such as the proposed swarm intelligence approach.

Yang and Bagrodia (2009) propose a different approach to dynamic routing, in which vehicles keep estimates of travel times for each road segment and augment these estimates using a weighted average when a new observation is received from another vehicle. The proposed algorithm also uses information ageing, in which data that is beyond a specific age is disregarded when received by a vehicle. This is useful as the data may not be applicable if the observation was made much earlier, as was seen in the results presented above. Yang and Bagrodia also identified the fact that using only road segment travel times as information can result in poor performance, as this information is updated infrequently in the case of extreme traffic jams (if a vehicle never reaches the end of a road segment, it will not send any information). For this reason, Yang and Bagrodia also broadcast 'vehicle information' relating to the recent travel speed of the vehicle for use in route computation. Figure 43 shows the results of using only travel time information, only vehicle information, or both over three metrics. These figures clearly show that using only travel time information results in an increase



(a) Number of vehicles arrived (b) Average travel time (c) Average delay time

Figure 43: Comparison of system performance using only travel time information, only vehicle information, and both types of information (Yang and Bagrodia, 2009)

of travel time/delay, resulting in a smaller number of vehicles arriving at their destination. Using only vehicle information increases the performance of the system, and using both forms of information seems to further increase system performance by a small amount. As with the bee-based routing algorithm discussed above, these results demonstrate the importance of real-time accurate information when making routing decisions in a traffic network.

Dobre et al. (2012) presents a centralized approach to dynamic route assignment in traffic networks. Within this system, vehicle sensors generate observations regarding traffic state/travel times and report these to intelligent traffic light devices (ITLs). ITLs compute aggregate information from these observations and forward their data to a centralized server which is responsible for assigning vehicle routes. The central server is responsible for analysing all data from the network and assigning routes to all vehicles within the network. The reasoning offered by the authors for this design decision, is that many vehicles would choose to take the same route if an alternate, faster route was discovered and announced to all vehicles. The central server, then, is responsible for enforcing coordination among vehicles to prevent any alternate routes from becoming over-saturated by vehicles who change routes. The problem with this system is that the collection/analysis of data and the calculation/assignment of routes for an entire network will carry an extremely large computational cost. Scaling such a system to address a city-sized traffic network in real-time would be difficult or impossible.

5.6 Summary

This section has presented a number of control methods that may be implemented within a VANET system. When considering a comprehensive traffic control solution (i.e., one capable of controlling all vehicles and control devices within the network) a number of desired properties can be identified from the section's discussion. The importance of each of these properties is outlined below:

Dynamic

As traffic conditions are continuously changing, a traffic control system must be capable of identifying and adapting to changes as they occur. This may include dynamically varying speed limits for the purposes of safety and improved traffic flow, modifying traffic signal plans to more efficiently serve current queues, or modifying vehicle routes within the network to avoid problem areas and alleviate congestion.

Cooperative

A traffic system consists of an extremely large number of entities interacting amongst each other. For this reason, a decision by one agent within the system cannot be made without considering the effect that the decision may have on other parts of the network. For example, if a selfish intersection control agent drastically modifies its signal plan to alleviate local traffic congestion, this congestion may simply be moved to the next intersection along the road. Furthermore, if the newly congested intersection does not address the situation quickly, traffic may backup enough to affect the performance at the original intersection as well. For this reason, intelligent control agents within the system must cooperate with each other to achieve the highest level of performance.

Decentralized

Decentralized systems are typically more robust and scalable than their centralized counterparts. For these two reasons, a decentralized solution is highly desirable within the traffic control domain. Using a decentralized approach to traffic control will allow for the system to scale to city-sized problems, while also allowing the system to operate in situations such as localized power or technology failures.

Predictive

The vast majority of previously proposed traffic control solutions are reactive in nature. Ideally, though, a traffic control solution should be capable of predicting short-term fluctuations in traffic volumes. Prediction can also be useful when considering a dynamic routing solution, as vehicles may predict current and future travel times based on the information made available through VANET communication.

6 Trust

A VANET system can be considered a social system, in which the information shared among participants affects the decisions of others. For this reason, trust can play an important role in the system's performance and security. Within the traffic system, misinformation could be classified into two categories: inadvertent and purposeful.

Inadvertent Misinformation

As much of the information originally introduced into a VANET is proposed to come from sensor data within a vehicle, malfunctioning sensors could result in inadvertent misinformation being introduced to the system. For example, a vehicle with a malfunctioning speed sensor may report the average speed on a road segment to me significantly different than it truly is.

Purposeful Misinformation

Purposeful misinformation is the conscious introduction of incorrect/ misleading information into the system by a malicious entity. There are a number of possible scenarios in which introducing false information may be advantageous to someone. A vehicle heading to a parking

> garage may constantly report that the parking garage was recently full, leaving other vehicles to search for a different parking space. This vehicle then would be more likely to find a spot available when it arrives at the parking garage. Business owners may attempt to convince vehicles that driving past their store would result in a shorter trip time than taking an alternate route. It is easy to imagine many other scenarios in which a person may benefit from introducing false information into the VANET system.

If VANETs are to become a useful tool within society, solutions must be developed to deal with these sorts of trust problems. The review of trust-based VANET research is beyond the scope of this work; however, Zhang (2011) offers a comprehensive review of trust management solutions in VANETs, including VANET trust challenges, desired trust management system characteristics, and current trust management approaches.

7 Future

This work has discussed a number of different approaches to intelligent traffic control and data dissemination within VANETs. This section will discuss important future research directions in both of these areas.

7.1 Data Dissemination

Section 3 identified a number of desirable characteristics for a data dissemination system to possess. Three of the most important of these characteristics were the ability to adapt to high/low traffic volumes, to decrease the amount of data transferred between vehicles, and to quickly propagate useful information to vehicles within the network. There are a number of possible efforts which could lead to improvement of data dissemination overall by addressing these three key concerns.

While a number of works have proposed algorithms which can operate using both multi-hop and single-hop communication, future work in this area could improve the identification of which method to use and how to use it. For example, data that is of short-term relevance within a small area (i.e., an emergency braking event) could be transmitted using multi-hop communication. This does not present a problem, as any vehicle that does not receive the message due to network disconnection is most likely not concerned with the event. On the other hand, data regarding slow moving and congested traffic may be useful over a longer time interval or to vehicles in other parts of the network. This data, then, could be transmitted using single-hop communication, decreasing the bandwidth used in comparison to the contention phase of many multi-hop protocols. This problem is really centred around the contextualization of data within the network. Further research may examine different types of data that may arise in VANETs, how this data can be contextualized, and how this context can be used to improve dissemination.

Similar to contextualization, geographic data could be used within a dissemination algorithm to further improve its performance. For example, consider a vehicle (the source) that wishes to distribute its travel time information to the entire network. Typically, the source vehicle will communicate its observed data between itself and a number of other vehicles (the destinations). Using a small amount of extra data, however, a destination vehicle can share the location it is travelling to (and possibly route information) with the source. This small amount of data could consist of exact coordinates or general areas within the network that each vehicle is travelling toward. This extra data allows the source vehicle to identify several parts of the network that should receive the data in the future if it is sent to the specified destination vehicle. Using this approach, the source vehicle can estimate which areas in the network the data will reach. Due to the highly dynamic movement of data within the network, however, this estimate may prove inaccurate in practice. Future investigation should consider what (if any) effect this approach to data dissemination may have.

Geographic data could also be used to decrease the amount of data transferred between vehicles or to better prioritize different pieces of data. For example, vehicles travelling north generally have no use for information regarding traffic to the south of its current location. Sharing general destinations/routes between vehicles, then, could be used to discriminate between data that is absolutely necessary, beneficial, or unnecessary to share between vehicles. Of course, when considering the global goals of dissemination, it may still be pertinent to share data that is not useful to a specific vehicle, if that data may be useful to other vehicles in the future. Future work, then, should focus on investigating this type of trade-off to determine the best globally performing solution.

Finally, research could investigate the effects that additional infrastructure may have on data dissemination within a network. This may involve varying the amounts of infrastructure, as well as investigating different infrastructure behaviour (i.e., storage facility, active forwarder, query point).

7.2 Traffic Control

While traffic signal control has been approached from a wide variety of perspectives, one problem that is still difficult to solve is the offsetting of traffic lights within a network. This offset determines at what point within the traffic light cycle the signal plan begins, allowing for some level of synchronization between consecutive/neighbouring traffic lights. One way in which this offset optimization could be improved is through the sharing of state information between nearby traffic lights. Examples of possible useful information to be shared are current traffic volumes at neighbour intersections and intended/estimated actions of vehicles at neighbouring intersections (i.e., how many plan to turn left/right). Using a VANET, this information could be transferred from intersection to intersection using vehicles as intermediary communication devices. This sharing of information could allow for a higher level of coordination between intersections, resulting in improved traffic flow. Assuming this information is made available through a VANET system, future work then needs to answer questions such as: how can this information be used to improve traffic signal plans, how much information should be shared between intersections, what information should be shared between intersections, and should information from intersections further away be incorporated into decision making?

One of the proposed advantages of reservation-based signal control algorithms (Section 5.4.3), is that they decrease the amount of wasted time that can come as a result of fixed signal phase cycles. However, as discussed in Section 5.4.3, these systems are also difficult to implement and generally require all vehicles in the network to take part in the system. Another way in which signal plans could be optimized is to allow for real-time selection of phases based on the current traffic state. To explain this concept, first consider a simple fixed-phase cycle traffic light, which first allows north/south traffic to flow, then east/west traffic. This process repeats indefinitely, with the lengths of each phase being set by some control mechanism (intelligent or otherwise). A real-time adaptive system could instead assign green lights to any set of non-competing traffic flows (flows that can travel through the intersection at the same time without interfering with each other) based on the current traffic state, as determined through communication between vehicles and control infrastructure. Future work should consider this 'cycle-less' approach to traffic signal control, which allows for a much more adaptive solution then solely setting cycle lengths, phase lengths and offset times.

Another major area of research for traffic control lies in dynamic vehicle route selection. Future work in this area should focus on how real-time information can be used to determine optimal route selection. Vehicles within the network should constantly be evaluating both their current route and a set of possible alternate routes based on the locally stored traffic state estimate. Beyond single vehicle route optimization, future research may also consider the role infrastructure may play in globally optimizing route selection. If vehicles inform the infrastructure of their chosen routes, it may allow for a prediction of future traffic volumes and travel times. Using these predictions, infrastructure units could alert drivers of potential future problems on their routes, or redirect some drivers to different routes that are expected to be better in the future. This would allow route assignment to become proactive in nature, selecting routes to avoid causing problems in the first place, as opposed to the typical reactive system which only re-routes vehicles when traffic congestion has already formed.

8 Conclusion

This document has introduced and discussed vehicular ad hoc networks (VANETs) as a tool for improving the intelligent control of vehicles within a traffic network. It has been proposed that the use of VANET technology will lead to improved traffic efficiency, resulting in decreased travel times, better use of constrained resources (i.e., roadways and intersections) and safer driving.

Throughout this discussion, the document has outlined a number of different algorithms used for both data dissemination within a VANET, as well as for the intelligent control of traffic and traffic control devices. Throughout the discussion of algorithms, a number of characteristics that should be considered when designing these algorithms were identified. These key characteristics are outlined below for dissemination and control algorithms:

Dissemination Algorithms

The effectiveness of a dissemination algorithm can be measured by how quickly it propagates data throughout the network and how useful the data it propagates is to the vehicles that receive it. For this reason, the speed and utility of data disseminated throughout the network should be maximized. Adapting to the volume of data/vehicles currently in an area will help an algorithm make the most efficient use of limited bandwidth, and will also allow the algorithm to determine parameters to maximize transmission velocity. As vehicles will typically have a large number of traffic state observations stored in memory, a dissemination algorithm must have the capability of identifying the most useful data to transmit to neighbouring vehicles (data selection) or of merging multiple reports into a single observation (data aggregation). Finally, to further increase efficiency, a dissemination algorithm should be capable of identifying and eliminating unreliable or unnecessary data (due to age, misinformation, etc.).

Traffic Control Algorithms

As traffic state is highly dynamic, one of the most important aspects of any real-time traffic control solution is that it be capable of quickly adapting to changes in the environment. Using a decentralized approach to control allows a system to quickly process changes in localized regions and generate effective control schemes. A decentralized approach also allows a control system to be both scalable and robust. Another important aspect of the traffic control domain is the fact that local decisions can carry global consequences (i.e., efficient traffic control at one intersection may lead to overloaded situations at nearby intersections). For this reason, a network-wide traffic control system should allow for cooperation between various distributed control agents. Fortunately, a VANET system allows for communication of data between different entities within the network, which could facilitate cooperation between geographically separated control agents. Finally, a traffic control algorithm should be predictive in nature, rather than reactive. The predictive abilities of a traffic control system could include forecasting traffic volumes using filtering techniques (i.e., alpha-beta or Kalman filtering), predicting future state through communication between neighbouring controllers, or estimating travel times based on observations stored in memory.

This document also discussed communication simulators, which are used to model and investigate wireless communication systems, including VANETs. A number of important components that must be considered when selecting a communications simulator for VANET research were identified. One of the main requirements of any communication simulation software selected is the ability to model realistic vehicle mobility and radio communications. However, there is generally an accepted trade-off between realism and computation time, so some in many cases a certain level of realism may be sacrificed to allow for feasible computation times. Another important aspect of a communication simulator is the possibility of integration with a realistic traffic simulator. As discussed in Section 4.1.4, coupling communication and traffic simulations allows for a much more diverse range of applications to be investigated. Finally, another important consideration that should be made when selecting a simulation environment is the ease of use/implementation. This may include the available documentation, the programming language/ knowledge required, as well as existing tools for generating necessary components of a simulation.

The main goals for the remainder of this project can be broken down into three stages, as outlined below:

1. Simulator Identification: This part of the project will involve identifying a suitable communication/traffic simulation environment to be used in investigating the developed algorithms. Along with identification, this stage will also include the development of (or identification

of already existing) realistic traffic scenarios. These scenarios should capture realistic vehicle mobility over realistic vehicle networks.

2. Dissemination Algorithms: This stage of the project will consider possible improvements of existing dissemination algorithms for VANETs. The main focus of this investigation will be to identify ways in which data can be contextualized within a VANET, and how an algorithm can capitalize on this contextualization to improve dissemination speed and efficiency. Proposed algorithms will be evaluated within simulation of the realistic scenarios developed in the previous phase of the project.

3. Real-Time Control: This phase of the project will investigate possible real-time control strategies that may be realized through the use of VANET data dissemination. One of the main foci of this phase will be on real-time route selection by vehicles using observations stored in memory, as well as through communication with possible roadside infrastructure. This stage of the project will also investigate possible improvements to existing intelligent traffic signal control algorithms, which may include further research in the areas of market and reservation-based control approaches, as well as 'cycle-less' real-time signal switching.

References

Christian Adler, S Eichler, Timo Kosch, Christoph Schroth, and Markus Strassbergert. Self-organized and context-adaptive information diffusion in vehicular ad hoc networks. In *Wireless Communication Systems, 2006. ISWCS '06. 3rd International Symposium on*, pages 307–311, 2006. ISBN 1424403987.

Assad Al Alam, A. Gattami, and K.H. Johansson. An experimental study on the fuel reduction potential of heavy duty vehicle platooning. In *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, pages 306–311. Ieee, September 2010.

Salim Allal and Saadi Boudjit. Geocast Routing Protocols for VANETs: Survey and Guidelines. In *2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, pages 323–328. Ieee, July 2012. ISBN 978-1-4673-1328-5.

K. Almejalli, K. Dahal, and M.A. Hossain. Intelligent Traffic Control Decision Support System. In *Applications of Evolutionary Computing*, volume 4448 of *Lecture Notes in Computer Science*, pages 688–701. Springer Berlin/Heidelberg, 2007a.

K. Almejalli, K. Dahal, and M.A. Hossain. GA-Based Learning Algorithms to Identify Fuzzy Rules for Fuzzy Neural Networks. *Seventh International*

Conference on Intelligent Systems Design and Applications (ISDA 2007), pages 289–296, October 2007b.

K. Almejalli, K. Dahal, and M.A. Hossain. Real Time Identification of Road Traffic Control Measures. In *Advances in Computational Intelligence in Transport, Logistics, and Supply Chain Management*, volume 144 of *Studies in Computational Intelligence*, pages 63–80. Springer Berlin/Heidelberg, 2008.

K. Almejalli, K. Dahal, and M.A. Hossain. An intelligent multi-agent approach for road traffic management systems. *18th IEEE International Conference on Control Applications*, pages 825–830, July 2009.

H Alshaer and Eric Horlait. An optimized adaptive broadcast scheme for inter-vehicle communication. *Vehicular Technology Conference, 2005. VTC 2005-Spring. 2005 IEEE 61st*, 5:2840–2844, 2005.

Alessandro Amoroso, Gustavo Marfia, and Marco Roccetti. Going realistic and optimal: A distributed multi-hop broadcast algorithm for vehicular safety. *Computer Networks*, 55(10):2504–2519, July 2011. ISSN 13891286. doi: 10.1016/j.comnet.2011.04.011. URL `http://linkinghub.elsevier.com/retrieve/pii/S138912861100140X`.

G. Balan and S. Luke. History-based traffic control. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 616–621. ACM, 2006.

MA Berlin and Sheila Anand. Driving Hazards Message Propagation Using Road Side Infrastructure in VANETs. *Advances in Computer Science and Information Technology. Computer Science and Information Technology*, 86(1):1–8, 2012. URL `http://www.springerlink.com/index/T43G86N317312K8G.pdf`.

James Bernsen and D. Manivannan. Unicast routing protocols for vehicular ad hoc networks: A critical comparison and classification. *Pervasive and Mobile Computing*, 5(1):1–18, February 2009. ISSN 15741192.

Christian Bettstetter, Hannes Hartenstein, and Xavier Pérez-Costa. Stochastic Properties of the Random Waypoint Mobility Model. *Wireless Networks*, 10(5):555–567, September 2004. ISSN 1022-0038.

J.J. Blum, a. Eskandarian, and L.J. Hoffman. Challenges of Intervehicle Ad Hoc Networks. *IEEE Transactions on Intelligent Transportation Systems*, 5(4):347–351, December 2004. ISSN 1524-9050.

Eric Bonabeau, Marco Dorigo, and Guy Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems*. Oxford, 1999.

Stefano Busanelli, G. Ferrari, and S. Panichpapiboon. Efficient broadcasting in IEEE 802.11 networks through irresponsible forwarding. In *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, pages 1–6, 2009. ISBN 9781424441488.

OMJ Carsten and FN Tate. Intelligent speed adaptation: accident savings and costbenefit analysis. *Accident Analysis & Prevention*, 37(3):407–416, May 2005.

S. Chiu and S. Chand. Adaptive traffic signal control using fuzzy logic. In *Fuzzy Systems, 1993., Second IEEE International Conference on*, pages 1371–1376. IEEE, 1993a.

S. Chiu and S. Chand. Self-organizing traffic control via fuzzy logic. In *Decision and Control, 1993., Proceedings of the 32nd IEEE Conference on*, pages 1897–1902. IEEE, 1993b.

Chih-hsun Chou and Jen-Chao Teng. A fuzzy logic controller for traffic junction signals. *Information Sciences*, 143(1-4):73–97, 2002.

M.C. Choy, R.L. Cheu, D. Srinivasan, and F. Logi. Real-time coordinated signal control using agents with online reinforcement learning. *Proc. 82nd Annual Meeting of the Transportation Research Board*, pages 1–21, 2003a.

M.C. Choy, D. Srinivasan, and R.L. Cheu. Cooperative, hybrid agent architecture for real-time traffic signal control. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 33(5):597–607, 2003b.

Seung-Bae Cools, C. Gershenson, and B. D'Hooghe. Self-Organizing Traffic Lights: A Realistic Simulation. In *Advances in Applied Self-organizing*

Systems, Advanced Information and Knowledge Processing, pages 41–50. Springer London, 2008.

J. Cuena. Knowledge-based models for adaptive traffic management systems. *Transportation Research Part C: Emerging Technologies*, 3(5):311–337, October 1995.

B.C. da Silva, E.W. Basso, A.L.C. Bazzan, and P.M. Engel. Dealing with non-stationary environments using context detection. *Proceedings of the 23rd international conference on Machine learning - ICML '06*, pages 217–224, 2006.

D. de Oliveira, A.L.C. Bazzan, B.C. da Silva, E.W. Basso, L. Nunes, R. Rossetti, E. de Oliveira, R. da Silva, and L. Lamb. Reinforcement learning based control of traffic lights in non-stationary environments: a case study in a microscopic simulator. In *Proceedings of the 4th European Workshop on Multi-Agent Systems (EUMAS06)*, pages 31–42, 2006.

B. De Schutter, S.P. Hoogendoorn, H. Schuurman, and S. Stramigioli. A multi-agent case-based traffic control scenario evaluation system. In *Intelligent Transportation Systems, 2003. Proceedings. The IEEE 6th International Conference on*, volume 1, pages 678–683. IEEE, 2003.

H Diab, M.G. Chavez Grunewald, I. Ben Makhlouf, D. Abel, and S. Kowalewski. A testing platform for cooperative vehicle platoon controllers. In *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, pages 1718–1723. Ieee, September 2010.

Ciprian Dobre, Andrei Ichimescu, and Valentin Cristea. Adaptive Traffic Optimization. In *2012 Sixth International Conference on Complex, Intelligent, and Software Intensive Systems*, pages 761–766. Ieee, July 2012. ISBN 978-1-4673-1233-2. doi: 10.1109/CISIS.2012.112. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6245666`.

K. Dresner. Multiagent traffic management: Opportunities for multiagent learning. *Learning and Adaption in Multi-Agent Systems*, 3898:129–138, 2006.

K. Dresner and P. Stone. Multiagent traffic management: A reservation-based intersection control mechanism. In *AAMAS '04 Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 2*, volume 2, pages 530–537. IEEE Computer Society, 2004.

K. Dresner and P. Stone. Multiagent traffic management: An improved intersection control mechanism. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 471–477. ACM, 2005.

K. Dresner and P. Stone. Traffic intersections of the future. *AAAI'06 proceed-*

ings of the 21st national conference on Artificial intelligence, 2:1593–1596, 2006.

K. Dresner and P. Stone. A multiagent approach to autonomous intersection management. *Journal of Artificial Intelligence Research*, 31:591–656, 2008.

Stephan Eichler, Christoph Schroth, Timo Kosch, and Markus Strassberger. Strategies for context-adaptive message dissemination in vehicular ad hoc networks. In *Mobile and Ubiquitous Systems: Networking & Services, 2006 Third Annual International Conference on*, pages 1–9, 2006. ISBN 0780397924.

Madeleine El-Zaher, Franck Gechter, P. Gruer, and M. Hajjar. A new linear platoon model based on reactive multi-agent systems. In *Tools with Artificial Intelligence (ICTAI), 2011 23rd IEEE International Conference on*, pages 898–899. Ieee, November 2011.

Elena Fasolo, Andrea Zanella, and Michele Zorzi. An effective broadcast scheme for alert message propagation in vehicular ad hoc networks. *Communications, 2006. ICC'06. ...*, 9:3960–3965, 2006.

Michel Ferreira, Ricardo Fernandes, Hugo Conceição, Wantanee Viriyasitavat, and Ozan K. Tonguz. Self-organized traffic control. In *Proceedings of the seventh ACM international workshop on VehiculAr InterNETworking - VANET '10*, pages 85–90, New York, New York, USA, 2010. ACM Press. ISBN 9781450301459.

Marco Fiore, J Harri, Fethi Filali, and Christian Bonnet. Vehicular mobility simulation for VANETs. In *Simulation Symposium, ... Simulation Symposium, 2007. ANSS '07. 40th Annual*, pages 301–309, 2007. ISBN 0769528147.

Takeshi Fujiki, Masayuki Kirimura, Takaaki Umedu, and Teruo Higashino. Efficient Acquisition of Local Traffic Information using Inter-Vehicle Communication with Queries. In *Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE*, pages 241–246. Ieee, September 2007. ISBN 978-1-4244-1395-9.

C. Gershenson. Self-organizing Traffic Lights. *Complex Systems*, 16:29–53, 2005.

Eugenio Giordano, Enzo De Sena, Giovanni Pau, and Mario Gerla. VERGILIUS: A Scenario Generator for VANET. In *Vehicular Technology Conference (VTC 2010-Spring), 2010 IEEE 71st*, pages 1–5. Ieee, 2010.

GloMoSim. `http://pcl.cs.ucla.edu/projects/glomosim/`, October 2012. Accessed: October 3rd, 2012.

Balkrishna Sharma Gukhool and Soumaya Cherkaoui. IEEE 802.11p modeling in NS-2. In *2008 33rd IEEE Conference on Local Computer Networks (LCN)*, pages 622–626. IEEE, October 2008.

Jianhua He, Hsiao-hwa Chen, Thomas Chen, and Wenqing Cheng. Adaptive congestion control for DSRC vehicle networks. *IEEE Communications Letters*, 14(2):127–129, February 2010. ISSN 1089-7798.

A. Hegyi, B. De Schutter, S. Hoogendoorn, R. Babuska, H. van Zuylen, and H. Schuurman. A fuzzy decision support system for traffic control centers. *ITSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings (Cat. No.01TH8585)*, pages 358–363, 2001.

A Hegyi, S.P. Hoogendoorn, M. Schreuder, H. Stoelhorst, and F. Viti. SPECIALIST: A dynamic speed limit control algorithm based on shock wave theory. In *Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on*, pages 827–832. Ieee, October 2008.

T.H. Heung, T.K. Ho, and Y.F. Fung. Coordinated road-junction traffic control by dynamic programming. *Intelligent Transportation Systems, IEEE Transactions on*, 6(3):341–350, 2005.

J Hoebeke, I Moerman, B Dhoedt, and P Demeester. An Overview of Mobile Ad Hoc Networks: Applications and Challenges. *Journal of the Communications Network*, 3(3):60–66, 2004.

J.H. Holland. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence.* University of Michigan Press, Ann Arbor, 1975.

R. Hoyer and U. Jumar. An advanced fuzzy controller for traffic lights. *Annual Review in Automatic Programming*, 19:67–72, 1994.

J.G. Jetcheva, Y.-C. Hu, S. PalChaudhuri, A.K. Saha, and D.B. Johnson. Design and Evaluation of a Metropolitan Area Multitier Wireless Ad Hoc Network Architecture. In *Mobile Computing Systems and Applications, 2003. Proceedings. Fifth IEEE Workshop on*, pages 32–43, 2003. ISBN 0769519954.

Daniel Jiang and Luca Delgrossi. IEEE 802.11p: Towards an international standard for wireless access in vehicular environments. In *Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE*, pages 2036–2040, 2008. ISBN 9781424416455.

JiST/SWANS. `http://jist.ece.cornell.edu/`, October 2012. Accessed: October 3rd, 2012.

Luo Junhai, Ye Danxia, Xue Liu, and Fan Mingyu. A survey of multicast routing protocols for mobile Ad-Hoc networks. *IEEE Communications Surveys & Tutorials*, 11(1):78–91, 2009. ISSN 1553-877X.

L.P. Kaelbling, M.L. Littman, and A.W. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.

Anan Kaku, Masakazu Mukai, and Taketoshi Kawabe. A centralized control system for ecological vehicle platooning using linear quadratic regulator theory. *Artificial Life and Robotics*, July 2012. ISSN 1433-5298. doi: 10.1007/s10015-012-0019-3.

Frank Kargl and Elmar Schoch. Simulation of MANETs : A Qualitative Comparison between JiST / SWANS and ns-2. In *MobiEval '07 Proceedings of the 1st international workshop on System evaluation for mobile platforms*, pages 41–46, 2007. ISBN 9781595937629.

Feliz Kristianto Karnadi, Zhi Hai Mo, and Kun-chan Lan. Rapid Generation of Realistic Mobility Models for VANET. In *Wireless Communications and Networking Conference, 2007.WCNC 2007. IEEE*, pages 2506–2511. Ieee, 2007. ISBN 1-4244-0658-7.

Tomoya Kitani, Takashi Shinkawa, Naoki Shibata, Keiichi Yasumoto, Minoru Ito, and Teruo Higashino. Efficient VANET-based Traffic Information Sharing using Buses on Regular Routes. In *Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE*, pages 3031–3036, 2008. ISBN 9781424416455.

G Korkmaz, Eylem Ekici, and F Ozguner. An efficient fully ad-hoc multi-hop broadcast protocol for inter-vehicular communication systems. In *Communications, 2006. ICC '06. IEEE International Conference on*, volume 1, pages 423–428, 2006. ISBN 1424403553.

Gökhan Korkmaz, Eylem Ekici, Füsun Özgüner, and Ümit Özgüner. Urban multi-hop broadcast protocol for inter-vehicle communication systems. In *Proceedings of the first ACM workshop on Vehicular ad hoc networks - VANET '04*, pages 76–85, New York, New York, USA, 2004. ACM Press.

S. Lämmer and D. Helbing. Self-control of traffic lights and vehicle flows in urban road networks. *Journal of Statistical Mechanics: Theory and Experiment*, 4, 2008.

J.H. Lee and H. Lee-Kwang. Distributed and cooperative fuzzy controllers for traffic intersections group. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 29(2):263–271, 1999.

Da Li, Hongyu Huang, Xu Li, Minglu Li, and Feilong Tang. A Distance-Based Directional Broadcast Protocol for Urban Vehicular Ad Hoc Network. In *2007 International Conference on Wireless Communications, Networking and Mobile Computing*, pages 1520–1523. Ieee, September 2007. ISBN 1-4244-1311-7.

Ming Li, Kai Zeng, and Wenjing Lou. Opportunistic broadcast of event-driven warning messages in Vehicular Ad Hoc Networks with lossy links. *Computer Networks*, 55(10):2443–2464, July 2011. ISSN 13891286. doi: 10.1016/j.comnet.2011.04.005. URL `http://linkinghub.elsevier.com/retrieve/pii/S1389128611001344`.

FJ Martinez, CK Toh, J.-C. Cano, C.T. Calafate, and P. Manzoni. Realistic radio propagation models (RPMs) for VANET simulations. In *Wireless Communications and Networking Conference, 2009. WCNC 2009. IEEE*, pages 1–6, 2009a. ISBN 9781424429486.

FJ Martinez, CK Toh, JC Cano, CT Calafate, and P Manzoni. A survey and comparative study of simulators for vehicular ad hoc networks (VANETs). *Wireless Communications and Mobile Computing*, 11(7):813–828, 2009b.

FJ Martinez, Manuel Fogue, M. Coll, J. Cano, C.T. Calafate, and P. Manzoni. Assessing the Impact of a Realistic Radio Propagation Model on VANET Scenarios using Real Maps. In *Network Computing and Applications (NCA), 2010 9th IEEE International Symposium on*, pages 132–139, 2010. ISBN 9780769541181.

Dave McKenney. Distributed and Adaptive Traffic Signal Control Within a Realistic Traffic Simulation. Master's thesis, Carleton University, 2011.

D.J. Montana. Strongly typed genetic programming. *Evolutionary computation*, 3(2):199–230, 1995.

D.J. Montana and S. Czerwinski. Evolving control laws for a network of traffic signals. In *GECCO '96 Proceedings of the First Annual Conference on Genetic Programming*, pages 333–338. MIT Press, 1996.

MoreVTS. `http://sourceforge.net/projects/morevts/`, April 2011.

Y. Murat and E. Gedizlioglu. A fuzzy logic multi-phased signal control model for isolated junctions. *Transportation Research Part C: Emerging Technologies*, 13(1):19–36, February 2005.

T. Nadeem, S. Dashtinezhad, L Chunyuan, and L. Iftode. Trafficview: a scalable traffic monitoring system. In *Mobile Data Management, 2004.*

Proceedings. 2004 IEEE International Conference on, pages 13–26. Ieee, 2004. ISBN 0-7695-2070-7. doi: 10.1109/MDM.2004.1263039.

Valeri Naoumov and Thomas Gross. Simulation of large ad hoc networks. In *MSWIM '03 Proceedings of the 6th ACM international workshop on Modeling analysis and simulation of wireless and mobile systems*, pages 50–57, 2003. ISBN 1581137664.

M Nekovee and BB Bogason. Reliable and Efficient Information Dissemination in Intermittently Connected Vehicular Adhoc Networks. *Vehicular Technology Conference, 2007. VTC2007-Spring. IEEE 65th*, pages 2486–2490, 2007.

J. Niittymaki and M. Pursula. Signal control using fuzzy logic. *Fuzzy sets and systems*, 116(1):11–22, 2000.

ns-2. `http://www.isi.edu/nsnam/ns/`, October 2012. Accessed: October 3rd, 2012.

NS2 Simulation with 802.11p. `https://sites.google.com/site/nahoons2/english-con/ns2-simulation-with-802-11p`, October 2012.

T Osafune, Lan Lin, and Massimiliano Lenardi. Multi-hop vehicular broadcast (MHVB). In *ITS Telecommunications Proceedings, 2006 6th International Conference on*, pages 757–760, 2006. ISBN 0780395867.

K Ozbay, H Nassif, and S Goel. Propagation characteristics of dynamic information collected by in-vehicle sensors in a vehicular network. In *Intelligent Vehicles Symposium, 2007 IEEE*, pages 1089–1094, 2007. ISBN 1424410681.

Sooksan Panichpapiboon and Gianluigi Ferrari. Irresponsible forwarding. In *ITS Telecommunications, 2008. ITST 2008. 8th International Conference on*, pages 311–316. Ieee, October 2008.

Sooksan Panichpapiboon and Wasan Pattara-atikom. A Review of Information Dissemination Protocols for Vehicular Ad Hoc Networks. *Communications Surveys & Tutorials, IEEE*, 14(3):784–798, 2012.

C.P. Pappis and E.H. Mamdani. A Fuzzy Logic Controller for a Trafc Junction. *Systems, Man and Cybernetics, IEEE Transactions on*, 7(10):707–717, 1977.

CE Perkins and EM Royer. Ad-hoc on-demand distance vector routing. In *Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA '99. Second IEEE Workshop on*, pages 90–100, 1999.

C.E. Perkins, E.M. Royer, S.R. Das, and M.K. Marina. Performance comparison of two on-demand routing protocols for ad hoc networks. *Personal Communications, IEEE*, 8(1):16–28, 2001. ISSN 10709916.

M.L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* Wiley Series in Probability and Mathematical Statistics: Applied Probability and Statistics. John Wiley & Sons, Inc., 1994.

A. Rathi. A control scheme for high density sectors. *Transportation Research Part B: Methodological*, 22(2):81–101, 1988.

J. Sánchez, M. Galán, and E. Rubio. Genetic algorithms and cellular automata: A new architecture for traffic light cycles optimization. In *Evolutionary Computation, 2004. CEC2004. Congress on*, volume 2, pages 1668–1674. IEEE, 2004.

J. Sánchez, M. Galán, and E. Rubio. Applying a Traffic Lights Evolutionary Optimization Technique to a Real Case: "Las Ramblas" Area in Santa Cruz de Tenerife. In *IEEE Transactions on Evolutionary Computation*, volume 12, pages 25–40. IEEE, 2008.

J. Sánchez, M. Galán, and E. Rubio. Traffic signal optimization in "la almozara" district in saragossa under congestion conditions, using genetic algorithms, traffic microsimulation, and cluster computing. *Intelligent Transportation Systems, IEEE Transactions on*, 11:132–141, March 2010.

Ramon S. Schwartz, Rafael R. R. Barbosa, Nirvana Meratnia, Geert Heijenk, and Hans Scholten. A Simple and Robust Dissemination protocol for VANETs. In *2010 European Wireless Conference (EW)*, pages 214–222. Ieee, 2010. ISBN 978-1-4244-5999-5.

Pravin Shankar, Tamer Nadeem, J. Rosca, and L. Iftode. Cars: Context-aware rate selection for vehicular networks. In *Network Protocols, 2008. ICNP 2008. IEEE International Conference on*, pages 1–12, 2008.

Ajit Singh, Mukesh Kumar, Rahul Rishi, and DK Madan. A Relative Study of MANET and VANET: Its Applications , Broadcasting Approaches and Challenging Issues. *Advances in Networks and Communications*, 132(3): 627–632, 2011.

Michael Slavik and Imad Mahgoub. Stochastic Broadcast for VANET. In *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, pages 1–5, 2010. ISBN 9781424451760.

C Sommer and F Dressler. Progressing toward realistic mobility models in VANET simulations. *Communications Magazine, IEEE*, 46(11):132–137, 2008.

J.C. Spall and D.C. Chin. A model-free approach to optimal signal light timing for system-wide traffic control. In *Proceedings of 1994 33rd IEEE Conference on Decision and Control*, pages 1868–1875. IEEE, 1994.

J.C. Spall and D.C. Chin. Traffic-responsive signal timing for system-wide traffic control. *Transportation Research Part C: Emerging Technologies*, 5 (3-4):153–163, August 1997.

D. Srinivasan, M.C. Choy, and R.L. Cheu. Neural Networks for Real-Time Traffic Signal Control. *IEEE Transactions on Intelligent Transportation Systems*, 7(3):261–272, September 2006.

M. Steingröver, R. Schouten, S. Peelen, E. Nijhuis, and B. Bakker. Reinforcement learning of traffic light controllers adapting to traffic congestion. In

Proceedings of the 17th Belgium-Netherlands Conference on Artificial Intelligence (BNAIC 2005), pages 216–223, 2005.

I Stojmenovic, editor. *Handbook of wireless networks and mobile computing.* Wiley, 2002. ISBN 0471419028.

Yuki Sugiyama, Minoru Fukui, Macoto Kikuchi, Katsuya Hasebe, Akihiro Nakayama, Katsuhiro Nishinari, Shin-ichi Tadaki, and Satoshi Yukawa. Traffic jams without bottlenecks: experimental evidence for the physical mechanism of the formation of a jam. *New Journal of Physics*, 10, 2008.

Mostafa M. I. Taha and Yassin M. Y. Hasan. VANET-DSRC Protocol for Reliable Broadcasting of Life Safety Messages. In *Signal Processing and Information Technology, 2007 IEEE International Symposium on*, pages 104–109. Ieee, December 2007. ISBN 978-1-4244-1834-3.

The City of Los Angeles Transportation Profile. `ladot.lacity.org/pdf/PDF10.pdf`, October 2012. Accessed: October 31st, 2012.

Tiger Products - Geopgraphy - US Censur Bureau. October 2012.

TraNS - Traffic and Network Simulation Environment. October 2012.

UCLA Parsec Programming Language. `http://pcl.cs.ucla.edu/projects/parsec/`, October 2012.

R Uzcategui and G Acosta-Marum. WAVE: A Tutorial. *Communications Magazine, IEEE*, 47(5):126–133, 2009.

M. Vasirani and S. Ossowski. A market-inspired approach to reservation-based urban road traffic management. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pages 617–624. International Foundation for Autonomous Agents and Multiagent Systems, 2009.

AM Vegni and TDC Little. A message propagation model for hybrid vehicular communication protocols. In *Communication Systems Networks and Digital Signal Processing (CSNDSP), 2010 7th International Symposium on*, pages 382–386, 2010. ISBN 9781861353696.

H Wedde, Muddassar Farooq, and Yue Zhang. BeeHive: An efficient fault-tolerant routing algorithm inspired by honey bee behavior. *Ant Colony Optimization and Swarm Intelligence*, 3172:83–94, 2004.

Horst F. Wedde, Sebastian Lehnhoff, and Bernhard van Bonn. Highly dynamic and scalable VANET routing for avoiding traffic congestions. In *Proceedings of the fourth ACM international workshop on Vehicular ad hoc networks - VANET '07*, pages 81–82, New York, New York, USA, 2007. ACM Press. ISBN 9781595937391.

Axel Wegener, H Hellbruck, S. Fischer, C. Schmidt, and S. Fekete. AutoCast: An adaptive data dissemination protocol for traffic information systems. In *Vehicular Technology Conference, 2007. VTC-2007 Fall. 2007 IEEE 66th*, pages 1947–1951, 2007. ISBN 1424402646.

W. Wei and Y. Zhang. FL-FN based traffic signal control. In *Fuzzy Systems, 2002. FUZZ-IEEE'02. Proceedings of the 2002 IEEE International Conference on*, pages 296–300, 2002.

W. Wei, Y. Zhang, J.B. Mbede, Z. Zhang, and S. Jingyan. Traffic signal control using fuzzy logic and MOGA. *Systems, Man, and Cybernetics, 2001 IEEE International Conference on*, 2:1335–1340, 2001.

M.P. Wellman. A market-oriented programming environment and its application to distributed multicommodity flow problems. *Journal of Artificial Intelligence Research*, 1:1–23, 1993.

M.A. Wiering. Multi-agent reinforcement learning for traffic light control. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 1151–1158, 2000.

M.A. Wiering, J. Van Veenen, J. Vreeken, and A. Koopman. Intelligent traffic light control. *ERCIM News, European Research Consortium for Informatics and Mathematics*, 53:40–41, 2003.

M.A. Wiering, J. Vreeken, J. Van Veenen, and A. Koopman. Simulation and optimization of traffic in a city. *Intelligent Vehicles Symposium, 2004 IEEE*, pages 453–458, 2004.

Wikipedia - Routing. `http://en.wikipedia.org/wiki/Routing`, October 2012.

U. Wilensky. NetLogo. `http://ccl.northwestern.edu/netlogo`. Center for Connected Learning and Computer-Based Modeling, Northwestern University. Evanston, IL., 1999.

U. Wilensky. NetLogo Traffic Grid model. `http://ccl.northwestern.edu/netlogo/models/TrafficGrid`. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL., 2003.

L Wischhof, A Ebner, and H Rohling. Information dissemination in self-organizing intervehicle networks. *Intelligent Transportation Systems, IEEE Transactions on*, 6(1):90–101, 2005. ISSN 1524-9050.

Lars Wischhof, André Ebner, and Hermann Rohling. Self-organizing traffic information system based on car-to-car communication: Prototype implementation. In *WIT 2004 - 1st International Workshop on Intelligent Transportation*, pages 49–53, 2004.

N. Wisitpongphan, O.K. Tonguz, J.S. Parikh, P. Mudalige, and V. Bai, F.Sadekar. Broadcast storm mitigation techniques in vehicular ad hoc networks. *Wireless Communications, IEEE*, 14(6):84–94, 2007.

Jia Wu, Abdeljalil Abbas-Turki, and Abdellah El Moudni. Discrete Methods for Urban Intersection Traffic Controlling. In *VTC Spring 2009 - IEEE 69th Vehicular Technology Conference*, pages 1–5. Ieee, April 2009.

Huaying Xu and Matthew Barth. An Adaptive Dissemination Mechanism for Inter-Vehicle Communication-Based Decentralized Traffic Information

Systems. In *Intelligent Transportation Systems Conference, 2006. ITSC '06. IEEE*, pages 1207–1213, 2006. ISBN 1424400945.

Yi Yang and Rajive Bagrodia. Evaluation of VANET-based advanced intelligent transportation systems. In *VANET '09 Proceedings of the sixth ACM international workshop on VehiculAr InterNETworking*, pages 3–12, 2009. ISBN 9781605587370.

S Yousefi, MS Mousavi, and Mahmood Fathy. Vehicular ad hoc networks (VANETs): challenges and perspectives. In *ITS Telecommunications Proceedings, 2006 6th International Conference on*, pages 761–766, 2006.

Sherali Zeadally, Ray Hunt, Yuh-Shyan Chen, Angela Irwin, and Aamir Hassan. Vehicular ad hoc networks (VANETS): status, results, and challenges. *Telecommunication Systems*, pages 1–25, December 2010. ISSN 1018-4864.

Jie Zhang. A Survey on Trust Management for VANETs. In *2011 IEEE International Conference on Advanced Information Networking and Applications*, pages 105–112. Ieee, March 2011. ISBN 978-1-61284-313-1.

Ting Zhong, Bo Xu, and Ouri Wolfson. Disseminating real-time traffic information in vehicular ad-hoc networks. In *Intelligent Vehicles Symposium, 2008 IEEE*, pages 1056–1061, 2008. ISBN 9781424425693.