Master of Science in Network Computing

Department of Information Technology and Telecommunications

DATA DISSEMINATION
IN VEHICULAR AD HOC NETWORKS (VANET)

MEGA ANASTASIA

SUPERVISOR PROFESSOR:

NICHOLAS S. SAMARAS

LARISSA SPRING 2012

Master of Science in Network Computing

Department of Information Technology and Telecommunications

THESIS DISSERTATION

DATA DISSEMINATION
IN VEHICULAR AD HOC NETWORKS (VANET)

MEGA ANASTASIA

Inquiry Commission:

1. Reviewer : Nicholas S. Samaras
2. Reviewer : Konstantinos Kokkinos
3. Reviewer :

ABSTRACT

With the proliferation of sensor nodes and the development in wireless communication technologies, Wireless Sensor Networks (WSNs) have gained worldwide attention in recent years. They facilitate monitoring and controlling of physical environments from remote locations with great accuracy and represent a significant improvement over wired sensor networks. Their function is to collect and disseminate critical data, while their position does not need to be engineered or predetermined, in contrast to the wired ones. This also means that WSN protocols must possess self-organizing capabilities, in addition to energy-aware characteristics, given the energy constraints imposed by these networks. Several special mechanisms need to be implemented in all the network layers, from physical to application, in order to ensure a long-lasting operation of the deployed WSN. Some of the most important topics in WSN are OS’s, security and mobility.

In this thesis we cope with the topic of mobility in WSN. More specifically we deal with data dissemination in vehicular ad-hoc networks (VANET) and we provide a comprehensive overview of the main techniques used in this field. The concept of data dissemination is wide and meaningful, and within this work we generally refer to it as the process of spreading some amount of data over a distributed wireless network, which is a superset of a VANET. The goal is to research how some parameters of a network (packet loss, packet delay, throughput) are affected due to different routing protocols, different speeds etc.

In general data exchanging on the roads is becoming more and more interesting, as the number of vehicles equipped with computing technologies and wireless communication devices is poised to increase dramatically. Communications between vehicles and within the same vehicle (inter-vehicle, or InV) is becoming a promising field of research and we are moving closer to the vision of intelligent transportation systems (ITS), which can enable a wide range of applications, such as collision avoidance, emergency message dissemination, dynamic route scheduling, real-time traffic condition monitoring.

Through the simulations and the results for different communication modes, it is proved that there is no efficient routing protocol for all parameters that we studied such as packet loss percentage and transferred throughput due to different characteristics of routing protocols (i.e.

AODV,DSDV,DSR). We observe that the density of the network (50,100,150 nodes), the speed value and the communication modes are important and affect the results.

INDEX TERMS

VANET, simulation, inter-vehicle communication, vehicular network, ad hoc network, packet loss, packet delay, packet jitter, transferred throughput ,VANET routing protocols.

To my parents and

my sister

ACKNOWLEDGEMENTS

Reaching the end of this two-year journey, and looking back I see that all the hard work and efforts of these months worth. I feel that through this master, but and from any form of education and knowledge I have improved as a scientist and as a person.

On deep gratitude I would like to thank first of all my supervisor Mr. Nicholas Samaras principle because it gave me the opportunity to write this diplomatic work, and then for their invaluable help, guidance, the absolute understanding that showed me all this time and excellent cooperation. I would to thank my parents and all family members who energized and supported me to start this journey. Last but not least, I would like to thank my friend Alexander Keramidas for the valuable help.

Mega Anastasia

Larissa, May 2012

Table of Contents

TABLE OF ITEMS

CHAPTER 1

INTRODUCTION

Over the past few years, the technology needed to support vehicular ad-hoc networks (VANETs) has evolved quickly and has been adopted and supported by many automobile manufacturers (*e.g.* Honda, Toyota, etc.). This industrial support has resulted in many ideas for applications targeting VANETs as their platform. These applications can be classified into various categories (*e.g.* safety, informational and entertainment).

Vehicular networks are usually designed according to prefixed goals. The extant literature has explored two primary varieties of communications, *infrastructure-to-vehicle* (*i2v*) and *vehicle-to-vehicle* (*v2v*). While in the first vehicles exchange their data with infrastructures mainly for information or data gathering applications [1], in the second variety they communicate with each other for safety, security or dissemination applications [2]. In addition, the literature explored a third type of communication, *infrastructure-to-vehicle- to-vehicle* (*i2v2v*). This approach merges the typologies of communication mentioned above to mainly deploy a system for internet content distribution through vehicles as shown in [3] and in [4].

Statistics from the Road Safety Cell (RSC) of the Bangladesh Road Transport Authority (BRTA) show the annual fatality rate in road accidents in Bangladesh is 85.6 per10,000 vehicles which compares to rates of below 3 per 10,000 vehicles in most developed countries [5]. Another study estimates the national cost of such road traffic crashes in Bangladesh at Taka 45 billion (US$ 76 million) which is more than 1.6% of the country’s GDP (Gross Domestic Product) [6]. Consequently, the need for having an efficient traffic controlling and monitoring system is very demanding. Deployment of smart sensors along with the roadside made many developed countries possible to collect live data or to monitor irresponsible vehicle violating the speed limit. In case of Bangladesh, placing sensor nodes at the identified spots with the addition of wireless communications could help for the development of a smart traffic monitoring system. Each sensor needs to be placed within predetermined optimum distance ranges, following a special node. This special node would act as local base station equipped with data transmission capabilities to the headquarters. Once the sensed data are received, central station might broadcast the message through radio stations.

Regardless of their category, most of these applications require some common vehicular data, *i.e.* speed and position (X,Y), to be able to function properly. The overall system is similar to the proposed flood controlling system, except sensor placement locations. Apart from above mentioned system, a simple traffic signal system could also be equipped with intelligent sensing devices at the road intersections. The smart sensing device would gather information of upcoming objects towards the intersection; perform scheduling to determine the time-to-wait (TTW) interval for signals to be changed. Time-to-wait is determined by the time gap between different crossing objects. Hence, automated signal changing would never keep the motorist waiting in one side for a longer time. Not to mention, TTW could also be human operated for special purposes like medical, military or other emergency situations.

The data characteristics (packet loss percentage, packet delay, packet jitter etc.) are very important for every network and especially for VANET, these are very crucial due to the fact that have to do with human's lives. So the goal is to work out how these characteristics change when we diversify the speed, the nodes , the routing protocol and the communication.

1.1 Thesis Organisation

In this section we give a brief insights on this research work by showing the organization of the other Chapters. In detail, we have:

Chapter 2: Preliminary concepts – Background

This chapter is meant to introduce the basic concepts such as intelligent transportation system, intelligent vehicles and analyze vanet basics and the requirements that are needed in the applications. In the other words the chapter tries to make sense the necessary meanings about "Vanets".

Chapter 3: State of the art

We briefly present related work pertaining to data compression and aggregation, position verification, data dissemination.

Chapter 4: Mobility models in vehicular networks

This Chapter describes several mobility models used to simulate vehicular ad-hoc networks.

Chapter 5: Problem statement and Methodology choice

In this chapter, we refer the problem, the parameters that are involved and the questions that maybe be answered. Also, we present a survey of the most popular vehicular network simulators and mobility simulators. Furthermore, we become an introduction to the simulators that will be used and after that we give details about the creation of the map etc.

Chapter 6: Simulation models and results

This is the main chapter because contains the plots that extract after the simulation stage in Linux platform and the results.

Chapter 7: Conclusion

It contains the overall conclusion and possible future work in order the thesis to be expanded.

I hope you enjoy this thesis until the end as I do.

CHAPTER 2

PRELIMINARY CONCEPTS- BACKGROUNG

Through the last years, control systems for the automotive industry have moved from the analog to the digital domain. Networked Electronic Control Units (ECUs) are increasingly being deployed in automobiles to realize diverse functions such as engine management, air-bag deployment, and even in intelligent brake systems. At the same time, emerging vehicular networks in the forms of Intra-Vehicle (InV), Vehicle-to-Vehicle (V2V), and Vehicle-to-Infrastructure (V2I) communications are fast becoming reality and will enable a variety of applications for safety, traffic efficiency, driver assistance, as well as infotainment to be incorporated into modern automobile designs.

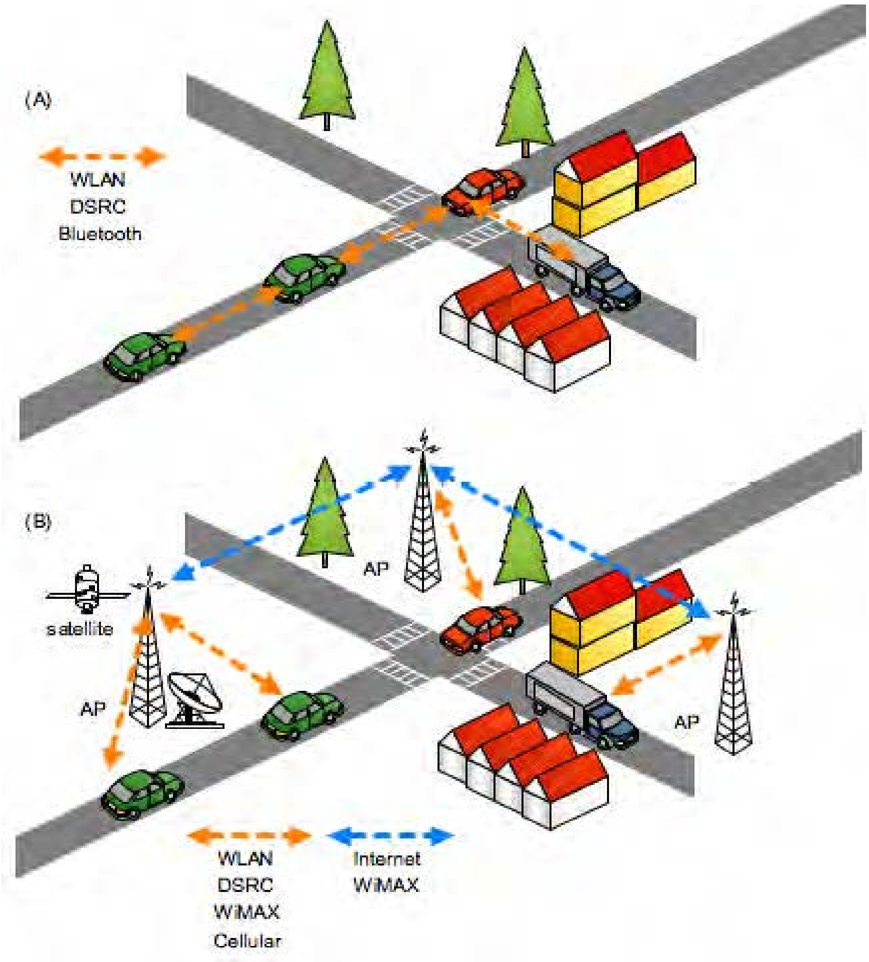


Figure 2.0.1 - Communications schemes in Vanets

The figure above shows the communication schemes typically used in a VANET: (a) Vehicle-to-Vehicle (V2V), (b) Vehicle-to-Infrastructure (V2I) and Infrastructure-to-Vehicle (I2V).

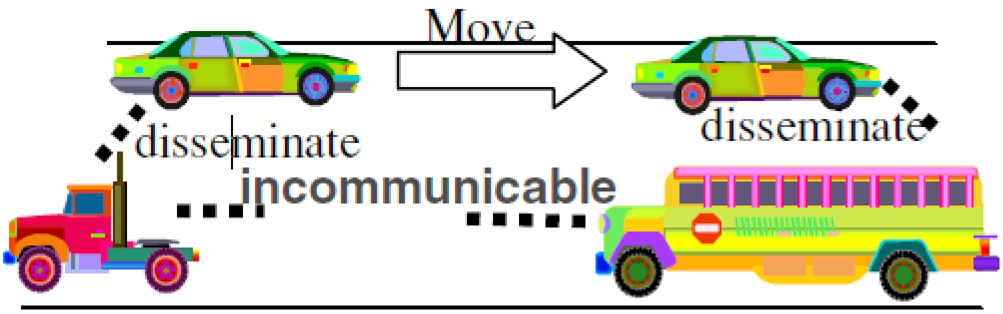


Figure 2.0.2 Inter-vehicle Ad-hoc Communication

Needless to say these communications in vehicles tend to do true an intelligent transportation system. In order to become clear what it is the intelligent transportation system we give some details below.

2.1 Network Architectures

MANETs generally do not rely on fixed infrastructure for communication and dissemination of information. VANETs follow the same principle and apply it to the highly dynamic environment of surface transportation. As shown in Figure 1, the architecture of VANETs falls within three categories: pure cellular/WLAN, pure ad hoc, and hybrid. VANETs may use fixed cellular gateways and WLAN access points at traffic intersections to connect to the Internet, gather traffic information or for routing purposes.

The network architecture under this scenario is a pure cellular or WLAN structure as shown in Figure 2.1(a). VANETs can combine both cellular network and WLAN to form the networks so that a WLAN is used where an access point is available and a 3 connection otherwise. Stationary or fixed gateways around the sides of roads could provide connectivity to mobile nodes but are eventually unfeasible considering the infrastructure costs involved. In such a scenario, all vehicles and roadside wireless devices can form a mobile ad hoc network (Figure 2.1(b)) to perform vehicle-to-vehicle communications and achieve certain goals, such as blind crossing (a crossing without light control). A hybrid architecture (Figure 2.1(c)) of combining cellular, WLAN and ad hoc networks together has also been a possible solution for VANETs.

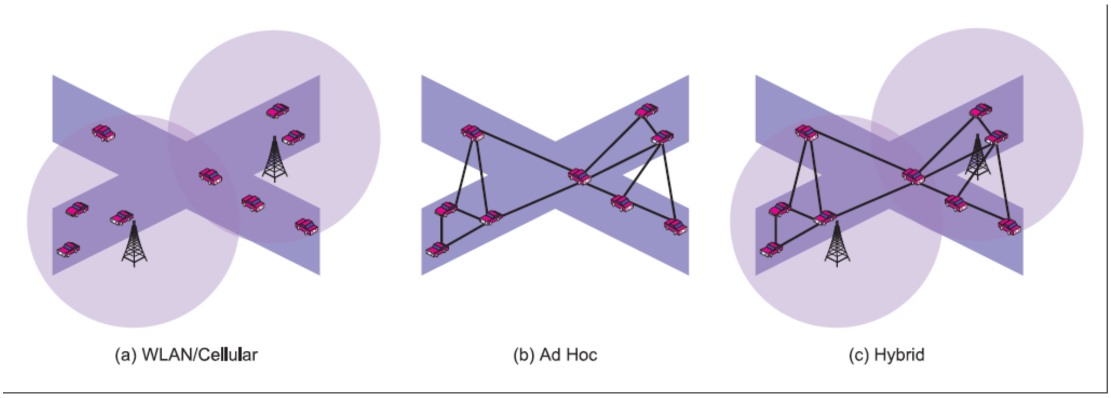


Figure 2.1 - Three possible network architectures for Vanets

2.2 Intelligent Transportation System

The Intelligent Transportation System (ITS) program is a federal initiative to improve the safety and economics of public transportation using advanced communication technologies (USDOT Intelligent Transportation System). The ITS program belongs to the United States Department of Transportation (USDOT). The focus is the development of two components: intelligent vehicles and intelligent infrastructures for the creation of an overall intelligent transportation system that integrates the two components. Between the two components, there are 16 types of supporting technologies that the ITS program defines. The distribution of the technologies is provided next [5].

2.3 Intelligent Infrastructure and Intelligent Vehicles

An intelligent infrastructure [5] requires applications that provide arterial management, freeway management, transmit management, incident management, emergency management, electronic [toll] payment, traveler information, information management, crash prevention and safety, roadway operations and maintenance, road weather management, commercial vehicle operations, and intermodal freight. The details for each of these applications are outside the scope of this thesis but at a high level the focus is the aggregation of traffic data for analysis, management, and dissemination.

The intelligent vehicles component of the ITS require three safety systems: collision avoidance, collision notification, and driver assistance. Collision avoidance systems will use sensors to monitor the vehicle’s surrounding so that the driver can be alerted if a collision could occur. Some examples are forward collision warning and road departure warning systems. The ITS program has documented several interesting studies regarding the intelligent vehicles component and collision avoidance systems. These are federal studies (USDOT IVI 2002) known as Vehicle Infrastructure Integration (VII) and Integrated Vehicle Based Safety Systems (IVBSS) [5].

The VII initiative concept of operations will use vehicle-vehicle and vehicle-road communication that would lower the rate of accidents resulting from roadway departures and at intersections. The ITS states that 21,000 of the 43,000 annual deaths occur because of roadway departure and intersection related incidents. The concept of operations would enable a vehicle to send and receive message to the devices on the roadway and intersections. This could prevent accidents and supply more data for traffic analysis.

The IVBSS initiative concept of operations will embed multiple sensors in vehicles to provide warnings of hazardous conditions to the driver. The IVBSS study investigates on the best way to communicate these warnings to the driver and test plans for prototype safety systems. The scenarios the IVBSS addresses are read-end, run-off-road, and lane change crashes. For rear-end collision avoidance, a proposed safety system will be able to detect objects in front of the vehicle and react by initiating vehicle brakes. The technologies that can be leveraged are GPS, forward looking radar, and cameras onboard the vehicle. For road departure collision avoidance, a proposed safety system will be able to detect road geometry in order to give warnings to the driver if the vehicle is deviating from the lane of travel. Finally, to avoid lane change and merge collisions, a proposed safety system that can detect the presence of adjacent vehicles could warn the driver during a lane change maneuver. Sensors that could detect vehicles are forward, rear, and side-looking radar and vision-based cameras. The challenge of the three avoidance systems is to avoid false alarms and driver confusion regarding the warnings from these systems [5].

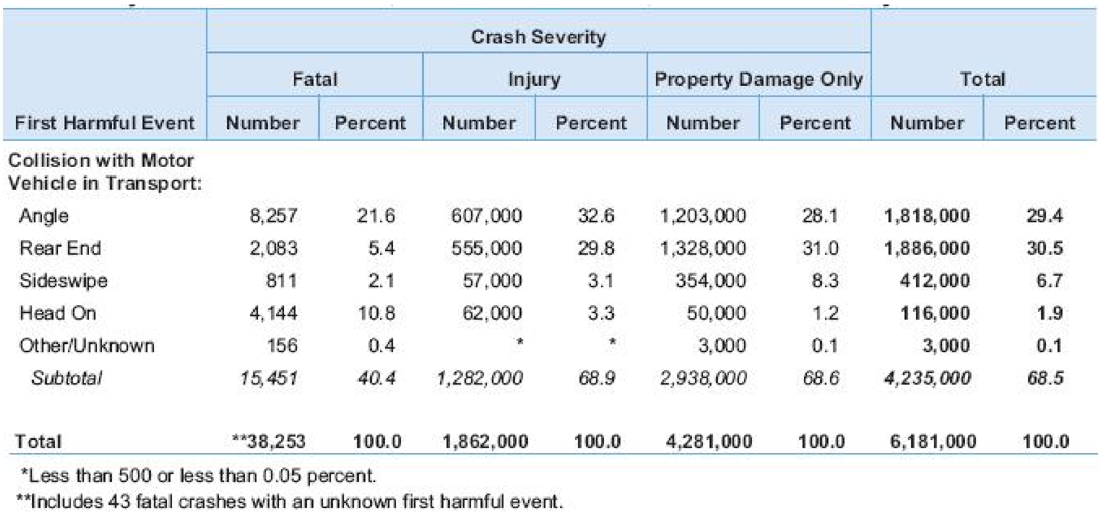
To conclude, the goal of the ITS program can be paraphrased as a guidance of functions and data flows, namely an architecture and standards, that are required to be implemented to be a contributor to the nationwide intelligent transportation system. The ITS program also provides a wealth of research and development efforts to develop safety and traffic detection prototypes.

While some implementing technologies are defined by the ITS program, such as Dedicated Short Range Communication (DSRC), other technologies are being created independently. This permits the greatest flexibility for newer implementing technology so long as the solutions meet the requirements stated in the ITS architecture [5], [7].

2.4 Traffic Statistics

In this section, statistics for motor vehicle crashes provide evidence that accidents truly impact the health and well being of more people than just what is announced on the news. It is this author's hope that newer safety systems based on the principle of sharing vehicle data such as demonstrated in the prototype can reduce the number of injuries and fatalities that occur during motor vehicle operation. In fact, the National Highway Traffic Safety Administration, NHTSA, has conducted studies that forecasts the positive effect newer safety systems would have on crash statistics. These are presented at the end of the section [5].

The following table is an excerpt from the 2004 Final Traffic Safety Facts Annual Report, which provides statistics on the number of rear-end crashes in the United States in 2004 (NCSA 2004). The number of fatal rear-end events was 2083 people, and over 500,000 people were injured [5].



	Crash Severity							
	Fatal		Injury		Property Damage Only		Total	
First Harmful Event	Number	Percent	Number	Percent	Number	Percent	Number	Percent
Collision with Motor Vehicle in Transport:								
Angle	8,257	21.6	607,000	32.6	1,203,000	28.1	1,818,000	29.4
Rear End	2,083	5.4	555,000	29.8	1,328,000	31.0	1,886,000	30.5
Sideswipe	811	2.1	57,000	3.1	354,000	8.3	412,000	6.7
Head On	4,144	10.8	62,000	3.3	50,000	1.2	116,000	1.9
Other/Unknown	156	0.4	*	*	3,000	0.1	3,000	0.1
Subtotal	*15,451*	*40.4*	*1,282,000*	*68.9*	*2,938,000*	*68.6*	*4,235,000*	*68.5*
Total	****38,253**	**100.0**	**1,862,000**	**100.0**	**4,281,000**	**100.0**	**6,181,000**	**100.0**

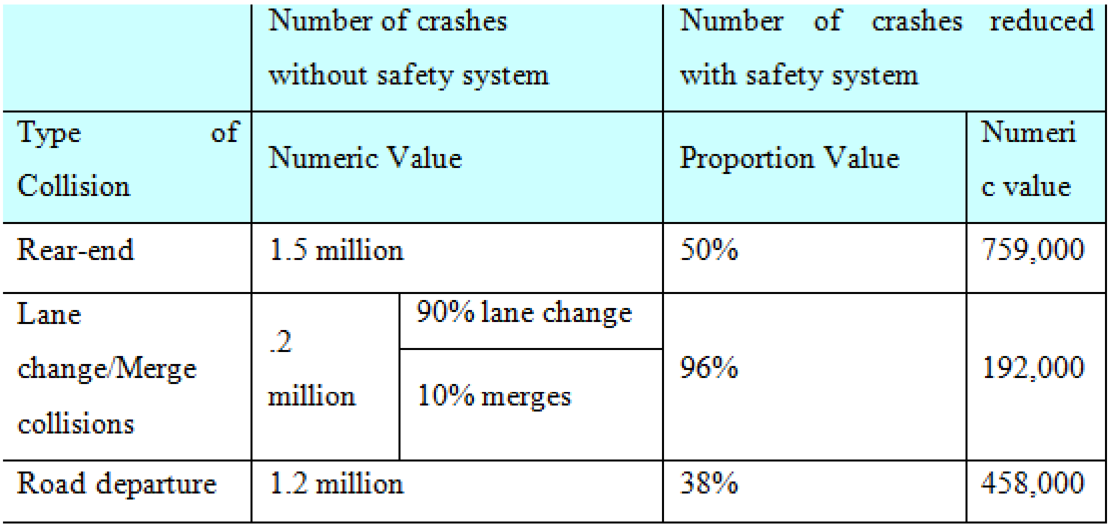
*Less than 500 or less than 0.05 percent.
**Includes 43 fatal crashes with an unknown first harmful event.

Table 1 - 2004 rear-end statistic

The trends in rear-end collisions that have been observed by the ITS Intelligent Vehicles Initiative (USDOT IVI 8 Major Problem Areas) are that rear-end collisions account for one in four crashes or over 1.5 million crashes a year. The NHTSA has estimated that driver warning systems would be effective in preventing nearly 50 percent of rear-end crashes or 759,000 crashes each year [5].

Lane change and merge collisions account for 1/25 crashes. Ninety percent are caused by lane changes and ten percent are caused by merges. A safety system that could detect neighboring cars positions could benefit 192,000 of the 200,000 lane change and merge collisions each year [5].

For vehicles that depart the road completely, a system that could warn the driver if the vehicle is leaving the lane or direction of travel could reduce the statistic that one in five crashes result from a roadway departure. NHTSA estimates that a safety system has the potential of preventing approximately 458,000 of the 1.2 million crashes each year. A summary chart of these statistics has been provided below [5].



	Number of crashes without safety system		Number of crashes reduced with safety system	
Type of Collision	Numeric Value		Proportion Value	Numeric value
Rear-end	1.5 million		50%	759,000
Lane change/Merge collisions	.2 million	90% lane change 10% merges	96%	192,000
Road departure	1.2 million		38%	458,000

Table 2 - Safety systems can reduce crashes

In summary, travel on roadways can be very dangerous. The reality of this danger is indicated in the statistics provided in this section. Variables such as gender, age, automobile type, etc. lead to more statistics.

2.5 VANET Basics

A VANET is a special kind of network in which the vehicles represent the network nodes. The vehicles can communicate with each other in addition to communicating with available supporting infrastructure along the road. Vehicle-to-vehicle (V2V) communication helps in sharing the information within the VANET, while vehicle-to-infrastructure (V2I) communication extends the VANET to include other designated networks (*e.g.* a centralized network or the Internet) and allows sharing of information with them.

VANET applications can be divided into three main categories: safety, informational and entertainment. Safety applications include collision warning [7,8, 9] and merge assistance [10]. Informational applications include notification of upcoming traffic conditions [11] and roadway hazards [12], as well as gathering and disseminating weather information [13]. Reducing the number of accidents can in turn reduce the number of traffic jams, which could reduce the level of environmental impact (Fig. 2.4). The third type of applications, entertainment, includes Internet access [14], multimedia streaming [15], and P2P file sharing [16, 17].

Many VANET applications require each vehicle to share its data (*e.g.* speed and location) with its neighbors through broadcasting a message containing such data. Sending these messages to farther distances will waste the bandwidth and may cause a broadcast storm problem based on the traffic density [18]. So, to share the data with vehicles at farther distances efficiently, many data aggregation techniques have been proposed.

Data aggregation has been proposed in VANETs to solve the bandwidth utilization problem. The basic idea is to gather information about many vehicles into a single frame. Data aggregation techniques can be classified as *syntactic* or *semantic* [19]. Syntactic aggregation uses a technique to compress or encode the data from multiple vehicles in order to fit the data into a single frame. This results in lower overhead than sending each message individually. In semantic aggregation, data from individual vehicles is summarized. For instance, instead of reporting the exact position of five vehicles, only the fact that five vehicles exist is reported. The trade-off is a much smaller message in exchange for a loss of precise data.

Efficient data dissemination in VANETs is an important problem that needs to be handled carefully. The importance of this problem has attracted many researchers to study it and, as a consequence, many data dissemination techniques have been proposed. The proposed techniques can be categorized according to the communication method used V2I, V2V, and hybrid of the two.

In the infrastructure-based techniques, vehicles mainly communicate through infrastructure units that can be road side units (RSUs) [14, 20–22], embedded sensor belts in the road pavement [11], or cellular networks. V2I approaches depend upon a pervasive infrastructure that may not become a reality due to its high cost. V2V techniques for data dissemination can depend on data routing [23–25] or on broadcasting. In the majority of VANET applications (especially safety applications), the exchanged messages have no specific destination but have all the surrounding vehicles as the targeted destinations. This makes broadcast the most suitable method for dissemination. The third category of dissemination techniques combines both V2I and V2V based on their availability [26, 27].

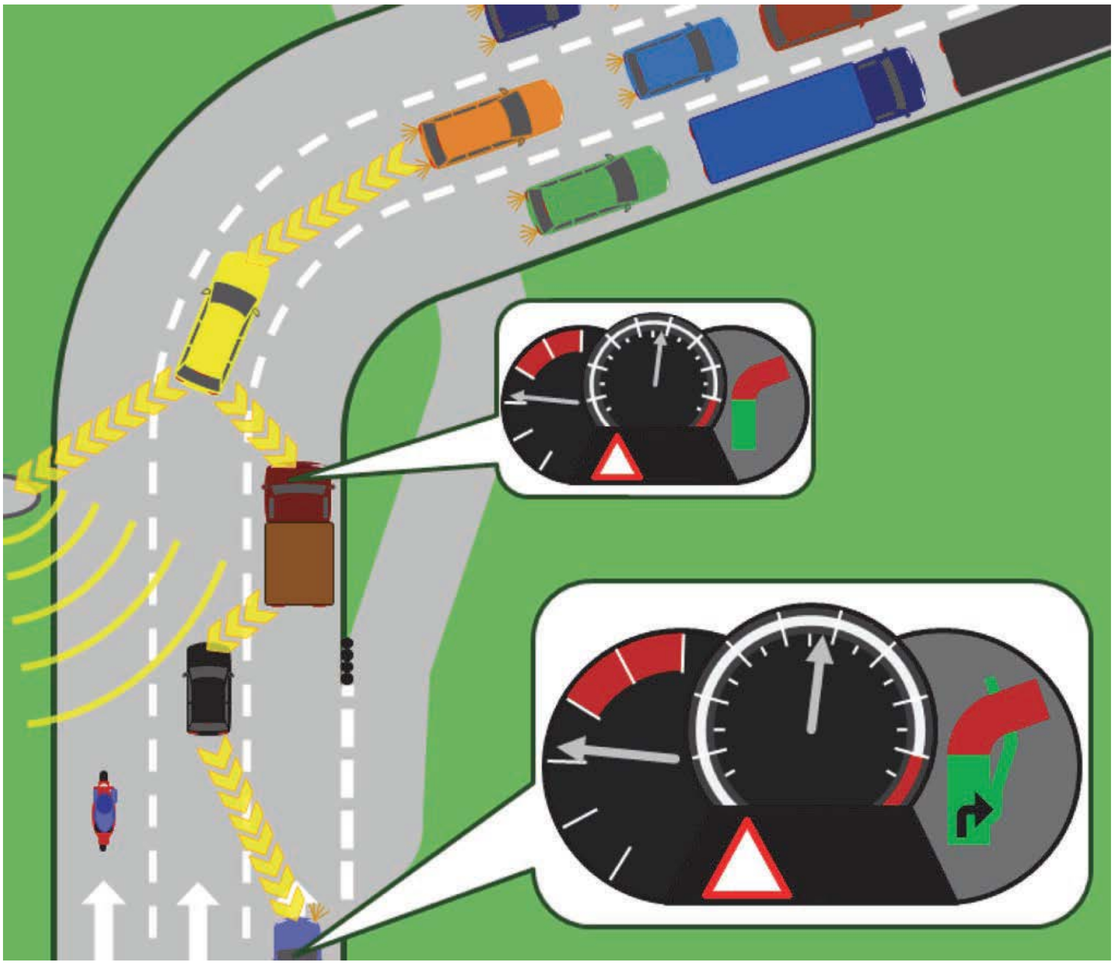


Figure 2.4 - By vehicle-to-vehicle and vehicle-to-roadside communication, accidents can be avoided and traffic efficiency can be increased.

The MAC layer architecture in IEEE 802.11 has two methods for accessing the medium, distributed coordination function (DCF) and point coordination function (PCF). PCF is only useful in the case of infrastructure network configuration, so it is not applicable for VANETs. So, the fundamental medium access method in VANETs is DCF. DCF relies on carrier sense multiple access with collision avoidance (CSMA/CA) for coordinating the medium access. The carrier sense in CSMA/CA can be either performed using physical mechanisms within the physical layer or virtually by the MAC layer using the RTS/CTS mechanism. RTS/CTS (Request to Send/ Clear to Send) is simply a medium reservation mechanism in which the node that has data to send (the source) will first send an RTS frame indicating the reservation time and identifying the destination node. Once the destination receives the RTS frame, it sends a CTS frame that when received by the source triggers sending the data frames. Meanwhile, all the other nodes within the transmission range should stay silent till the reservation time expires.

To carry vehicle data beyond the original sender’s transmission range, data rebroadcast will be needed. Flooding is the typical method for data broadcasting. Basic flooding is defined as whenever a node receives a frame, it will re-broadcast it. This may result in redundant re-broadcasts because the neighbors of the rebroadcasting node may have already received the original frame. Moreover, when multiple nodes in the same vicinity receive a frame, they will all re-broadcast it, causing severe contention on the channel. In addition to the previous two problems, collisions are highly probable due to the lack of RTS/CTS and collision detection mechanisms. So, blindly flooding will waste bandwidth by sending redundant frames that will probably collide. These three problems are collectively known as the *broadcast storm problem* [18]. So, there should be a more efficient method for handling re-broadcast of information to enhance bandwidth utilization and avoid the broadcast storm problem.

2.6 Communication schemes in VANET

Communication of vehicles can be divided in two main groups: vehicle to vehicle communications (V2V) or inter-vehicle communications (InV), and communications with the infrastructure (V2I). In the literature, many authors use V2I to denote both data flow directions, however, according to the use of several technologies for one or the other communication pattern, it is more correct to distinguish between V2I and I2V (infrastructure to vehicle communications). It is important to consider this whole set of communication possibilities for vehicles because, depending on the application or service necessities, we will have to decide among one of the available wireless network technologies. Apart from the communication pattern covered, wireless communication technologies can be divided into those which establish 1-to-1 physical links, and those which consider 1-to-n broadcast ones. In this last case, some kind of access point is in charge of sharing out the available bandwidth among the clients. This bandwidth, though, could become insufficient when the number of served nodes increases inside the coverage area. Due to this, the tendency in short-range wireless technologies is taking advantage of the available bandwidth, sharing it among a small number of users because, anyway, the coverage is small. On the contrary, wide-range technologies must share the available bandwidth among much more users. However, short-range wireless media lack stability, due to the small accessible area. It is also important to remark how V2V communications are obtained by means of 1-to-1 technologies, and communications with the infrastructure are commonly created using the 1-to-n ones.

2.7 Overview of communication system

In order to make clear the communication system, it will be presented the CAR-2-X (with this term it is intended car-to-car and car-to-infrastructure communication) system architecture and details of Geocast protocols (Geocast refers to the delivery of information to a group of destinations in a network identifed by their geographical locations. It is a specialized form of multicast used by some routing protocols for mobile ad hoc networks. Geocast routing protocols are extremely important in vehicular networks.) that serve as a basic building block for CAR-2-X communication in many European R&D projects. As shown in Figure 2.6, the CAR-2-X communication system consists of three domains: the in-vehicle domain, the ad hoc domain, and the infrastructure domain. The in-vehicle domain refers to a network logically composed of an On-Board Unit (OBU, which is responsible for CAR-2-X communication) and (potentially multiple) Application Units (AUs). It also provides communication services to AUs and forwards data on behalf of other OBUs in the ad hoc domain. An OBU is equipped with at least a single network device for short-range wireless communications based on IEEE 802.11p radio technology, and may also be equipped with more network devices, for example, for non-safety communications, based on other radio technologies such as IEEE 802.11a/b/g/n . An AU is typically a dedicated device that executes a single or a set of applications and uses the OBUs communication capabilities. An AU can be an integrated part of a vehicle and be permanently connected to an OBU. It can also be a portable device such as laptop, PDA, or game pad that can dynamically attach to and detach from) an OBU. An AU and an OBU are usually connected with a wired connection, but the connection can also be wireless, using Bluetooth, WUSB, or UWB (Wireless USB is a short-range, high-bandwidth wireless radio communication protocol created by the Wireless USB Promoter Group. Wireless USB is sometimes abbreviated as "WUSB", although the USB Implementers Forum discourages this practice and instead prefers to call the technology "Certified Wireless USB" to differentiate it from competitors. Wireless USB is based on the WiMedia Alliance's Ultra-WideBand (UWB) common radio platform, which is capable of sending 480 Mbit/s at distances up to 3 meters and 110 Mbit/s at up to 10 meters. It was designed to operate in the 3.1 to 10.6 GHz frequency range, although local regulatory policies may restrict the legal operating range for any given country.)

The ad hoc domain, or vehicular ad hoc network (VANET), is composed of vehicles equipped with OBUs and stationary units along the road, termed road-side units (RSUs). OBUs form a mobile ad hoc network (MANET), which allows communications among nodes in a fully distributed manner without the need for centralized coordination. OBUs directly communicate if wireless connectivity exists among them.

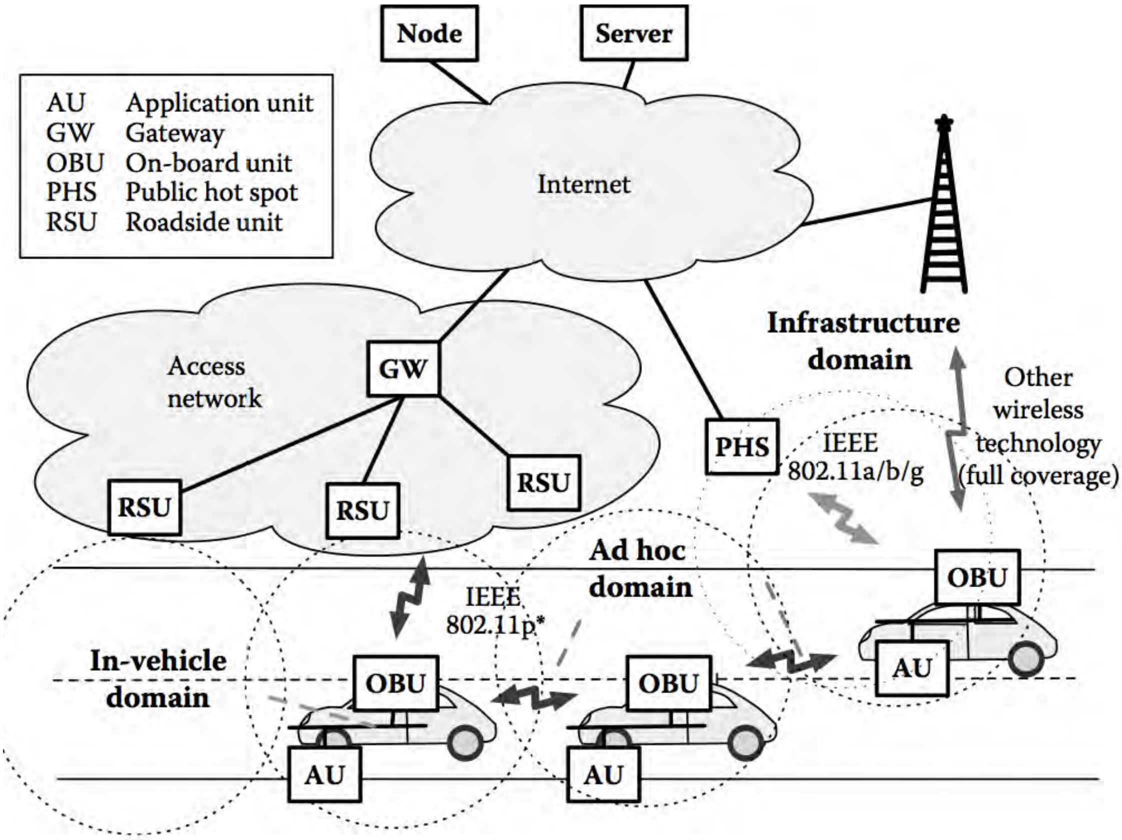


Figure 2.6 - CAR-2-X communication scheme

In above picture a typical VANET scenario showing a CAR-2-X communication system and involved protocols of the IEEE 802 family. As shown in the figure, the CAR-2-X communication system consists of three domains: the in-vehicle domain, the ad hoc domain, and the infrastructure domain.

In the case of no direct connectivity, dedicated routing protocols allow multihop communications, where data are forwarded from one OBU to another, until they reach the destination. An RSU can be attached to an infrastructure network, which in turn can be connected to the Internet. As a result, RSUs may allow OBUs to access the infrastructure. In this way it is possible for AUs registered with an OBU to communicate with any host on the Internet, when at least one infrastructure-connected RSU is available.

2.8 The need of Geocast protocols

Geocast is basically an ad hoc routing protocol using geographical positions for data transfer. Its basic principles were originally proposed as an alternative to pure topology-base internetworking [28] and in mobile ad hoc networks [29]. Geocast assumes that vehicles acquire information about their position via GPS or any other positioning system. Every vehicle periodically advertises this information to its neighbor vehicles and a vehicle is thus informed about all other vehicles located within its direct communication range. If a vehicle intends to send data to a known target geographic location, it chooses another vehicle as a message relay, which is located in the direction towards the target position. The same procedure is executed by every vehicle on the multihop path until the destination is reached. This approach does not require establishment and maintenance of routes. Instead, packets are forwarded on the y based on the most recent geographic positions. In detail, Geocast assumes that every node knows its geographical position and maintains a location table containing other nodes and their geographical positions. Geocast supports point-to-point and point-to-multipoint communication.Core protocol components of Geocast are beaconing, a location service, and forwarding. With beaconing, nodes periodically broadcast short packets with their ID, current geographical position, speed, and heading. On reception of a beacon, a node stores the information in its location table. The location service resolves a nodes ID to its current position. When a node needs to know another nodes position that is currently not available in its location table, it issues a location query message with the sought node ID, sequence number, and hop limit. Neighboring nodes rebroadcast this message until it reaches the sought node (or the hop limit). If the request is not a duplicate, the sought node answers with a location reply message carrying its current position and a time stamp. On reception of the location reply, the originating node updates its location table. Forwarding basically means relaying a packet towards the destination. The most innovative method for distribution of information enabled by geographical routing is to target data packets to certain geographical areas.

Geocast routing [30] is basically a location-based multicast routing. The objective of a geocast routing is to deliver the packet from a source node to all other nodes with a specified [31],[32] geographical region (*Zone of Relevance*, ZOR). Many VANET applications will benefit from geocast routing. For example, a vehicle identifies itself as crashed by vehicular sensors that detect events like airbag ignition, then it can report the accident instantly to nearby vehicles. Vehicles outside the ZOR are not alerted to avoid unnecessary and hasty reactions. In this kind of scenarios the source node usually resides inside the ZOR. See Figure 2.7.2 for an illustration of difference among unicast, broadcast and geocast in VANETs.

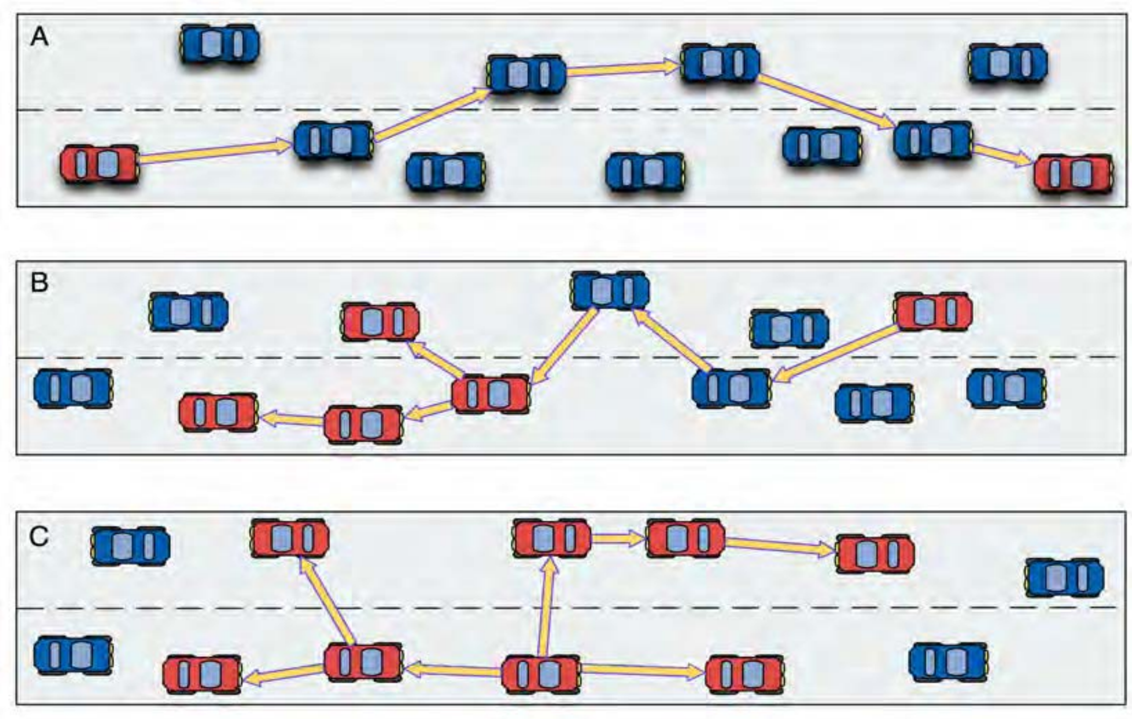


Figure 2.7.1 - Geocasting forwarding schemes. Typical geocasting forwarding schemes: unicast (A), broadcast (B), and topologically scoped broadcast (C).

To conclude, in general Geocast includes the following forwarding schemes:

- GeoUnicast

According to this scheme, when a node wishes to send a unicast packet, it first determines the destinations position (by location table look-up or the location service) and then forwards the data packet to the node towards the destination, which in turn reforwards the packet along the path until the packet reaches the destination.

- GeoBroadcast

In this scheme, data packets are distributed by flooding where nodes rebroadcast the packets if they are located in the geographical region determined by the packets. This simple flooding scheme is enhanced with techniques based on packet numbering to alleviate the effects of so called broadcast storms. Broadcast storms (a typical problem in wireless ad hoc networks) occur when multiple nodes simultaneously rebroadcast a data packet that they have just received. GeoAnycast is similar to GeoBroadcast but addresses a single node (i.e., any node) in a geographical area.

- Topologically scoped broadcast (TSB)

According to this scheme, data packets are broadcasted from a source to all nodes in the n-hop neighborhood. Single-hop broadcast are a specific case of TSB and are used to send periodic messages (beacons or heartbeats).

2.9 Security at VANET

Security is also a very crucial problem in VANETs but in this thesis we do not deal with this aspect extensively. Vehicular network challenges include technical problems like key distribution as well as more abstract difficulties.

Authentication versus Privacy: In a vehicular network, we would like to bind each driver to a single identity to prevent Sybil [35] or other spoofing attacks. However, drivers value their privacy and are unlikely to adopt systems that require them to abandon their anonymity.

Balancing privacy concerns with security needs will require codifying legal, societal and practical considerations. Most countries have widely divergent laws concerning their citizens' right to privacy. Authentication schemes must also weigh societal expectations of privacy against practical considerations [34].

Availability: For many applications, vehicular networks will require real-time, or near real-time, responses as well as hard realtime guarantees. While some applications may tolerate some margin in their response times, they will all typically require faster responses than those expected in traditional sensor networks, or even ad hoc networks.

However, attempts to meet real-time demands typically make applications vulnerable to Denial of Service (DoS) attacks. In the deceleration application, a delay of even seconds can render the message meaningless [34].

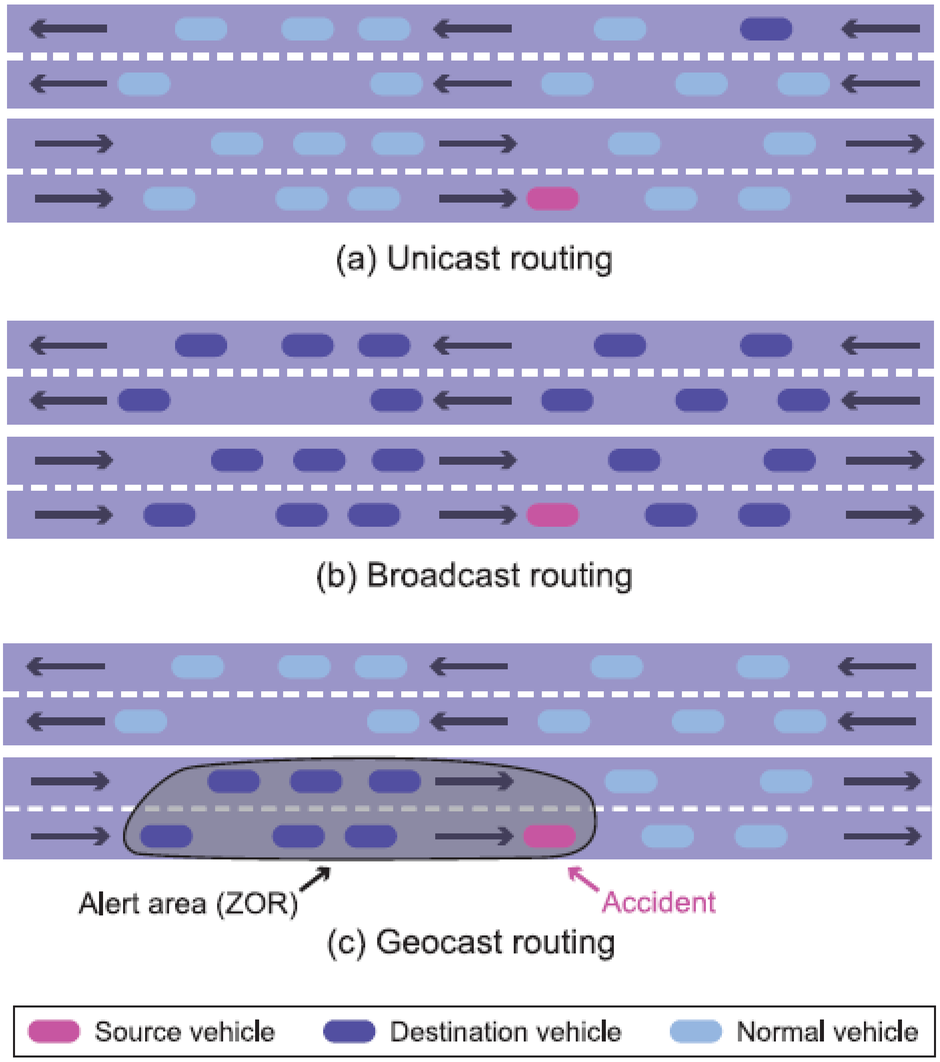


Figure 2.7.2 - Different communications scenarios in Vanets

Low Tolerance for Errors: Many applications use protocols that rely on probabilistic schemes to provide security. However, given the life-or-death nature of many proposed vehicular applications, even a small probability of error will be unacceptable. Furthermore, for many applications, security must focus on prevention of attacks, rather than detection and recovery. In an ad hoc network, it may suffice to detect an attack and alert the user, leaving recovery and clean-up to the humans. However in many, safety related, vehicular network applications, detection will be insufficient, since by the time the driver can react, the warning may be too late [34].

Key Distribution: Key distribution is often a fundamental building block for security protocols. In vehicular networks, distribution poses several significant challenges. First, vehicles are manufactured by many different companies, so installing keys at the factory would require coordination and interoperability between manufacturers. If manufacturers are unable or unwilling to agree on standards for key distribution, then we could turn to government-based distribution. However, without a system for key distribution applications like traffic congestion detection may be vulnerable to spoofing [34].

So, as far as security is concerned, in order to make vehicular networks viable and acceptable to consumers, we need to establish secure protocols that satisfy the stringent requirements of this application space. Designing secure protocols is complicated by the seemingly conflicting requirements of consumers, automobile manufacturers, and government, particularly when trying to provide strong vehicle identification while protecting driver privacy.

CHAPTER 3

STATE OF THE ART

In this chapter we present the approaches and solutions proposed by others to handle data dissemination, data compression and aggregation, and position verification problems in VANETs. Although there are many solutions that can handle these problems in MANETs, they cannot be directly applied to VANETs because VANETs have some very different characteristics from MANETs, namely higher mobility speeds and restricted network topologies and vehicle movement.

3.1 Data dissemination in VANETs

As we have mentioned, broadcast is the most appropriate data dissemination technique in VANETs especially for safety-related applications. One of the most common problems in broadcast is the broadcast storm problem [18]. Various solutions have been proposed to handle the broadcast storm problem in mobile ad-hoc networks (MANETs) [18, 35, 36], but most of them are not applicable in case of VANETs. The design of the MANET techniques depends on the node limited speed, but in VANETs the nodes (the vehicles) can move with much higher speeds. The approaches that have been proposed to alleviate the broadcast storm problem in dense VANETs can be classified as probability-based, timer-based and priority based approaches. The main idea behind the probability-based techniques is that whenever a vehicle receives a frame, it will re-broadcast it after holding it for a certain waiting time, according to a probability p that depends on how far this vehicle is from the sending vehicle. Distant vehicles will have higher probability p to re-broadcast the received frames than nearby vehicles. This technique is called weighted p-persistence broadcasting and has been proposed by Wisitpongphan *et al.* [37].

There are two versions of the timer-based techniques: slotted time and continuous time. Wisitpongphan *et al.* [37] proposed the slotted 1-persistence broadcasting technique, dividing the waiting time into slots. When each vehicle receives a frame, it is assigned a time slot during which it should re-broadcast the frame with probability 1 if no one else had re-broadcasted it. As the distance increases, the receiving vehicle is assigned a shorter r broadcast time slot. Also, the same authors proposed a slotted *p*-persistence broadcasting technique, in which during the assigned time slot the vehicle will re-broadcast the frame with a pre-determined probability *p*.

Briesemeister *et al.* [38] proposed a continuous time version of the timer-based technique, called role-based multicast. In this technique, whenever a vehicle receives a frame, it waits for a certain amount of time before re-broadcasting it. The longer the distance from the sender, the shorter the waiting time is. This technique was only concerned with the sparsely connected networks to maximize the message reachability and does not handle the broadcast storm problem. Bachir *et al.* [39] proposed Inter- Vehicle Geocast (IVG), based on the same idea presented by Briesemeister *et al.* [38], but it handles the dense network situation.

In general, timer-based techniques have shown superiority over the probability based techniques in mitigating the broadcast storm problem. In dense VANETs, the timer-based techniques re-broadcast the frame sooner (early re-broadcast) because the re-broadcast waiting time lessens with increasing distance from the sending vehicle, while the waiting time is constant in the probability-based techniques. Also, having an early re-broadcast will reduce the channel contention and redundant rebroadcasts because whoever hears the re-broadcast will cease its own re-broadcast. Within the timer-based techniques, the time continuous version known as IVG is better than the slotted time techniques, because the slotted versions restrict the rebroadcast to be initiated only at certain times, which increases channel contention.

Although IVG mitigates the broadcast storm problem, it suffers from the spatial broadcast storm problem, as it does not utilize the network topology information in the re-broadcast decision.

In both of the timer-based and probability-based techniques the farthest vehicle(s) within transmission range of the original sender will be selected to be the re-broadcaster(s). Based on that, all the vehicles at the boundary of transmission range will re-broadcast the frame at the same time. This will cause a broadcast storm locally at the boundary. Although the generated storm is local, it will affect the overall system performance. We term this the *spatial broadcast storm problem*.

Torrent-Moreno *et al.* [40] proposed a priority-based broadcast scheme in which nodes that have a time-critical message to send will be assigned a higher priority to access the channel. In general, the priority-based techniques categorize the network nodes into multiple classes with different priorities and schedules frame transmission accordingly. This technique reduces the contention on the channel access by allowing the higher priority nodes to access the channel before the other nodes, but it does not solve the broadcast storm problem.

Tonguz *et al.* [41] provided a comprehensive framework (DV-CAST) to handle broadcasting in VANETs considering three possible traffic densities (dense, regular, and sparse). In dense traffic, they suggest using one of the timer-based techniques [37], while in sparse traffic they suggest using role-based multicast [38]. They did not suggest a specific technique to be used in case of regular density traffic since they define regular density as a mix of some vehicles sensing dense traffic and some vehicles sensing sparse traffic. So, each vehicle will use the technique appropriate to what it has sensed. Even though DV-CAST appears to be a complete solution, it is still vulnerable to the spatial broadcast storm problem. Our contribution of *p*-IVG was designed to specifically address the spatial broadcast storm problem.

Furthermore, the extant literature presents many proposed solutions for data dissemination over vehicular networks (introduced in [42] and [43]). The most interesting solution exploits network coding in order to obtain high bandwidth efficiency. However, packet loss can deeply affect the performance of these methods. This problem has been recently addressed in [44], where errors and erasures can be tolerated. This method generates codes on a vector space and avoids the overhead of the encoded set description while simultaneously providing highly efficient transmissions. Unfortunately, this is not a desirable t for our scenario as the recoding phase of network coding approach is computationally more expensive than a smart selection of the received information to be forwarded to other nodes.

Byers et al. in [45] study the content reconciliation problem in wired networks. In their study, two nodes compute the resemblance of their information in order to understand if the communication can be useful. The authors also present appropriate tools for an effective content reconciliation. Although the methods presented in their paper are very interesting, we argue that the performance in wireless vehicular networks would be poor. For example, to perform the content reconciliation methods, nodes would be required to periodically exchange packets to update their neighbors on its own representation of information. Additional time would also be required to compute the information that is to be sent. In sum, the content reconciliation phase would require too much time.

Yet another solution, Bullet, is found in [46]. This method can use either rateless or multiple descriptions coding to efficiently disseminate data. It efficiently solves the problem of choosing the best nodes to talk to in an overlay mesh network. Moreover, it uses a simplified version of

TCP to fulfill the requirements of multimedia transmission that is also TC friendly. Performance analysis shows that Bullet is efficient on wired networks but again perform poorly in vehicular networks for reasons mentioned earlier.

The work in [47] simplifies the content reconciliation phase by sequentially labeling the received packets and by exchanging the sequence number ranges in order to perform a simple content reconciliation. Unfortunately, this solution can work only in wired networks where packet loss is almost negligible. In fact, when loss happens, the number of sub ranges of packets to be transmitted grows, thus decreasing the efficiency.

Finally, a method for P2P streaming in wired networks, rStream, is presented in [48]. R-Stream is based on the encoding of the source using rateless codes. According to this method, every node forwards the information after having decoded it. Furthermore, availability of information is assured if each node generates a set of symbols that is independent of the others in the network. This allows symbols to be received from several nodes at the same time without the need for content reconciliation. Unfortunately, a similar approach in vehicular networks would perform poorly because it is based on the implicit assumption that connections between source node and interested nodes have sufficient time to receive all the encoded information needed for recovery. However, this may not happen in vehicular networks because a node can be connected to APs or vehicles for only a limited time. Therefore, the transmission of the encoded information may not be quickly completed.

Localization Systems were studied from the viewpoint of Vehicular Ad Hoc Networks (VANets). It is showed how GPS receivers, the most common source of localization information in VANets, can become erroneous or unavailable in a number of situations. Furthermore it is argued that future localization systems for VANets are likely to use some kind of Data Fusion technique in order to provide position information for vehicles that is accurate and robust enough to be applied in VANet critical applications. Finally, it is shown how Data Fusion techniques can be used to compute an accurate position based on a number of relatively inaccurate position estimations [49].

[50]Based on simulation results, the simulator (NETSTREAM) calculates packet collision ratios and success rates of ad-hoc communications among vehicles per second, and evaluates how fast each vehicle can obtain its destination route information. After the stage of simulation, Saito, Funai, Umedu, Higashino [50] have proposed an inter-vehicle ad-hoc communication protocol density (SDRP (Speed Dependent Random Protocol) for road information dissemination. In

SDRP, according to the vehicle speed *v*, random transmission interval is calculated between the minimum value *min(v)* and the maximum value *max(v)*)with suitable dissemination intervals depending on vehicles' speeds and traffic.

In [51], it becomes an investigation of the effectiveness of AODV and GPRS in an inner city environment and on a highway segment. This evaluation is based on traces obtained from a microscopic vehicle traffic simulation on the real road maps of Switzerland. To increase the credibility of our study, we model the irregular radio channel behavior by the probabilistic Shadowing signal propagation model. VANETs pose unique challenges to a routing protocol. AODV and GPSR exhibit serious performance problems in the VANET scenarios investigated here. The packet delivery ratio of both protocols stays very low and varies from 5-20 % depending on the simulated scenario. One way to address the performance problems is to use a preferred group broadcasting strategy (PGB) to split and merge hops. This strategy significantly reduces broadcasting load and noticeably improves performance of AODV in VANETs. The Advanced Greedy Forwarding (AGF) technique improves the effectiveness of greedy forwarding and, as a result, the performance of GPSR in VANETs. Packet delivery ratio of GPSR enhanced with AGF is up to 10 times better than of the standard version of this protocol. So, through [51], it is discovered two improvements that increase the packet delivery ratio and reduce the delay until the first packet arrives.

Tubaishat, Shang and Shi [52] propose a traffic control system using wireless sensor networks. The new decentralized system depends on the traffic information collected from the wireless sensor network to achieve a real time adaptive traffic control. The goal is to maximize the flow of vehicles and reduce the waiting time while maintaining fairness among the other traffic lights.

Xu, et al. [53] propose the design of layer-2 protocols for a vehicle to send safety messages to other vehicles. The target is to send vehicle safety messages with high reliability and low delay. The communication is one-to-many, local, and geo-significant. The vehicular communication network is ad-hoc, highly mobile, and with large numbers of contending nodes. The protocols are compatible with the Dedicated Short Range Communications (DSRC) multi-channel architecture. The sensitivity of the protocol performance is evaluated under various offered traffic and vehicular traffic flows. The results show the approach is feasible for vehicle safety messages in DSRC.

According to Jose Santa and Gómez-Skarmeta [54], inter-vehicle communications are, nevertheless, achieved using 802.11 technologies, through vehicular ad-hoc networks (VANETs), which have been applied in safety solutions over all. [54] Tries to make the reader aware of the potential of cellular networks not only in V2I, but also for infrastructure to vehicle (I2V) and vehicle to vehicle (V2V) communications. The results over a real prototype prove the feasibility of cellular networks (CN) for lots of vehicular services, dealing with the latency obtained from V2I, I2V and V2V data transmissions.

Hoehmann and Kummert [55] present a new concept of combining a simulator for wireless sensor networks and a simulator of road traffic scenarios. Based on this new enhanced simulation environment they developed a first safety application which determines the trajectories of surrounding vehicles and estimates potential collisions.

Tsai et al. [56] through their experiments support that ZigBee is a viable and promising technology for implementing an intra-car wireless sensor network. After the results that they take from the experiments, they then propose a set of detection algorithms and an adaptive strategy that can adjust to channel conditions for improving the error performance of the wireless channel, and preliminary evaluation results are presented.

Tsai et al. [57] report different aspects of a *statistical analysis* of four representations in car wireless channels based on the received power data collected from a Binary Phase Shift Keying (BPSK) transmission experiment. It is shown that the communication channel between the base station and a sensor placed under the engine compartment is the worst in terms of stability, average fade duration, and fade proportion, while the channel between the base station and a sensor placed in the trunk and the channel between the base station and a sensor placed on the hood are the best. They also show that the 4 representative in-car wireless channels can satisfy the maximum packet delay requirement of less than 500 ms and the trunk channel and the in-the-engine-compartment channels can satisfy the requirement of up to 98% packet reception rate. Finally, we can tell that they define wireless communication reliability via the following two performance metrics: packet reception rate and maximum packet delay.

Yousef, Karaki and Shatnawi [58] present an adaptive traffic control system based on a new traffic infrastructure using Wireless Sensor Network (WSN) and using new techniques for controlling the traffic flow sequences. The system has the potential to revolutionize traffic surveillance and control technology because of its low cost and potential for large scale deployment. The proposed system consists of two parts: WSN and a control box (*e.g.* base-station) running control algorithms. The WSN, which consists of a group of traffic sensor nodes (TSNs), is designed to provide the traffic communication infrastructure and to facilitate easy and large deployment of traffic systems. Simulation results show the efficiency of the proposed scheme in solving traffic congestion in terms of the average waiting time and average queue length on the isolated (single) intersection and efficient global traffic flow control on multiple intersections.

Biswas, Tatchikou, and Dion [59] present an overview of vehicle cooperative collision avoidance (CCA) application using the emerging Dedicated Short Range Communication (DSRC) infrastructure for inter-vehicle wireless networking. The concept of CCA has been introduced with an overview, and its implementation issues have been analyzed in light of specific requirements from the MAC and routing-layer protocols of the underlying wireless networks. In order to explain the interactions between CCA and its underlying networking protocols, they present an example of the safety performance of CCA using simulated vehicle crash experiments. The results from these experiments are also used to demonstrate the need for network data prioritization for safety-critical applications such as CCA.

Ekici, Gu and Bozdag [60] give an overview of proposals that utilize mobile communication devices in WSNs and then, two new approaches are introduced (Mobile base station (MBS) and Mobile data collector (MDC)). In addition, they propose a new solution suite to calculate paths for mobile devices that collect information from sensors, based on knowledge of geographical data generation rates. In other words, they introduce an offline heuristic called the Partitioning Based Scheduling (PBS) algorithm that computes periodic paths of a mobile element (ME) to avoid sensor data loss at low ME speeds. They also present the Multihop Route to Mobile Element (MRME) algorithm that extends PBS to deliver urgent messages to MEs within specific delay bounds.

There also exist performance analyses of general information dissemination. These, however, deal only with highway scenarios. The situation on a highway cannot be directly compared to an inner-city environment, because highways are practically one dimensional, while in a city the number of junctions and intersections is typically high. Moreover, the driving speed and the traffic pattern are also largely different, which can severely influence the dissemination speed. In [61], the authors study information dissemination in VANETs depending on the number of cars in a scenario. However, they analyze only unidirectional traffic along a single road. A study of a one-dimensional highway scenario is presented in [62]. The authors assess the network utilization analytically, and conduct simulations regarding this aspect as well as the dissemination performance. In particular, they focus on the question whether oncoming traffic should be used for information transport or not.

Xu *et al.* propose an opportunistic dissemination (OD) scheme [63] that is similar to gossip [64]–[66]. In this approach, the data center periodically broadcasts some data, which will be received and stored by passing vehicles. Whenever two vehicles move into the transmission range of each other, they exchange data. This scheme does not rely on any infrastructure and, hence, is suitable for highly dynamic VANETs. However, after a data item has been propagated into the network, it is hard to timely remove the outdated information, particularly when it is frequently updated.

In addition, the performance of the OD scheme is poor in areas with high vehicle density due to media access control (MAC)- layer collisions [67]. This can easily lead to severe congestion and significantly reduce the data delivery ratio. To mitigate the excessive transmissions and congestion, Korkmaz *et al.* [68] propose a link-layer broadcast protocol to help disseminate the data. The protocol relies on link-layer acknowledge mechanisms to improve the reliability of the multihop broadcast. More specifically, only one vehicle is used to forward and acknowledge the broadcast packet to reduce the broadcast storm problem. However, in the case of network congestion, the link-layer solution is not enough. Furthermore, since many information sources may exist in a given urban area, the amount of broadcasted data from these sources can easily consume the limited bandwidth. Thus, it is important to study the maximum amount of data that can be disseminated in a given area.

CHAPTER 4

MOBILITY MODELS IN VEHICULAR NETWORKS

Software development for vehicular scenarios is a very complex process because of the many factors that can impact on the result, ranging from the mobility of the nodes to the radio transmission and the end-to-end delay. In order to overcome or at least reduce the probability of failures in terms of application functionalities, simulation becomes a very important and mandatory step in software design before any implementation. Moreover, simulations are fast, cheap, repeatable and make it possible to investigate the influence of single parameter variations. A large number of network nodes can be simulated which is not feasible in real-world experiments. In case of new protocols' design, it is imperative to use a mobility model that accurately represents the mobile nodes (MNs) that will eventually utilize the given protocol. Only in this type of scenario it is possible to determine whether or not the proposed protocol will be useful once implemented. The faithfulness of the simulation results is proportional to the realism of the parameters and the accuracy of the models used in the simulation, in particular, the mobility model (MM) which defines the movements of the mobile nodes within the simulated area during the simulation.

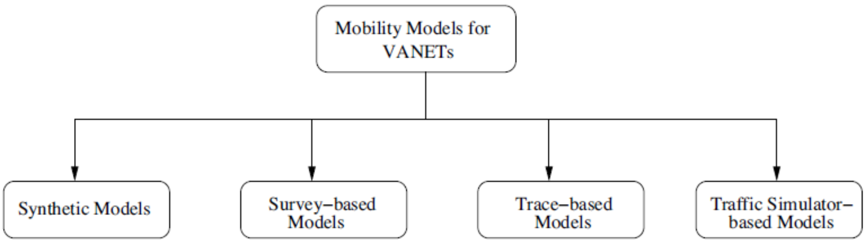


Figure 4.0 - Classification of Vehicular Mobility Models

4.1 Manet’s mobility models

4.1.1 Entity Models

In this type of models, the movement of each node is defined separately and independently from the others. Each node moves by itself, following its own parameters. The first model of this category was described by Albert Einstein in 1926 [69] in order to mimic the erratic movement that many entities show in nature by moving in extremely unpredictable ways and it is called Random Walk Mobility Model (RWMM). In this mobility model, that is widely used and sometimes referred to as Brownian motion, a MN moves from its current location to a new location by randomly choosing a direction and speed in which to travel. The new speed and direction are both chosen from predefined ranges, [speedmin, speedmax] and [0, 2π] respectively. Each movement in the Random Walk Mobility Model occurs in either a constant time interval t or a constant distance traveled d , at the end of which a new direction and speed are calculated. If a MN, which moves according to this model, reaches a simulation boundary, it bounces off the simulation border with an angle determined by the incoming direction. The MN then continues along this new path. Many derivatives of the Random Walk Mobility Model have been developed including the 1-D, 2-D, 3-D, and d-D walks. In 1921, Polya proved that a random walk on a one or two dimensional surface returns to the origin with complete certainty [70]. This characteristic ensures that the random walk represents a mobility model that tests the movements of entities around their starting points, without worry of the entities wandering away never to return. One of the main issues of RWMM is that it is a memoryless mobility pattern because it retains no knowledge concerning its past locations and speed values, thus generating unrealistic movements such as sudden stops and sharp turns.

Generally, the Random Walk Mobility Model is a memoryless mobility pattern because it retains no knowledge concerning its past locations and speed values [71]. The current speed and direction of an MN is independent of its past speed and direction [72]. This characteristic can generate unrealistic movements such as sudden stops and sharp turns (see Figure 4.1.1.a).

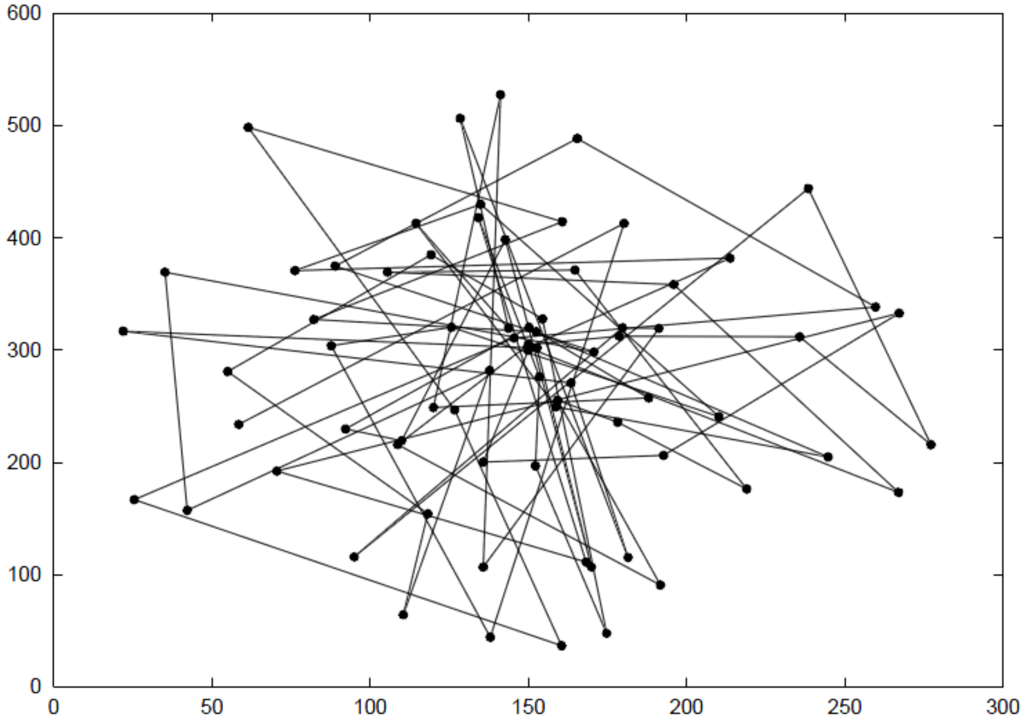


Figure 4.1.1.a - Travelling pattern of an MN using the 2-D Random Walk Mobility Model (time)

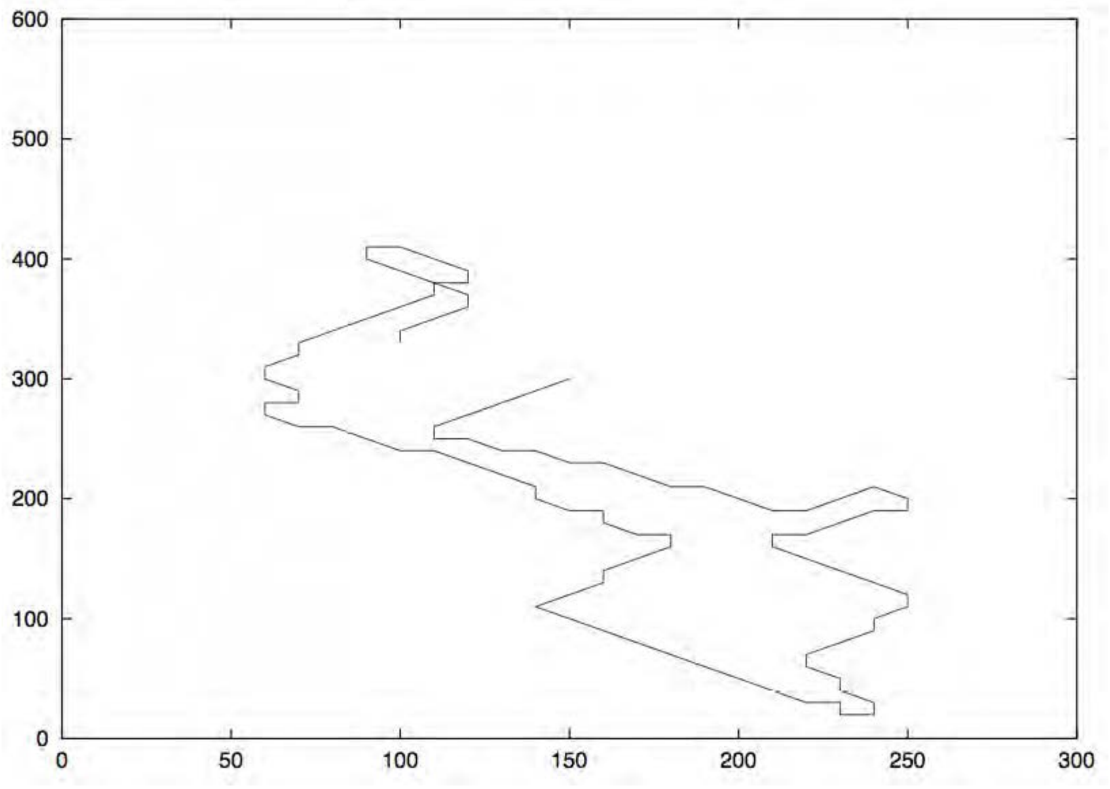


Figure 4.1.1.b - Random Walk Mobility model. Travelling pattern of a MN using a probabilistic version of RWM

The most popular random mobility model is the Random Waypoint (RWP) [73], largely used for simulating ad hoc networks and available in many simulators like ns2, GloMoSim and Qualnet. According to this model, a MN stays in a location and waits for a certain amount of time Once this time expires, the MN chooses a new random location within the simulation area and than travels towards it at a selected speed, uniformly distributed in a predefine interval. Usually, MNs are initially distributed randomly and uniformly in the simulation area but it is important to realize that this is not representative of the manner in which node distribute themselves when moving. One of the most important parameter in RW simulations is the average MN neighbor percentage, given as the cumulative percentage of a total MNs that are a given MN's neighbor.

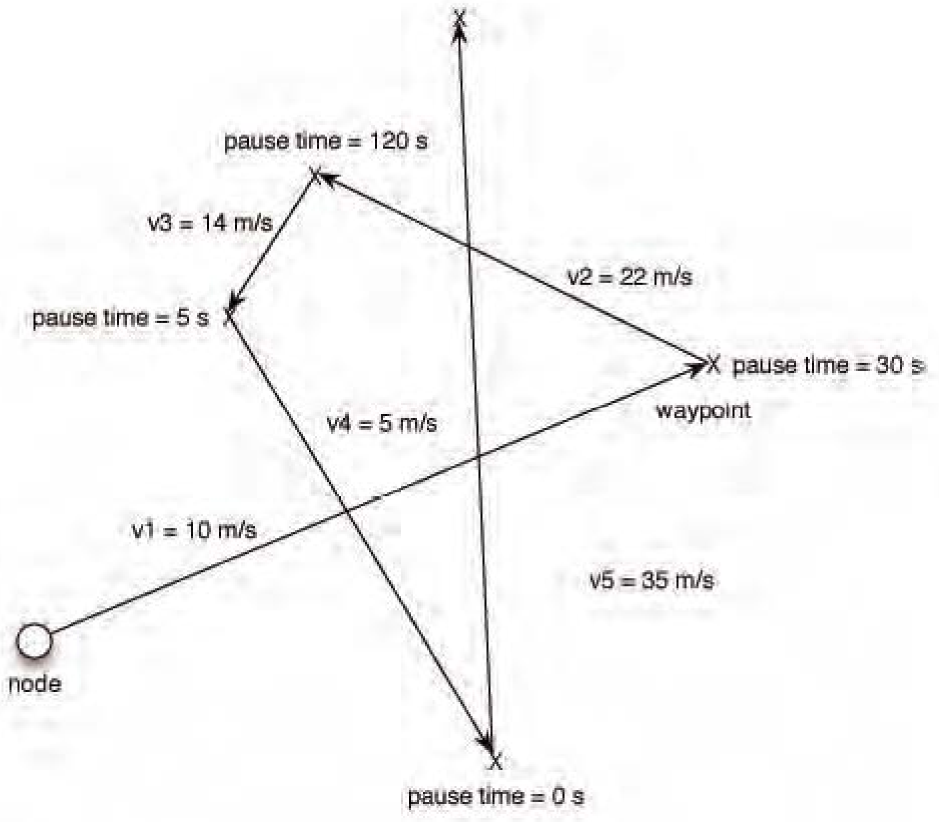


Figure 4.1.1.c - Random-Waypoint model In RWP, mobile nodes choose randomnly a destination and move towards it with a constant velocity

4.1.2 Group Models

This class of mobility models represents movements of groups of nodes whose actions are completely independent of each other and describes very well many scenarios such as in tourist trips, where groups of tourists move together to visit particular monuments or in battlefields, where group of soldiers in a military scenario may be assigned the task of searching a particular plot of land in order to destroy land mines, capture enemy attackers, or simply work together in a cooperative manner to accomplish a common goal. Moreover, group models can easily and effective depict scenarios such as avalanche rescue, where the responding team consisting of human and canine members work cooperatively. The most general model of this group is the Reference Point Group Mobility Model (RPGM) [74], which represents a random motion of a group of nodes as well as a random motion of each individual within the group. Group movements are based upon the path traveled by a logical center for the group, which is used to calculate group motion via a group motion vector, G~M (see figure 4.1.2.a). The motion of the group center completely characterizes the movement of its corresponding group of MNs, including their direction and speed. Individual MNs randomly move about their own predefined reference points, whose movements depend on the group movement.

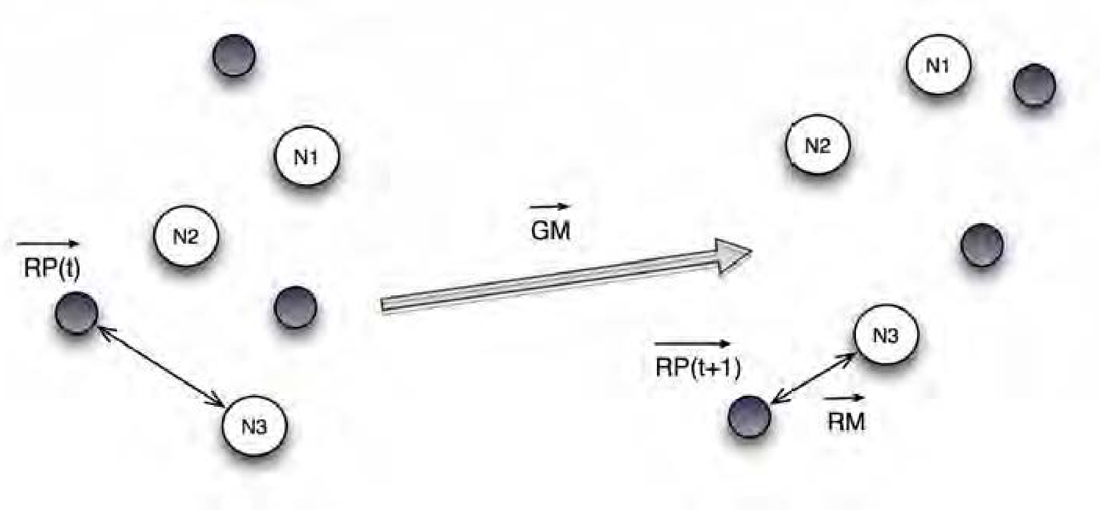


Figure 4.1.2.a - RPGM model Movements of three nodes according to the RPGM mobility model

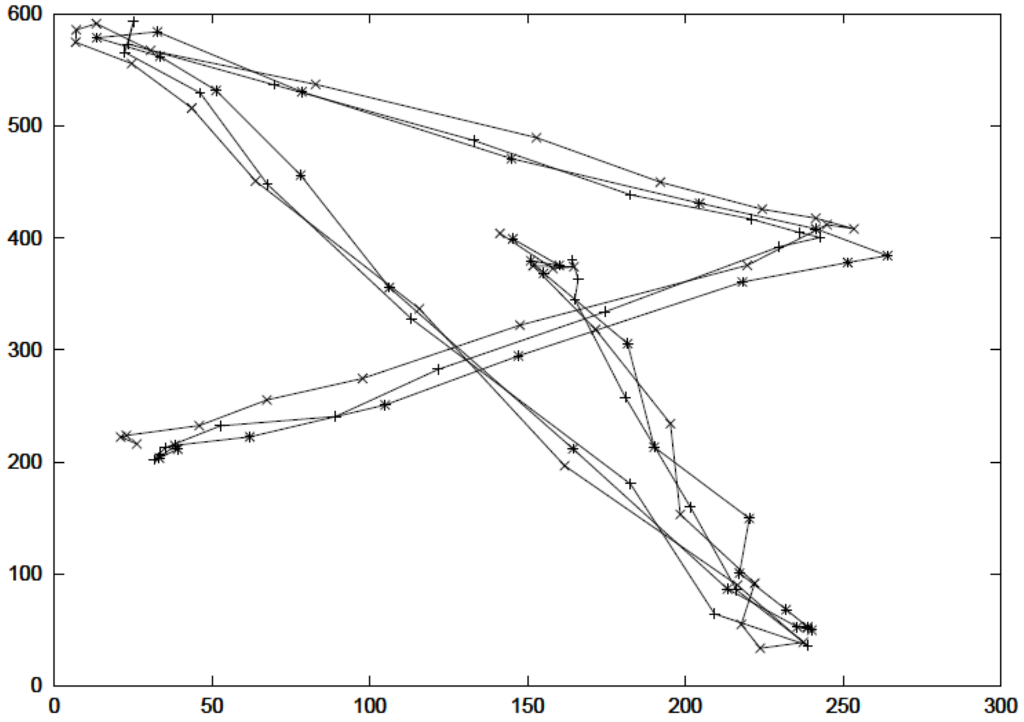


Figure 4.1.2.b Travelling pattern of one group (three MNs) using the RPGM model

4.2 VANETs mobility models

A critical aspect in VANETs is the need for a mobility model which reflects, as close as possible, the real behavior of vehicular traffic.

When dealing with vehicular mobility modeling, it is worth to introduce two concepts: macro-mobility and micro-mobility.

With the former, we mean all the macroscopic aspects which influence vehicular traffic, such as: road topology, constraining car movements, per-road speed limits, number of lanes, overtaking, safety rules and traffic signs establishing the intersections' crossing rules.

Without doubt, it would be desirable for a trustworthy vanet simulation that both micro and macro mobility descriptions be jointly considered. However, many mobility models employed for VANETs ignore these guidelines, thus failing in reproducing peculiar aspects of vehicular motion such as car acceleration and deceleration in presence of nearby vehicles, queuing at road intersections clustering caused by semaphores, vehicular congestion and traffic jams.

4.2.1 Factors affecting mobility

Mobility pattern of nodes in a VANET can significantly influence route discovery, maintenance, reconstruction, consistency and caching mechanism and this can obviously affect data dissemination protocols. For example, static or slow moving vehicles tend to hold the topology configuration, as they can be assimilated to "semi-static" and reliable relaying nodes; on the contrary, fast moving vehicles can cause highly changeable topology demanding frequent route reorganization and packet losses. In detail, the following factors can be considered in modeling VANETs' mobility:

- **Block size:** if the block (that is the smallest area surrounded by streets) is extended, it causes few intersections and, then, in turn, the frequency of which the vehicles stop decreases

- **Streets layout:** streets force nodes to confine their movements to well-defined paths, determining the spatial distribution of nodes and their connectivity. Moreover, streets can have either single or multiple lanes and can allow one-way or two-way traffic

- **Average speed:** vehicle's speed determines how quickly its position changes, which in turn determines the rate of network topology change. The speed limit of each road, as well as acceleration and deceleration, also directly affects the average speed of vehicles and how often the existing routes are broken or new routes are established

- **Interdependent vehicular motion:** in sophisticated mobility models, the movement of each vehicle is influenced by the movement pattern of its surrounding vehicle (minimum safety distance, lane changing, overtaking)

- **Traffic control mechanism:** stop signs and traffic lights affect the mobility and can result in the formation of clusters and queues. Furthermore, reduced mobility implies more static nodes and then, slower rates of route changes in the network. On the contrary, vehicles' clusters can affect network performance due to channel contention and longer network partitions.

CHAPTER 5

PROBLEM STATEMENT AND METHODOLOGY CHOICE

Vehicular traffic monitoring is collecting the statistics of roads which will answer questions like:

- ✓ How many vehicles are moving on the road?
- ✓ Time at which road is getting congested?
- ✓ Which roads are acting as traffic bottlenecks?
- ✓ Which path is less congested among the available paths?
- ✓ How we can detect accidents? (accident identification)

Having information about current situation of a particular road may help drivers to take alternate paths, efficiency of traffic lights can be increased by integrating these readings with traffic lights operation, and also this data will help the government to take necessary steps to control the traffic, drivers to take less congested paths, to increase efficiency of traffic lights.

In real world, traffic lights are used to regulate traffic flow moving in different directions. The existence of traffic lights tends to create a “clustering” effect. In other words, places where there is a traffic light are likely to have a higher node density since vehicles are forced to stop at the traffic light to wait for the light to turn green. Intuitively, a high node density might improve the network connectivity. On the other hand, a higher node density might also suggest a higher chance for packet collision since more nodes might be transmitting at the same time. In addition, the distance between two adjacent traffic lights can have a significant effect on the network connectivity. Specifically, the network can be “fragmented” by the traffic lights when the radio transmission range is smaller than the distance between two adjacent clusters. In other words, a link breakage can happen when the inter-cluster distance is larger than the radio coverage.[75]

Nowadays monitoring and controlling vehicular traffic is one of the major challenges for many countries and it is said that we are moving closer to the vision of intelligent transportation systems (ITS), which can enable a wide range of applications, such as collision avoidance, emergency message dissemination, dynamic route scheduling and real-time traffic condition monitoring.

The vehicular traffic data collected can be served to users through FM radios, mobile phones and Internet. Vehicular traffic monitoring with wireless sensor networks is fairly new which includes deployment of sensors on road that monitor traffic movements. The collected data is uploaded to nearest base station with the help of radio. The base station uploads the collected data to a centralized server.

The problem, on which the thesis is based, is to design and deploy a wireless sensor network to monitor traffic in some scenarios using one of available simulation packets such as GrooveNet, SWANS++, TraNS, Qualnet etc. In order to give answers how some parameters vary by changing speed, nodes, different communications and routing protocols, some traffic simulation scenarios will be implemented, such as traffic congestion in crossroads, in train junctions, in traffic lights junctions etc.. Needless to say that in Greece existing infrastructures are quite old, the whole effort will have a difficulty in practical implementation situation that is the structure of the roads and the fact that the roads do not have lane discipline.

There are two possible ways of placing sensors to monitor traffic

1. On Road Mount: Sensors will be placed inside the road by digging small holes.

2. Side Mount: Sensors will be attached to the poles on two sides of the road.

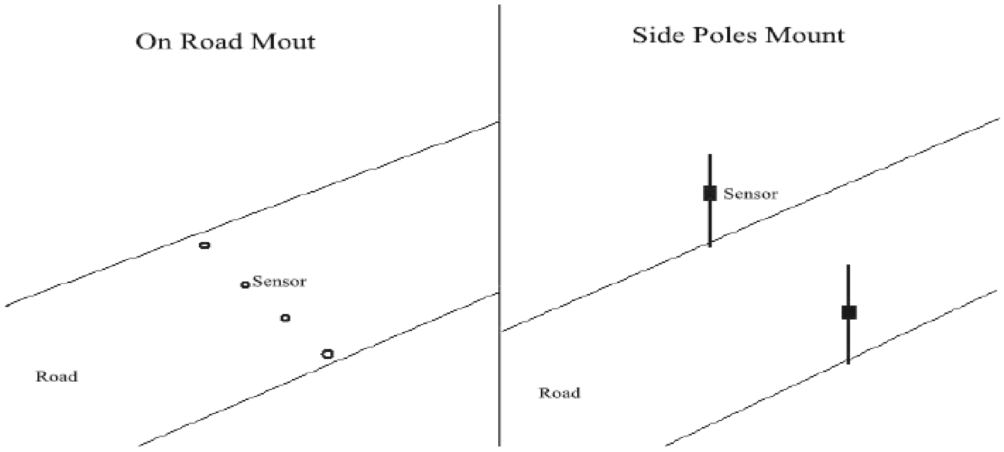


Figure 5.0 - Two ways of placing sensors on the roads

5.1 Data-parameters involved

Data and parameters that are involved through the scenarios:

- ***Data aggregation**:* The vehicles have to pass on the data sent by the neighbors to other neighbors of its coverage area. This increases the number of packets to be sent by a vehicle. Therefore, data aggregation techniques are applied to reduce such overheads. Data aggregation is an interesting approach, which reduces the number of packets transmitted drastically by combining several messages related to the same event into one aggregate message. For example, the records about two vehicles can be replaced by a single record with little error, if the vehicles are very close to each other and move with relatively the same speed.

- ***Data validation**:* A vehicle may send the data it has observed directly (assuming that a vehicle always trusts the data it has gathered itself) to its neighbors. Sometimes malicious vehicles may send the incorrect information to confuse the users. For example, a malicious node may send the false accident information and divert all the vehicles on other roads, which may sometimes lead to traffic congestion. In such situation data validation techniques must be applied before passing on the received information to other nodes.

- ***Data dissemination**:* Data dissemination can be defined as broadcasting information about itself and the other vehicles it knows about. Each time a vehicle receives information broadcasted by another vehicle, it updates its stored information accordingly, and defers forwarding the information to the next broadcast period, at which time it broadcasts its updated information. The dissemination mechanism should be scalable, since the number of broadcast messages is limited, and they do not flood the network. VANET characteristics like high-speed node movement, frequent topology change, and short connection lifetime especially with multi-hop paths needs some typical data dissemination models for VANETs.

- ***Routing**:* Since the topology of the network is constantly changing, the issue of routing packets between any pair of nodes becomes a challenging task. Most protocols should be based on reactive routing instead of proactive. Multicast routing is another challenge because the multicast tree is no longer static due to the random movement of nodes within the network.

- ***Network congestion**:* Congestion control in VANETs is a challenging issue. The Internet is based on an end-to-end paradigm, where the transport protocol (e.g. TCP) instances at the endpoints detect overload conditions at intermediate nodes. In case of congestion, the source reduces its data rate. However, in VANETs the topology changes within seconds and a congested node used for forwarding a few seconds ago might not be used at all at the point in time when the source reacts to the congestion.

Some of the important applications of VANETs are message and file delivery, location dependent services, Internet connectivity, information and warning functions, co-operative assistance systems, safety services (like emergency breaking, accidents, passing assistance, security distance warning, etc.) traffic monitoring, etc. [76].

5.2 Data Dissemination Techniques

Data broadcast is an attractive solution for large scale data dissemination. In contrast to unicast, where a data item must be transmitted many times to answer multiple requests, broadcast has the potential to satisfy all outstanding requests for the same data item with a single response. In general, there are two major data broadcast approaches [22,62,77, 78]: push-based and pull-based. In push-based broadcast, the roadside unit (RSU) broadcasts the whole or part of the database periodically according to a static broadcast program, which is based on historical data access statistics or a set of pre-defined request profiles. All vehicles listen passively to the broadcast channel to retrieve data items of interest without sending any request. Usually, push-based algorithms bear the advantage of achieving optimal or near optimal solutions by using some prior knowledge such as data access patterns to design broadcast programs. It is believed that broadcast-based applications have the potential of bootstrapping vehicular ad-hoc networks. The goal of the data broadcast or push communication model is to exchange information which may have parameter or data like speed, position and direction the vehicle on regular basis among a set of vehicles which may also be moving in order to enable each individual vehicle to view and access traffic conditions.

There are two main mechanisms to achieve this goal [22]. In the flooding mechanism, each vehicle periodically broadcasts its own data or information. Whenever another vehicle receives abroad cast message, it stores and immediately forward sit by rebroadcasting. This mechanism is not scalable because in high traffic density a large number of messages will flood over the network. In dissemination, each vehicle broadcast information of itself as well the information of all the vehicles available with it. Each vehicle receives information broadcasted by another vehicle, updates accordingly and defers forwarding the information to the next broadcast period, at which time it broadcasts its uploaded information. This mechanism overcomes the disadvantage of flooding and therefore scalable, since the number of broadcast messages is limited because network is not flooded by them. In pull-based broadcast, commonly known as on demand broadcast, the RSU disseminates data items in response to explicit requests submitted by vehicles. Compared to its push-based counterpart, pull based is more scalable to large size databases.

Moreover, the absence of assumption on data access patterns makes it more adaptable to dynamic workload environments. However there are various parameters in vehicular networks which need to be kept in mind so as to implement either of the approach. In the research literature, many methods of data delivery are presented and, we can distinguish the following data dissemination approaches:

(1) Opportunistic: Under most highway scenarios VANETs tend to be disconnected. To overcome the limitation imposed by lack of connectivity, opportunistic communication is proposed using cars as data mules. Information is pulled from other vehicles or the infrastructure as a target vehicle encounters them [79, 80];

(2) Vehicle-Assisted: It adopts the idea of carry and forward, where a moving vehicle carries the packet until a new vehicle moves into its vicinity and forwards the packet. A vehicle carries information with it and delivers it either to the infrastructure or to other vehicles when it encounters them. This process involves mobility in addition to wireless transmissions in order to disseminate the information [22];

(3) Co-operative: Vehicles can download partial units of some content and then share them afterwards to obtain the complete content. This method is particularly suitable for content

dissemination (where the amount of information is rather important in terms of le size) and it was adopted to develop dissemination protocol based on rate-less codes.

5.3 Routing

Generally, a routing protocol plays a vital role in scheduling of data in communication. Since the operational principles of MANETs and VANETs resembles therefore most of the routing algorithms which were applicable to MANETs has been studied and modified as the difference is in high speed mobility and the nature of unpredictability of their movements. Since each node has a limited transmission range, messages often have to be forwarded by other nodes in a VANET. The routing protocols can broadly be categorized into two classes:

1. Topology–Based Routing;

2. Position-Based Routing or Geographic Routing.

Topology-Based Routing uses the data/information about links that exist in the network to perform packet forwarding whereas the Geographic routing or position based routing uses the information of the neighbouring location to perform packet forwarding. Taxonomy of various routing protocols in VANETs has been shown in figure 5.3. below.

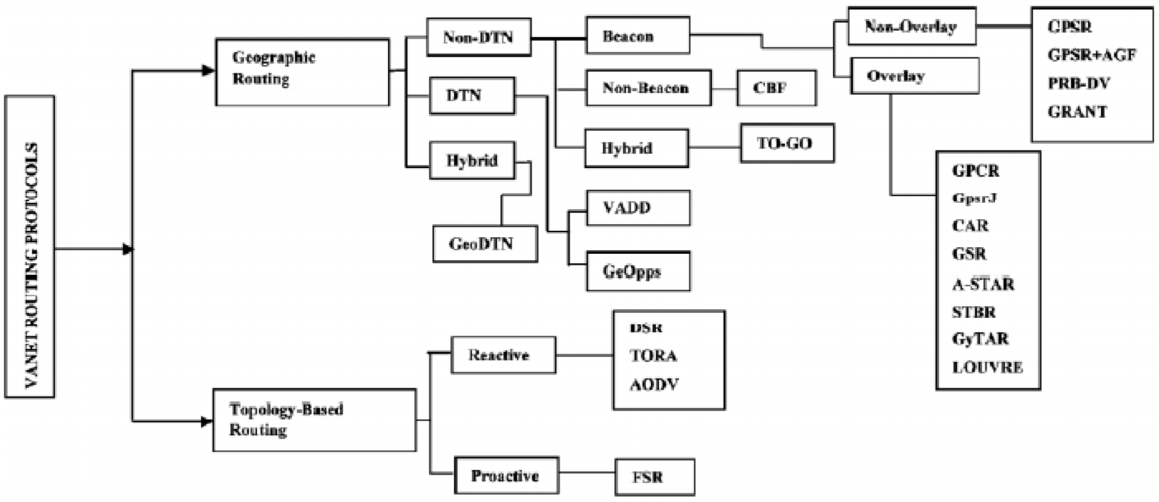


Figure 5.3 - Routing Protocols in VANETs

Topology-based routing protocols use the information about the links that exists in the network to perform packet forwarding. On the other hand, location based routing the forwarding decisions is based on a nodes location. They can be sub-divided into proactive and reactive approaches.

Proactive algorithms employ classical routing strategies such as distance-vector routing (e.g. DSDV) or link-state routing (e.g. OLSR and TBRPF). Proactive algorithms maintain routing information about the available paths in the network even if these paths are not currently used. The main drawback of this approach is that the maintenance of unused paths may occupy a significant part of the available bandwidth if the network topology changes frequently. In response to the maintenance problem, reactive routing protocols were developed. In general a proactive routing protocol periodically creates the new routes of each pair of vehicles. It basically suffers from the complexity of how to determine the optimal period for route creation and update. Too short periods make the protocol suffer from high overhead. Conversely, too long periods make the protocol suffer from frequent route failures.

Reactive routing protocols maintain only the routes that are currently in use, thereby reducing the burden on the network when only a small subset of the available routes is in use. In location-based routing, forwarding decisions are based on the location of the forwarding node in relation to the location of the source and destination nodes. Location-based routing protocols consist of location services and geographic forwarding. Geographic forwarding takes advantage of a topological assumption that works well for wireless ad hoc networks: nodes that are physically close are likely to be close in the network topology also. Each node learns its own geographic position using a mechanism such as GPS, and periodically announces its presence, position, and velocity to its neighbors. Thus each node maintains a table of its current neighbor’s identities and geographic positions. In contrast to a proactive protocol, a reactive one creates a new route only when the existing one is broken. Therefore, its overhead is lower than that of a proactive protocol, but the number of route failures is higher. In addition, it also lacks the ability to determine a better route due to lack of periodic updates of routing information.

5.4 Simulation stage

5.4.1 Vehicular Network Simulators

With network simulators it is possible to easily design new protocols and analyze their impact on a large number of network nodes under several traffic and environmental conditions that may reflect in specific mobility and propagation models. The most important feature which characterizes simulation is, obviously, the repeatability, which allows studying the overall performances of a given protocol by varying simply the simulation parameters.

Both in academic and industry research, researchers almost always must resort to simulation as the expense of actual deployment would be too high. Unfortunately, there is no standard vehicular networks simulator and the common practice is to generate a mobility trace using a vehicular mobility simulator and then input this trace to the network simulator. Further, the right choice of the mobility simulator is important as performance in vehicular networks highly depends on the connectivity which, in turn, is affected from the nodes' movements.

Needless to say, in order to simulate a VANET we need both the networking and the mobility component. As we already said, in most cases, these two functionalities are provided by two independent simulators so that researchers build a topology and produce a trace of vehicle movements using a mobility simulator.

5.4.2 Mobility simulators

The mobility simulator generates the mobility of vehicles and records the vehicular movements into trace files. VANET mobility simulators are used to generate traces of the vehicles' motion that can be usually saved and subsequently imported into a network simulator in order to study the performances of the protocol/application. However, it is important to generate realistic movement traces in order to rigorously evaluate VANET protocols because the overall performances depend on the connectivity which, in turn, relies on the movement traces.

5.4.3 Network simulators

These applications allow simulating (at different level of detail) the ISO-OSI layers, taking into account the propagation and fading effect of the radio signals. As described earlier, a mobility simulator is generally used to produce node movement traces that are then fed to the network simulator. The network simulator then controls the communications between the mobile nodes. As these network simulators support wireless communication, most of them include at least a simple node mobility model, which includes the following models: Random Drunken Model, Random Waypoint Model, Trace file.

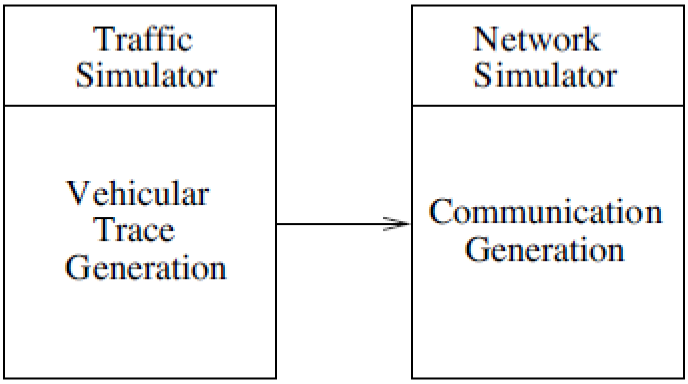


Figure 5.4.1 Interaction between Network and Traffic Simulators: The Isolated Case

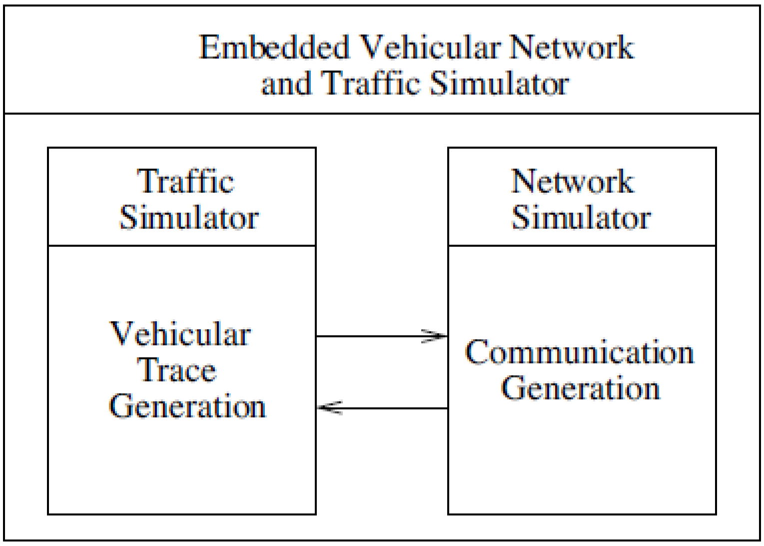


Figure 5.4.2 - Interaction between Network and Traffic Simulators: The Embedded Case

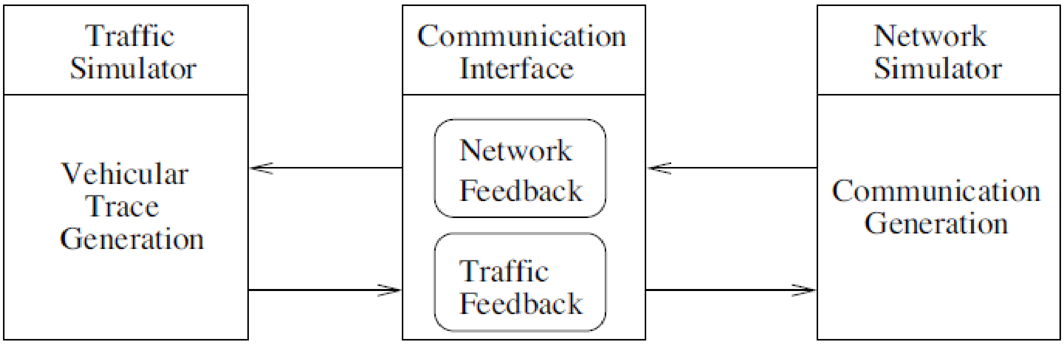


Figure 5.4.3 - Interaction between Network and Traffic Simulators: The Federated Case

5.4.4 Survey on VANET simulators

Without a doubt, the existence of a standard VANET simulator that the entire research community trusts would enhance the research quality and shorten the development cycle of any VANET application. So in order to achieve this goal, much effort has been exerted to develop a VANET-specific simulator by integrating a network simulator with a mobility generator because the simulation of VANET applications not only requires simulating the wireless communication between the vehicles, but also requires simulating the mobility of the vehicles.

Vehicular traffic models are typically classified into three categories based on traffic granularity: macroscopic, mesoscopic, and microscopic. Macroscopic models deal with traffic as flows, while mesoscopic models are concerned with the movement of whole platoons of vehicles. Microscopic models handle the movement of each vehicle in the traffic flow, thus they are the most suitable for VANET applications. Many microscopic models have been developed. The most widespread ones are the SK model [81], the Cellular Automaton (CA) model [82], and the IDM/MOBIL model [82] [83]. Fiore *et al.* [83] evaluated the realism of several mobility models and recommended that only realistic car-following models, such as IDM, be used in VANET simulations. There have been multiple efforts for developing mobility simulators. SUMO [84] is considered one of the pioneers in mobility simulators that can generate vehicle mobility traces. CanuMobiSim [85] is another mobility simulator that can also generate vehicle mobility traces. There is a variety of network simulators that are being used in the network research community. Some researchers prefer open source simulators like GloMoSim [86], OMNet++ [87], and ns-2 [83], while others prefer the commercial simulators, such as QualNet [89] and OPNET [90], to get better support and customization. Although ns-2 is an open source simulator, it is the most widely-used network simulator in the research community [91]. But, ns-2 suffers from problems when simulating large numbers of nodes. SWANS [92] was developed to be a scalable alternative to ns-2 for simulating wireless networks. Based on comparisons [93] between SWANS, GloMoSim and ns-2, SWANS was found to be the most scalable, the most efficient in memory usage, and fastest in runtime. In addition, Kargl *et al.* [93] validated the network model in SWAN against ns-2. They showed that along with better performance, SWANS delivered similar results a ns-2, at least for the network components that were implemented in both. Unfortunately, in most cases these two components of VANET simulation (wireless network/communication simulation and vehicular mobility simulation) have been decoupled. Both vehicular mobility and wireless communication have large communities concerned with their modeling an simulation, so high quality simulators exist in each of these areas. The problem is in merging the two types of simulators. An ideal VANET simulator would consist of two sub-simulators, a network simulator to simulate the wireless communication between the vehicles and a traffic simulator to simulate the vehicles' mobility. VANET applications that run at the top level of the network simulator can then be categorized according to how these two sub-simulators need to communicate.

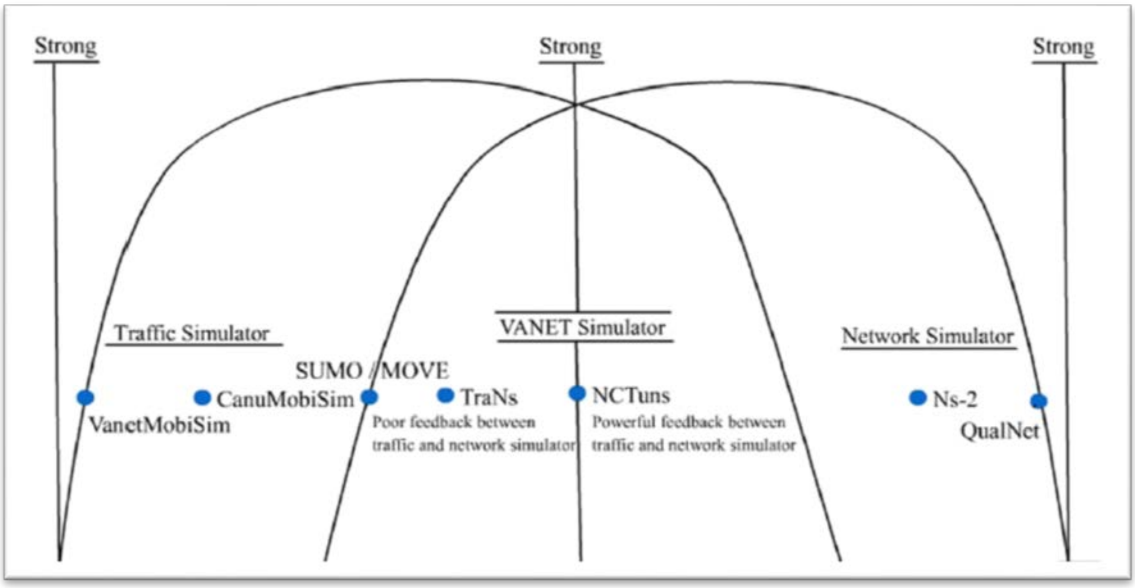
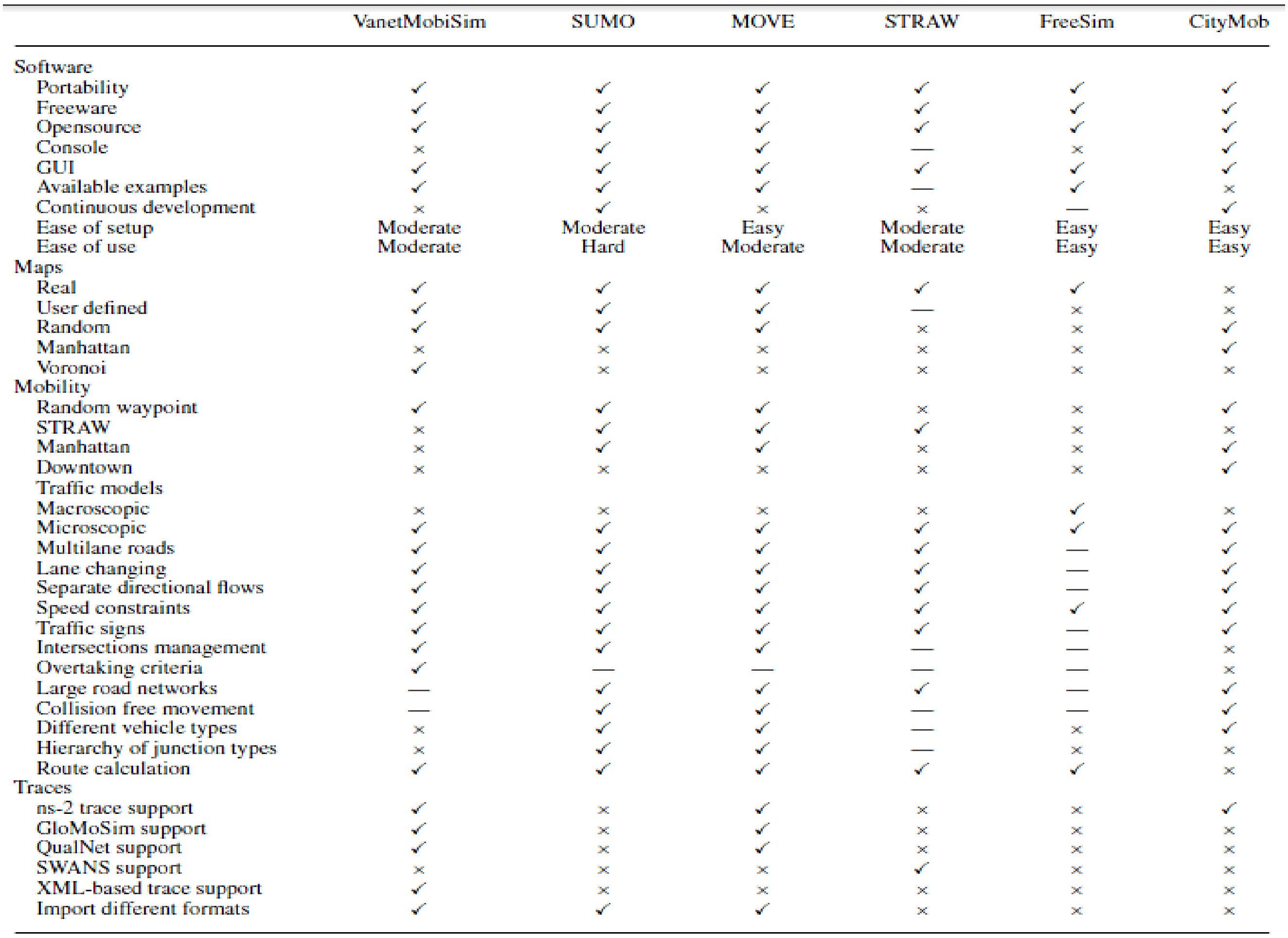


Figure 5.4.4 - Strength of traffic, VANET and network simulators



	VanetMobiSim	SUMO	MOVE	STRAW	FreeSim	CityMob
Software						
Portability	✓	✓	✓	✓	✓	✓
Freeware	✓	✓	✓	✓	✓	✓
Opensource	✓	✓	✓	✓	✓	✓
Console	×	✓	✓	—	×	✓
GUI	✓	✓	✓	✓	✓	✓
Available examples	✓	✓	✓	—	✓	×
Continuous development	×	✓	×	×	—	✓
Ease of setup	Moderate	Moderate	Easy	Moderate	Easy	Easy
Ease of use	Moderate	Hard	Moderate	Moderate	Easy	Easy
Maps						
Real	✓	✓	✓	✓	✓	×
User defined	✓	✓	✓	—	×	×
Random	✓	✓	✓	×	×	✓
Manhattan	×	×	×	×	×	✓
Voronoi	✓	×	×	×	×	×
Mobility						
Random waypoint	✓	✓	✓	×	×	✓
STRAW	×	✓	✓	✓	×	×
Manhattan	×	✓	✓	×	×	✓
Downtown	×	×	×	×	×	✓
Traffic models						
Macroscopic	×	×	×	×	✓	×
Microscopic	✓	✓	✓	✓	✓	✓
Multilane roads	✓	✓	✓	✓	—	✓
Lane changing	✓	✓	✓	✓	—	✓
Separate directional flows	✓	✓	✓	✓	—	✓
Speed constraints	✓	✓	✓	✓	✓	✓
Traffic signs	✓	✓	✓	✓	—	✓
Intersections management	✓	✓	✓	—	—	×
Overtaking criteria	✓	—	—	—	—	×
Large road networks	—	✓	✓	✓	—	✓
Collision free movement	—	✓	✓	—	—	✓
Different vehicle types	×	✓	✓	—	×	✓
Hierarchy of junction types	×	✓	✓	—	×	×
Route calculation	✓	✓	✓	✓	✓	×
Traces						
ns-2 trace support	✓	×	✓	×	×	✓
GloMoSim support	✓	×	✓	×	×	×
QualNet support	✓	×	✓	×	×	×
SWANS support	×	×	×	✓	×	×
XML-based trace support	✓	×	×	×	×	×
Import different formats	✓	✓	✓	×	×	×

Table 1 - A comparison of some mobility generators

5.4.5 Scalability of VANET simulators

When simulating a complex scenario including VANET networks, with both mobility and signal transmission and propagation models, it might be valuable to understand how far you can go with your simulation. This means that the most important parameters that every researcher should keep in mind before approaching any kind of simulation are scalability and overall performance. Scalability, which is, generally speaking, the ability to either handle growing amounts of work in a graceful manner or to be readily enlarged, gives to the researcher an idea on the maximum scenario he is allowed to simulate, in terms of number of mobile nodes, terrain dimensions, number of simultaneous protocols and details' level of the simulation's outcome. For example, one may simply think to simulate a large network of thousands of nodes with many details per network layer but if the simulation time or the memory consumption start growing too fast, this may turn out to be unfeasible. Needless to say that we must have in mind a lot of parameters in order to have a successful simulation stage.

Trying to compare two network simulators is not an easy task to accomplish, first of all because of the different implementations of both kernel and protocols, which may result in completely different resource consumption. Secondly, it should also be considered the detail's level of the simulation itself. In order to gain some experimental results of the performance trend in Qualnet and Ns2 (for a quick comparison), we implemented in both the simulators the same simulation set-up. The difference in the range of mobile nodes is due to the limitation in the Ns2 scalability which, compared to Qualnet, does not allow simulation of thousands of nodes. In fact, in [94] it was proved that Ns2 does not scale well for wireless sensor network (and then it makes sense to extend the results for Vanet, which is a specialization of a sensor network).

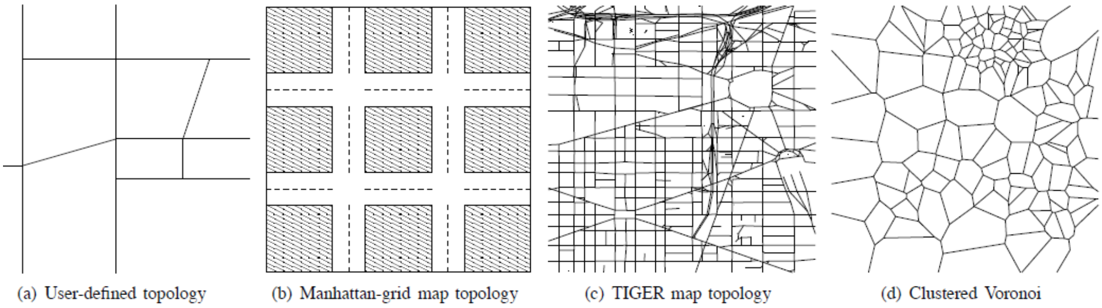
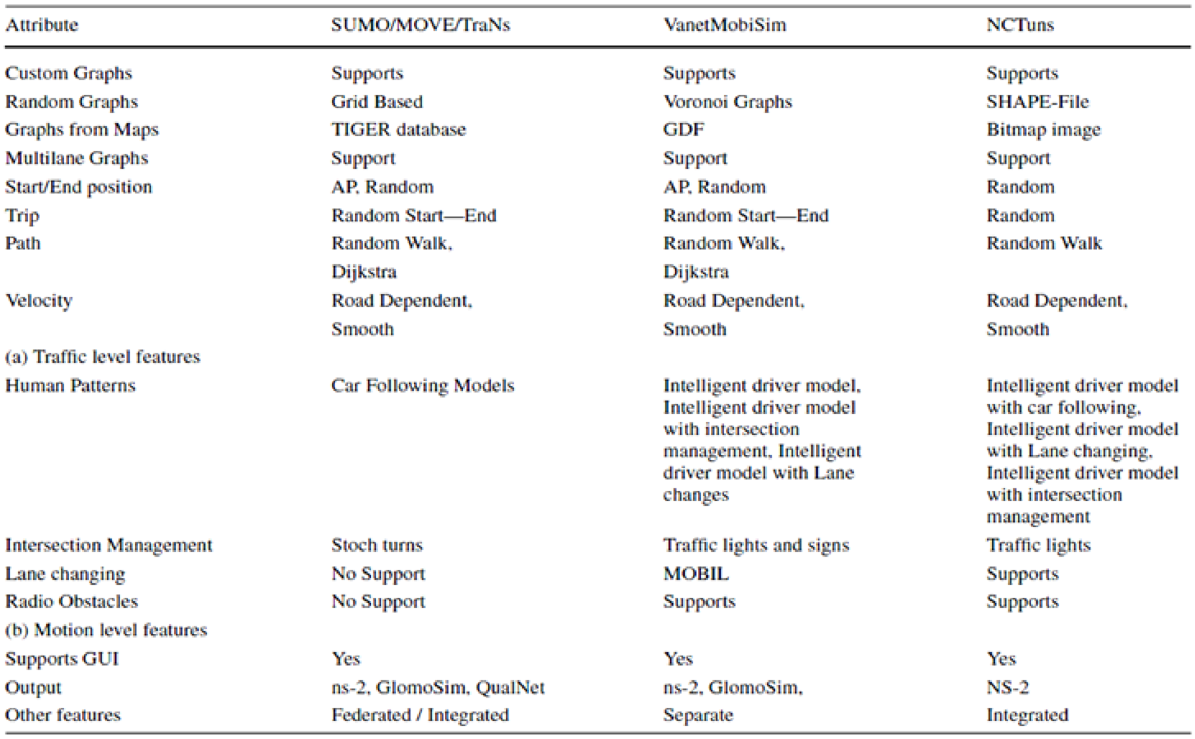

(a) User-defined topology (b) Manhattan-grid map topology (c) TIGER map topology (d) Clustered Voronoi

Figure 5.4.5 - Road topologies examples

5.5 The choice of the simulator

Through the survey between the different simulators, the combination of **SUMO**, **MOVE** and **NS2** is selected.

THE SUMO SIMULATOR:"Simulation of Urban MObility" (SUMO) [95] is an open source, highly portable, microscopic road traffic simulation package designed to handle large road networks. Its main features include collision free vehicle movement, different vehicle types, single-vehicle routing, multi-lane streets with lane changing, junction-based right-of-way rules, hierarchy of junction types, an openGL graphical user interface (GUI), and dynamic routing. SUMO can manage large environments, i.e., 10 000 streets.



Attribute	SUMO/MOVE/TraNs	VanetMobiSim	NCTuns
Custom Graphs	Supports	Supports	Supports
Random Graphs	Grid Based	Voronoi Graphs	SHAPE-File
Graphs from Maps	TIGER database	GDF	Bitmap image
Multilane Graphs	Support	Support	Support
Start/End position	AP, Random	AP, Random	Random
Trip	Random Start—End	Random Start—End	Random
Path	Random Walk, Dijkstra	Random Walk, Dijkstra	Random Walk
Velocity	Road Dependent, Smooth	Road Dependent, Smooth	Road Dependent, Smooth
(a) Traffic level features			
Human Patterns	Car Following Models	Intelligent driver model, Intelligent driver model with intersection management, Intelligent driver model with Lane changes	Intelligent driver model with car following, Intelligent driver model with Lane changing, Intelligent driver model with intersection management
Intersection Management	Stoch turns	Traffic lights and signs	Traffic lights
Lane changing	No Support	MOBIL	Supports
Radio Obstacles	No Support	Supports	Supports
(b) Motion level features			
Supports GUI	Yes	Yes	Yes
Output	ns-2, GlomoSim, QualNet	ns-2, GlomoSim,	NS-2
Other features	Federated / Integrated	Separate	Integrated

Table 3 - Traffic and Motion level features of SUMO, MOVE, TraNs, VanetMobiSim nad NCTuns simulators

The mobility generated by SUMO [96] is based on road networks (e.g., street maps supplemented by positions and semantics of traffic signs) where movements between source and destination roads are determined by a shortest path algorithm or, for example, traffic counting data. After the installation of the packets, we go on to the creation of the map and the different scenarios.

THE MOVE SIMULATOR :As far as MOVE (The MObility model generator for VEhicular networks) is concerned, is a Java-based application built on SUMO [84] with GUI support. MOVE supports a very good visualization tool and focuses mainly on traffic level features. In addition, it also supports custom graphs defined by the user as well as random generated graphs. But with random generated graphs, it restricts the node movement to a grid (i.e., the node should only move on the grid). MOVE is composed of a Map editor and a Vehicular Movement editor [97]. In real world, a driver normally has to decide his moving direction at an intersection. He can choose to either go straight, turn left, or turn right. MOVE allows user to define the turning probability of different directions at each intersection (e.g. 0.7 to turn left,0.5 to go straight and 0.3 to turn right) in the Vehicle Movement Editor [75]. The Map editor creates topological maps for network scenarios and the vehicular movement editor generates movement patterns automatically or use those defined by the users in the editor. MOVE can also generate its own mobility model but the results obtained are not satisfactory as compared to that of standard mobility models. The problem accompanied with this mobility model is the lack of support for large networks (i.e., its packet delivery ratio drops as the number of nodes increases). Moreover multiple radio interfaces are not supported by larger networks [97].

Except for the existence of traffic lights and driver route choice, another parameter that will concern us, is the overtaking behavior. In real world, a faster vehicle can overtake some other slower ones when overtaking is allowed on a multi-lane road. Overtaking behavior can have a great effect on the network topology and should be considered. Specifically, when overtaking behavior is not allowed, it usually results in a chain-like topology and a shorter and uniform intervehicle distance (the uniform distance is due to the that the vehicle needs to maintain a safe distance from the adjacent cars), which often suggests a better network connectivity [75].

To sum up, it is showed that details of mobility models such as the existence of traffic lights, driver route choice and car overtaking behavior can have a drastic impact on the VANET simulation results. Furthermore, it is argued that the faithfulness of simulation results is proportional to the realism of the parameters and the models used in the simulations. Therefore, selecting appropriate level of details in the mobility model for a VANET simulation is a very important yet challenging task.

NS-2 SIMULATOR: NS (version 2) is an object-oriented, discrete event driven network simulator developed at UC Berkely written in C++ and OTcl. NS is primarily useful for simulating local and wide area networks. NS is the most popular choice of simulator used in research papers appearing in select conferences like Sigcomm. Although NS is fairly easy to use once you get to know the simulator, it is quite difficult for a first time user, because there are few user-friendly manuals. Even though there is a lot of documentation written by the developers which has in depth explanation of the simulator, it is written with the depth of a skilled NS user. NS is constantly maintained and updated by its large user base and a small group of developers at ISI [98-101].

Historical issues: NS began as a variant of the REAL network simulator in 1989 and has evolved substantially over the past few years. In 1995 ns development was supported by DARPA through the VINT project at LBL, Xerox PARC, UCB, and USC/ISI. Currently ns development is support through DARPA with SAMAN and through NSF with CONSER, both in collaboration with other researchers including ACIRI. Ns has always included substantal contributions from other researchers, including wireless code from the UCB Daedelus and CMU Monarch projects and Sun Microsystems. For documentation on recent changes, see the version 2 change log [99], [101].

Nam Tool: Nam is a Tcl/TK based animation tool for viewing network simulation traces and real world packet traces. It supports topology layout, packet level animation, and various data inspection tools. Nam began at LBL. It has evolved substantially over the past few years. The nam development effort was an ongoing collaboration with the VINT project [101], [102].

Ns together with its companion, *nam*, form a very powerful set of tools for teaching networking concepts. *ns* contains all the IP protocols typically covered in undergraduate and most graduate courses, and many experimental protocols contributed by its ever-expanding users base. With *nam*, these protocols can **visualized** as animations [99], [101].

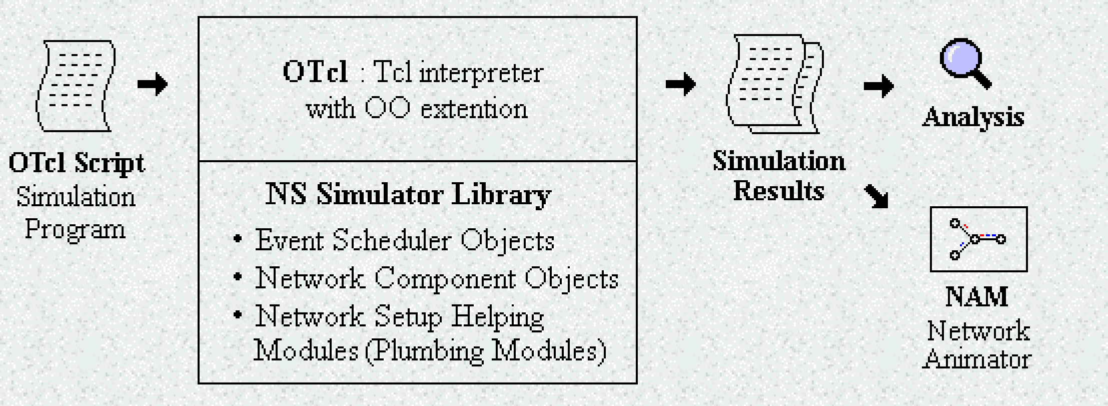


Figure 5.5.0 - Simplified User's View of NS

As shown in Figure 5.5.0, in a simplified user's view, NS is Object-oriented Tcl (OTcl) script interpreter that has a simulation event scheduler and network component object libraries, and network setup (plumbing) module libraries (actually, plumbing modules are implemented as member functions of the base simulator object). In other words, to use NS, you program in OTcl script language. To setup and run a simulation network, a user should write an OTcl script that initiates an event scheduler, sets up the network topology using the network objects and the plumbing functions in the library, and tells traffic sources when to start and stop transmitting packets through the event scheduler. The term "plumbing" is used for a network setup, because setting up a network is plumbing possible data paths among network objects by setting the "neighbor" pointer of an object to the address of an appropriate object. When a user wants to make a new network object, he or she can easily make an object either by writing a new object or by making a compound object from the object library, and plumb the data path through the object. This may sound like complicated job, but the plumbing OTcl modules actually make the job very easy. The power of NS comes from this plumbing [98].

5.6 Before the running stage

To begin with, there is need to download and install some software-environment in order to use SUMO and MOVE, as we have already said above.

***1. Mobility Generation*:**

This part of the software (called MOVE - MObility model generator for VEhicular networks) will generate the mobility model created by SUMO. Firstly you can select "Mobility Model" on the main top level menu (Figure 5.5.1).

Step 1: To click “*File*” then choose “*Set SUMO Binaries Path*”(Figure 5.5.3).

Step 2: Please you set SUMO binaries path. For instance, we set sumo path that is located at **“C:\Users\Anastasia\Documents\Vanet Simulators\sumo-0.13.1\bin**

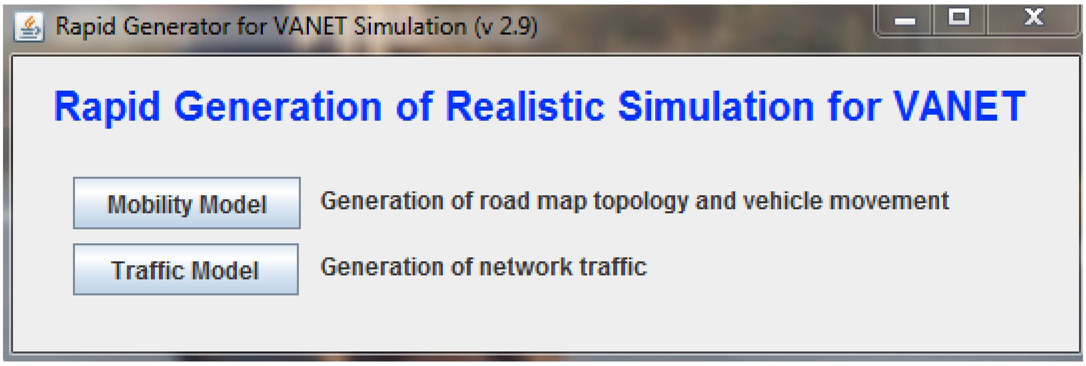
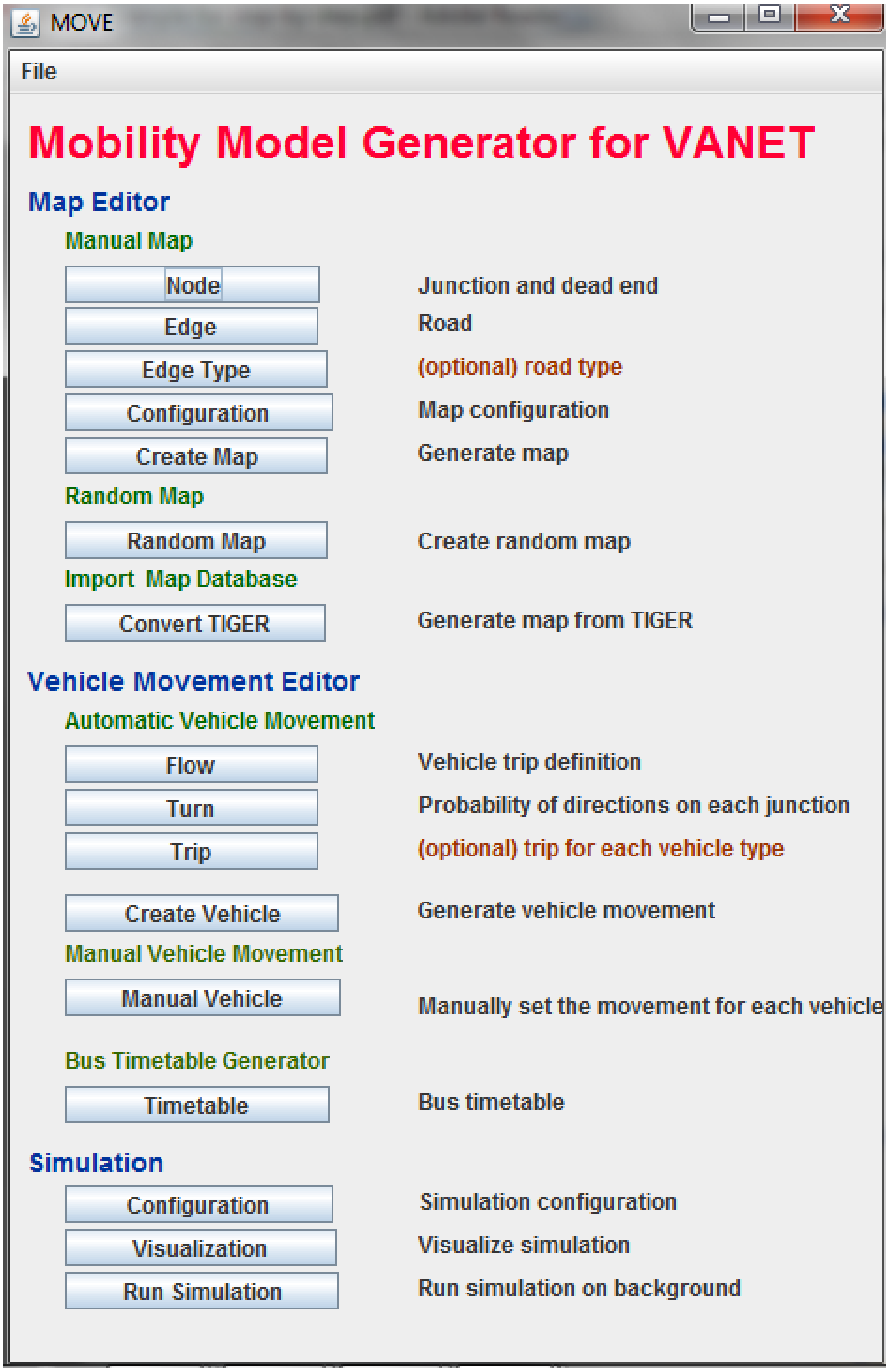


Figure 5.5.1 - Rapid Generation of Realistic Simulation for Vanet



Mobility Model Generator for VANET

Map Editor

Manual Map

Node	Junction and dead end
Edge	Road
Edge Type	(optional) road type
Configuration	Map configuration
Create Map	Generate map

Random Map

Random Map	Create random map

Import Map Database

Convert TIGER	Generate map from TIGER

Vehicle Movement Editor

Automatic Vehicle Movement

Flow	Vehicle trip definition
Turn	Probability of directions on each junction
Trip	(optional) trip for each vehicle type
Create Vehicle	Generate vehicle movement

Manual Vehicle Movement

Manual Vehicle	Manually set the movement for each vehicle

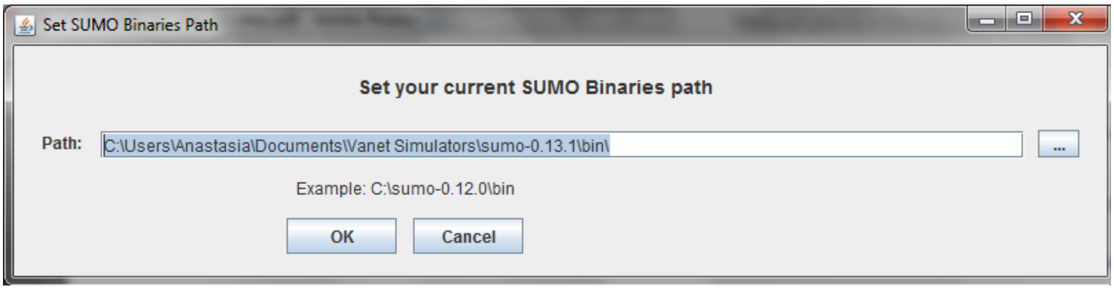
Bus Timetable Generator

Timetable	Bus timetable

Simulation

Configuration	Simulation configuration
Visualization	Visualize simulation
Run Simulation	Run simulation on background

Figure 5.5.2 - Mobility Model



Path: C:\Users\Anastasia\Documents\Vanet Simulators\sumo-0.13.1\bin\

Example: C:\sumo-0.12.0\bin

Figure 5.5.3 - SUMO Binaries Path

2. Map Generation

Step 1: Manually create your own map nodes (ex_NODE.nod.xml)

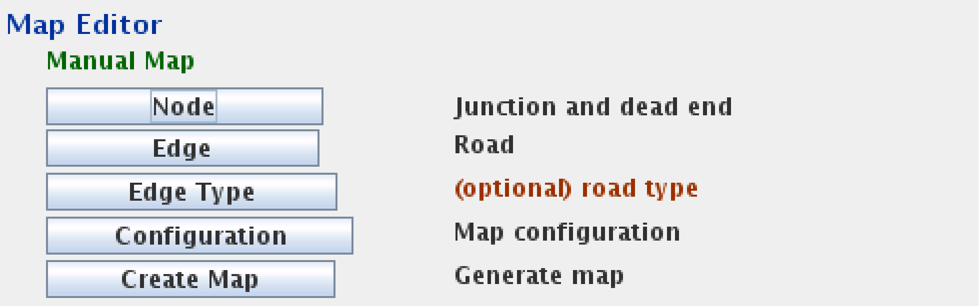
First, select “Node” from MOVE main menu (Figure 5.5.4 , 5.5.5). This is simply where all the map nodes are. Then select ‘File’ -> ‘save’ or ‘save as’ when you are done editing. Save the file as <name>.nod.xml (example.nod.xml) (Figure 5.5.6).

Step 2: Manually create your own map edge (ex_EDGE.edg.xml). First, select “Edge” from MOVE main menu. This is similar to node editor. This is where you can specify all the roads (a road will create a connection between two nodes created previously).

Step 3: Map configuration editor (ex_Map.netc.cfg). After that, choose “Configuration” from MOVE main menu.

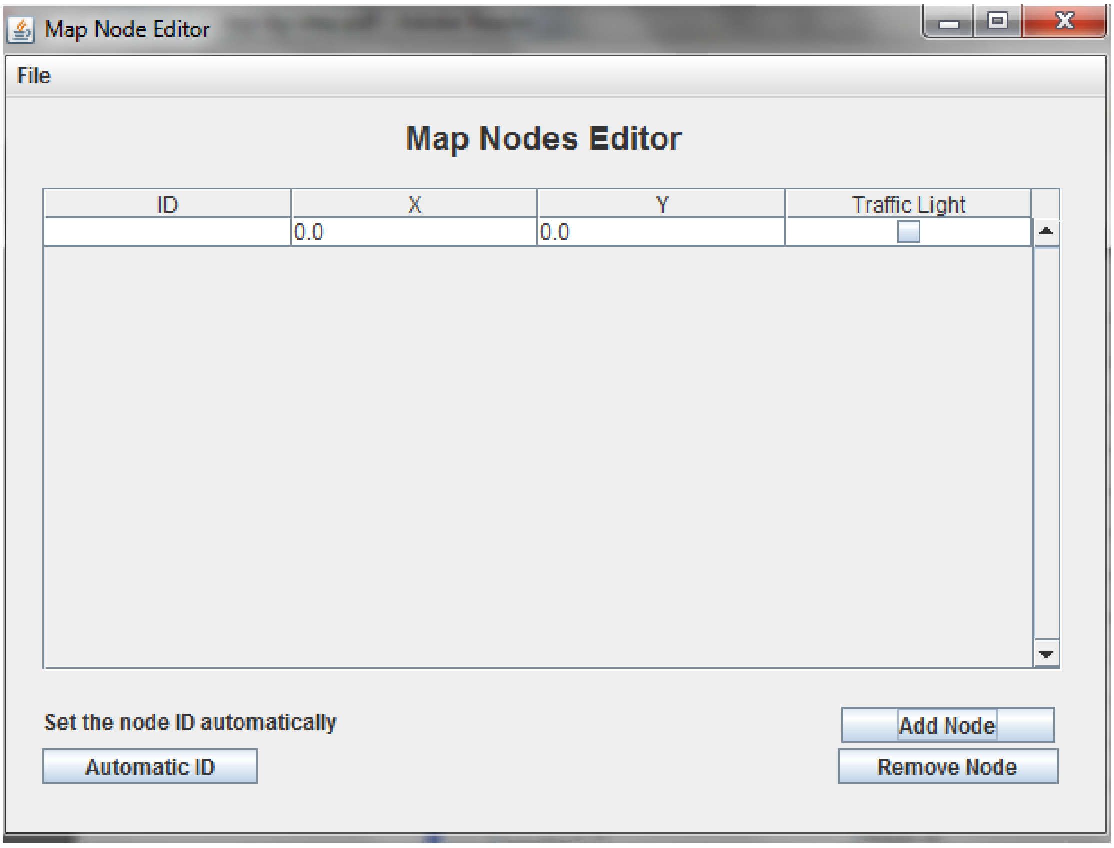
Step 4: Generate the Map (ex_Map.net.xml). Finally, select “Create Map” from MOVE main menu.

Simply select the netc file (**ex_Map.netc.cfg**) and click OK. A <name>.net.xml (**ex_Map.net.xml**) will be automatically generated. This is your map file.(Figure 5.5.7)



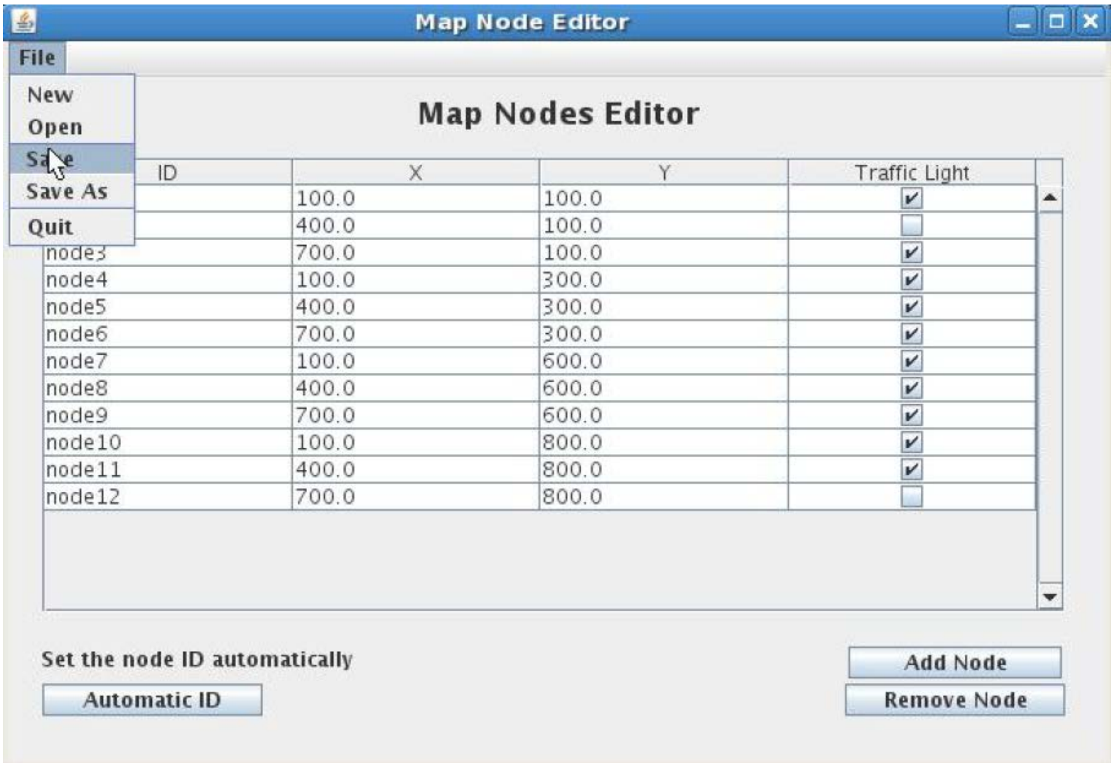
Node — Junction and dead end
Edge — Road
Edge Type — (optional) road type
Configuration — Map configuration
Create Map — Generate map

Figure 5.5.4 - Creation of map nodes



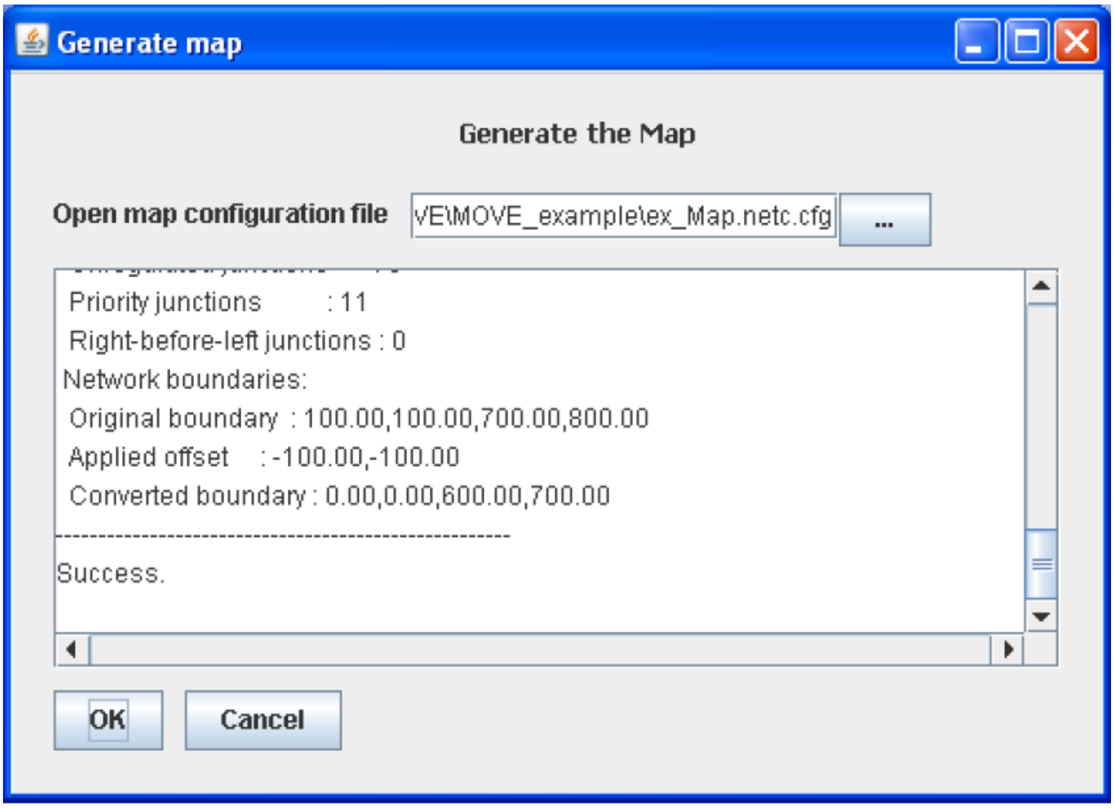
ID	X	Y	Traffic Light
	0.0	0.0	☐

Figure 5.5.5 - Map Node Editor



ID	X	Y	Traffic Light
	100.0	100.0	☑
	400.0	100.0	☐
node3	700.0	100.0	☑
node4	100.0	300.0	☑
node5	400.0	300.0	☑
node6	700.0	300.0	☑
node7	100.0	600.0	☑
node8	400.0	600.0	☑
node9	700.0	600.0	☑
node10	100.0	800.0	☑
node11	400.0	800.0	☑
node12	700.0	800.0	☐

Figure 5.5.6 - How to save the map node editor



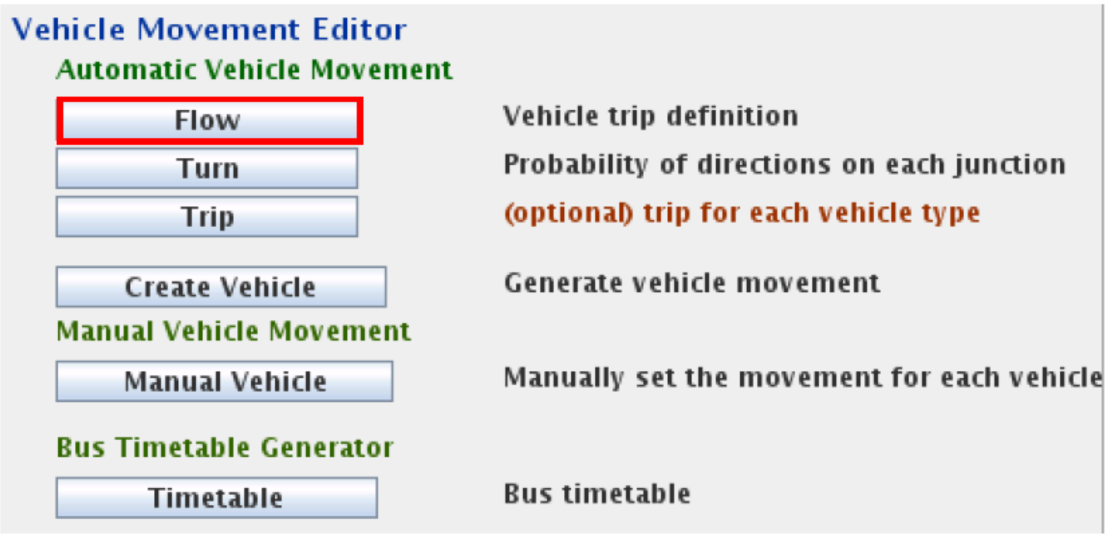
Generate the Map

Open map configuration file: VE\MOVE_example\ex_Map.netc.cfg

```
 Priority junctions        : 11
 Right-before-left junctions : 0
Network boundaries:
 Original boundary  : 100.00,100.00,700.00,800.00
 Applied offset     : -100.00,-100.00
 Converted boundary : 0.00,0.00,600.00,700.00
--------------------------------------------------
Success.
```

Figure 5.5.7 - Generate Map

3. Vehicle movements generation: In order to define some parameters (flow definition, junction turning ratio, create vehicle) we use the below window (Figure 5.5.8)



Vehicle Movement Editor

Automatic Vehicle Movement

Flow — Vehicle trip definition
Turn — Probability of directions on each junction
Trip — (optional) trip for each vehicle type
Create Vehicle — Generate vehicle movement

Manual Vehicle Movement

Manual Vehicle — Manually set the movement for each vehicle

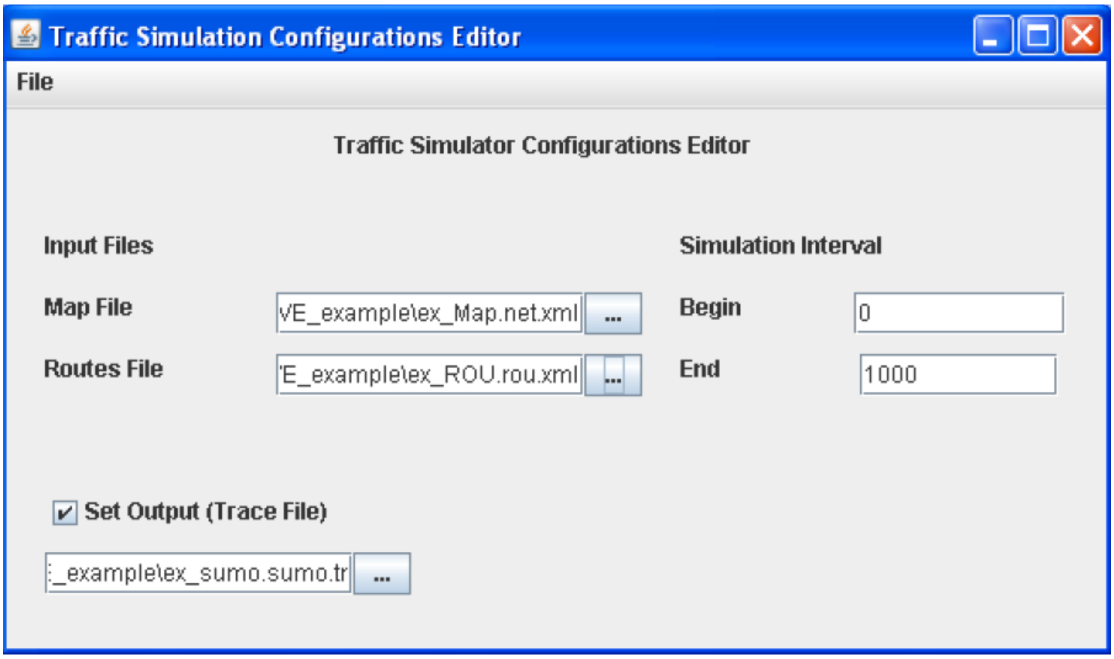
Bus Timetable Generator

Timetable — Bus timetable

Figure 5.5.8 - Vehicle Movement Editor

4. Simulation setup: After the map and movement is complete, you will need to specify the configurations of the simulation. Select “Configuration” at the bottom on MOVE main menu (Figure 5.5.9).

Figure 5.5.9 - Simulation



Traffic Simulator Configurations Editor

Input Files

Map File: VE_example\ex_Map.net.xml

Routes File: E_example\ex_ROU.rou.xml

Simulation Interval

Begin: 0

End: 1000

☑ Set Output (Trace File)

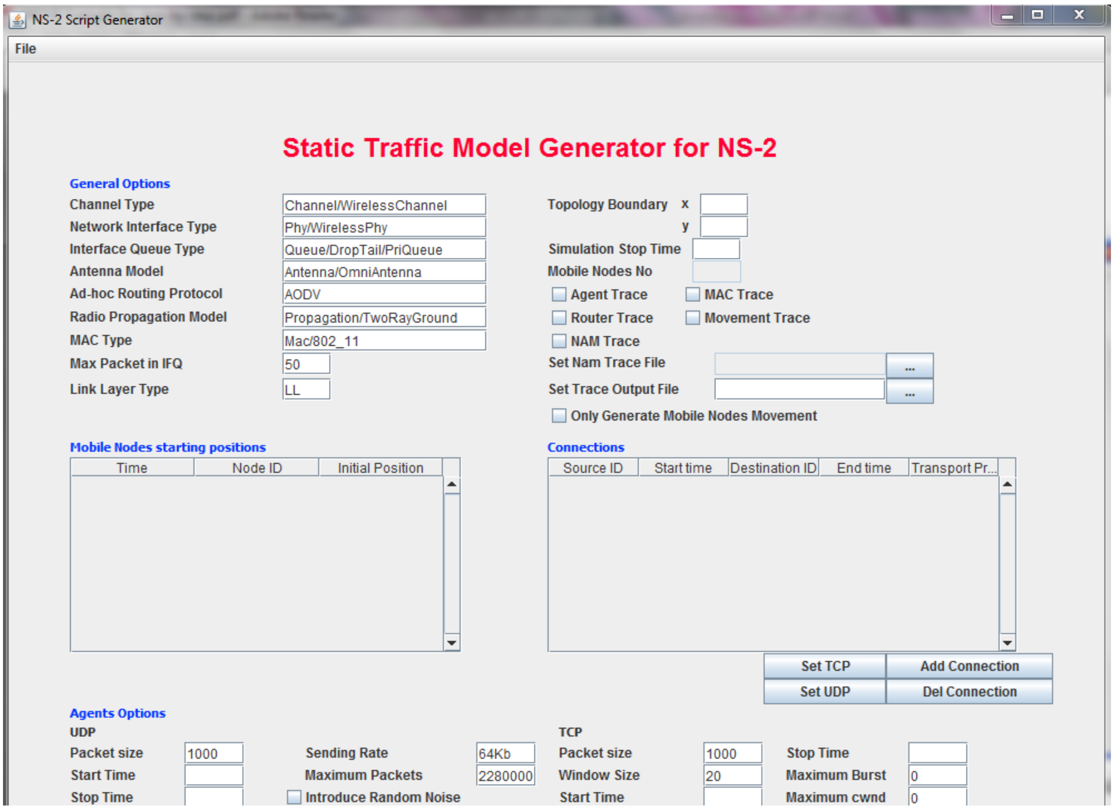
:_example\ex_sumo.sumo.tr

Figure 5.5.10 - Traffic Simulation Editor

The above picture defines all the last details about input files and the simulation interval before the visualization of simulation.

Figure 5.5.11 - Simulation

But in order to extract the output files there is need to install NS2 (Network Simulator) and define some parameters (traffic model generator) as we see below. This editor will generate the traffic simulation file (a tcl file) for NS-2 simulation tool. First import MOVE Trace (eg. **ex_SUMOTRACE.sumo.tr**) and .net.xml (eg.. **ex_Map.net.xml**) file for script generator.



Channel Type: Channel/WirelessChannel
Network Interface Type: Phy/WirelessPhy
Interface Queue Type: Queue/DropTail/PriQueue
Antenna Model: Antenna/OmniAntenna
Ad-hoc Routing Protocol: AODV
Radio Propagation Model: Propagation/TwoRayGround
MAC Type: Mac/802_11
Max Packet in IFQ: 50
Link Layer Type: LL

Topology Boundary x:
y:
Simulation Stop Time:
Mobile Nodes No:
☐ Agent Trace ☐ MAC Trace
☐ Router Trace ☐ Movement Trace
☐ NAM Trace
Set Nam Trace File:
Set Trace Output File:
☐ Only Generate Mobile Nodes Movement

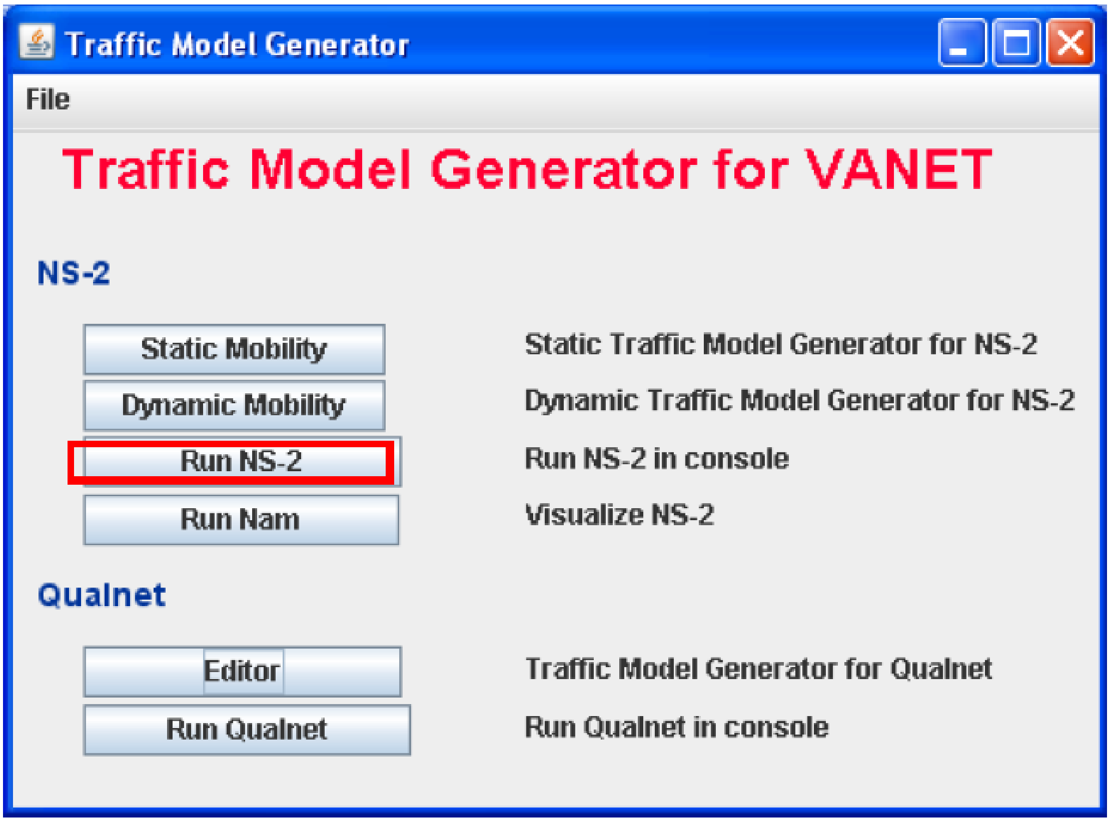
UDP
Packet size: 1000
Start Time:
Stop Time:
Sending Rate: 64Kb
Maximum Packets: 2280000
☐ Introduce Random Noise

TCP
Packet size: 1000
Window Size: 20
Start Time:
Stop Time:
Maximum Burst: 0
Maximum cwnd: 0

Figure 5.5.12 - NS-2 Script Generator

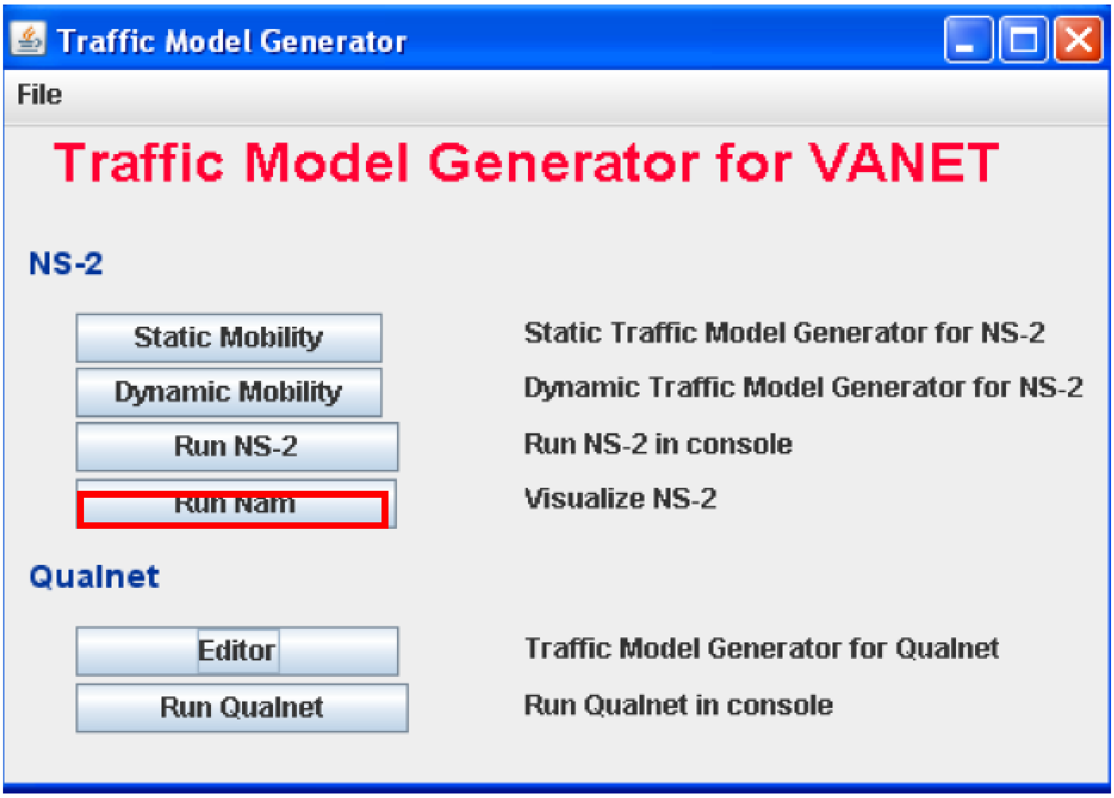
After that we run NS2 and visualize the simulation as we see in the figures 5.5.13 and 5.5.14. After select “Run NS-2” runs NS-2 in console (figure 5.5.13). Finally, you can call the NAM trace runner from the main menu (figure 5.5.14). In Figure 5.5.15 we see the NAM environment and we have the opportunity to play with it and see the actual movements.

NOTE: NS-2 runs only in Linux platform. The same happens with NAM.



Static Traffic Model Generator for NS-2
Dynamic Traffic Model Generator for NS-2
Run NS-2 in console
Visualize NS-2
Traffic Model Generator for Qualnet
Run Qualnet in console

Figure 5.5.13 - Run NS-2



Static Traffic Model Generator for NS-2
Dynamic Traffic Model Generator for NS-2
Run NS-2 in console
Visualize NS-2
Traffic Model Generator for Qualnet
Run Qualnet in console

Figure 5.5.14 - Run NAM

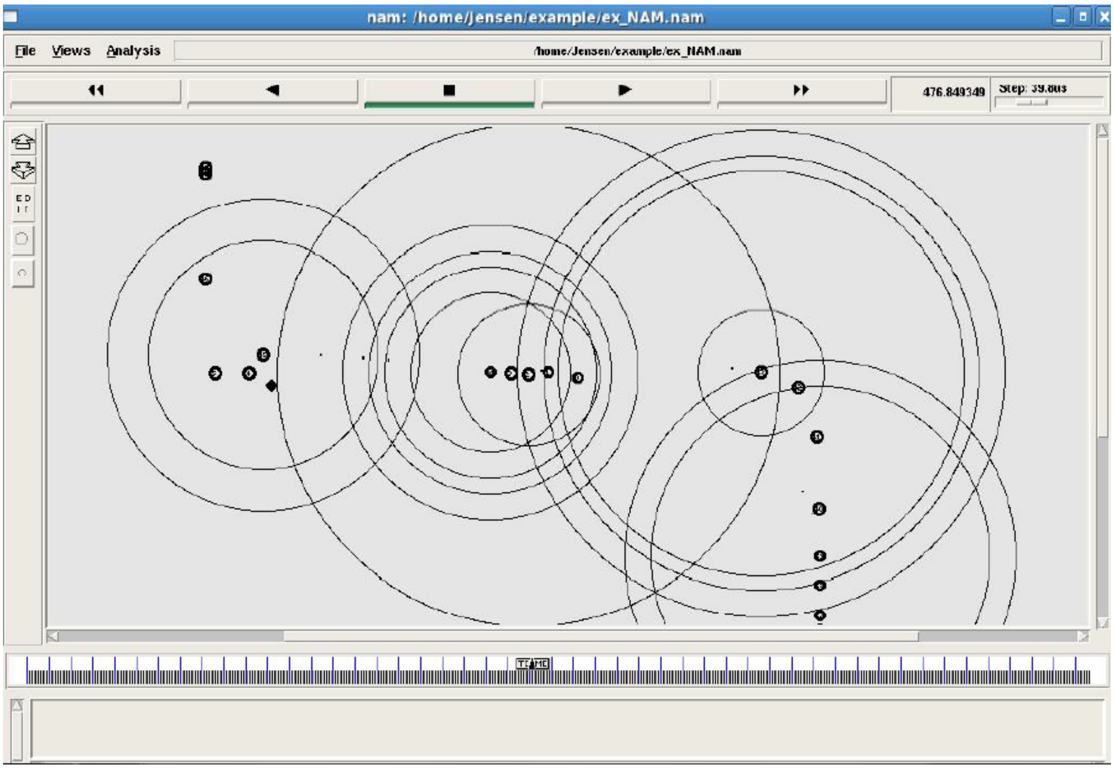


Figure 5.5.15 - NAM environment

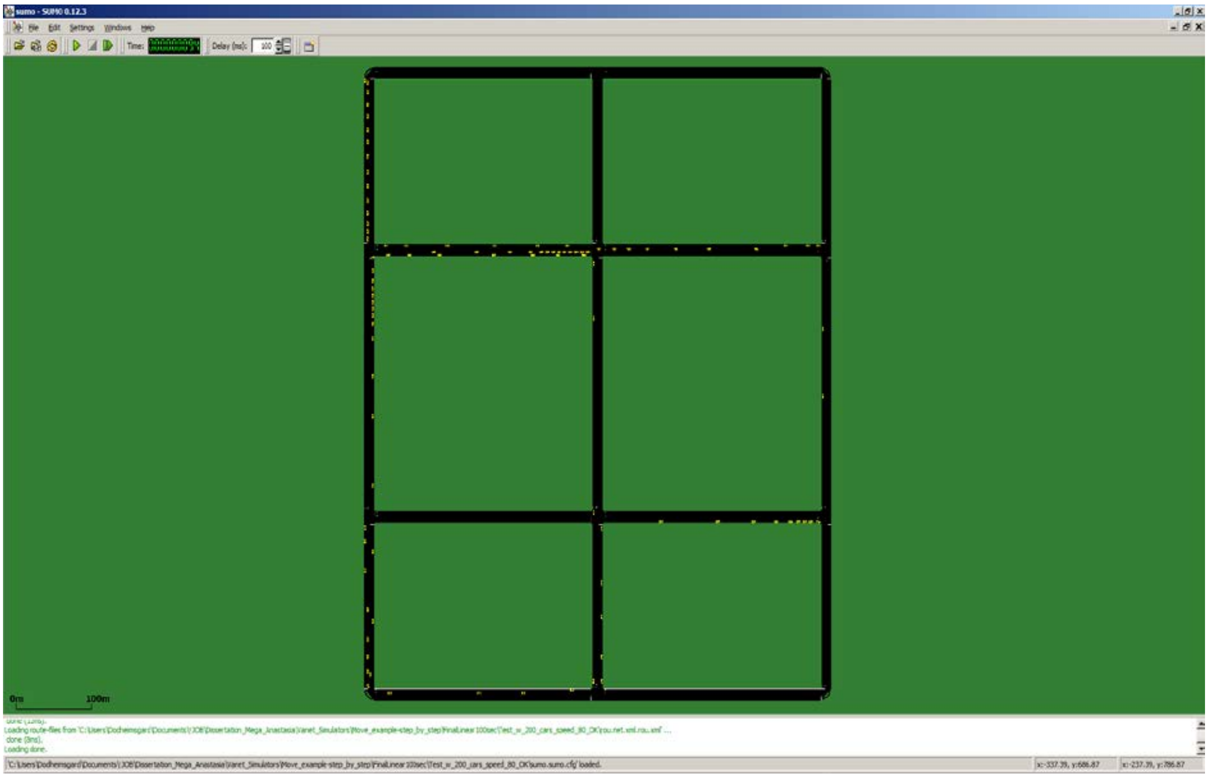
CHAPTER 6

SIMULATION MODELS AND RESULTS

To sum up, it was used one map and two scenarios, linear-neighbor scenario in which every node can communicate with the neighbor node and random in which every node can communicate with everyone. Furthermore, for both scenarios we have 50,100, 150 nodes and different velocity values 40m/s (~10km/h), 80 m/s (~20km/h),360 m/s(~100km/h). As far as the simulation time is concerned, it remains the same for all scenarios (200sec.). The fact that we want to compare different routing protocols urges me to use 3 different protocols (AODV, DSDV, DSR). The map we use appears below. Its dimensions are (654,70 X 570,80) m. When we talk about speed, we mean the maximum speed of the edges and not the speed of every node-vehicle. The table below (Table 4) depicts all the details that concern the simulation stage.

Platform	Linux and Windows 7 x64
NS version	2.32
Simulation time	200 sec
Number of nodes	50,100,150
Communication Type	CBR (Constant Bit Rate)
CBR Packet size	512 bytes
Simulation Area size	654,70 X 570,80 m
Node speed	40,80,360 m/s
Routing protocol	AODV, DSDV,DSR

Table 4 - Simulation Setup



```
done (12ms).
Loading route-files from 'C:\Users\Dodheimsgard\Documents\JOB\Dissertation_Mega_Anastasia\Vanet_Simulators\Move_example-step_by_step\FinalLinear 100sec\Test_w_200_cars_speed_80_OK\rou.net.xml.rou.xml' ...
 done (3ms).
Loading done.
'C:\Users\Dodheimsgard\Documents\JOB\Dissertation_Mega_Anastasia\Vanet_Simulators\Move_example-step_by_step\FinalLinear 100sec\Test_w_200_cars_speed_80_OK\sumo.sumo.cfg' loaded.
```

Figure 6.1 - Sumo map

In picture 6.2, we show a lot of vehicles, the different two way street, the traffic lights etc.

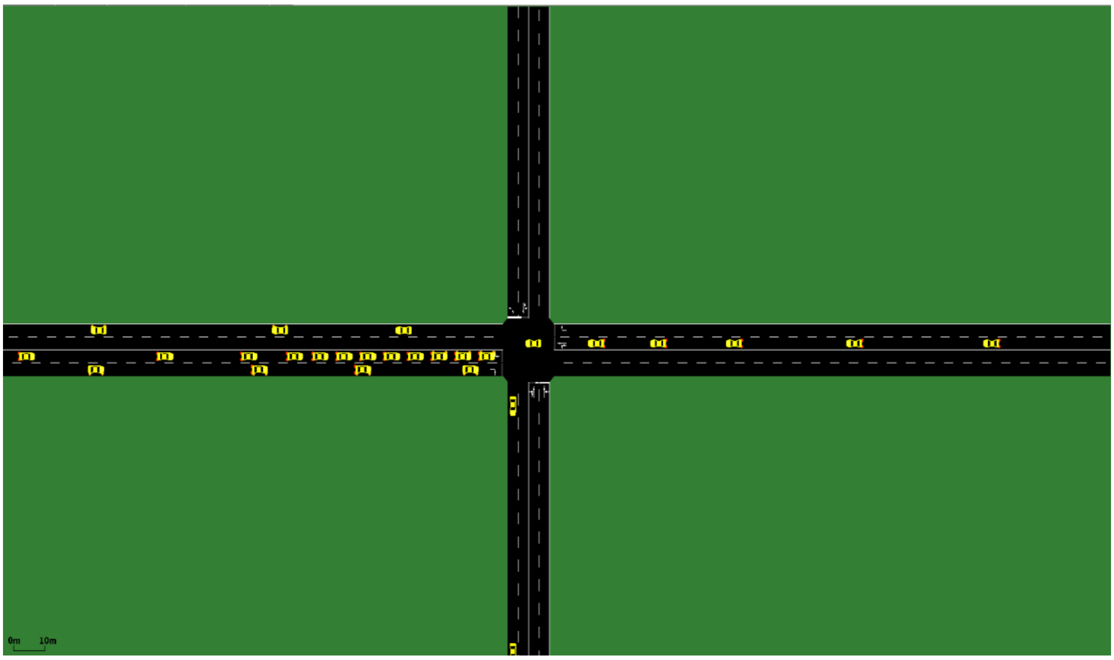


Figure 6.2 - A part of the map

After the different simulations and in order to make plots we take the below results. These results have arisen by using the application NS-2 Visual Trace Analyzer 0.2.72 in which load the tcl files and after that by selecting simulation we load the scenario file node movements (trace files) and so we take the results of the tables below.

The different routing protocols that we use extract different types of packets. The AODV exports one type of packet (CBR). But on the other hand the DSDV and the DSR protocols export two types of packets CBR and message packets for the first and CBR and DSR packets for the last. So for the plots, we use only the CBR type of packet in order to draw conclusions for the same things. It is true that in every comparison, there is need to compare the same things.

In order to understand the CBR, it is needed to give some details about this. Constant bit rate encoding means that the rate at which a codec's output data should be consumed is constant. CBR is useful for streaming multimedia content on limited capacity channels since it is the maximum bit rate that matters, not the average, so CBR would be used to take advantage of all of the capacity.

6.1 Routing protocols

An ad hoc routing protocol is a convention, or standard, that controls how nodes decide which way to route packets in between computing devices in a mobile ad hoc network. The routing protocol of VANET can be classified into two categories such as Topology based routing protocols and Position based routing protocols. Existing unicast routing protocols of VANET is not capable to meet every traffic scenarios. They have some pros and cons.

6.1.1 Ad hoc On-Demand Distance Vector (AODV)

The Ad hoc On-Demand Distance Vector (AODV) [103] algorithm enables dynamic, self-starting, multihop routing between participating mobile nodes wishing to establish and maintain an ad hoc network. AODV allows mobile nodes to obtain routes quickly for new destinations, and does not require nodes to maintain routes to destinations that are not in active communication. AODV allows mobile nodes to respond to link breakages and changes in network topology in a timely manner. The operation of AODV is loop-free, and by avoiding the Bellman-Ford "counting to infinity" problem offers quick convergence when the ad hoc network topology changes (typically, when a node moves in the network). When links break, AODV causes the affected set of nodes to be notified so that they are able to invalidate the routes using the lost link. Route Requests (RREQs), Route Replies (RREPs) and Route Errors (RERRs) are message types defined by AODV.

6.1.2 Dynamic Source Routing (DSR)

The Dynamic Source Routing protocol (DSR) is (Perkins, 2007), an on demand routing protocol. DSR is a simple and efficient routing protocol designed specifically for use in multi-hop wireless ad hoc networks of mobile nodes. Using DSR, the network is completely self-organizing and self-configuring, requiring no existing network infrastructure or administration. The DSR protocol is composed of two main mechanisms that work together to allow the discovery and maintenance of source routes in the ad hoc network [104]:

- Route Discovery is the mechanism by which a node S wishing to send a packet to a destination node D obtains a source route to D. Route Discovery is used only when S attempts to send a packet to D and does not already know a route to D.
- Route Maintenance is the mechanism by which node S is able to detect, while using a source route to D, if the network topology has changed such that it can no longer use its route to D because a link along the route no longer works. When Route Maintenance indicates a source route is broken, S can attempt to use any other route it happens to know to D, or it can invoke Route Discovery again to find a new route for subsequent packets to D. Route Maintenance for this route is used only when S is actually sending packets to D.

6.1.3 Destination-Sequenced Distance-Vector Routing (DSDV)

Destination-Sequenced Distance-Vector Routing (DSDV) is a table-driven routing scheme for ad hoc mobile networks based on the Bellman- Ford algorithm. It was developed by C. Perkins and

P.Bhagwat in 1994. It eliminates route looping, increases convergence speed, and reduces control message overhead.

In DSDV, each node maintains a next-hop table, which it exchanges with its neighbors. There are two types of next-hop table exchanges.

- Periodic full-table broadcast
- Event-driven incremental updating

The relative frequency of the full-table broadcast and the incremental updating is determined by the node mobility. In each data packet sent during a next-hop table broadcast or incremental updating, the source node appends a sequence number. This sequence number is propagated by all nodes receiving the corresponding distance-vector updates, and is stored in the next-hop table entry of these nodes. A node, after receiving a new next-hop table from its neighbor, updates its route to a destination only if the new sequence number is larger than the recorded one, or if the new sequence number is the same as the recorded one, but the new route is shorter. In order to further reduce the control message overhead, a settling time is estimated for each route. A node updates to its neighbors with a new route only if the settling time of the route has expired and the route remains optimal [105].

Type	Nodes	Speed(m/s)	Protocol	Generated Packets	Size (MB)	Dropped Packets	Size	%	Packet Delay (sec) AVG	Packet Jitter (sec) AVG	Throughtput (KBPS) Transferred AVG	Throughtput (KBPS) Generated AVG	Dropped Packets Loss By Destination	Dropped Packets Loss By Destination Multicast	Dropped Packets Loss by Link	Total Dropped Packets	
Linear	50	40	AODV	6675	1,000	2708	0,555	40,569%	0,4418195900	0,025198074	4,000	7,000	0	0	0	2708	CBR
	50	80		6675	1,000	3071	0,630	46,007%	0,4473012500	0,019035427	4,000	7,000	0	0	0	3071	
	50	360		6675	1,000	2682	0,550	40,180%	0,4300033300	0,017541846	4,000	7,000	0	0	0	2682	
	100	40		5750	1,000	1354	0,278	23,548%	0,7776266500	0,031844851	5,000	6,000	0	0	0	1354	
	100	80		5750	1,000	2210	0,453	38,452%	1,2117029000	0,054252227	4,000	6,000	1	0	1	2211	
	100	360		5750	1,000	1533	0,314	26,661%	1,3010589000	0,049076034	5,000	6,000	0	0	0	1533	
	150	40		12451	2,000	5849	1,000	46,976%	0,7447120500	0,528339010	7,000	13,000	0	0	0	5849	
	150	80		12728	3,000	5577	1,000	43,817%	0,5682290200	0,289581730	8,000	13,000	0	0	0	5577	
	150	360		12451	2,000	5103	1,000	40,985%	0,7108801200	0,359322590	8,000	13,000	0	0	0	5103	

Table 5 – One neighbor simulation with AODV protocol- CBR Type

Type	Nodes	Speed(m/s)	Protocol	Generated Packets	Size (MB)	Dropped Packets	Size	%	Packet Delay (sec) AVG	Packet Jitter (sec) AVG	Throughtput (KBPS) Transferred AVG	Throughtput (KBPS) Generated AVG	Dropped Packets Loss By Destination	Dropped Packets Loss By Destination Multicast	Dropped Packets Loss by Link	Total Dropped Packets	
Random	50	40	AODV	6675	1,000	2616	0,536	39,191%	0,4659632300	0,422687920	5,000	8,000	0	0	0	2616	CBR
	50	80		6675	1,000	2063	0,423	30,906%	0,2531587000	0,188220520	6,000	8,000	0	0	0	2063	
	50	360		6675	1,000	2536	0,520	37,993%	0,4163646100	0,351857410	5,000	8,000	0	0	0	2536	
	100	40		5750	1,000	1590	0,326	27,652%	0,5558108200	0,363310580	5,000	6,000	0	0	0	1590	
	100	80		5750	1,000	1721	0,353	29,930%	1,1229912000	0,844740190	4,000	6,000	0	0	0	1721	
	100	360		5750	1,000	1139	0,234	19,809%	0,6916785000	0,451556880	5,000	6,000	0	0	0	1139	
	150	40		5544	1,000	2465	0,506	44,462%	1,3622815000	1,481401700	3,000	6,000	0	0	0	2465	
	150	80		5544	1,000	1634	0,335	29,473%	2,1660295000	1,881558200	4,000	6,000	0	0	0	1634	
	150	360		5544	1,000	2191	0,449	39,520%	1,0752525000	0,939856850	4,000	6,000	0	0	0	2191	

Table 6 – Random simulation with AODV protocol – CBR type

Type	Nodes	Speed(m/s)	Protocol	Generated Packets	Size (MB)	Dropped Packets	Size	%	Packet Delay (sec) AVG	Packet Jitter (sec) AVG	Throughtput (KBPS) Transferred AVG	Throughtput (KBPS) Generated AVG	Dropped Packets Loss By Destination	Dropped Packets Loss By Destination Multicast	Dropped Packets Loss by Link	Total Dropped Packets	
Linear	50	40	DSDV	6675	1,000	5950	1,000	89,139%	0,2439618100	0,292293860	0,885	7,000	0	0	0	5950	CBR
	50	80		6675	1,000	6007	1,000	90,007%	0,3773217100	0,447604610	0,877	7,000	1	0	1	6008	
	50	360		6675	1,000	5742	1,000	86,037%	0,2220371300	0,288997520	1,000	7,000	1	0	1	5743	
	100	40		5750	1,000	5241	1,000	91,148%	0,6270020300	0,544538200	0,616	6,000	0	0	0	5241	
	100	80		5750	1,000	5614	1,000	97,652%	1,8945354000	0,923302830	0,156	6,000	1	0	1	5615	
	100	360		5750	1,000	5503	1,000	95,722%	1,4557618000	1,131543100	0,279	6,000	1	0	1	5504	
	150	40		12451	2,000	11461	2,000	92,049%	0,3734176900	0,436159170	1,000	13,000	0	0	0	11461	
	150	80		12728	3,000	11710	2,000	92,002%	0,5937098700	0,568638520	1,000	13,000	0	0	0	11710	
	150	360		12451	2,000	11511	2,000	92,450%	0,3433900360	0,434068950	1,000	13,000	0	0	0	11511	

Table 7 – One neighbor simulation with DSDV protocol- CBR type

Type	Nodes	Speed(m/s)	Protocol	Generated Packets	Size (MB)	Dropped Packets	Size	%	Packet Delay (sec) AVG	Packet Jitter (sec) AVG	Throughtput (KBPS) Transferred AVG	Throughtput (KBPS) Generated AVG	Dropped Packets Loss By Destination	Dropped Packets Loss By Destination Multicast	Dropped Packets Loss by Link	Total Dropped Packets	
Linear	50	40	DSDV	2854	0,487	132	0,027	4,660%	0,0023767660	0,,0016922546	0,037	2,000	0	1	1	133	Message
	50	80		2440	0,461	127	0,032	5,287%	0,0032676203	0,002257481	0,059	2,000	0	2	2	129	
	50	360		2303	0,450	139	0,031	6,079%	0,0033083700	0,002468989	0,087	2,000	0	1	1	140	
	100	40		7301	2,000	1080	0,313	14,820%	0,0171020550	0,015001466	0,146	11,000	0	2	2	1082	
	100	80		7684	2,000	1312	0,378	17,087%	0,0206357740	0,011703002	0,223	11,000	0	1	1	1313	
	100	360		7167	2,000	1061	0,308	14,804%	0,0103265390	0,006700533	0,356	11,000	0	0	0	1061	
	150	40		10579	4,000	2317	0,905	21,930%	0,0085904285	0,008306452	0,169	21,000	0	3	3	2320	
	150	80		11789	5,000	2935	1,000	24,913%	0,0193677150	0,013309398	0,097	24,000	0	2	2	2937	
	150	360		10018	4,000	2267	0,905	22,639%	0,0196065460	0,011818051	0,132	21,000	0	1	1	2268	

Table 8 – One neighbor simulation with DSDV protocol – Message type

Type	Nodes	Speed(m/s)	Protocol	Generated Packets	Size (MB)	Dropped Packets	Size	%	Packet Delay (sec) AVG	Packet Jitter (sec) AVG	Throughtput (KBPS) Transferred AVG	Throughtput (KBPS) Generated AVG	Dropped Packets Loss By Destination	Dropped Packets Loss By Destination Multicast	Dropped Packets Loss by Link	Total Dropped Packets	
Random	50	40	DSDV	6675	1,000	4119	0,845	61,708%	0,0043765292	0,000879323	3,000	8,000	0	0	0	4119	CBR
	50	80		6675	1,000	4546	0,932	68,105%	0,0040328219	0,000529731	3,000	8,000	0	0	0	4546	
	50	360		6675	1,000	4144	0,850	62,082%	0,0044504923	0,001088258	3,000	8,000	0	0	0	4144	
	100	40		5750	1,000	3941	0,808	68,539%	0,1786380400	0,177951580	2,000	6,000	0	0	0	3941	
	100	80		5750	1,000	3802	0,780	66,139%	0,1244938820	0,007261136	2,000	6,000	1	0	1	3803	
	100	360		5750	1,000	3918	0,803	68,139%	0,0069333511	0,004047406	2,000	6,000	0	0	0	3918	
	150	40		5544	1,000	4130	0,847	74,495%	0,0629655750	0,107456060	2,000	6,000	0	0	0	4130	
	150	80		5544	1,000	3980	0,816	71,789%	0,0169154370	0,014705860	2,000	6,000	0	0	0	3980	
	150	360		3507	0,719	2483	0,506	71,286%	0,0264170820	0,031183518	2,000	5,000	17	0	17	2500	

Table 9 – Random simulation with DSDV protocol –CBR type

Type	Nodes	Speed(m/s)	Protocol	Generated Packets	Size (MB)	Dropped Packets	Size	%	Packet Delay (sec) AVG	Packet Jitter (sec) AVG	Throughtput (KBPS) Transferred AVG	Throughtput (KBPS) Generated AVG	Dropped Packets Loss By Destination	Dropped Packets Loss By Destination Multicast	Dropped Packets Loss by Link	Total Dropped Packets	
Random	50	40	DSDV	2436	0,473	168	0,370	6,979%	0,0022271301	0,001261664	0,032	2,000	0	2	2	170	Message
	50	80		2544	0,480	129	0,028	5,149%	0,0022912658	0,001501080	0,068	2,000	0	2	2	131	
	50	360		2313	0,446	150	0,034	6,615%	0,0043040221	0,004000214	0,099	2,000	0	3	3	153	
	100	40		7264	2,000	1090	0,315	15,019%	0,0222143090	0,016864404	0,115	11,000	0	1	1	1091	
	100	80		7670	2,000	1110	0,334	14,472%	0,0152768840	0,008632618	0,149	12,000	0	0	0	1110	
	100	360		7207	2,000	1071	0,320	14,888%	0,0080967186	0,006847136	0,333	11,000	0	2	2	1073	
	150	40		8863	4,000	1880	0,768	21,234%	0,0097388370	0,010248966	0,141	19,000	0	2	2	1882	
	150	80		10710	4,000	2351	0,914	21,951%	0,0168723550	0,010700929	0,100	22,000	0	0	0	2351	
	150	360		6142	2,000	1701	0,592	27,988%	0,0161869670	0,010503351	0,092	16,000	0	18	18	1719	

Table 10 – Random simulation with DSDV protocol – Message type

Type	Nodes	Speed(m/s)	Protocol	Generated Packets	Size (MB)	Dropped Packets	Size	%	Packet Delay (sec) AVG	Packet Jitter (sec) AVG	Throughtput (KBPS) Transferred AVG	Throughtput (KBPS) Generated AVG	Dropped Packets Loss By Destination	Dropped Packets Loss By Destination Multicast	Dropped Packets Loss by Link	Total Dropped Packets	
Linear	50	40	DSR	6675	1,000	2792	0,565	42,397%	2,2836312000	0,239819060	4,000	7,000	38	0	38	2830	CBR
	50	80		6675	1,000	3439	0,698	52,015%	3,0732223000	0,317998440	4,000	7,000	33	0	33	3472	
	50	360		6675	1,000	2803	0,587	42,172%	1,9245232000	0,153482970	4,000	7,000	12	0	12	2815	
	100	40		5750	1,000	4679	0,918	84,904%	5,9634393000	0,742721840	1,000	6,000	203	0	203	4882	
	100	80		5750	1,000	2987	0,612	52,035%	3,1924902000	0,523613700	3,000	6,000	5	0	5	2992	
	100	360		5750	1,000	4262	0,833	77,600%	4,2006728000	0,683367530	2,000	6,000	200	0	200	4462	
	150	40		12451	2,000	10595	2,000	86,716%	5,3210650000	0,757229340	2,000	13,000	202	0	202	10797	
	150	80		12728	3,000	11504	2,000	92,190%	9,3471136000	1,472004900	3,000	13,000	230	0	230	11734	
	150	360		12451	2,000	10799	2,000	88,378%	6,3844723000	1,324380600	3,000	13,000	205	0	205	11004	

Table 11 – One neighbor simulation with DSR protocol – CBR type

Type	Nodes	Speed(m/s)	Protocol	Generated Packets	Size (MB)	Dropped Packets	Size	%	Packet Delay (sec) AVG	Packet Jitter (sec) AVG	Throughtput (KBPS) Transferred AVG	Throughtput (KBPS) Generated AVG	Dropped Packets Loss By Destination	Dropped Packets Loss By Destination Multicast	Dropped Packets Loss by Link	Total Dropped Packets	
Linear	50	40	DSR	3126	0,162	676	0,038	21,689%	0,1885320500	0,270476780	0,615	0,835	0	2	2	678	DSR
	50	80		3190	0,165	620	0,034	19,530%	0,2132945000	0,324391190	0,634	0,851	1	2	3	623	
	50	360		2258	0,113	429	0,022	19,221%	0,1400392300	0,172181240	0,436	0,579	3	2	5	434	
	100	40		41976	2,000	20499	1,000	48,983%	2,8841649000	2,851863500	6,000	11,000	62	0	62	20561	
	100	80		12857	0,707	3589	0,212	28,125%	0,7448906100	0,856218760	3,000	4,000	26	1	27	3616	
	100	360		30568	2,000	12986	0,707	42,626%	2,0025975000	2,158796200	5,000	8,000	44	0	44	13030	
	150	40		41587	2,000	19443	1,000	46,957%	4,6210000000	4,093000000	6,000	11,000	84	1	85	19528	
	150	80		101192	5,000	69905	3,000	69,168%	5,2471840000	3,791148800	8,000	26,000	85	2	87	69992	
	150	360		87862	4,000	59183	3,000	67,468%	5,6296299000	3,677814600	7,000	22,000	96	0	96	59279	

Table 12 – One neighbor simulation with DSR protocol – DSR type

Type	Nodes	Speed(m/s)	Protocol	Generated Packets	Size (MB)	Dropped Packets	Size	%	Packet Delay (sec) AVG	Packet Jitter (sec) AVG	Throughtput (KBPS) Transferred AVG	Throughtput (KBPS) Generated AVG	Dropped Packets Loss By Destination	Dropped Packets Loss By Destination Multicast	Dropped Packets Loss by Link	Total Dropped Packets	
Random	50	40	DSR	6675	1,000	2730	0,560	40,899%	0,6248403100	0,308923150	5,000	8,000	0	0	0	2730	CBR
	50	80		6675	1,000	2774	0,560	42,172%	1,2121169000	1,063133100	5,000	8,000	41	0	41	2815	
	50	360		6675	1,000	3190	0,653	47,880%	0,7184139500	0,636381170	4,000	8,000	6	0	6	3196	
	100	40		5750	1,000	3707	0,746	65,687%	3,4353249000	2,088362000	2,000	6,000	70	0	70	3777	
	100	80		5750	1,000	2308	0,473	40,209%	1,2131730000	0,736355250	4,000	6,000	4	0	4	2312	
	100	360		5750	1,000	4039	0,815	71,357%	4,1977588000	0,972703620	2,000	6,000	64	0	64	4103	
	150	40		5544	1,000	3485	0,702	63,943%	4,4708643000	1,840916400	2,000	6,000	60	0	60	3545	
	150	80		5544	1,000	4206	0,806	80,844%	4,0833213000	0,650758700	2,000	6,000	276	0	276	4482	
	150	360		5544	1,000	2741	0,561	49,513%	1,6933116000	1,140406300	3,000	6,000	4	0	4	2745	

Table 13 – Random simulation with DSR protocol – CBR type

Type	Nodes	Speed(m/s)	Protocol	Generated Packets	Size (MB)	Dropped Packets	Size	%	Packet Delay (sec) AVG	Packet Jitter (sec) AVG	Throughtput (KBPS) Transferred AVG	Throughtput (KBPS) Generated AVG	Dropped Packets Loss By Destination	Dropped Packets Loss By Destination Multicast	Dropped Packets Loss by Link	Total Dropped Packets	
Random	50	40	DSR	2304	0,115	392	0,020	17,014%	0,3433964500	0,325877110	0,476	0,657	0	0	0	392	DSR
	50	80		3146	0,159	519	0,029	16,497%	0,2439544900	0,271215550	0,730	0,922	0	0	0	519	
	50	360		3557	0,180	724	0,039	20,410%	0,1642028900	0,208700710	0,700	1,000	1	1	2	726	
	100	40		16782	0,915	3897	0,223	23,317%	0,6929420000	0,918450760	4,000	5,000	16	0	16	3913	
	100	80		5491	0,282	853	0,046	15,607%	0,3796632800	0,517927470	1,000	1,000	4	0	4	857	
	100	360		38103	2,000	17924	0,987	47,149%	2,6144208000	2,646430800	5,000	10,000	40	1	41	17965	
	150	40		27360	1,000	7270	0,397	26,809%	2,0456205000	2,251428700	5,000	7,000	65	0	65	7335	
	150	80		103029	5,000	70367	4,000	68,382%	4,7102981000	3,826472800	8,000	27,000	84	2	86	70453	
	150	360		10920	0,592	2089	0,123	19,130%	0,4312240600	0,458718730	2,000	3,000	0	0	0	2089	

Table 14 –Random simulation with DSR protocol – DSR type

6.2 Packet loss percentage

This dissertation has to do with data dissemination so its goal is to understand how the parameters of the network change by taking into account different routing protocols, different nodes, different speeds and different communications.

First of all we deal with the packet loss percentage in all scenarios.

Packet loss occurs when one or more packets of data travelling across a computer network fail to reach their destination. Packet loss can be caused by a number of factors including signal degradation over the network medium due to multi-path fading, packet drop because of channel congestion, corrupted packets rejected in-transit, faulty networking hardware, faulty network drivers or normal routing routines (such as DSR in ad-hoc networks). However, it is important to note that packet loss does not always indicate a problem. If the latency and the packet loss at the destination hop are acceptable then the hops prior to that one don't matter.

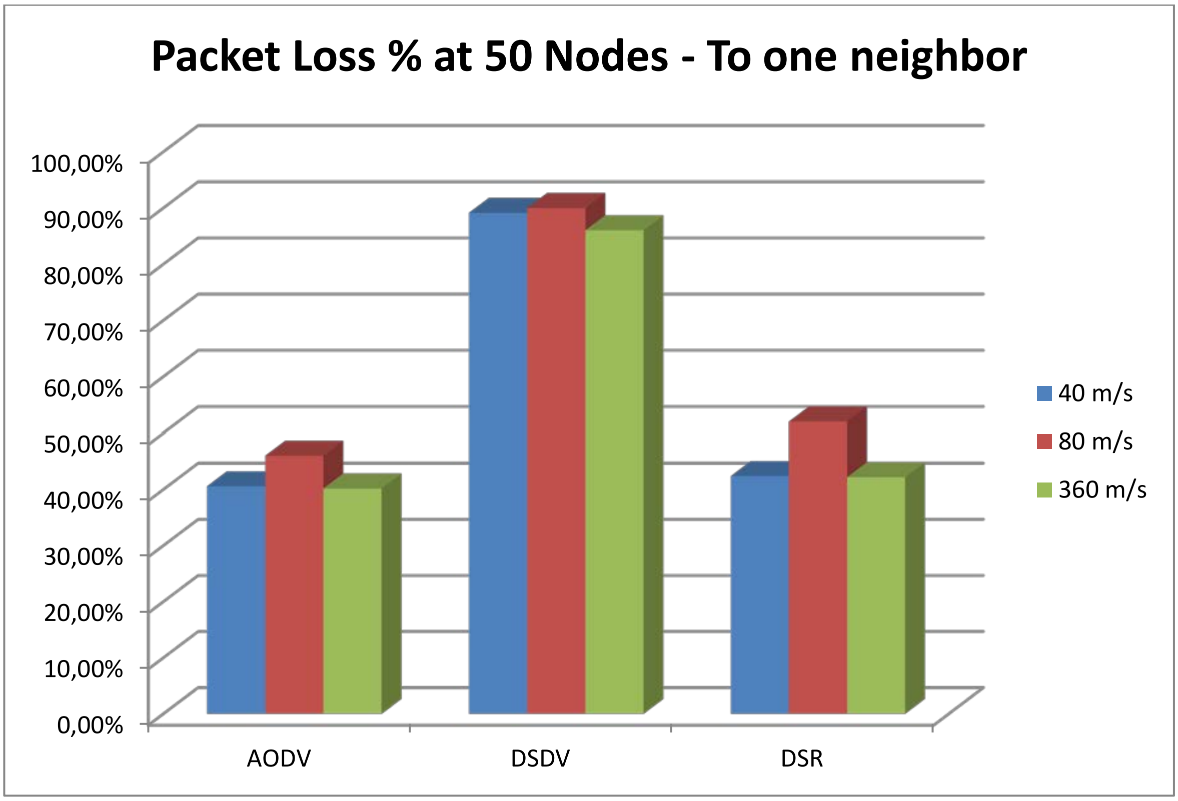


Figure 6.3 - Packet loss % at 50 Nodes – o one neighbor

As we can see in the picture above, for all speeds the AODV protocol seems to be the best routing protocol at the scenario of 50 nodes in communication in which every node can communicate with its neighbor. On the other hand the DSDV protocol has the biggest percentage of packet loss for all speeds and we can observe that the percentage tends to overcome the 90%. Another observation is that the different protocols behave with the same manner in all speeds so in 80 m/s occurs the biggest packet loss (Figure 6.3).

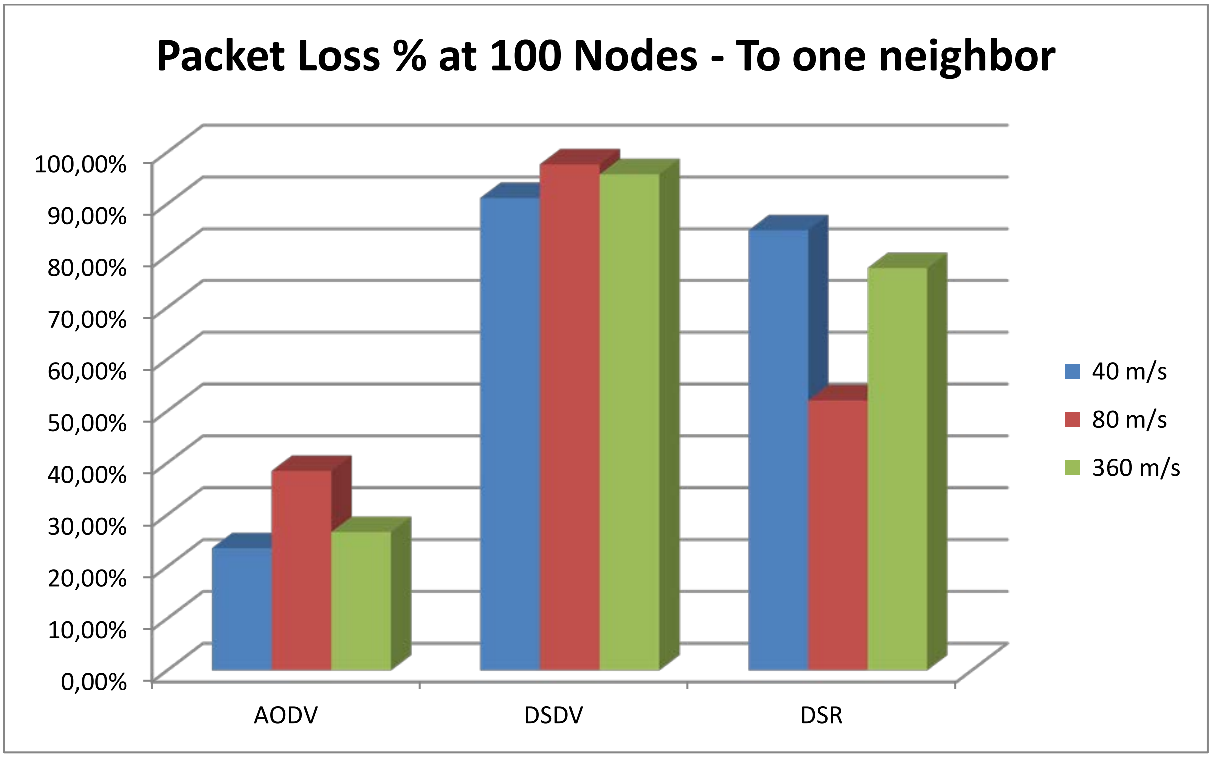


Figure 6.4 - Packet loss % at 100 Nodes – to one neighbor

At the scenario of 100 nodes, the AODV protocol remains the more efficient with the smaller percentage of packet loss. For DSDV protocol for the different speeds the packet loss percentage varies between 91.15% and 97.65%. The AODV and the DSR protocols act differently, the first has the biggest packet loss for 80m/s and the second for the same speed has the smallest packet loss (Figure 6.4).

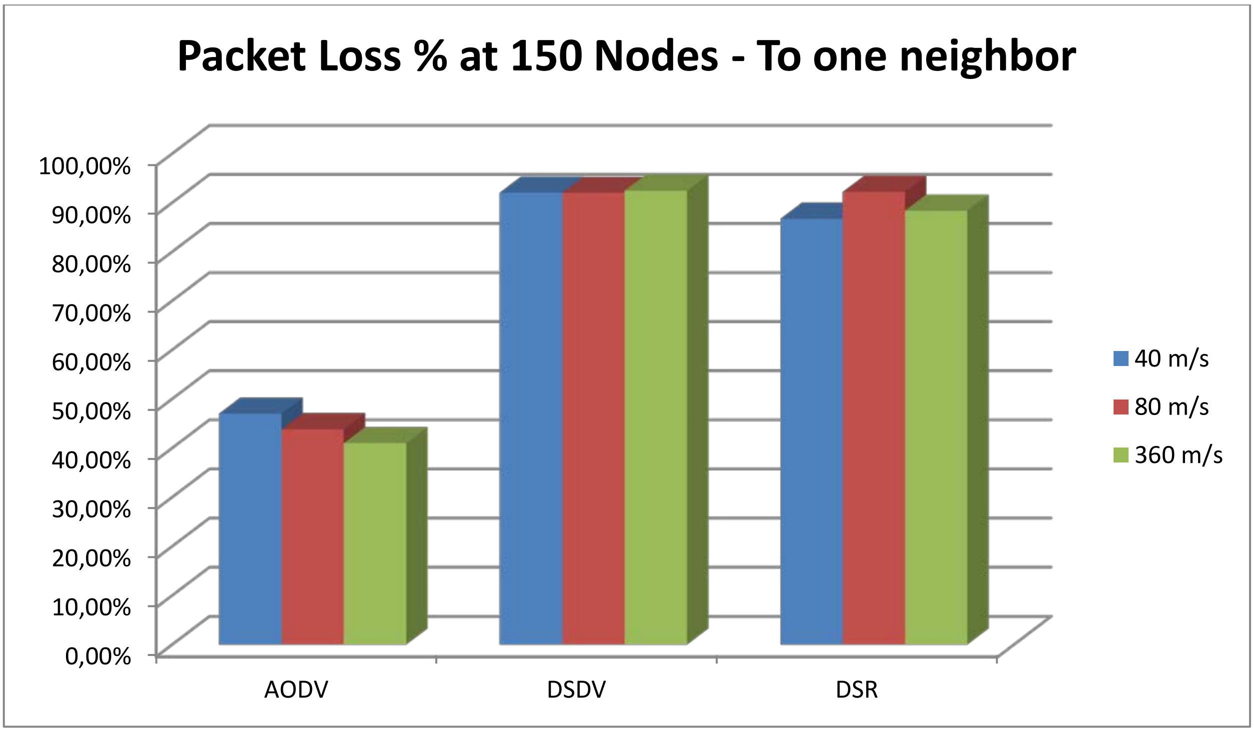


Figure 6.5 - Packet loss % at 150 Nodes – to one neighbor

At the scenario of 150 nodes the AODV remains the preferable protocol and at 360 m/s has the smallest loss. Last but not least the other two protocols have losses of about 90% (Figure 6.5).

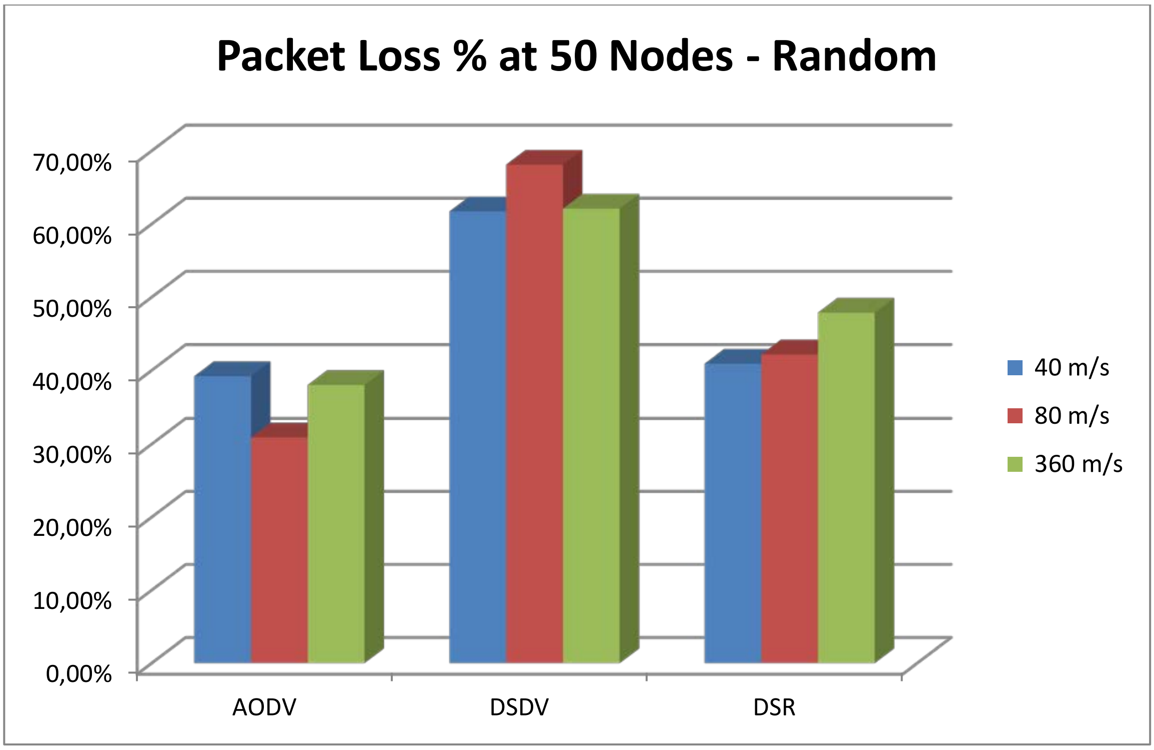


Figure 6.6 - Packet loss % at 50 Nodes – Random

At random communication (every node can communicate with everyone), the DSDV has the worst performance for all speeds. The AODV is the best protocol but the percentage of packet loss has increased compared with "to one neighbor" communication. It has the smallest percentage at 80m/s but this doesn't happen in other protocols (DSDV and DSR at 40m/s) (Figure 6.6).

At Figure 6.7 we observe that for the AODV protocol (100 nodes) at speed of 360 m/s, we have fewer loss. On the other hand, for the other protocols at speed of 80m/s happens the above. The percentage of both protocols (DSDV, DSR) ranges between 66% and 71%. An exception is the percentage at 80m/s for DSR protocol.

A similar picture is the Figure 6.8 where the AODV remains the best protocol but presents the lowest percentage at speed of 80m/s. The DSDV demonstrates the same performance for 150 nodes (72-74%).

So for the packet loss percentage for both communications the AODV protocol has fewer losses for both communications (one neighbor, random). As far as the nodes are concerned, at 100 nodes we have the smaller percentage of packet loss (Figure 6.4, 6.7).

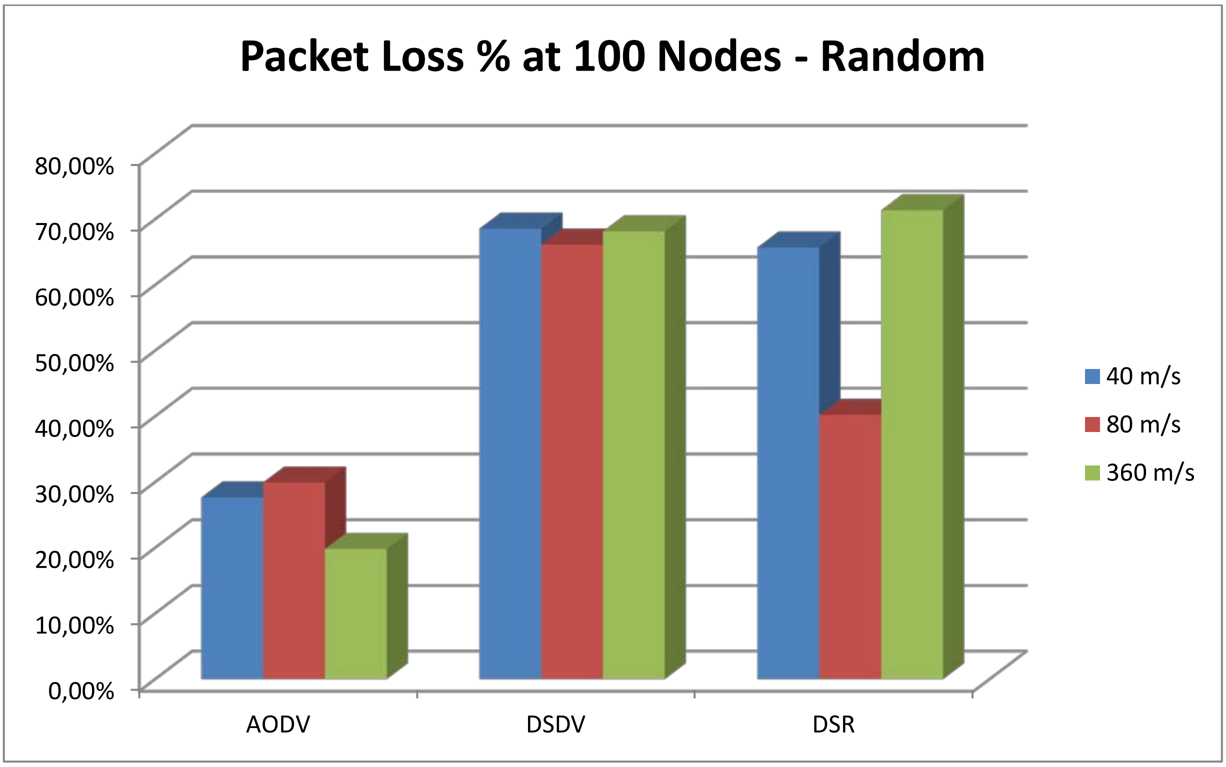


Figure 6.7 - Packet loss % at 100 Nodes -Random

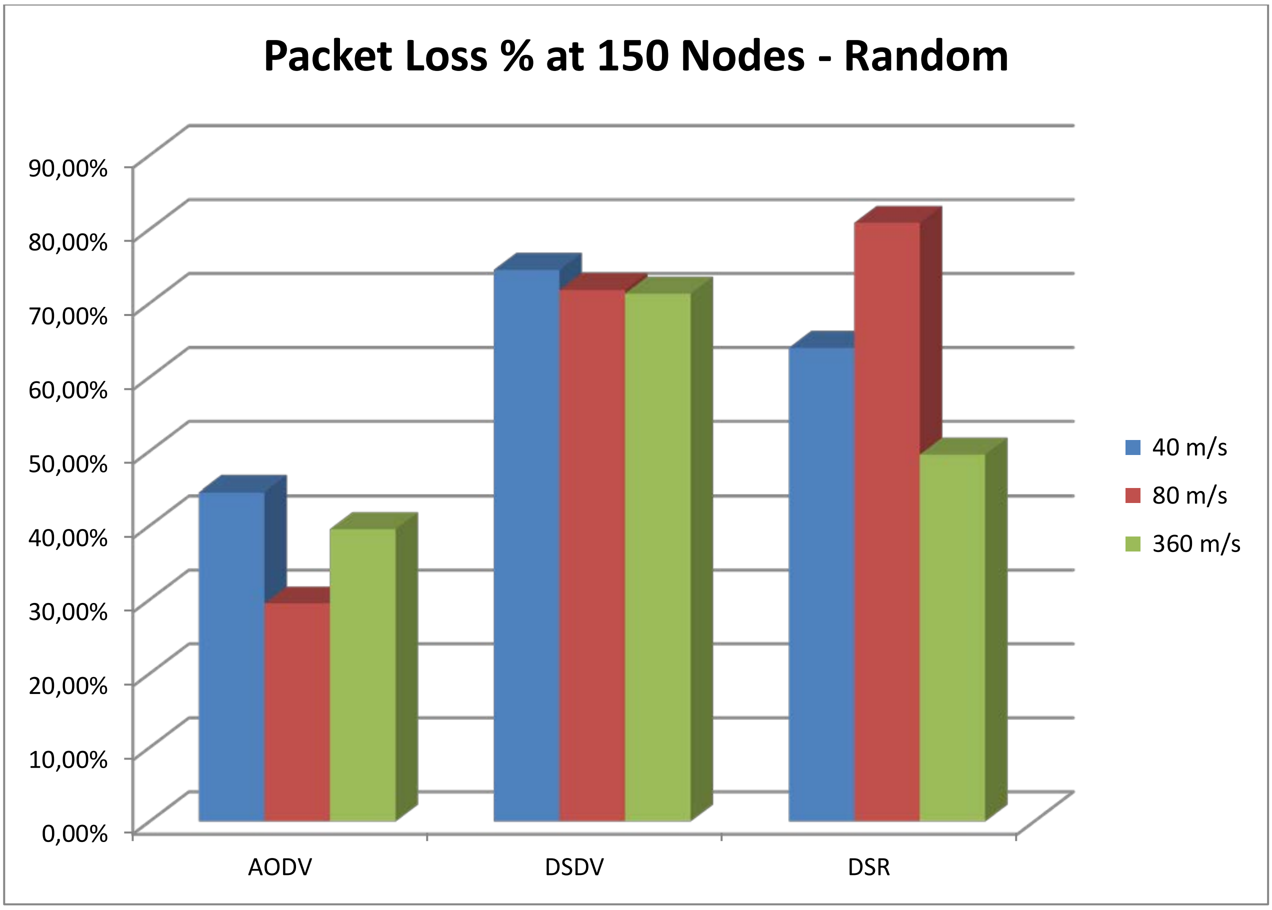


Figure 6.8 - Packet loss % at 150 Nodes – Random

6.3 Packet Delay

Another parameter that is needed to be taken into account in every network is the average packet delay. In computer networking, packet delay is the difference in end-to-end one-way delay between selected packets in a flow with any lost packets being ignored. The effect is sometimes referred to as jitter, although the definition is an imprecise fit. This size measured in seconds.

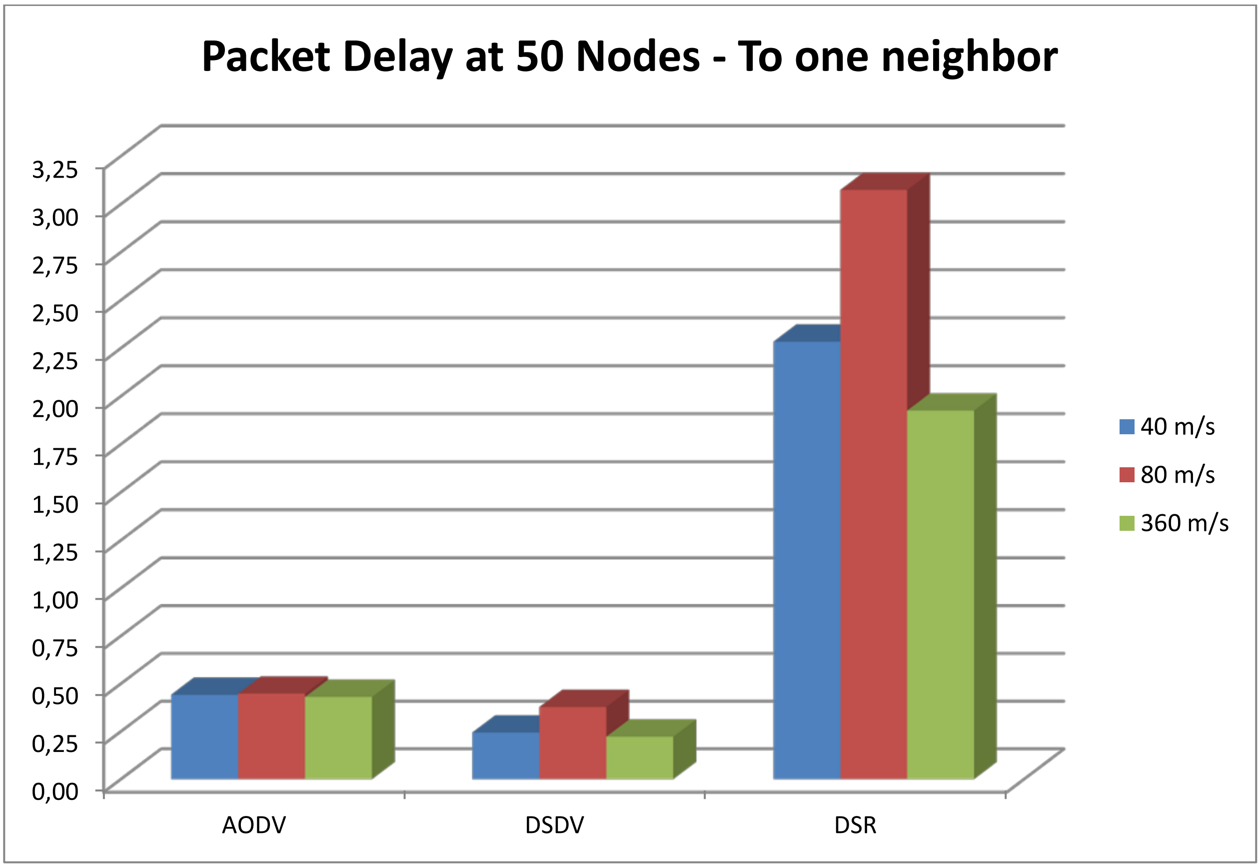


Figure 6.9 - Packet Delay at 50 Nodes – to one neighbor

In order to choose the more efficient protocol, we want the minimum packet delay. This happens (Figure 6.9) for DSDV protocol, followed by AODV and then DSR. At 360 m/s we have the minimum packet delay.

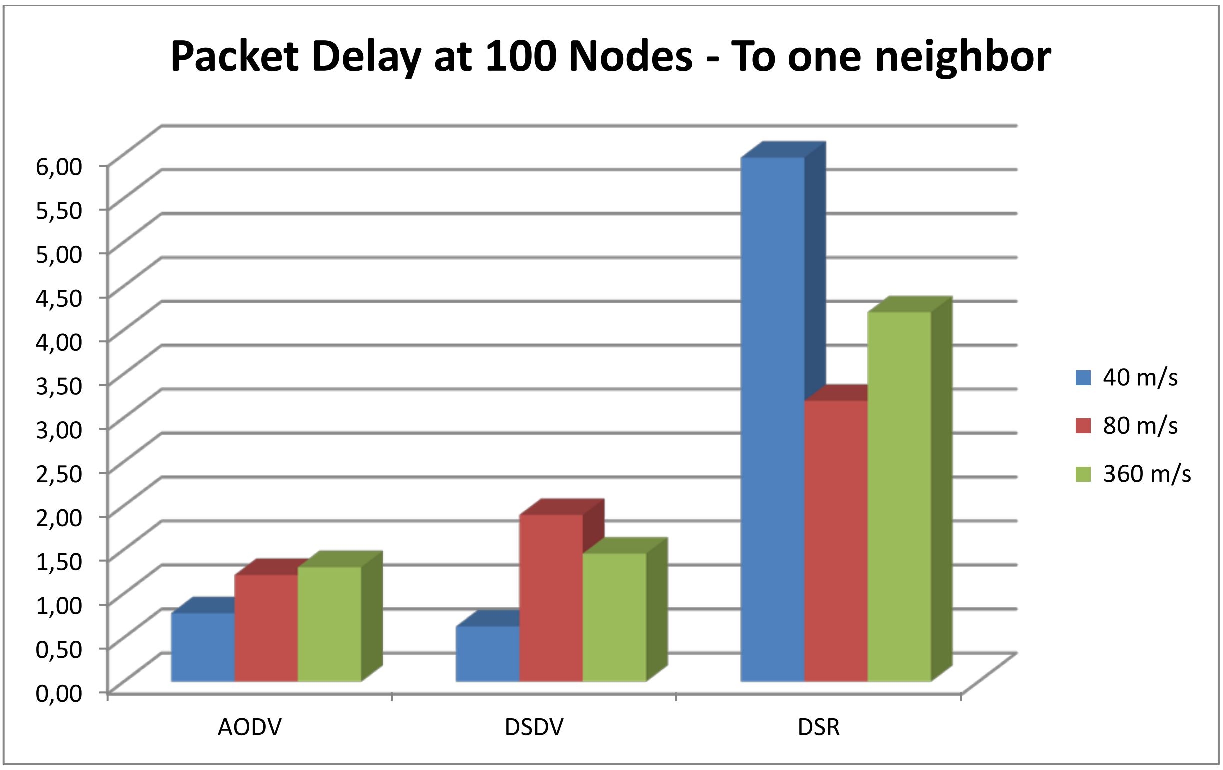


Figure 6.10 - Packet Delay at 100 Nodes – to one neighbor

For 100 nodes (Figure 6.10) the minimum packet delay has the DSDV protocol at the speed of 40m/s, followed by the AODV at the same speed. Then AODV at the speed of 80m/s, after this AODV at the speed of 360 m/s and followed by the other combinations of protocol and speed. Also for 100 nodes we select the DSDV only for the speed of 40m/s.

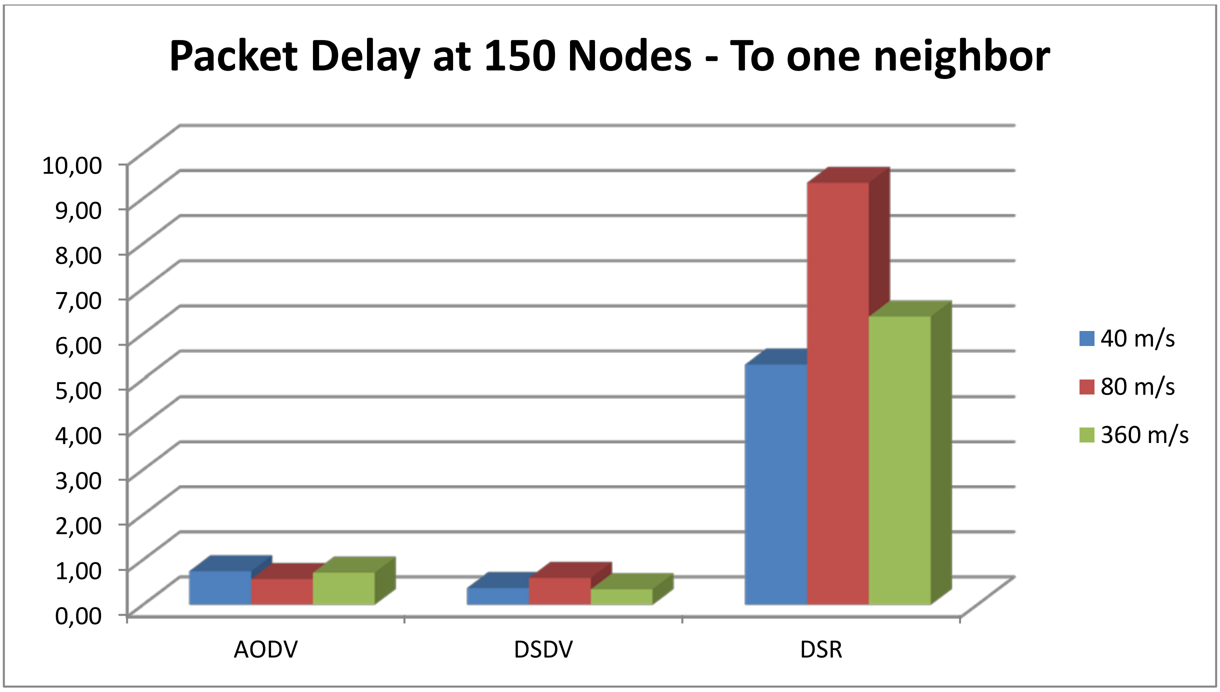


Figure 6.11 Packet Delay at 150 Nodes – to one neighbor

At 150 nodes (Figure 6.11) we have less packet delay for AODV and DSDV than 50 and 100 nodes and the DSDV remains the protocol with the best performance.

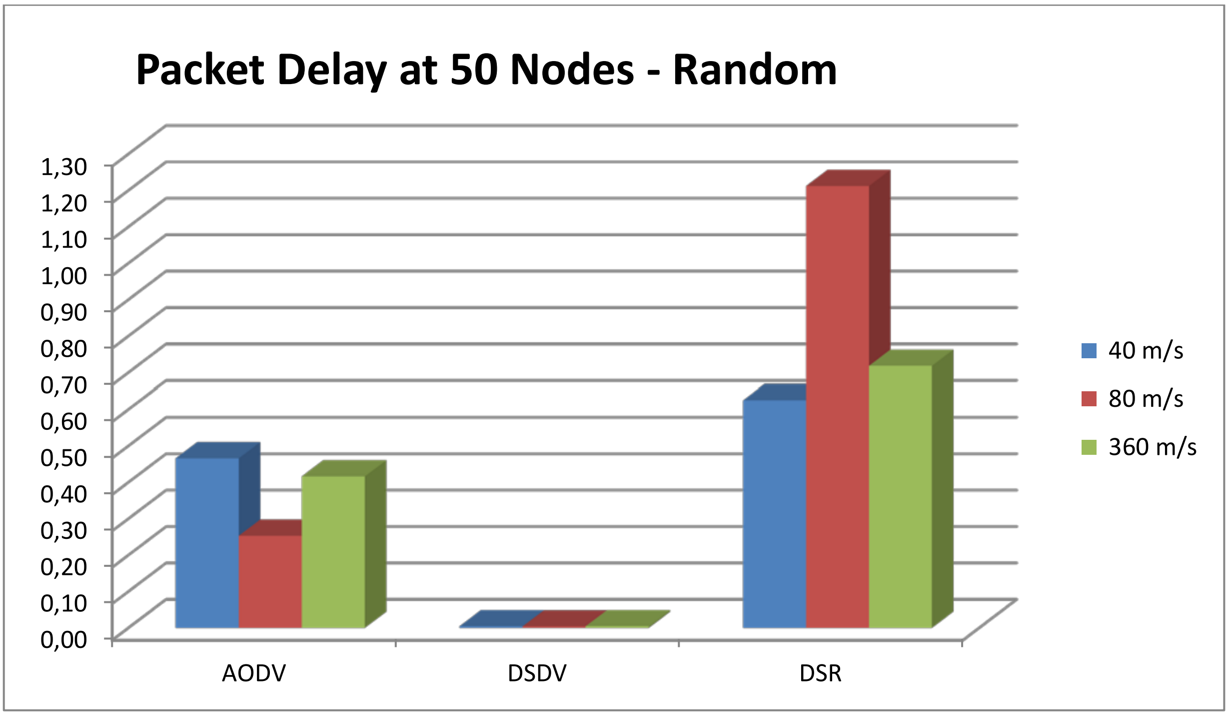


Figure 6.12 - Packet Delay at 50 Nodes – Random

For random communication (Figure 6.12) the DSDV protocol has minimal delay for every speed value. The other protocols act the same as for one neighbor communication at 150 nodes.

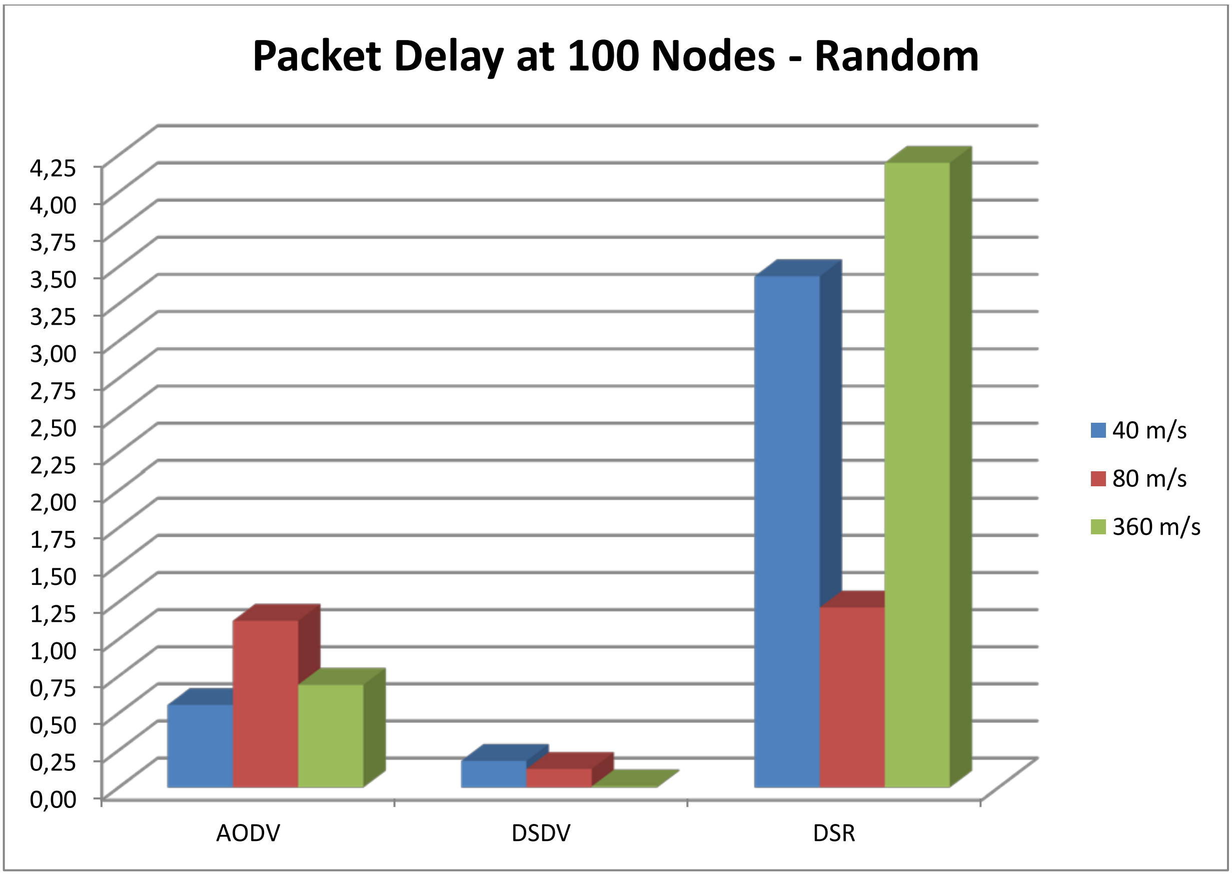


Figure 6.13 - Packet Delay at 100 Nodes – Random

For 100 nodes (Figure 6.13), packet delay for DSDV has increased negligibly (0,01 sec to 0,18 sec) and as a result remains the more efficient protocol from the three. For speeds of 80m/s except for DSDV, we choose DSR while for speeds of 40m/s and 360m/s the choice with the smallest delay is AODV.

At 150 nodes (Figure 6.14) we observe the same performance as at 50 nodes (minimal delays). At speed of 360m/s we have the smallest packet delay for AODV and DSR.

To sum up for the amount of packet delay, the DSDV is the more efficient routing protocol for both communications (to one neighbor and random). This fact strongly appears at random scenario.

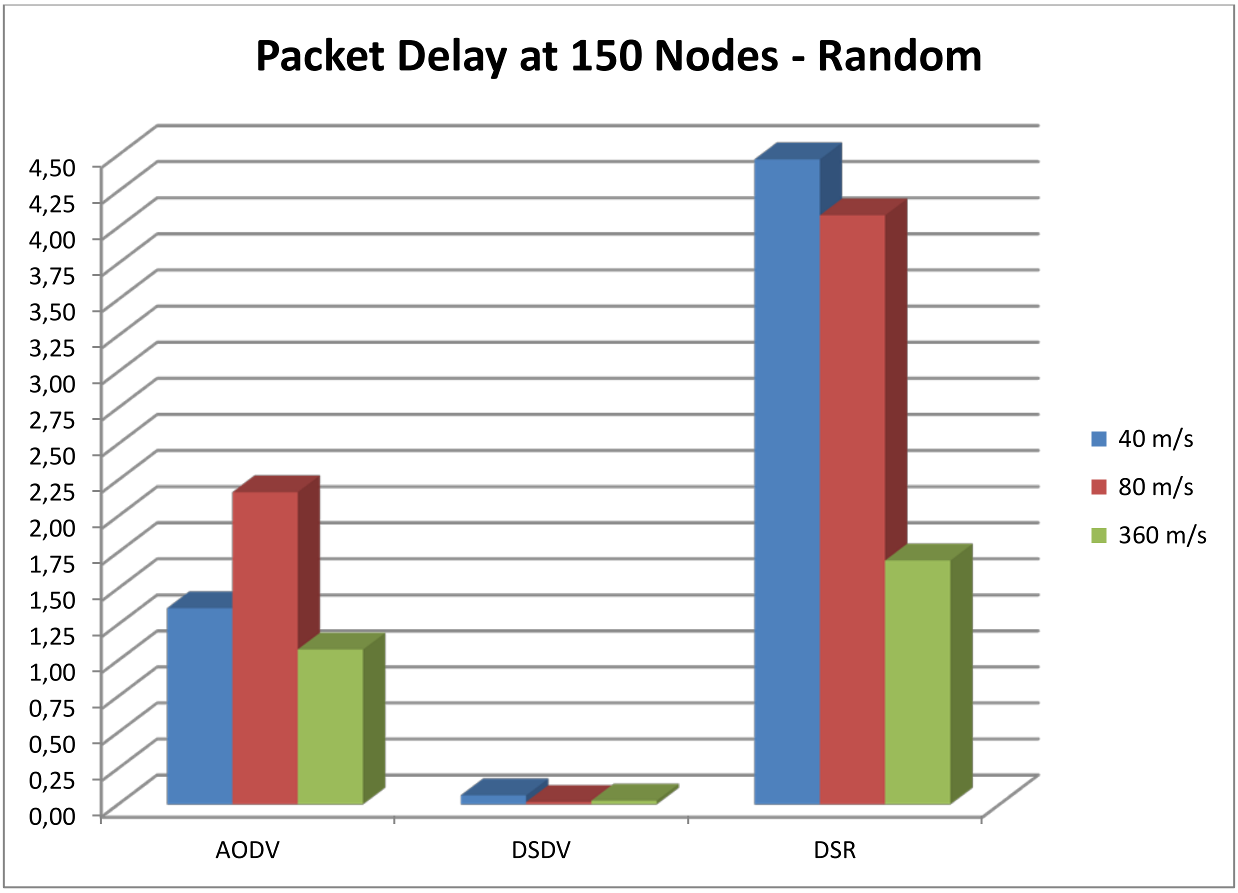


Figure 6.14 - Packet Delay at 150 Nodes – Random

6.4 Packet Jitter

In general, packet jitter is an incomprehensible concept for the majority of people but it is an important parameter for every network.

Jitter is the undesired deviation from true periodicity of an assumed periodic signal in electronics and telecommunications, often in relation to a reference clock source. Jitter may be observed in characteristics such as the frequency of successive pulses, the signal amplitude, or phase of periodic signals.

In the context of computer networks, the term jitter is often used as a measure of the variability over time of the packet latency across a network. A network with constant latency has no variation (or jitter). Packet jitter is expressed as an average of the deviation from the network mean latency. However, for this use, the term is imprecise. The standards-based term is *packet delay variation* (PDV). PDV is an important quality of service factor in assessment of network performance. This size measured in seconds.

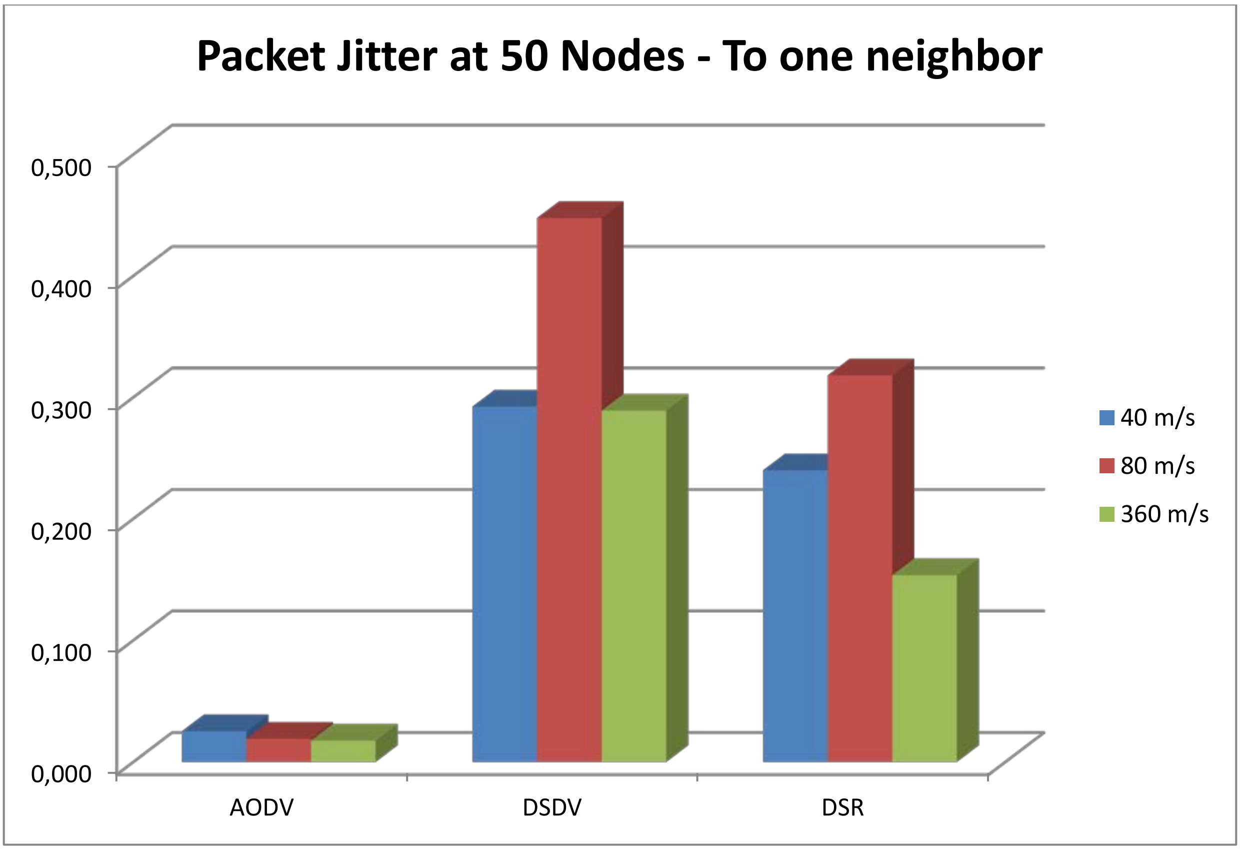


Figure 6.15 - Packet Jitter at 50 Nodes – to one neighbor

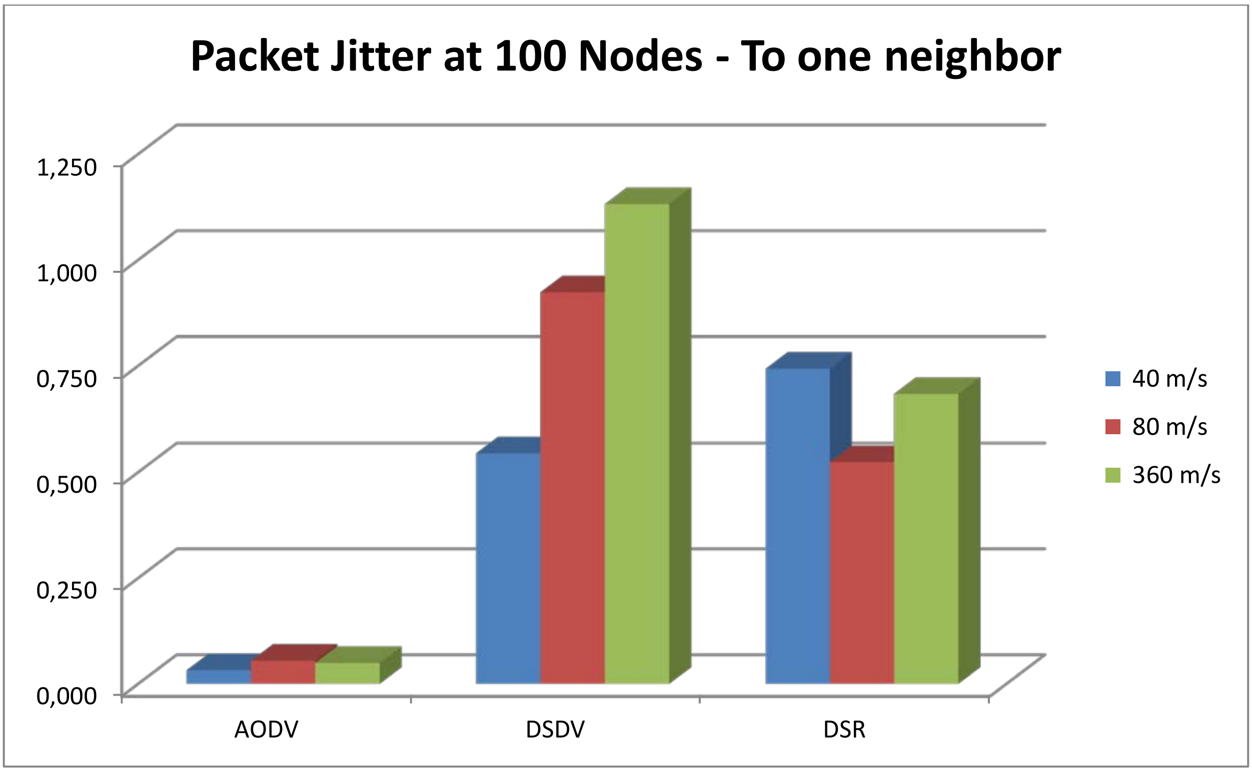


Figure 6.16 - Packet Jitter at 100 Nodes – to one neighbor

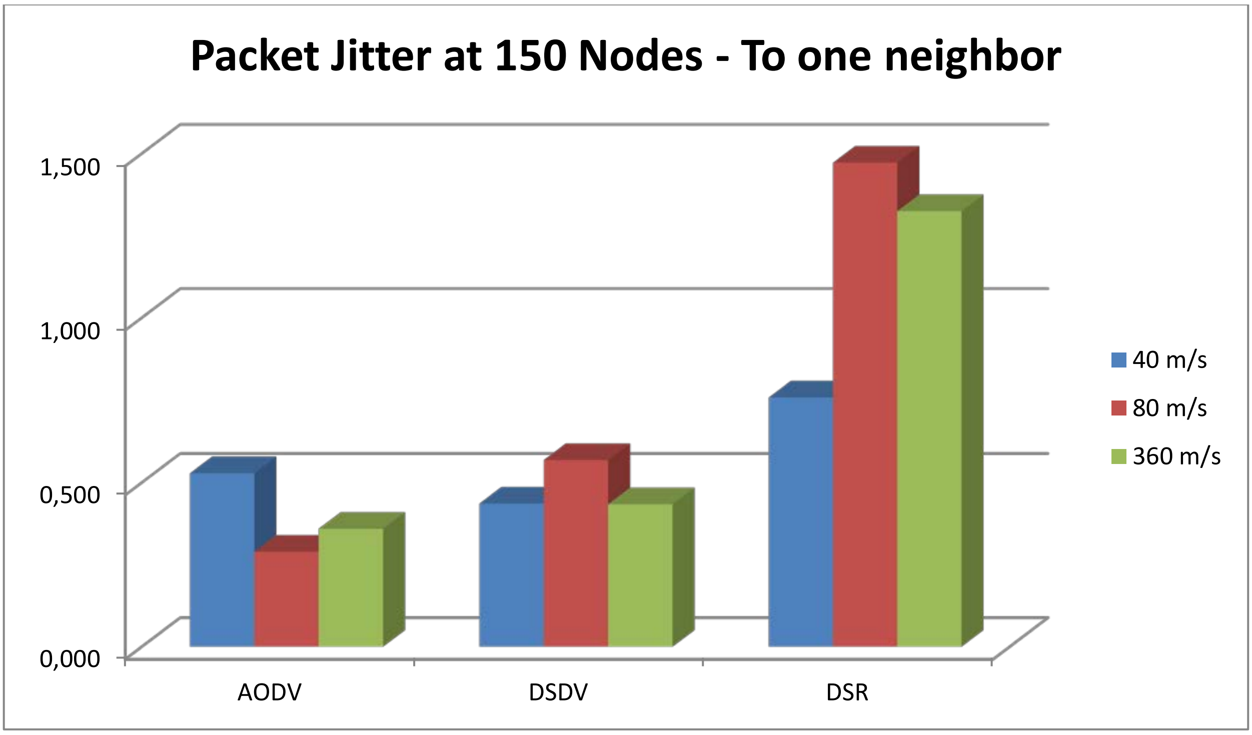


Figure 6.17 - Packet Jitter at 150 Nodes – to one neighbor

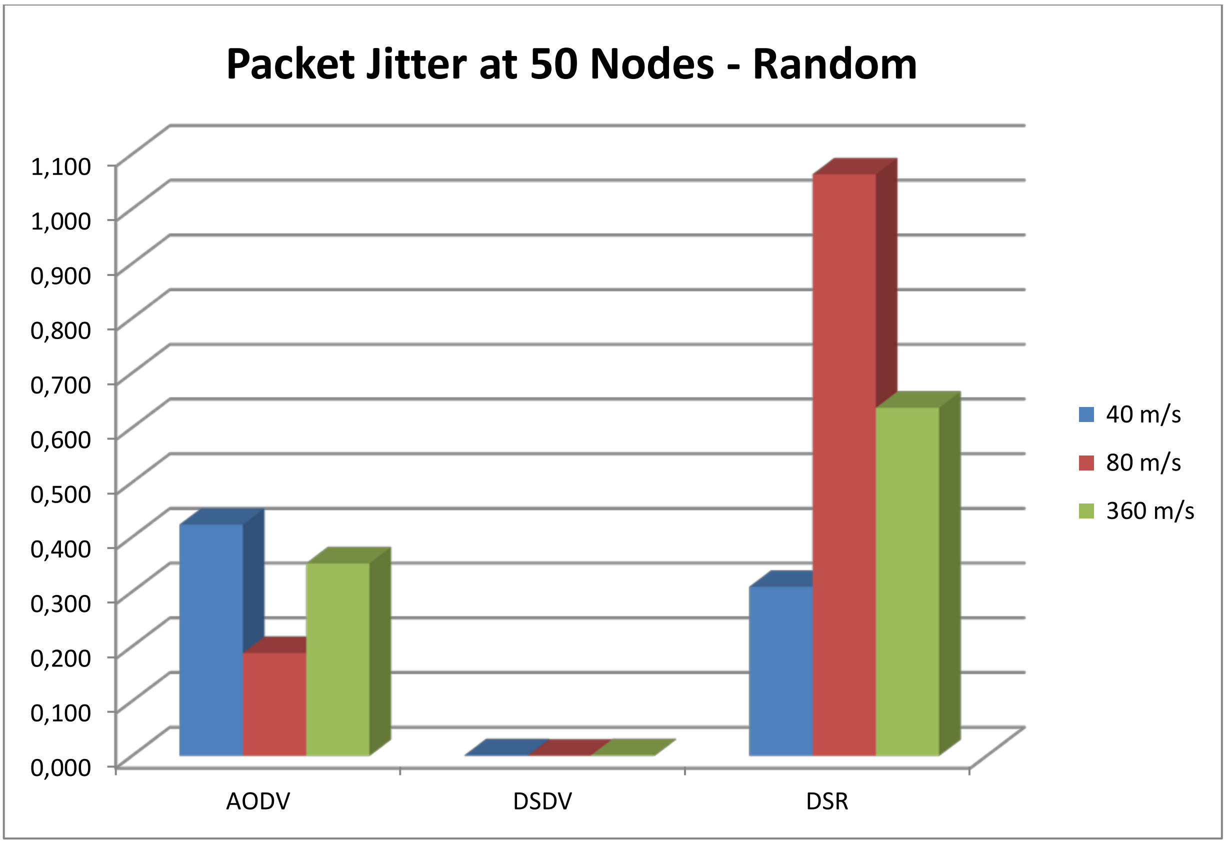


Figure 6.18 - Packet Jitter at 50 Nodes – Random

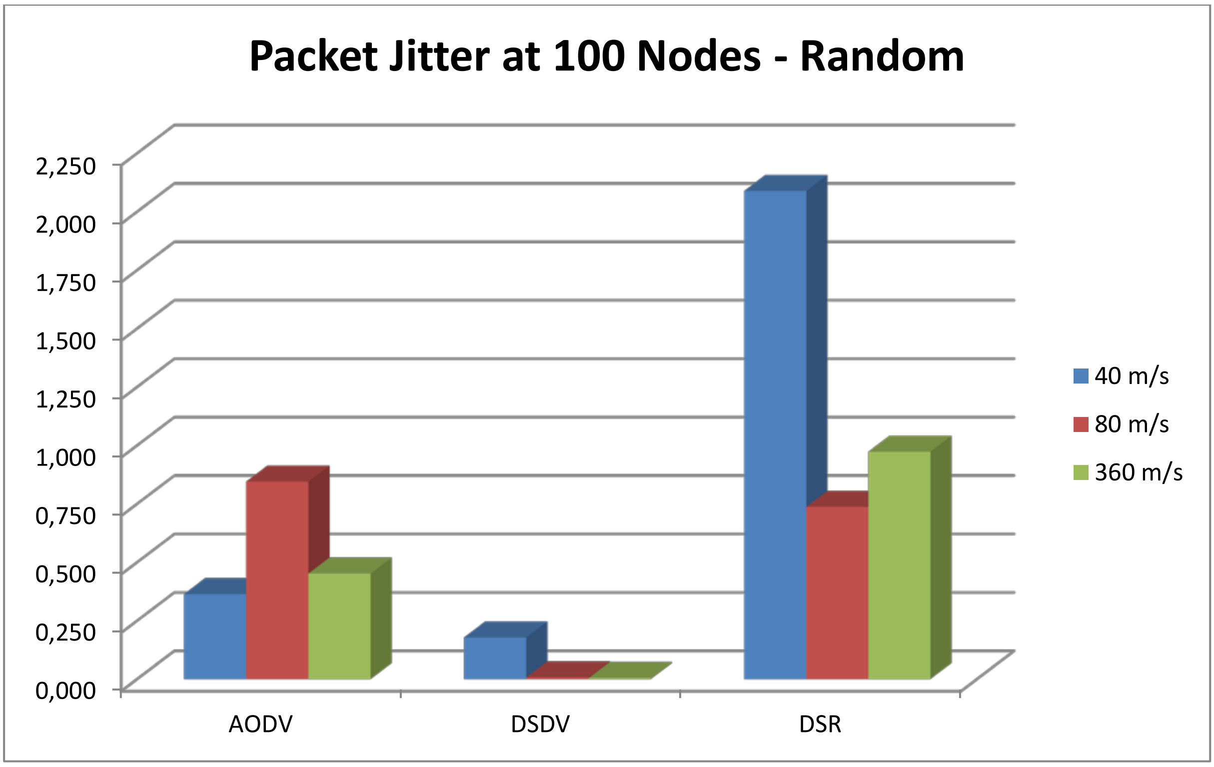


Figure 6.19 - Packet Jitter at 100 Nodes – Random

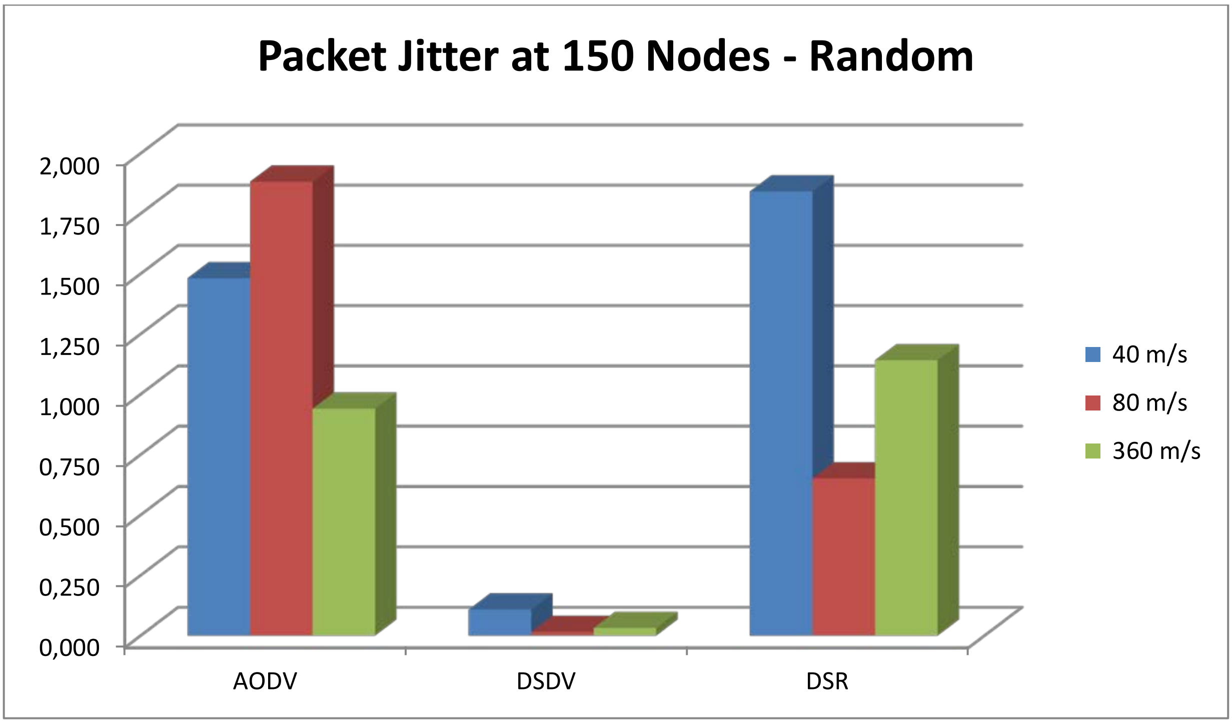


Figure 6.20 - Packet Jitter at 150 Nodes – Random

Both figures (6.19 and 6.20) represent approximately the same results. In other words, the biggest packet jitter for AODV is at 80m/s, the DSR and DSDV protocols have the same performance at 100 and 150 nodes. Another observation is that the DSDV is the most preferable routing protocol due to the small packet jitter (Figure 6.18 and 6.20).

For “to one neighbor” scenario we select the first protocol (AODV) due to the small number (sec) of packet jitter but this doesn’t happen for random communication where the DSDV protocol is the predominant.

6.5 Throughput Transferred

Throughput has been a measure of the comparative effectiveness of large commercial computers that run many programs concurrently. This size measures in kilobits per second.

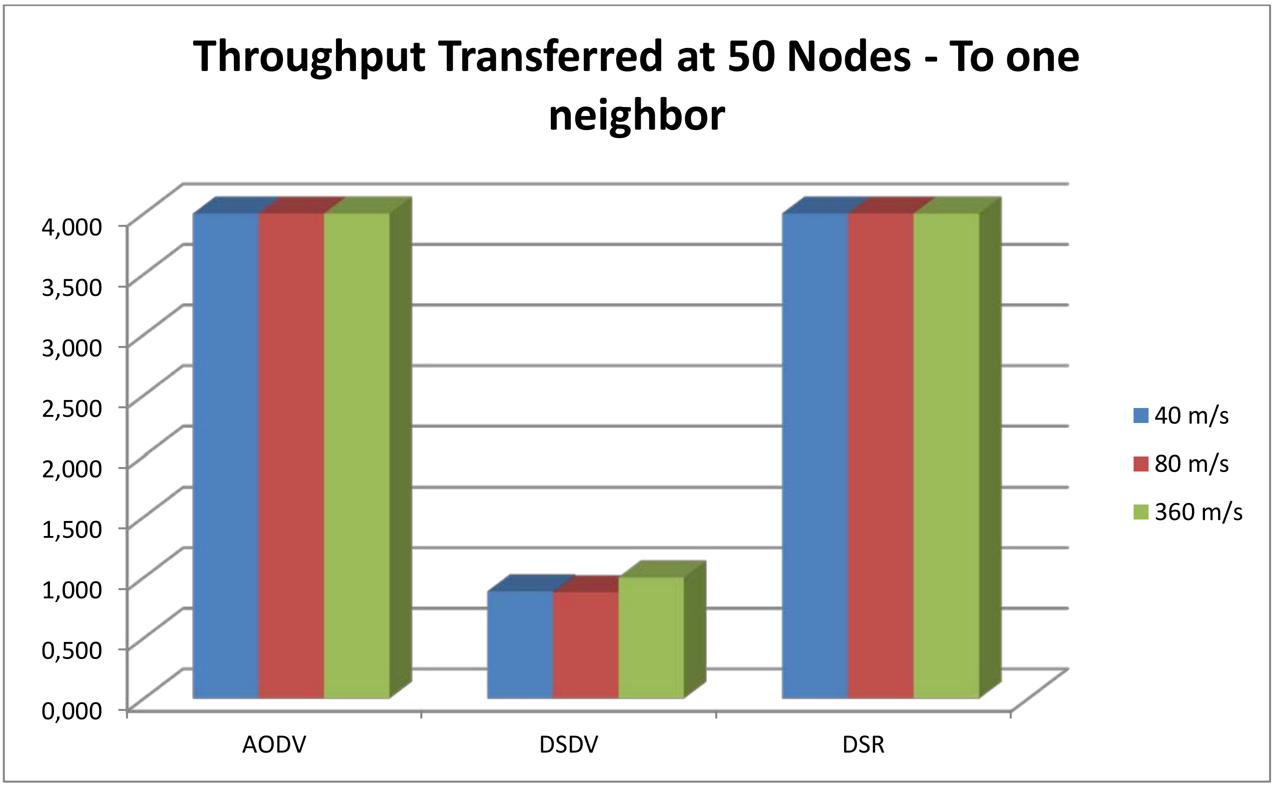


Figure 6.21 - Throughput Transferred at 50 Nodes – to one neighbor

At 50 nodes (Figure 6.21) both protocols (AODV, DSR) produce the same values at all velocity values (4,000 kbps). For DSDV protocol almost the same results appear so it has almost the same throughput at all speeds. So we can tell that the speed cannot play an important role for this simulation (To one neighbor- 50 nodes).

In Figure 6.22, we do not observe the same, the DSR protocol presents throughput transferred of about 1,000-3,000 kbps. For “To one neighbor” scenario the DSDV protocol has the minimum throughput transferred (Figures 6.21 to 6.23). Last but not least AODV has the best statistical results for 150 nodes (7,000-8,000 kbps) (Figures 6.21 to 6.23).

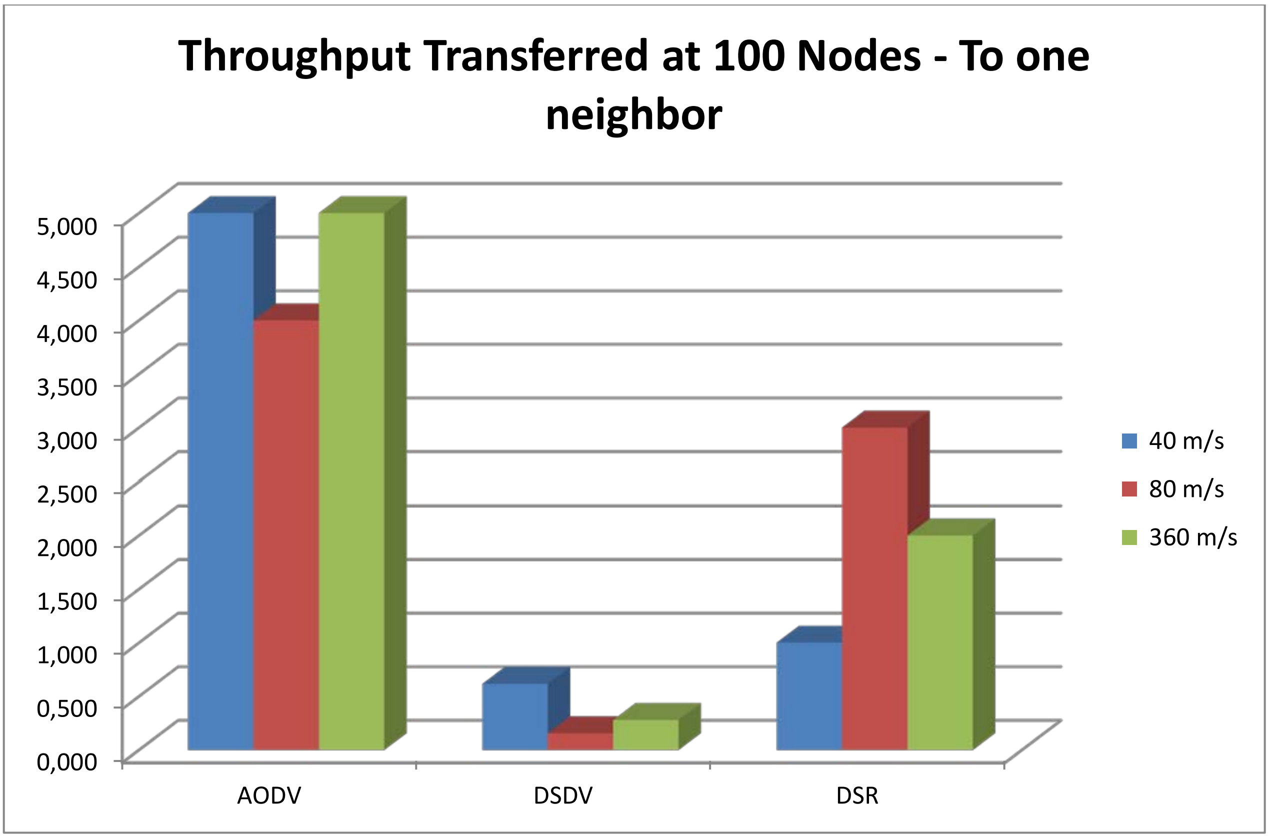


Figure 6.22 - Throughput Transferred at 100 Nodes – to one neighbor

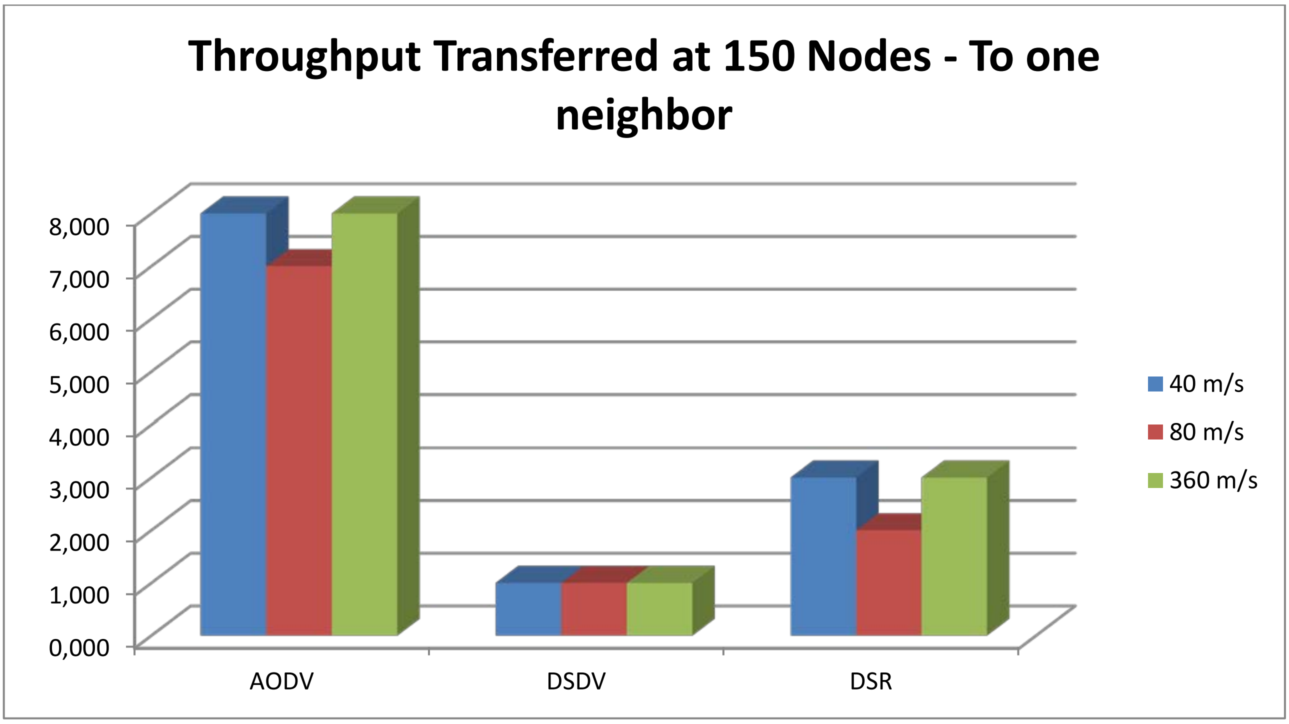


Figure 6.23 - Throughput Transferred at 150 Nodes – to one neighbor

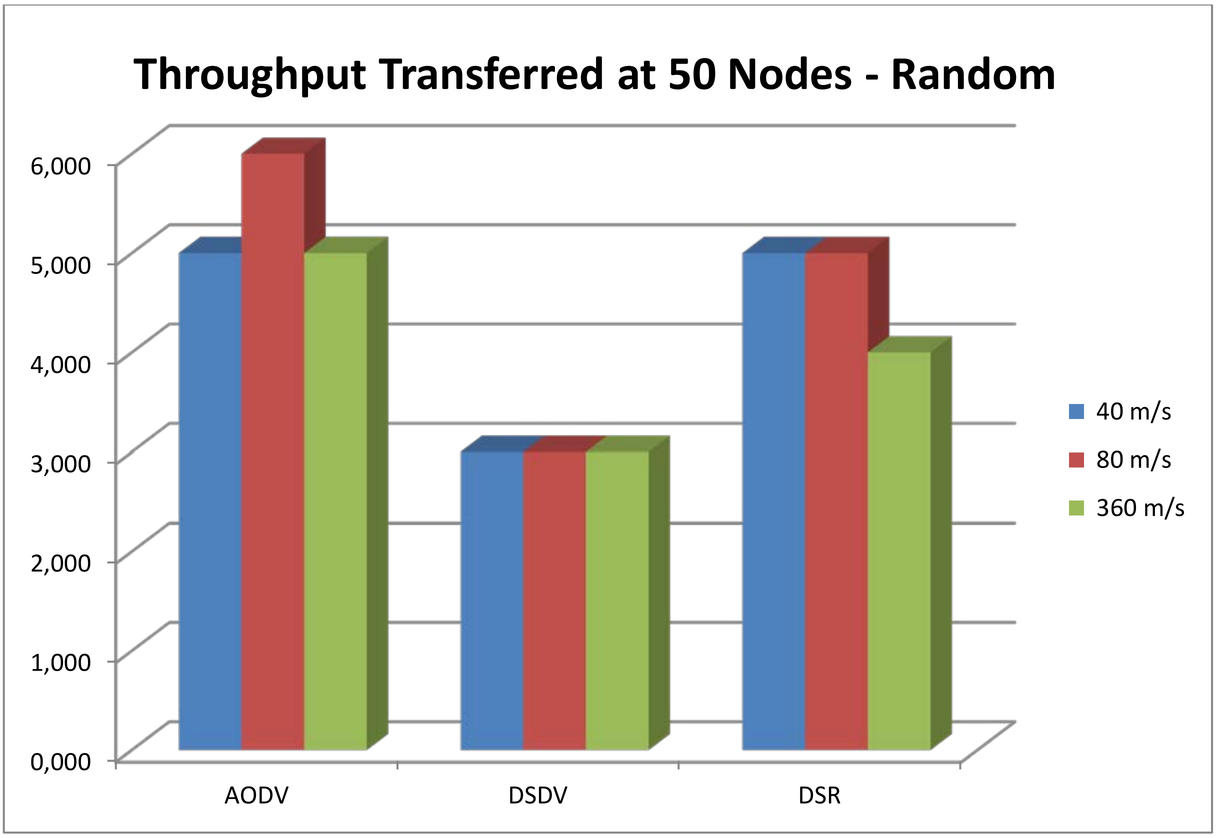


Figure 6.24 - Throughput Transferred at 50 Nodes – Random

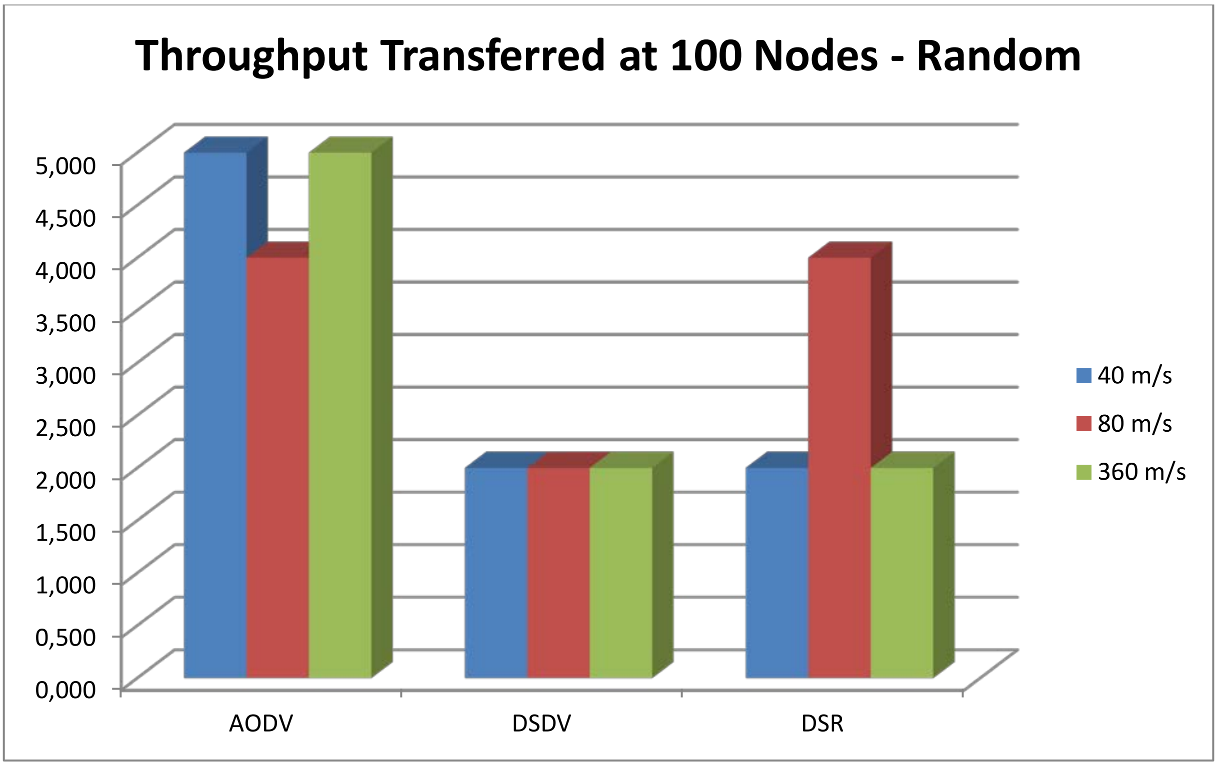


Figure 6.25 - Throughput Transferred at 100 Nodes – Random

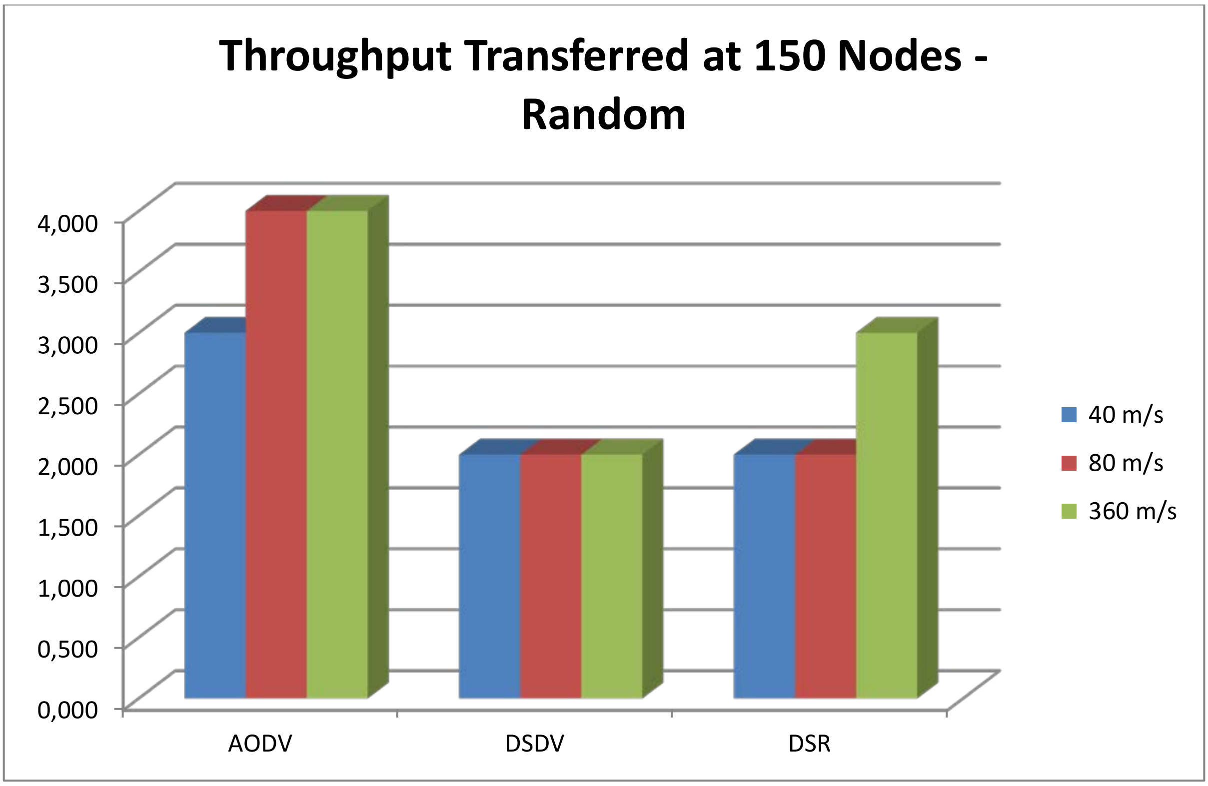


Figure 6.26 - Throughput Transferred at 150 Nodes - Random

As for the parameter of throughput transferred for random scenario, the biggest number of transferred throughput (8,000kbps) is observed at the speed of 80m/s for 50 nodes (Figure 6.24). For 50 and 100 nodes (Figures 6.24 and 6.25) at speed of 40 and 360 m/s, we observe exactly the same result.

Summarizing , for communication one node to its neighbor node the minimum packet loss percentage is observed at 100 nodes with AODV protocol at 40m/s while at the random communication, minimum percentage is observed at 100 nodes at 360m/s. As far as packet delay is concerned, the DSDV protocol is more preferable because of the smallest value at sec (0, 22) and this happens at 50 nodes at speed of 360m/s (To one neighbor scenario). In the random communication the DSDV remains the “best” protocol at 100 nodes where at every velocity values, we observe minimal packet delay. Regarding packet jitter the AODV protocol demonstrates the best performance (one-to-one neighbor node) at 50 nodes at 360m/s. But in the other communication models does not happen the same. The DSDV protocol is the more efficient at 50 nodes at 80m/s. Last but not least, for the transferred throughput we notice that the AODV “to one neighbor” scenario at 150 nodes at speeds of 40m/s and 360 m/s has the same measurement (8,000 kbps). At the random communication, we observe something different that forces us not to be able to select a protocol because both DSDV and DSR produce approximately the same measurements at 100 and 150 nodes at every speed value.

CHAPTER 7

CONCLUSION

In this thesis we are concerned with the problem of data dissemination in vehicular networks. Achieving this goal in a wireless environment usually involves lots of challenging issues mainly due to the characteristics of the medium and the lack of synchronization and organization among nodes.

This is even more challenging in vehicular networks, as the nodes can move at high speeds in a wide area surrounded by buildings and other architectural structures which can, in turn, affect the propagation of the radio signal.

As a consequence, we create a map as big as a city block so as to represent the real world, as close as possible. So, the size of the map is (654, 70 X 570, 80 m). We deal with V2V communication by using different routing protocols (i.e. AODV,DSDV,DSR),different communication modes (one node to one neighbor node, one node to a random node) and various speeds in order to determine the routing protocol with the best performance.

The results were collected through extensive simulations combining three simulation tools such as MOVE, SUMO and NS2. MOVE is based on an open source micro-traffic simulator SUMO and allows the user to quickly generate realistic mobility models for vehicular network simulations. Additionally, MOVE provides an interface to automatically generate simulation scripts for ns-2.

The collected results show that there is no efficient protocol for all parameters that we studied (packet loss, packet delay, jitter, throughput transferred) due to different simulation scenarios and characteristics of each protocol.

Regarding packet loss percentage and packet delay, the AODV protocol demonstrates the best performance at both types of communication and at sparse and dense environments (50 to 150 nodes). An exception for the above measures is packet delay where DSDV is the more preferable protocol at “to one neighbor” scenario in sparse environment (50 nodes) and at the same scenario in dense environment (150 nodes) and velocity values of 40m/s and 360m/s. Through the results that concern the packet jitter, at “to one neighbor” scenario for both environments (sparse and dense) the AODV seems to be the most efficient protocol. On the other hand, in random communication the DSDV protocol is the predominant. Last, when transferred throughput is concerned, the DSDV protocol demonstrates the best performance from the other two protocols. An exception is that in random scenario the DSR protocol for dense environment (100 and 150 nodes) appears the same results at every speed value.

7.1 Future work

In addition to the subjects studied in this thesis there are still many challenging open issues. However, we want to clarify that the list of open problems identified here is not exhaustive. We hope, at least, that this study contributes to feed the debate and open the window of new research directions. Regarding the disconnected network problem there is also a crucial research challenge for developing a reliable and efficient routing protocol that can support highly diverse network topologies.

A possible extension of this thesis is to deal with the optimization of the network that is used. So it could be to minimize the packet loss percentage for every scenario. Furthermore another possible work for the future is to investigate with EstiNet technologies, which is an expert company on developing products and solutions for network simulation, emulation, planning, and cloud computing. However to use this program, a university license must be acquired. [106]

REFERENCES

[1] J. T. Isaac, J. S. Camara, S. Zeadally and J. T. Marquez, “A secure vehicle-to-roadside communication payment protocol in vehicular ad hoc networks,” *Comput. Commun.*, vol. 31, no. 10, 2008.

[2] X. Yang, J. Lin, F. Zhao and N. H. Vaidya, “A vehicle-to-vehicle communication protocol for cooperative collision warning,” in *MOBIQUITOUS ’04,* 2004.

[3] A. Nandan, S. Das, G. Pau, M. Gerla, M.Y. Sanadidi, “Cooperative downloading in vehicular ad-hocwireless networks,” in *IEEE WONS 2005*, 2005.

[4] J. Byers, J. Considine, M. Mitzenmacher, S. Rost, “Informed content delivery across adaptive overlay networks,” in *ACM SIGCOMM 2002*, Pittsburgh-Pennsylvania - USA, 2002.

[3] Rahman, A.K.M.F., “The Burden of Road Traffic Injuries in South Asia:A Commentary”, JCPSP, Vol. 14 (12), 2004, pp. 707-708.

[4] http://www.crp-bangladesh.org/issues_roadsafety.htm

[5] Ricardo Joseph Estevez ,”Emergency Braking Notification System: Applying Technology of Wireless Sensor Networks to Automobiles”, PhD thesis the Graduate Faculty of the University of Colorado at Colorado Springs Master of Science Department of Computer Science 2006.

[6] Tamer El Batt, Siddhartha Goel, Gavin Holland, Harihan Krishnan, and Jayendra

Parikh. “Cooperative collision warning using dedicated short range wireless communication”. In *Proceedings of ACM VANET*, pages 1–9, Los Angeles, CA,

September 2006.

[7] Raymond Tatchikou, Subir Biswas, and Francois Dion. “Cooperative vehicle

collision avoidance using inter-vehicle packet forwarding”. In *Proceedings of*

the IEEE Global Telecommunications Conference (GLOBECOM), pages 2762–

2766, St. Louis, MO, December 2005.

[8] Ziyuan Wang, Lars Kulik, and Kotagiri Ramamohanarao. “Proactive traffic

merging strategies for sensor-enabled cars”. In *Proceedings of ACM VANET*, pages 39–48, Montreal, Quebec, Canada, September 2007.

[9] Mahmoud Abuelela, Stephan Olariu, and Michele C. Weigle. “NOTICE: An architecture for notification of traffic incidents”. In *Proceedings of IEEE VTCSpring*, pages 3001–3005, Singapore, May 2008.

[10] Jakob Eriksson, Lewis Girod, Bret Hull, Ryan Newton, Samuel Madden, and Hari Balakrishnan.“The pothole patrol: using a mobile sensor network for road surface monitoring”. In *Proceedings of the International Conference on Mobile systems, Applications, and Services (MobiSys)*, pages 29–39, Breckenridge, CO, June 2008.

[11]Clarus initiative. http://www.clarusinitiative.org, 2004.

[12] J¨org Ott and Dirk Kutscher. Drive-thru Internet: IEEE 802.11b for ”automobile” users. In *Proceedings of the Annual Joint Conference Conference of the IEEE Computer and Communications Societies (INFOCOM)*, pages 362–373, Hong Kong, March 2004.

[13] Joon-Sang Park, Uichin Lee, Soon Y. Oh, Mario Gerla, and Desmond S. Lun. “Emergency related video streaming in vanet using network coding”. In *Proceedings of ACM VANET*, pages 102–103, Los Angeles, CA, September 2006.

[14] Kevin C. Lee, Seung-Hoon Lee, Ryan Cheung, Uichin Lee, and Mario Gerla. “First experience with CarTorrent in a real vehicular ad hoc network testbed”. In *Proceedings of IEEE MOVE*, Anchorage, AK, May 2007.

[15] US Department of Transportation. Standard Specification for Telecommunications and Information Exchange Between Roadside and Vehicle Systems. ASTM E2213-03, August 2003.

[16] Sze-Yao Ni, Yu-Chee Tseng, Yuh-Shyan Chen, and Jang-Ping Sheu. “The broadcast storm problem in a mobile ad hoc network”. In *Proceedings of the ACM International Conference on Mobile Computing and Networking (MOBICOM)*, pages 151–162, Seattle, WA, August 1999.

[17] Fabio Picconi, Nishkam Ravi, Marco Gruteser, and Liviu Iftode. “Probabilistic

validation of aggregated data in vehicular ad-hoc networks”. In *Proceedings of ACM VANET*, pages 76–85, Los Angeles, CA, September 2006.

[18] Vladimir Bychkovsky, Bret Hull, Allen Miu, Hari Balakrishnan, and Samuel Madden. “A measurement study of vehicular internet access using in situ wifi networks”. In *Proceedings of the ACM International Conference on Mobile Computing and Networking (MOBICOM)*, pages 50–61, Los Angeles, CA, USA, 2006.

[19] David Hadaller, Srinivasan Keshav, Tim Brecht, and Shubham Agarwal. “Vehicular opportunistic communication under the microscope”. In *Proceedings of the International Conference on Mobile systems, Applications, and Services (MobiSys)*, pages 206–219, San Juan, Puerto Rico, 2007.

[20] Bret Hull, Vladimir Bychkovsky, Yang Zhang, Kevin Chen, Michel Goraczko, Allen Miu, Eugene Shih, Hari Balakrishnan, and Samuel Madden. “Cartel: a distributed mobile sensor computing system”. In *Proceedings of ACM SenSys*, pages 125–138, Boulder, CO, 2006.

[21] Dirk Reichardt, Maurizio Miglietta, Lino Moretti, Peter Morsink, andWolfgang Schulz. “Cartalk 2000: safe and comfortable driving based upon inter-vehicle communication”. In *Proceedings of the IEEE IV Symposium*, pages 545–550, June 2002.

[22] Jing Zhao and Guohong Cao. “Vadd: Vehicle-assisted data delivery in vehicular ad hoc networks”. *Proceedings of IEEE Transactions on Vehicular Technology*, 57(3):1910–1922, May 2008.

[23] G¨okhan Korkmaz, Eylem Ekici, and F¨usun ¨O zg¨uner. “An efficient fully ad-hoc multi-hop broadcast protocol for inter-vehicular communication systems”. In *Proceedings of IEEE ICC*, pages 423–428, June 2006.

[24] G¨okhan Korkmaz, Eylem Ekici, and F¨usun ¨O zg¨uner. “Urban multi-hop broadcast protocol for inter-vehicle communication systems”. In *Proceedings of ACM VANET*, pages 76–85, Philadelphia, PA, October 2004.

[25] Christian Lochert, Bj¨orn Scheuermann, Christian Wewetzer, Andreas Luebke, and Martin Mauve. “Data aggregation and roadside unit placement for a VANET traffic information system”. In *Proceedings of ACM VANET*, pages 58–65, San Francisco, CA, September 2008.

[26] John R. Douceur. “The Sybil attack”. In *First International Workshop on Peer-to-Peer Systems (IPTPS)*, March 2002.

[27] Bryan Parno, Adrian Perrig.”Challenges in securing vehicular networks”. This research was supported in part by CyLab at Carnegie Mellon under grant DAAD190210389 from the Army Research Office, and by an NDSEG Fellowship from the Department of Defense.

[28] Changdong Liu, M.J. Lee, and T.N. Saadawi. “A scalable multicast routing”, pages 983 {987 vol.2, nov 1997.

[29] Brad Karp and H. T. Kung. “GPSR: greedy perimeter stateless routing for wireless networks”.In MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking, pages 243{254, New York, NY, USA, 2000. ACM.

[30] C. Maihöfer, “A survey of geocast routing protocols,” *IEEE Communications Surveys & Tutorials*, vol. 6, no. 2, pp. 32–42, 2004.

[31] http://www.slideshare.net/gabitobalderas/a-survey-of-geographic-routing-protocols-for-vehicular-ad-hoc-networks-vanets

[32] http://www.slideshare.net/gabitobalderas/a-survey-of-geographic-routing-protocols-for-vehicular

[33] Yu-Chee Tseng, Sze-Yao Ni, and En-Yu Shih. “Adaptive approaches to relieving broadcast storms in a wireless multihop mobile ad hoc network”. *IEEE Transactions on Computers*, 52:545–557, 2003.

[34] Chunyu Hu, Yifei Hong, and Jennifer Hou. “On mitigating the broadcast storm problem with directional antennas”. In *Proceedings of IEEE ICC*, pages 104–110, Anchorage, AK, May 2003.

[35] NawaPorn Wisitpongphan, Ozan Tonguz, Jayendra Parikh, Priyantha Mudalige, Fan Bai, and Varsha Sadekar. “Broadcast storm mitigation techniques in vehicular ad hoc networks”. *IEEE Wireless Communications*, 14(6):84–94, December 2007.

[36] Linda Briesemeister and G¨unter Hommel. “Role-based multicast in highly mobile but sparsely connected ad hoc networks”. In *Proceedings of ACM MobiHoc*, pages 45–50, Boston, MA, August 2000.

[37] Abdelmalik Bachir and Abderrahim Benslimane. “A multicast protocol in ad Hoc networks: Inter-vehicle geocast”. In *Proceedings of IEEE VTC-Spring*, pages 2456–2460, Jeju Island, Korea, April 2003.

[38] Marc Torrent-Moreno, Daniel Jiang, and Hannes Hartenstein. “Broadcast reception

rates and effects of priority access in 802.11-based vehicular ad-hoc networks”. In *Proceedings of ACM VANET*, pages 10–18, Philadelphia, PA, October 2004.

[39] Ozan Tonguz, NawaPorn Wisitpongphan, Fan Bai, Priyantha Mudalige, and

Varsha Sadekar. “Broadcasting in VANET”. In *Proceedings of IEEE MOVE*, pages 7–12, Anchorage, AK, May 2007.

[40] P. Manzoni M. Sanchez. “A Java-based ad-hoc networks simulator”. SCS Western Multiconference, 1999.

[41] M.M. Zonoozi and P. Dassanayake. “User mobility modeling and characterization of mobility patterns”. Selected Areas in Communications, IEEE Journal on, 15(7):1239{1252, Sep 1997.

[42] Gianluca Grilli Andrea Tomatis Pasquale Cataldi, Mario Gerla. “A novel data dissemination method for vehicular networks with rateless codes”. Technical Repor TR090009, UCLA Computer Science Department, April 2009.

[43] Gianluca Grilli Andrea Tomatis Pasquale Cataldi, Mario Gerla. “Careless data dissemination in mobility environment with rateless codes”. Technical Report TR080023.

[44] Kschischang F. Ralf Koetter. “Coding for errors and erasures in random network coding”. In in Proc. IEEE Int. Symp. Information Theory, 2007.UCLA Computer Science Department, July 2008.

[45] J. Byers, J. Considine, M. Mitzenmacher, S. Rost. “Informed Content Delivery Across Adaptive Overlay Networks”. In ACM SIGCOMM 2002, Pittsburgh - Pennsylvania - USA, 2002.

[46] Dejan Kostic, Adolfo Rodriguez, Jeannie Albrecht, and Amin Vahdat. “Bullet: High Bandwidth Data Dissemination Using an Overlay Mesh”. In In SOSP, 2003.

[47] Petar Maymounkov and David Mazieres. “Rateless Codes and Big Downloads”. In IPTPS'03, 2003.

[48] C. Wu and B. Li. “rStream: resilient peer-to-peer streaming with rateless codes”. In Proceedings of the 13th annual ACM international conference on Multimedia, pages 307{310. ACM New York, NY, USA, 2005.

[49] Azzedine Boukerche, Horacio A.B.F. Oliveira, Eduardo F. Nakamura, and Antonio A.F. Loureiro. “Vehicular Ad Hoc Networks: A New Challenge for Localization-Based Systems”. Computer Communications, 31(12):2838 - 2849, 2008. Mobility Protocols for ITS/VANET.

[50] Masashi Saito, Mayuko Funai, Takaaki Umedu, Teruo Higashino. “Inter-Vehicle Ad-Hoc Communication Protocol For Acquiring Local Traffic Information”. Graduate School of Information Science and Technology, Osaka University- Japan.

[51] Valery Naumov, Rainer Baumann, and Thomas Gross. “An evaluation of inter-vehicle ad hoc networks based on realistic vehicular traces”. In MobiHoc '06: Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing, pages 108-119, Florence ,22-25 May, 2006.

[52] Malik Tubaishat, Yi Shang and Hongchi Shi.”Adaptive Traffic Light Control with Wireless Sensor Networks”. Indian Institute of Science, pages 187-191 , University of Missouri – Columbia 2007.

[53] Qing Xu, Tony Mak, Jeff Ko, and Raja Sengupta. “Vehicle-to-vehicle safety messaging in DSRC”. In VANET '04: Proceedings of the 1st ACM international workshop on Vehicular ad hoc net-works, pages 19-28, New York, NY, USA, 2004. ACM.

[54] José Santa, Antonio F. Gómez-Skarmeta. “Potential of Cellular Networks in Vehicular Communications”. University of Murcia, Spain 2006.

[55] Lars Hoehmann and Anton Kummert.” Mobility Support for Wireless Sensor Networks Simulations for Road Intersection Safety Applications”. *IEEE* Communications Magazine, pages 260-263, Germany, 2009.

[56] H.-M. Tsai, C. Saraydar, T. Talty, M. Ames, A. MacDonald, and 0. K. Tonguz, "ZigBee-based intra-car wireless sensor network," in Proc. *IEEE* International Conference on Communications (ICC 2007), June 2007.

[57] Hsin-Mu Tsai, Wantanee Viriyasitavatt, Ozan K. Tonguz, Cem Saraydar, Timothy Talty, and Andrew Macdonald. “Feasibility of In-car Wireless Sensor Networks: A Statistical Evaluation”. In Secon ’07 Proccedings *IEEE* Communication Society, pages 101-111, USA, 2007.

[58] Khalil M. Yousef, Jamal N. Al-Karaki and Ali M. Shatnawi. “Intelligent Traffic Light Flow Control System Using Wireless Sensors Networks”. Journal of Information Science and Engineering 26, pages 753-768, 2010.

[59] Subir Biswas, Raymond Tatchikou and Francois Dion.”Vehicle-to-Vehicle Wireless Communication Protocols for Enhancing Highway Traffic Safety”. *IEEE* Communications Magazine, pages 74-82, January 2006.

[60] Eylem Ekici, Yaoyao Gu, and Doruk Bozdag. “Mobility-Based Communication in Wireless Sensor Networks”. *IEEE* Communications Magazine, pages 56-62, July 2006.

[61] H. Wu, R. Fujimoto, and G. Riley, “Analytical models for information propagation in vehicle-to-vehicle networks,” in *VTC ’04-Fall: Proceedings of the 60th IEEE Vehicular Technology Conference*, vol. 6, 2004.

[62] T. Nadeem, P. Shankar, and L. Iftode, “A Comparative Study of Data Dissemination Models for VANETs,” in *MobiQuitous ’06: Proceedings of the Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services*, July 2006.

[63] B. Xu, A. Ouksel, and O. Wolfson, “Opportunistic resource exchange in inter-vehicle ad hoc networks,” in *Proc. IEEE Int. Conf. MDM*, 2004, pp. 4–12.

[64] A. Demers, D. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinehart, and D. Terry, “Epidemic algorithms for replicated database maintenance,” *Oper. Syst. Rev.*, vol. 22, no. 1, pp. 8–32, 1988.

[65] Z. Haas, J. Halpern, and L. Li, “Gossip-based ad hoc routing,” in *Proc. IEEE Infocom*, 2002, pp. 1707–1716.

[66] G. Karlsson, V. Lenders, and M. May, “Delay-tolerant broadcasting,” in *Proc. SIGCOMM Workshop CHANTS*, Pisa, Italy, 2006, pp. 197–204.

[67] *Wireless LAN Media Access Control (MAC) and Physical Layer (PHY Specifications*, IEEE Std. 802.11, 1999. [Online]. Available: http://grouper.ieee.org/groups/802/11

[68] G. Korkmaz, E. Ekici, F. Ozguner, and U. Ozguner, “Urban multi-hop broadcast protocol for inter-vehicle communication systems,” in *Proc. ACM VANET*, Oct. 2004, pp. 76–85.

[69] David B. Johnson and David A. Maltz. “Dynamic source routing in ad hoc wireless networks.In Mobile Computing”, pages 153{181. Kluwer Academic Publishers, 1996.

[70] Xiaoyan Hong, Mario Gerla, Guangyu Pei, and Ching-Chuan Chiang. “A group mobility model for ad hoc wireless networks”. In MSWiM '99: Proceedings of the 2nd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems, pages 53{60, New York, NY, USA, 1999. ACM.

[71] B. Liang and Z. Haas. “Predictive distance-based mobility management for PCS networks”. In *Proceedings of the Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, March 1999.

[72] X. Hong, M. Gerla, G. Pei, and C. Chiang. “A group mobility model for ad hoc wireless networks”. In *Proceedings of the ACM International Workshop on Modeling and Simulation of Wireless and Mobile Systems (MSWiM)*, August 1999.

[73] Stefan Kraus. “*Microscopic Modeling of Traffic Flow: Investigation of Collision Free Vehicle Dynamics*”. PhD thesis, Mathematisches Institut, Universit¨at zu K¨oln, 1998.

[74] Kai Nagel and Michael Schreckenberg. “A cellular automaton model for freeway traffic”. *J. Phys. I France*, 2:2221–2229, 1992.

[75] Kun-chan Lan and Chien-Ming Chou.“Realistic Mobility Models for Vehicular Ad hoc Network (VANET) Simulations”, In IEEE Communication Magazine August 2008.

[76] M. S. Kakkasageri and S. S. Manvi , “Intelligent Information Dissemination in Vehicular ad hoc Networks”, in International Journal of Ad hoc, Sensor & Ubiquitous Computing (IJASUC) Vol.2, No.1, March 2011.

[77] Y. Wu and G. Cao, “Stretch-optimal Scheduling for On demand Data Broadcasts.”, *In Proceeding of the 10th International Conference on Computer Communications and Networks (ICCCN 01’)*, Pages 500-504, 2001.

[78] Zhang Y, Zhao J, Cao G, “Roadcast: a Popularity Aware Content Sharing Scheme in Vanets” *In Proceedings of ICDCS’09*, Montreal.

[79] J. Ott and D. Kutscher, “Drive-thru Internet: IEEE 802.11b for Automobile Users”, *In IEEE Infocom*, 2004.

[80] D. Hadaller et al., “Vehicular Opportunistic Communication Under the Microscope”, *In Proceedings of ACM MobiSys*, 2007.

[81] Martin Treiber, Ansgar Hennecke, and Dirk Helbing. “Congested traffic states in empirical observations and microscopic simulations”. *Physical Review E*, 62(2):1805–1824, 2000.

[82] Arne Kesting, Martin Treiber, and Dirk Helbing. “MOBIL: General lanechanging

model for car-following models”. In *Proceedings of the Transportation Research Board Annual Meeting*, Washington, DC, January 2007.

[83] Marco Fiore, J´erˆome H¨arri, Fethi Filali, and Christian Bonnet. “Understanding vehicular mobility in network simulation”. In *Proceedings of the IEEE MoVeNet*, Pisa, Italy, October 2007.

[84] http://sumo.sourceforge.net.

[85] http://canu.informatik.uni-stuttgart.de.

[86] Xiang Zeng, Rajive Bagrodia, and Mario Gerla. “Glomosim: a library for parallel simulation of large-scale wireless networks”. *SIGSIM Simulation Digest*, 28(1):154–161, 1998.

[87] http://www.omnetpp.org.

[88] http://isi.edu/nsnam/ns.

[89] http://www.scalable-networks.com.

[90] http://www.opnet.com.

[91] Stuart Kurkowski, Tracy Camp, and Michael Colagrosso. “MANET simulation studies: The incredible”. *ACM Mobile Computing and Communications Review (MC2R)*, 9(4):50–61, 2005.

[92] JiST/SWANS. http://jist.ece.cornell.edu, 2004.

[93] Frank Kargl and Elmar Schoch. “Simulation of MANETs: A qualitative comparison between JiST/SWANS and ns-2”. In *Proceedings of the International Workshop on System Evaluation for Mobile Platforms (MobiEval)*, pages 41– 46, San Juan, PR, June 2007.

[94] Ming Yang Kumarawadu P. Ghenniwa H.H. Weiming Shen Yunjiao Xue, Ho Sung Lee. “Performance Evaluation of NS-2 Simulator for Wireless Sensor Networks”. pages 1372 {1375, April 2007.

[95] Krajzewicz D, Rossel C. Simulation of Urban Mobility (SUMO). German Aerospace Centre, 2007 Available at: http://sumo.sourceforge.net/index.shtml

[96] D. Krajzewicz, “Traffic Simulation with SUMO - Simulation of Urban Mobility,” in *Fundamentals of Traffic Simulation*, ser. International Series in Operations Research & Management Science, J. Barcelo, Ed. Springer New York, 2010, vol. 145, pp. 269–293.

[97] Karnadi, F., Mo, Z., & Lan, K. (2007). “Rapid generation of realistic mobility models for VANET”. In *Proceedings of the IEEE wireless communication and networking conference (WCNC’07)* (pp. 2506–2511), Hong Kong, March 2007.

[98] http://nile.wpi.edu/NS/

[99] http://www.isi.edu/nsnam/ns/

[100] http://www.isi.edu/nsnam/ns/edu/index.html

[101] http://www.isi.edu/nsnam/ns/doc/ns_doc.pdf

[102] http://www.isi.edu/nsnam/nam/

[103] Mobile Ad Hoc Networking Working Group – AODV, http://www.ietf.org/rfc/rfc3561.txt

[104] Mobile Ad Hoc Networking Working Group – DSR, http://www.ietf.org/rfc/rfc4728.txt.

[105] Zygmunt J. Haas, Jing Deng, Ben Liang, Panagiotis Papadimitratos, and S. Sajama “Wireless Ad Hoc Networks” Cornell University School of Electrical and Computer Engineering

[106] http://www.estinet.com/index.php

LIST OF ACRONYMS

AGF	Advanced Greedy Forwarding
AODV	Ad-Hoc On demand Distance Vector routing
AP	Access Point
AUs	Application Units
BPSK	Binary Phase Shift Keying
BRTA	Bangladesh Road Transport Authority
CA	Cellular Automaton
CAR-2-X	car to car + car to infrastructure communication
CBR	Constant Bit Rate
CCA	Cooperative Collision Avoidance
CDMA	Code Division Multiple Access
CMU	Carnegie Mellon University
CN	Cellular Networks
CONSER	Collaborative Simulation for Education and Research
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
DAPRA	Defense Advanced Research Project Agency
DCF	Distributed Coordinated Function
DSDV	Destination-Sequenced Distance-Vector Routing
DSR	Dynamic Source Routing
DSRC	Dedicated Short-Range Communications
ECUs	Electronic Control Units
GDP	Gross Domestic Product
GPRS	General Packet Radio Service

GPS	Global Positioning System
GPSR	Greedy Perimeter Stateless Protocol
IEEE	Institute of Electrical and Electronic Engineers
ITS	Intelligent Transportation System
I2V	Infrastructure to Vehicle
I2V2V	Infrastructure to Vehicle to Vehicle
I2I	Infrastructure to Infrastructure
InV	In-Vehicle Communications
IP	Internet Protocol
IVBSS	Integrated Based Safety Systems
IVG	Inter-Vehicle Geocast protocol
LAN	Local Area Network
LBL	Lawrence Berkeley National Laboratory
MAC	Media Access Control
MANET	Mobile Ad-Hoc Networks
MBS	Mobile Base Station
MDS	Mobile Data Collector
ME	Mobile Element
MM	Mobility Model
MN	Mobile Node
MOVE	MObility model generator for VEhicular networks
MRME	Multihop Route to Mobile Element
NSF	National Science Foundation
NS-2	Network Simulator 2
Nam	Network Animator
OBU	On Board data Unit

OD	Opportunistic Scheme
OLSR	Optimized Link State Routing Protocol
PBS	Partitioning Based Scheduling
PCF	Point Coordination Function
PGR	Preferred Broadcasting Strategy
P2P	Peer to Peer
QoS	Quality of Service
RF	Radio Frequency
RPGM	Reference Point Group Mobility Model
RSC	Road Safety Cell
RSU	Road Side Unit
RTS/CTS	Request to Clear/ Clear to Send
RWMM	Random Walk Mobility Model
RWP	Random Waypoint Model
SAMAN	Simulation Augmented by Measurement and Analysis for Networks
SDRP	Speed Dependent Random Protocol
SUMO	Simulation of Urban MObility
TBRPF	Topology broadcast based on reverse-path forwarding
TCP	Transfer Control Protocol
TSB	Topologically Scoped Broadcast
TSNs	Traffic Sensor Nodes
TTW	Time-to-wait
UCM	University of California Berkeley
USDOT	United States Department of Transportation
VANET	Vehicular Ad-Hoc Networks
VII	Vehicle Infrastructure Integration

VINT	Virtual InterNetwork Testbed
V2I	Vehicle to Infrastructure
V2V	Vehicle to Vehicle
WiFi	Wireless Fidelity
WLAN	Wireless Local Area Networks
WSN	Wireless Sensor Networks
ZOR	Zone of Relevance