

ADAPTIVE ROUTING ALGORITHM FOR VEHICULAR NETWORKS IN CITY
ENVIRONMENT

by

B. Loran Özşahakyan

B.S, Computer Engineering, Işık University, 2004

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in System and Control Engineering

Boğaziçi University

2011

ADAPTIVE ROUTING ALGORITHM FOR VEHICULAR NETWORKS IN CITY
ENVIRONMENT

APPROVED BY:

Assoc. Prof. Fatih Alagöz
(Thesis Supervisor)

Prof. Emin Anarım

Assoc. Prof. Tuna Tuğcu

DATE OF APPROVAL: 08.08.2011

ACKNOWLEDGEMENTS

I would like to thank my thesis advisor, Assoc. Prof. Fatih Alagöz, for his guidance, support and patience during my thesis. I would like to specially thank Gürkan Gür for his understanding and support. I also would like specially to thank my family for my life path and their endless love.

ABSTRACT

ADAPTIVE ROUTING ALGORITHM FOR VEHICULAR NETWORKS IN CITY ENVIRONMENT

The most known position based routing protocol for high mobile vehicular ad-hoc networks is the Greedy Perimeter Stateless Routing (GPSR) protocol. This protocol contains two routing modes, the greedy mode and the recovery mode, and creates a neighbor list to make a routing decision. The GPSR uses the greedy forwarding method whereas hops that follow are chosen based on nodes which are geographically closer to the destination node among the neighboring nodes. Nevertheless, forwarding fails if the current node is geographically the closest but unable to forward the packets to the destination. In this situation, the GPSR algorithm acts in recovery (perimeter) mode which is the repair strategy of this algorithm. Some enhancements such as the Greedy Perimeter Coordinator Routing (GPCR) and the Greedy Border Superiority Routing (GBSR) have been proposed. In this thesis, a routing protocol which works on real city maps and takes into account the actual movements of vehicles in city environment is recommended. We describe Adaptive Routing Algorithm For Vehicular Networks (ARAV) as a solution that improves the packet delivery ratio of GPSR and also improves the GPCR with road vehicle density information. ARAV differs from GPCR and GBSR, in that it uses routing algorithms adaptively in two routing modes. In ARAV greedy mode, we address the shortcomings of GPCR by considering the density of the paths. Packets are routed from one junction to another, the direction and position of hops are determined by taking the density of the paths into consideration. Our simulation results show that the proposed ARAV protocol outperforms the GPSR protocol in terms of packet delivery, throughput, delay in one successful transmission and greedy/perimeter ratio.

ÖZET

ŞEHİR ORTAMI İÇİN TASARSIZ ARAÇ AĞLARINDA UYARLANIR YOL SAPTAMA ALGORİTMASI

Tasarsız araç ağları arasında konuma dayalı en bilinen yönlendirme protokolü Greedy Perimeter Stateless Routing (GPSR) protokolüdür. GPSR iki farklı tarzda çalışır. Fırsatçı algoritma modu ve kurtarma algoritma modu. Yönlendirme kararı için GPSR kendi araç komşu düğümleri arasında bir liste oluşturur. Sonraki düğüm seçimi için, fırsatçı modunda, komşu düğümler arasında hedefe en yakın coğrafi noktada bulunan düğüm seçilir. Fakat bazı zamanlarda coğrafi olarak yakın olan düğüm seçildiği halde veri iletiminde başarısız olur. Bu durumda ileriye doğru gönderim yapılması olanaksızdır. GPSR algoritması kurtarma moduna geçer, bu mod GPSR'ın tamir stratejisini tanımlar. Diğer taraftan, tasarsız araç ağlarının topolojisi sebebiyle, GPSR birçok farklı olası yolların ve kavşakların olduğu şehir ortamında yeterince etkin değildir. Şimdiye kadar bir çok GPSR'ı geliştirme çalışmaları yapılmıştır, örnek olarak Greedy Perimeter Coordinator Routing (GPCR) ve Greedy Border Superiority Routing (GBSR) verilebilir. GPCR data paket yönlendirmesi için, yol topolojisinde bulunan kavşaklar üzerindeki düğümleri kullanır ve hep yönlenimi kavşaklar üzerinden yapar. GBSR ise yerel maksimum durumuyla karşılaştığı sırada düzlemsel çizge yaratmak yerine sınır üstünlüklü çizge yaratır. Bu tez içerisinde şehir topolojisini ve araçların gerçek hareketlerini dikkate alan bir yönlendirme protokolü önerilmektedir. Tasarsız araç ağları için uyarlanabilir yol saptama algoritması (ARAV) çözümü ile GPSR'a göre paket iletim oranı artmış, gecikme zamanı kısalmış, çıkan iş oranı artmış ve fırsatçı algoritmanın çalışma oranı GPSR'a göre artmıştır. ARAV algoritmasının fırsatçı modunda GPCR ile birlikte taşıt yoğunluk algoritması kullanılmıştır. ARAV, GPCR ve GBSR'dan uyarlanabilir yönlendirme algoritması ile farklılık gösterir. ARAV fırsatçı modunda GPCR algoritmasının eksiklerini taşıt yoğunluk algoritması ile geliştirmektedir.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	iii
ABSTRACT	iv
ÖZET	v
LIST OF FIGURES	viii
LIST OF TABLES	x
LIST OF SYMBOLS	xi
LIST OF ABBREVIATIONS	xii
1. INTRODUCTION	1
2. LITERATURE SURVEY	7
2.1. Vehicular Communications	7
2.1.1. Inter Vehicular Communications	8
2.1.2. Road Side Vehicular Communication	10
2.1.3. Hybrid Vehicular Communication	10
2.2. Standards for Wireless Access in Vehicular Ad-Hoc Networks	11
2.2.1. DSRC/WAVE	11
2.2.2. Cellular Networks	11
2.2.3. Mobile WIMAX 802.16e	12
2.2.4. WLAN	13
2.2.5. Infrared	13
2.3. Research Challenges in Vehicular Ad-Hoc Networks	14
2.4. Characteristics of Vehicular Ad-Hoc Networks	15
2.5. Routing Algorithms in Vehicular Ad-Hoc Networks	16
2.5.1. Topology-based Routing Protocols	17
2.5.2. Proactive (Table Driven)	17
2.5.3. Reactive (On Demand)	18
2.5.3.1. Ad Hoc On Demand Distance Vector Routing (AODV)	19
2.5.3.2. Dynamic Source Routing(DSR)	19
2.5.4. Position-based (Geographic) Routing Protocols	20
2.5.4.1. Greedy Perimeter Stateless Routing (GPSR)	20

2.5.4.2.	Greedy Perimeter Coordinator Routing (GPCR)	22
2.5.4.3.	Greedy Border Superiority Routing (GBSR)	23
2.5.5.	Performance Evaluation of Routing Protocols	24
3.	PROPOSED ROUTING ALGORITHM: ARAV	26
3.1.	Repair Strategy of the ARAV Algorithm	29
3.2.	Greedy Mode of the ARAV Algorithm	30
4.	PERFORMANCE EVALUATION	34
4.1.	Simulator Design	34
4.1.1.	JiST/SWANS	35
4.1.2.	Architecture of JiST/SWANS	37
4.1.2.1.	Architectural Differences in JiST	38
4.1.2.2.	Architectural Differences in SWANS	39
4.1.3.	Performance Analysis of JiST/SWANS Compared to ns-2	40
4.1.4.	Used Machines For Simulations	41
4.2.	Mobility Model Used for ARAV	42
4.2.1.	STRAW Mobility Model	43
4.2.1.1.	Intra-segment Mobility in STRAW	44
4.2.1.2.	Route Management and Execution in STRAW	45
4.2.2.	Implementation of STRAW in JiST/SWANS	46
4.3.	City Topology: The Census Bureau's TIGER System	47
4.4.	Experimental Results	49
5.	CONCLUSION and FUTURE WORK	57
	REFERENCES	59

LIST OF FIGURES

Figure 1.1.	GPSR two-mode routing in presence of local maximum.	4
Figure 2.1.	System overview of CAR-2-X communication.	7
Figure 2.2.	Taxonomy of vehicular communication systems.	9
Figure 2.3.	Single-hop, multi-hop IVC systems.	10
Figure 2.4.	Possible wireless vehicular networking scenarios.	14
Figure 2.5.	Taxonomy of various routing protocols in VANET.	18
Figure 2.6.	AODV route discovery.	19
Figure 2.7.	The weakness of recovery mode of GPSR in city scenarios.	22
Figure 3.1.	Proposed scheme.	27
Figure 3.2.	Flowchart of STALE flag algorithm.	28
Figure 3.3.	Repair strategy of the ARAV algorithm.	29
Figure 3.4.	Greedy Routing vs. ARAV Routing in the junction.	31
Figure 3.5.	The General Structure of ARAV algorithm.	33
Figure 4.1.	A taxonomy of VANET simulation software.	35

Figure 4.2.	The JiST system architecture.	37
Figure 4.3.	Processing time of CGGC routing.	41
Figure 4.4.	Illustration of vehicular mobility components and their interactions in STRAW.	44
Figure 4.5.	Map 1: Washington DC, USA street map.	49
Figure 4.6.	Average packet delivery ratio for different number of nodes and transmitters.	52
Figure 4.7.	Average number of hops for different number of nodes and trans- mitters.	53
Figure 4.8.	Average delay in one transmission for different number of nodes and transmitters.	55
Figure 4.9.	Average throughput in one transmission.	55
Figure 4.10.	Greedy perimeter ratio for different algorithms.	56

LIST OF TABLES

Table 2.1.	Wireless data link characteristics.	13
Table 3.1.	System parameters.	32
Table 4.1.	Simulation parameters.	50

LIST OF SYMBOLS

D_i	Distance from current node to the target node
D_j	Distance from next hop to the target node
D_k	D_j / D_i
V_i	Velocity vector
W_p	Calculated weighted score for a node on a path
α	Weight factor for position of node
β	Weight factor for direction of travel

LIST OF ABBREVIATIONS

AU	Application Unit
CBR	Constant Bit Rate
AODV	Ad Hoc On Demand Distance Vector Routing
C2C	Car 2 Car Communication Consortium
DSRC	Dedicated Short Range Communication
DSDV	Destination-Sequenced Distance-Vector
DSR	Dynamic Source Routing
GG	Gabriel Graph
GPRS	General Packet Radio Service
GPS	Global Positioning System
GBSR	Greedy Border Superiority Routing
GPCR	Greedy Perimeter Coordinator Routing
GPSR	Greedy Perimeter Stateless Routing
GIS	Geographic Information Software
IEEE	Institute of Electrical and Electronics Engineers
ITS	Intelligent Transportation System
IVC	Inter-Vehicle Communication systems
IVE	In-Vehicle Equipment
JIST	Java in Simulation Time
MANET	Mobile Ad Hoc Network
MOVE	Mobility For Vehicles
MIVC	Multi-hop IVC Systems
OBU	ON Board Unit
QoS	Quality of Service
RMI	Remote Method Invocation
RNG	Relative Neighborhood Graph
RVC	Roadside-to-Vehicle Communication

SRVC	Sparse RVC
RWIS	Road Weather Information System
RWP	Random Waypoint
RREQ	Route Request
RSU	Road Side Unit
SIVC	Single-Hop IVC Systems
SMAC	Smart Medium Access Controller
STRAW	StreetRandom Waypoint
SWANS	Scalable Wireless Ad hoc Network Simulator
TIGER	Topologically Integrated Geographic Encoding and Referencing
UMTS	Universal Mobile Telecommunication System
URVC	Ubiquitous RVC
V2V	Vehicle-to-Vehicle Communication
VANET	Vehicular Ad Hoc Network
WAVE	Wireless Access in the Vehicular Environment
WLAN	Wireless Local Area Network

1. INTRODUCTION

Development of Intelligent Transportation Systems (ITS) begins with mobile computing and wireless communication. Traffic congestion and a synergy of new information technology for simulation, real-time control, and communication networks cause the interest in ITS. Traffic congestion has been increasing worldwide because of the increased motorization, urbanization, population of growth, and changes in population density. What is going to be improved by ITS is the traffic safety and driving experience of passengers [1].

Although the number of accidents and casualties remain dauntingly high, personal safety systems like belt pretensioners or airbags have reached ubiquitous deployment. There were still more than 1.2 million accidents on European roads, costing more than 40000 lives according to the European Road Safety Charter 1, in 2007. The goal of The European Commission is to reduce the fatality number to 20000. Nevertheless, this goal will surely be missed because of today's technology, especially with the increase of traffic [1, 2].

Whereas those only mitigate the effects of an accident, active safety systems like ABS or ESP actively influence the driving behavior with the goal to keep it controllable in critical situations. This may help to prevent accidents or at least reduce their severeness. But even if those systems became widespread, they would not prevent any accidents. Active sensors like radar or lidar are one of the approaches. However, the concept of inter-vehicle communication is addressed in this work [2].

Digital data communication between vehicles is enabled by "Inter-Vehicular Communication" (IVC). IVC systems can improve the safety, efficiency, and comfort of everyday road travel. The ability to bypass the need for expensive infrastructure is their main advantage; the comparatively complex networking protocols and the need for significant penetration are their major drawbacks before their applications can become effective. They require a number of major classes of applications and several

types of services. In order to improve the safety and increase the efficiency on the transportation infrastructure [3] these networks collect and disseminate time-critical information [1].

Vehicular ad hoc networks (VANET) are wireless communication networks proposed to be used as parts of ITS. VANETs facilitate communication among vehicles, and between vehicles and roadside equipment [4]. Figure 2.2 depicts a classification of vehicular communication (VC) systems: Inter-Vehicle Communication systems (IVC), Roadside-to-vehicle communication (RVC) systems and hybrid vehicular communication (HVC) systems.

In this thesis we focus on pure IVC systems (i.e., without roadside equipment). IVC systems are completely infrastructure-free; only onboard units (OBUs) sometimes also called in-vehicle equipment (IVE) are needed. Depending on whether the information is retransmitted at intermediate hops or not, the difference between single-hop (SIVC) and multihop IVCs (MIVC) can be distinguished. SIVC systems are useful for applications requiring short-range communications. MIVC systems are more complex than SIVCs but can also support applications that require long-range communications [5]. In this thesis we focus on MIVC systems for adaptive routing algorithms.

Wireless Access for Vehicular Environment (WAVE) is devoted to vehicle-to-vehicle and vehicle-to-roadside communications. The main objective has clearly been to improve the overall safety of vehicular traffic. Promising traffic management solutions and on-board entertainment applications are also expected by the different bodies and projects [5, 6]. Cars and roadside units form a highly dynamic network called a Vehicular Ad Hoc Network (VANET), a special kind of Mobile Ad-Hoc Networks (MANETs) when equipped with WAVE communication devices [7].

Mobile Ad Hoc Network (MANET) consists of multi-hop wireless mobile nodes which communicate with each other without centralized control or any infrastructure. The wireless links broadcasting under this network are highly capable of having problems or errors and lack of serviceability due to mobility of nodes, interference cutoffs

and absence of infrastructure. Vehicular Ad Hoc Network (VANET) is a specific part of MANETs, dealing with wireless communications among traveling vehicles and roadside units [4, 8].

Countless numbers of routing protocols have been developed in MANETs but many do not fit into VANETs. Because VANETs particularly represent a challenging class of MANETs. VANET's are distributed, self-organized network protocols formed by moving vehicles. Thus they are characterized by very high node mobility and with limited degrees of freedom in mobility patterns. Topology-based routing and position-based routing are two categories of routing protocols. For performing packet forwarding, topology-based routing uses the information about the links which exist in the network. On the other hand, to perform packet forwarding position-based routing uses the neighboring location information [7].

Position-based routing has been found to be suitable for VANET environments. Usually, only the position information of the vehicles is required in order to perform data packet routing. This information can be achieved via a Global Positioning System (GPS), or by using other position determining techniques. One of the primary advantages of using position-based routing is that it does not require route maintenance which is very suitable for the highly mobile VANET environment [9].

The most popular position based routing protocol for the vehicular environment is the Greedy Perimeter Stateless Routing (GPSR) [10] protocol. The protocol contains two routing modes, the greedy mode and the recovery mode, and creates a neighbor list to make a routing decision. In GPSR, packets are forwarded using a greedy mode in normal situations. For this task, each node has to be aware of (i) its own position, (ii) its direct neighbors' position and (iii) the position of the final destination [10]. The GPSR uses the greedy forwarding method in which the hops that follow are chosen based on nodes which are geographically closer to the destination node among the neighbor nodes. Nevertheless, forwarding fails if the current node is geographically the closest but unable to forward the packets to the destination as shown in Figure 1.1. If a neighbor which is closer to the destination node does not exist, the intermediate

for VANET application because of the distributed and infrastructureless nature of mobile ad-hoc wireless communication and the relatively high mobility of vehicles. Because of the transmission range and the distance between nodes connectivity problem in a wireless network may be studied analytically. The delay is affected by the signal propagations, collisions, or congestions that might be encountered over the air layer. The distribution of the nodes over the area differs with the topology, that affects the node density in several parts of the terrain.

In this thesis, we describe Adaptive Routing Algorithm For Vehicular Networks (ARAV), a solution that improves the packet delivery ratio of GPSR and also improves the GPCR with road vehicle density information. ARAV differs from GPCR and GBSR in that it uses routing algorithms adaptively in two routing modes. In ARAV greedy mode, we address the shortcomings of GPCR by considering the density of the paths. Packets are routed from one junction to another and the direction and position of hops are determined by taking the density of the path into consideration. The General Structure of ARAV algorithm is shown in Figure 3.5.

In ARAV perimeter mode, we are using the same assumptions with GBSR in order to generate a border superior sub-graph. ARAV protocol eliminates the overlaid links from the neighbor list until both nodes are on the same road. Our simulation results show that the proposed ARAV protocol outperforms the GPSR protocol in terms of packet delivery ratio, throughput, greedy/perimeter ratio and reduces the hop count used in the perimeter compared to GPSR. As a result, it allows routing schemes to return to the greedy mode faster.

The well known network simulator JIST/SWANS [11–13] was used to perform the simulations. The ARAV protocol was compared with the existing routing protocol GPSR. STRAW (STreetRAndom Waypoint) [14] mobility model was used to simulate the movement pattern of moving vehicles on streets or roads. STRAW provides accurate simulation results by using a vehicular mobility model based on the operation of real vehicular traffic for real US city maps. STRAW is also written for the highly efficient JIST/SWANS discrete-event simulator.

For the simulations, we used a real city topology map from Topologically Integrated Geographic Encoding and Referencing system (TIGER) which is a part of Washington DC [15] street maps as shown in Figure 4.5. The movement of the nodes was generated with a dedicated vehicular traffic simulator and represents a real world movement pattern for this given scenario [16]. The nodes communicate using the IEEE 802.11 wireless LAN standard with a maximum range of 750m. For each simulation, a different number of sender-receiver pairs were randomly selected. The nodes move at an average velocity of up to 15 m/s.

The rest of the thesis is organized as follows. In Chapter 2, vehicular communication, routing algorithms and their challenges and characteristics in the literature are summarized. In Chapter 3, the proposed algorithm ARAV is introduced. The Network simulator JiST/SWANS, mobility models used, city topology and all experimental results are introduced in Chapter 4 and in Chapter 5 states the conclusions and future work.

2. LITERATURE SURVEY

2.1. Vehicular Communications

The vehicular communication system consists of three domains as shown in Figure 2.1 the in-vehicle domain, the ad hoc domain, and the infrastructure domain.

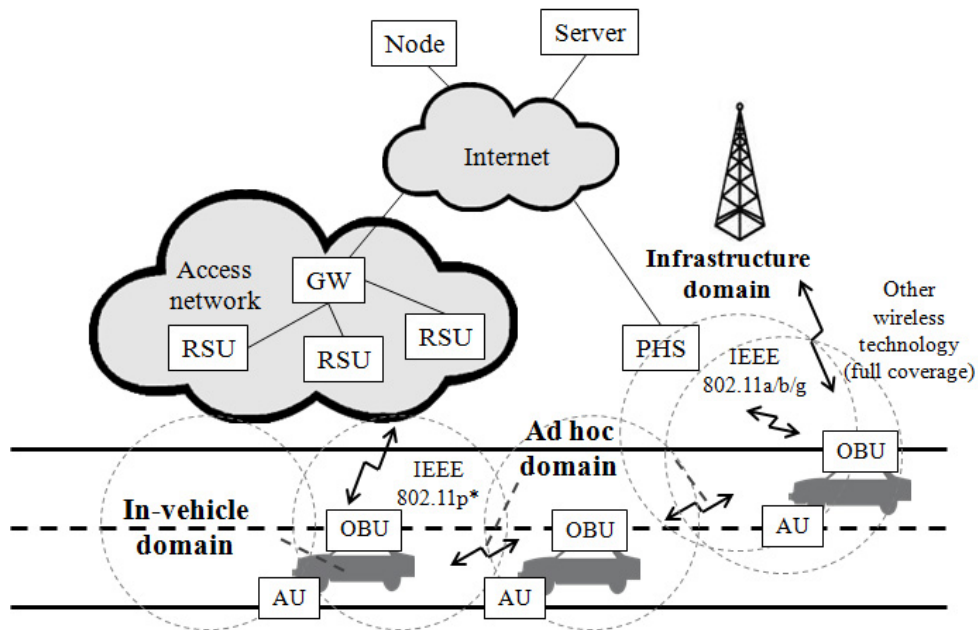


Figure 2.1. System overview of CAR-2-X communication.

The in-vehicle domain is based on a network which is logically composed of an on-board unit (OBU) and (potentially multiple) application units (AUs). The OBU has the responsibility of vehicular communication. It obtains communication services for AUs and forwards data in the name of other OBUs in the ad hoc domain. An OBU is composed of a single network device for short-range wireless communications which based on IEEE 802.11p radio technology, or it can be equipped with more network devices for non safety communications, based on other radio technologies like IEEE 802.11a/b/g/n. An AU is a dedicated device which uses the OBU's communication capabilities achieves a single or a set of applications. An AU is the integrated part of a vehicle and connected to an OBU. Furthermore, it may also be used as a portable device like laptop, PDA, or game pad that may dynamically attach to an OBU. Apart from

the wired connection, AU and an OBU can be connected via wireless with Bluetooth, WUSB, or UWB. AU and OB may reside in a single physical unit with a logical distinction [3, 4].

The vehicles equipped with OBUs and the stationary units along the road, termed road-side units (RSUs) are the components of the vehicular ad hoc network (VANET). OBUs create a mobile ad hoc network (MANET) that allows communications among nodes in a fully distributed manner without need for centralized coordination. OBUs may directly communicate if wireless connectivity exists among them. In case of no direct connectivity, dedicated routing protocols allow multihop communications, where data are forwarded from one OBU to another, as far as they reach the destination. An RSU may be attached to an infrastructure network. That in turn may be connected to the Internet. As a result, RSUs can allow OBUs to access the infrastructure. In this way AUs registered with an OBU can communicate with any host on the Internet, when at least one infrastructure-connected RSU is available [3, 4].

Figure 2.2 depicts a classification of vehicular communication (VC) systems: Inter-Vehicle Communication systems (IVC), Roadside-to-vehicle communication (RVC) systems and hybrid vehicular communication (HVC) systems.

2.1.1. Inter Vehicular Communications

Safety, efficiency, and comfort of everyday road travel can be improved by Inter-vehicle communication (IVC). Their ability to bypass the need for expensive infrastructure is the principal advantage; Complex networking protocols and the need for significant penetration before their applications can become effective is their main drawback . They require lots of main classes of applications and numerous types of services [1].

IVCs rely on direct communication amongst vehicles to satisfy the communication needs of a large number of applications like collision avoidance, passing assistance, and platooning. For internet access and other applications IVC systems may be sup-

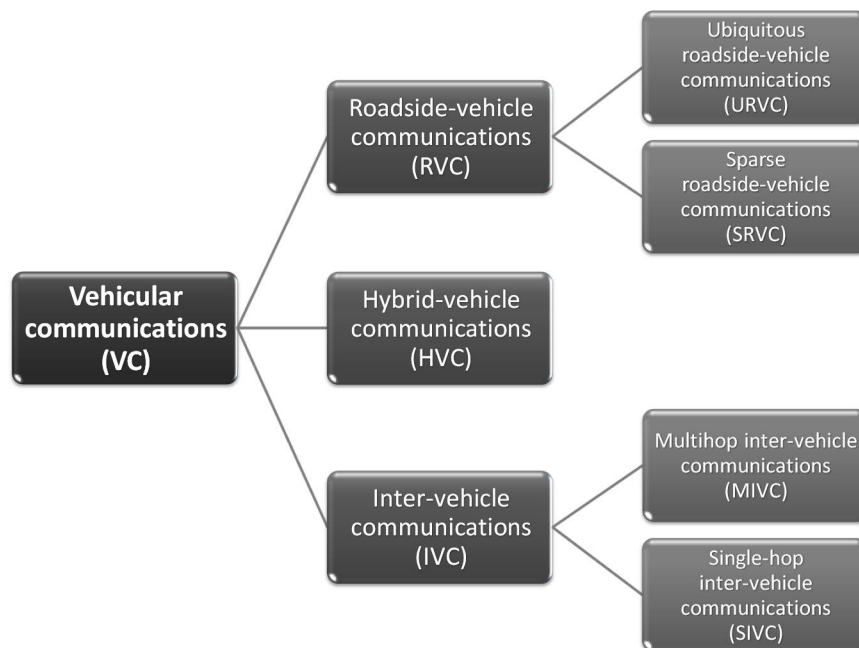


Figure 2.2. Taxonomy of vehicular communication systems.

plemented or replaced by roadside infrastructure [5].

IVC systems are totally infrastructure-free; OBUs sometimes also called in-vehicle equipment (IVE) are needed. The difference between single-hop (SIVC) and multihop IVCs (MIVC) is distinguished depending on whether the information is retransmitted at intermediate hops or not. SIVC systems are useful for applications requiring short-range communications. MIVC systems are more complex than SIVCs but they support applications which require long-range communications [3].

The main difference between SIVC and MIVC systems is shown in Figure 2.3. In a SIVC system, a vehicle, called vehicle A, can send a message only to other vehicles which are in its own transmission range, that makes vehicle B not to receive any messages. On the other hand, another vehicle with a MIVC system in it, called vehicle C, can relay the message so that vehicles that are not in the transmission range of vehicle A and vehicle B, can also receive the message. Therefore, a MIVC system requires a network layer with an ability of multihop routing.

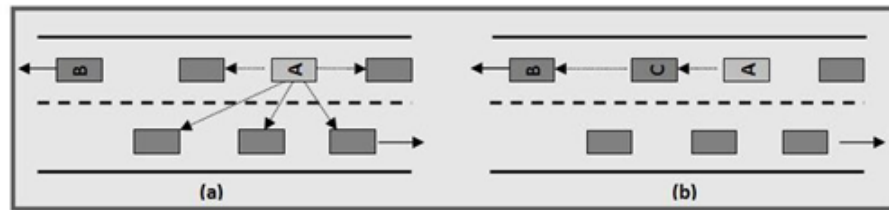


Figure 2.3. Single-hop, multi-hop IVC systems.

2.1.2. Road Side Vehicular Communication

RVC systems assume all communications which take place amongst roadside infrastructure and OBUs. The sparse RVC (SRVC) and the ubiquitous RVC (URVC) systems are the two different types of this infrastructure.

SRVC systems may provide communication services at hot spots. The examples of applications requiring an SRVC system are a busy intersection scheduling its traffic light, a gas station advertising its existence (and prices), and parking availability at an airport. An SRVC system can be deployed by stages, therefore not requiring substantial investments before any available benefits.

URVC systems require considerable investments to provide full coverage of substantial roadways. A URVC system provides all roads with high-speed communication that these applications unavailable for other systems.

2.1.3. Hybrid Vehicular Communication

HVC systems propose to extend the range of RVC systems. In HVC systems vehicles communicate with roadside infrastructure only when they are not in direct wireless range with other vehicles, like mobile routers. An HVC system enables the same applications Hybrid Vehicular Communication with a larger transmission range than an RVC system. It requires less roadside infrastructure than RVC. Nevertheless, network connectivity can not be guaranteed in scenarios with low vehicle density Systems.

2.2. Standards for Wireless Access in Vehicular Ad-Hoc Networks

Standards simplify product development, help reduce costs, and enable users to compare competing products. Requirements of interconnectivity and interoperability are guaranteed and the emergence of new products is verified with the use of VANET standards to enable rapid implementation of new technologies. There are several standards that relate to wireless access in vehicular environments. These standards range from protocols which apply to transponder equipment and communication protocols through to security specification, routing, addressing services, and interoperability protocols [17].

2.2.1. DSRC/WAVE

As mentioned in [4, 6] dedicated short-range communication (DSRC) is a short-to medium-range communication technology operating in the 5.9 GHz range. The Standards Committee E17.51 endorses a variation of the IEEE 802.11a MAC for the DSRC link. DSRC supports vehicle Emerging Vehicular Applications 6-3 speeds up to 120 mi/h, nominal transmission range of 300 m (up to 1000 m), and default data rate of 6 Mb/sec (up to 27 Mb/sec). This will enable operations related to the improvement of traffic flow, highway safety, and other intelligent transport system (ITS) applications in a variety of application environments called DSRC/WAVE [4].

DSRC has two different modes of operations: one is ad hoc mode characterized by distributed multihop networking (vehicle-vehicle), and the other is infrastructure mode characterized by a centralized mobile single-hop network (vehicle-gateway). Gateways can be connected to one another or to the Internet, and they may be equipped with computing and storage devices depending on the deployment scenarios.

2.2.2. Cellular Networks

Cellular systems have been improving rapidly to support the ever increasing demands of mobile networking. 2G systems like IS-95 and GSM support data communi-

cations at the maximum rate of 9.6 kb/sec. For providing higher rate data communications, GSM based systems use GPRS (<171 kb/sec) and EDGE (<384 kb/sec), and IS-95-based CDMA systems use 1xRTT (<141 kb/sec). Today 3G systems support much higher data rates.

UMTS/HSDPA provides maximum rates of 144 kb/sec, 384 kb/sec, and 2 Mb/sec under high-mobility, low-mobility, and stationary environments, respectively. CDMA2000 1xEvDO (Rev. A) provides 3 Mb/sec and 1.8 Mb/sec for down and up links, respectively. The average data rate perceived by users is much lower in practice: <128 kb/sec for GSM/EDGE and 512 kb/sec for 3G technologies. In the U.S. Verizon and Sprint provide 1xEvDO, and AT&T and T-Mobile provide GSM/EDGE [4].

2.2.3. Mobile WIMAX 802.16e

The goal of 802.16e or WiMAX (Worldwide Interoperability for Microwave Access) is to enable the delivery of last mile wireless broadband access (<40 Mb/sec) as an alternative to cable and xDSL, therefore providing wireless data over long distances. This will fill the gap amongst 3G and WLAN standards, providing the data rate (tens of Mb/sec), mobility (<60 km/h), and coverage (<10 km) required to deliver the Internet access to mobile clients. In its part, WiBro, developed in Korea based on 802.16e draft version 3, provides 1 km range communications at the maximum rate per user of 6 and 1 Mb/sec for down and up links [4].

It supports numerous service levels including guaranteed Quality of Service (QoS) for delay sensitive applications, and an intermediate QoS level for delay tolerant application which requires a minimum guaranteed data rate. The performance of WiBro networks is measured in a subway whose maximum speed is 90 km/h, and it's indicated that the average uplink and downlink speeds were 2 and 5.3 Mb/sec, respectively, and that the average packet delay (half RTT) was less than 100 msec, and almost all packets experienced delay below 200 msec except in the case when handoff s happened (>400 msec).

Table 2.1. Wireless data link characteristics.

Indicative wireless data link characteristics	Technology			
	802.11 p wave	Wi-Fi	Celular	Infrared
Bit rate	3-27 Mb/s	6-54 Mb/s	< 2 Mb/s	< 1 Mb/s < 2Mb/s
Communication range*	< 1000 m	< 100 m	< 15 km	< 100 m (CALM IR)
Transmission power for mobile (maximum)	760 mW (US) 2 W EIRP (EU)	100 mW	2000 mW (GSM) 3 80 mW (UMTS)	12800 W/Sr pulse peak
Channel bandwidth	10 MHz 20 MHz	1-40 MHz	25 MHz (GSM) 60 MHz (UMTS)	N/A (optical carrier)
Allocated spectrum	75 MHz (US) 30 MHz (EU)	50 MHz @ 2.5 GHz 300 MHz @ 5 GHz	(Operator dependent)	N/A (optical carrier)
Suitability for mobility	High	Low	High	Medium
Frequency band(s)	5.86-5.92 GHz	2.4 GHz, 5.2 GHz	800 MHz, 900 MHz 1800 MHz 1900 MHz	835-1035 nm
Standards	IEEE, ISO, ETSI	IEEE	ETSI, 3GPP	ISO
*The communication range depends on parameters such as data rate, power, bandwidth, and topography; values given in this table are estimates and may vary.				

2.2.4. WLAN

As mentioned in [4] WiFi or WLAN may support broadband wireless services. 802.11 a/g provides 54 Mb/sec and has a nominal transmission range of 38 m (indoor) and 140 m (outdoor). Despite its short radio range, its ubiquitous deployment makes WLAN an attractive method for supporting broadband wireless services. It has been used for a long time as a means of Internet access in vehicles, known as Wardriving. WLAN is unsuitable for high mobility.

2.2.5. Infrared

Infrared (IR) radiation is electromagnetic radiation with a wavelength between 0.7 and 300 micrometers, which equates to a frequency range from approximately 1 to 430 THz. [4] These different networks simplify different connectivity modes that are listed in Table 2.1.

If vehicles are only equipped with DSRC, we have an infrastructure-free mode (V2V only), infrastructure mode (V2I), and mixed mode (V2V and V2I) as shown in Figure 2.4a. If vehicles are only equipped with other broadband wireless access (i.e., cellular and WiMAX), we can have a scenario where vehicles can talk to each other via the Internet as shown in Figure 2.4b. When vehicles have both DSRC and other

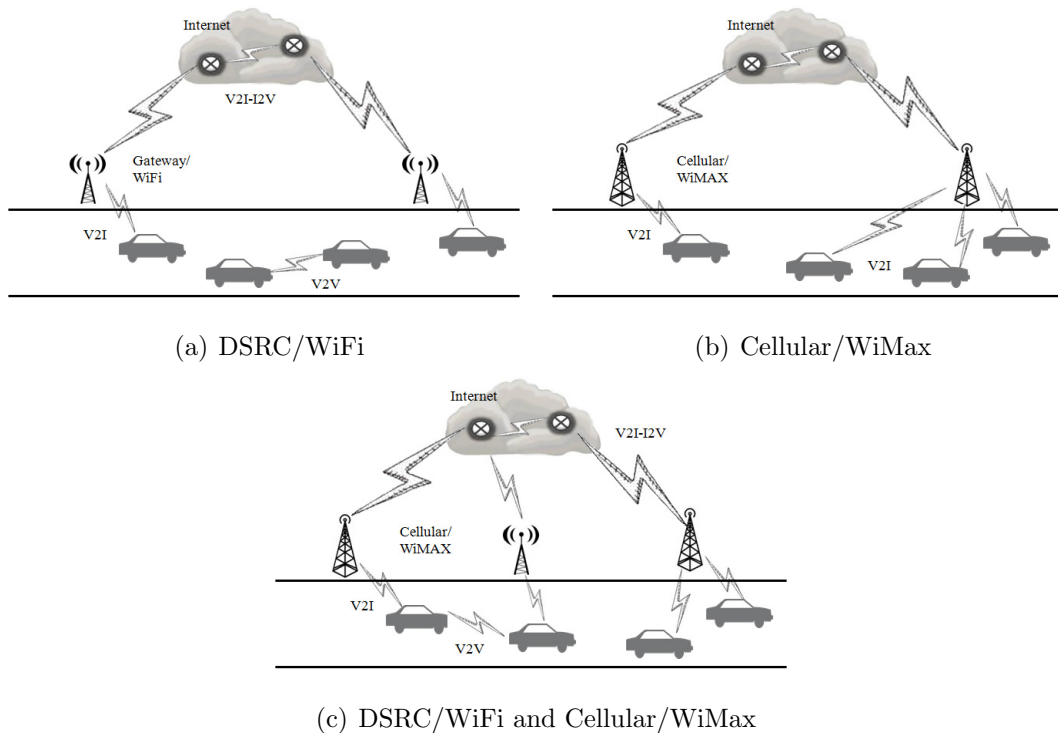


Figure 2.4. Possible wireless vehicular networking scenarios.

broadband wireless access methods, we can have a mixed access scenarios as shown in Figure 2.4c.

2.3. Research Challenges in Vehicular Ad-Hoc Networks

In this part, we discuss various VANET-related research challenges which still need further investigation and innovative solutions for enabling VANET infrastructures, communications, security, applications and services.

Routing has an important role in VANET applications. The high-speed mobility of vehicles and their rapidly changing topology results in conventional MANET routing protocols, are inadequate for dealing with unique vehicular environment. The intermediate nodes can not always be found amongst source and destination and end-to-end connectivity can not always be established. This motivates researchers to find scalable routing algorithms which are robust enough for the frequent path distributions caused by vehicle mobility as listed in [9, 18, 19], and may deliver better packet delivery ratio and improved throughput.

Since the vehicle speeds are numerous and driver behavior is unpredictable, we may not assume that an end-to-end connection always be established amongst a source and a destination. In a highway scenario, a routing route may not always be created amongst "isolated islands" of vehicles, thereby new forwarding concepts or caching schemes should be developed [20].

Efficient packet forwarding schemes are required because broadcasting is the principal means to propagate information in VANET applications. What cause contention which limits dramatically network throughput is the large number of duplicated packets resulting from the utilization of conventional flooding methods. A more effective way would be limiting the number of nodes which participate in the relaying process either by forming clusters or by using location-aware broadcasting in which packets are propagated only into a zone of relevance [20].

Fake messages or interference attempts by third-parties should be prevented. Many of the VANET applications are safety related and anonymity should be kept for protecting users' privacy.

Authentication versus privacy is the most confusing challenge in the VANET security area. Authentication of all messages is necessary to be sure that messages originate from actual vehicles. Imagine what can happen if a normal vehicle may transmit a warning beacon message like an ambulance just to clear its travelling lane. Furthermore, by investigating the pre-accident transmitted messages, higher authorities (e.g. police officers) should be able to determine causes of accidents. Although a third party may use this information to track vehicles of important persons remotely. Consequently, VANET technology requires a completely new bundle of security protocols [21].

2.4. Characteristics of Vehicular Ad-Hoc Networks

In designing protocols for the next generation vehicular network, nodes have significantly different characteristics and demands from those in traditional wireless

ad hoc networks deployed in infrastructureless environments. These differences have a significant impact on application infrastructures [4].

- Vehicles have higher power than a typical mobile computer. On-board batteries can provide power also it can be drawn from gasoline or alternative fuel engine.
- Vehicle mobility is restricted to one-dimensional road geometry. Moreover, in urban centers, buildings and similar obstacles limit the propagation of the communication signal and split the network topology [8].
- Factors such as road configuration, traffic laws, safety limits, and physical limits affect the mobility of vehicles. Drivers' behavior and interaction with each other also contribute to the vehicle mobility pattern [8].
- Vehicle mobility creates a highly dynamic topology. In heavy traffic congestion, vehicles may be within a couple of meters proximity of one another, whereas on a sparsely populated road the distance may be hundreds of meters. Traffic condition may change rapidly between congested and sparse due to traffic jams, accidents, and road constraints.
- VANETs are potentially large-scale networks.
- Vehicles can provide more resources such as large batteries, antennas, and processing power. Therefore, conserving such resources in VANETs is not a major concern. Thus, energy consumption is not an issue; instead transmission power has to be adjusted to maintain connectivity and to prevent interference.
- As they require little latency instead of high data rates, most of the intended VANET applications are safety related . For example, in case of a hard brake, a message carrying this situation has to be propagated in a 500 m radius area in 0.5 second. In this kind of applications, instead of average delay, the maximum delay is crucial.

2.5. Routing Algorithms in Vehicular Ad-Hoc Networks

Many routing protocols have been developed in MANETs. Most of them do not apply well to VANETs. VANETs are distributed, self-organizing communication

networks formed by moving vehicles. They are characterized by very high node mobility and limited degrees of freedom in mobility patterns [7].

A routing approach for a VANET should, therefore, show a high degree of adaptability with respect to the dynamics of the network, cope with scalability, and employ the broadcast nature of wireless networks. The MANET-style packet-forwarding protocols may still be used for relatively delay-tolerant applications such as Internet services. Since the cars of the future are likely to be equipped with an on-board GPS receiver and have access to digital maps, routing in VANET will be achieved using a position-based approach.

As shown in Figure 2.5, routing protocols designed for VANETs are broadly classified in two groups: topology-based routing and position-based routing protocols. In topology-based routing, the source node decides the trajectory that the message should take through the network to reach the destination. It uses the information about links that exist in the network to perform packet forwarding. Since link information changes in a regular basis, source routing suffers from routing route breaks. Position-based routing uses neighboring location information to perform packet forwarding. It forwards the packet to only one neighbor at a time, which is successively closer to the packet's destination [7].

2.5.1. Topology-based Routing Protocols

These routing protocols use links' information that exists in the network to perform packet forwarding. They can further be divided into proactive (table-driven) and reactive (on-demand) routing.

2.5.2. Proactive (Table Driven)

A distinct feature of the proactive routing is that the routing information such as the next forwarding hop is calculated in the background regardless of communication requests. Control packets are incessantly broadcast and flooded among nodes to keep

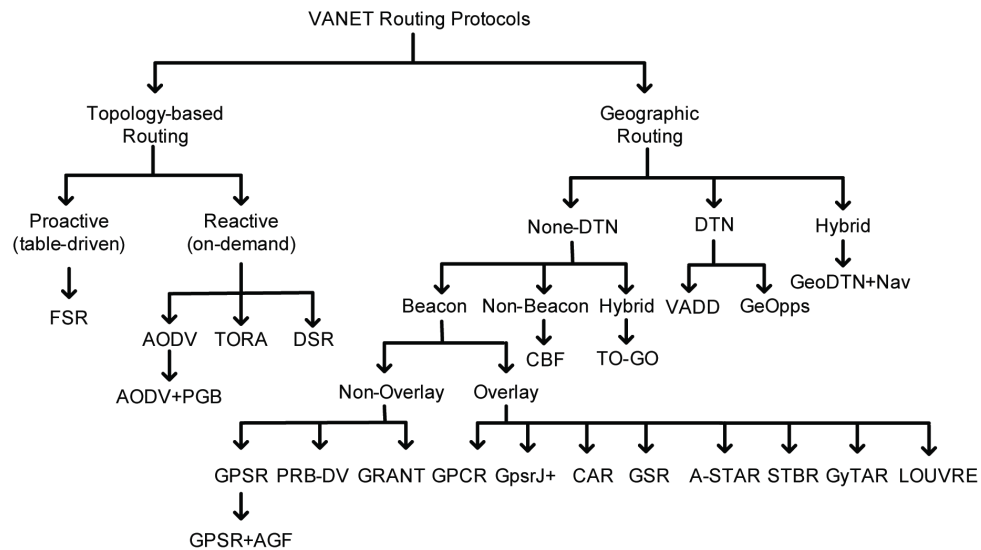


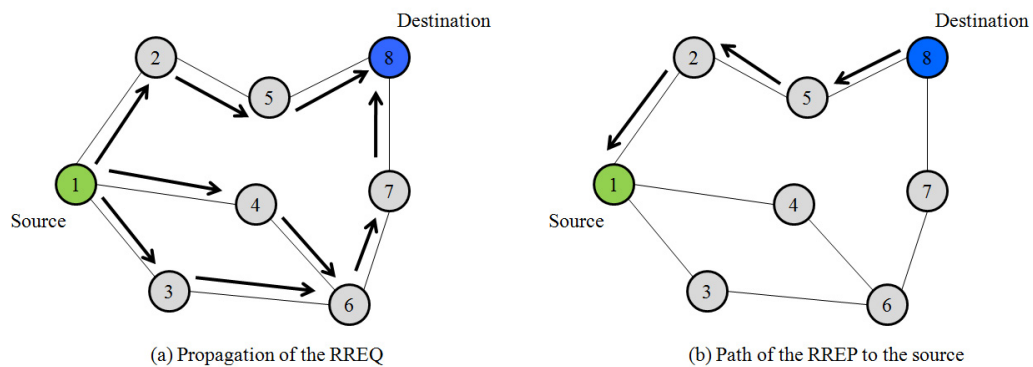
Figure 2.5. Taxonomy of various routing protocols in VANET.

the paths or the link states between any pair of nodes although some of paths are never used. A table is then constructed within a node and each entry in the table shows the next hop node toward a certain destination. The major advantage of the proactive routing protocols is that there is no need for route discovery. Route to the destination is maintained in the background and is always available upon lookup. In spite of its good property of providing low latency for real-time applications, the maintaining of unused paths in highly mobile VANETs occupies a critical part of the available bandwidth [7].

2.5.3. Reactive (On Demand)

Different from Proactive routing, Reactive routing creates a route only when it needs a node to communicate with another node. Routes are maintained when they are currently in use. In this way, burden on the network is reduced. Reactive routings have typical route discovery phase where query packets are flooded into the network in search of a path. The phase completes when a route is found.

2.5.3.1. Ad Hoc On Demand Distance Vector Routing (AODV). In Ad Hoc On Demand Distance Vector (AODV) (Perkins, 1999) routing, upon receipt of a broadcast query (RREQ), nodes record the address of the node sending the query in their routing table (Figure 2.6a). This procedure of recording is called backward learning. After arrival at the destination, a reply packet (RREP) is then sent through the complete path which is obtained from backward learning to the source (Figure 2.6b). At each stop of the path, the node would record its previous hop. In this way it determines the forward path from the source. Sending of reply and the flooding of query and determines a full duplex path. After the path has been established, it is maintained as long as the source uses it. A link failure will be reported recursively to the source and will in turn trigger another query-response procedure to find a new route [7].



(a) Propagation of the RREQ

(b) Path of the RREP to the source

Figure 2.6. AODV route discovery.

2.5.3.2. Dynamic Source Routing(DSR). DSR protocol is an on-demand routing protocol that is based on the concept of source routing. The protocol consists of two major phases: route discovery and route maintenance. In DSR, the query packet copies in its header the IDs of the intermediate nodes that it has traversed. The destination then get the entire path from the query packet (source routing), and uses it to respond to the source. As a result, the source can determine a path to the destination [4, 7].

There are two major differences between AODV and DSR. First one is that, AODV data packets carry the destination addresses, whereas in DSR, data packets carry the full routing information. This shows DSR has potentially more routing overheads than AODV. Moreover, as the network diameter increases, the amount of

overhead in the data packet will continue to increase. Second difference is that in AODV, route reply packets carry the destination address and the sequence number. In DSR, route reply packets carry the address of each node throughout the route [7].

2.5.4. Position-based (Geographic) Routing Protocols

In position-based routing, the forwarding decision is calculated based on the position of the destination and the position of the node's one-hop neighbors. The position of the destination is stored in the header of the packet which is sent by source. Position of the node's one-hop neighbors is obtained by the beacons which are sent periodically with random jitter. It is used to prevent collision in the network. Then, nodes that are within a node's radio range will become neighbors of the node. For position based routing protocols, each node is aware of its own position, its neighbors and also the position of the destination node [7, 22].

Position-based routing protocols do not exchange link state information and do not preserve established routes. In these protocols, route is determined based on the geographic location of neighboring nodes as the packet is forwarded. There is no need of link state exchange nor route setup. Position-based routing protocols are more compatible and powerful for highly dynamic environments like VANETs [7].

Figure 2.5 classifies Geographic routing into three different categories. Non-Delay Tolerant Network (non-DTN), Delay Tolerant Network (DTN), and hybrid. The non-DTN types of geographic routing protocols do not consider discontinuous connectivity and are only practical in densely populated VANETs. However DTN types of geographic routing protocols do consider disconnectivity. They are designed from the perspective that networks are disconnected by default.

2.5.4.1. Greedy Perimeter Stateless Routing (GPSR). The most popular position based routing protocol for the vehicular environment is the Greedy Perimeter Stateless Routing (GPSR) [10] protocol. In GPSR, packets are forwarded using a greedy mode in

normal situation. For this task, each node has to be aware of (i) its own position, (ii) its direct neighbors' position and (iii) the position of the final destination [10]. The GPSR uses the greedy forwarding method in which the hops that follow are chosen based on nodes which are geographically closer to the destination node among the neighbor nodes. Nevertheless, forwarding fails if the current node is geographically the closest but unable to forward the packets to the destination as shown in Figure 1.1. If a neighbor which is closer to destination node does not exist, the intermediate node itself is chosen and this situation is named as local maximum problem. In such a case no further progress is possible. In this situation, the GPSR algorithm acts in perimeter mode which is the repair strategy of this algorithm. The general aim of a repair strategy is to forward the packet to a node which is closer to the destination than the node where the packet encountered the local optimum.

In the perimeter mode of GPSR, the network topology is seen as a planar graph and packets are forwarded by traversing the graph using the right-hand rule [10]. Two planar graph algorithms, the RNG [23] and the GG [24], have been proposed in the GPSR to build the planar network graph [10]. The node selects a vertex from the graph using right hand rule [16]. This rule states that when a node first enters into the recovery mode, the VANET node should select the closest node with a minimum angle and toward which it has to move counter-clockwise to reach the destination node.

The packet will be consistently detoured until it reaches a node whose distance to the destination node is closer than the former recovery node [10]. When the packet reaches that node, the packet's mode is turns back to the greedy mode. These algorithms perform well when nodes are uniformly or randomly distributed. However, this is not a solution for VANET environment. Planar graph algorithms do not take into account whether a path is populated with vehicles in order to forward packets. For example, when cars follow each other closely along the road and a car faces the local maximum, the links on planar graph are only the two edges next to the car using either the RNG or the GG scheme. All crossed links in the original network topology are eliminated by those schemes. Figure 2.5.4.1 shows an example of planar graphs in a city scenario. Though the radio range is enough to forward a packet to a far away

distance, each node transmits the packet to the nearest node in the repair strategy. Therefore the short-hop distance results in higher hop counts, and higher hop counts intern results in higher forwarding delays.

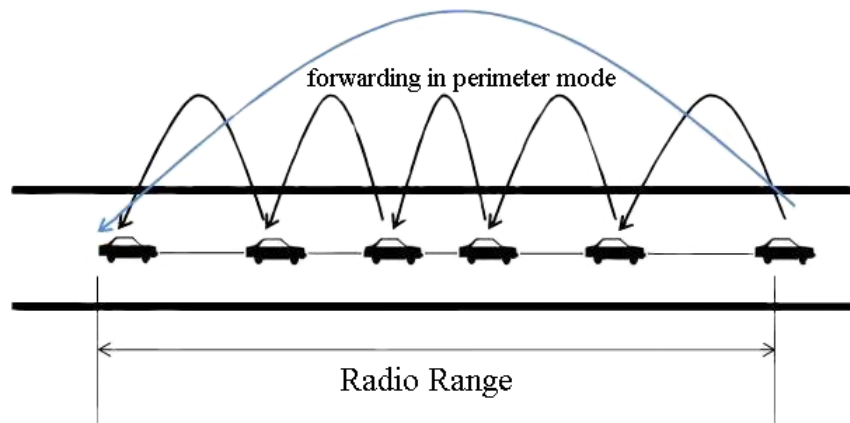


Figure 2.7. The weakness of recovery mode of GPSR in city scenarios.

As to the greedy mode in the GPSR, the shortest path algorithm is the simplest way to route data packets. By using Dijkstra's algorithm [25], the shortest path from a source to its destination can be precalculated. Nevertheless, this method is only suitable in an ideal condition as the shortest path is the best. This case does not apply to the city environment in general. Vehicles move with various velocities and directions according to the traffic rules and also there are various obstacles, i.e high rise buildings and road topologies consisting of many streets and junctions. However, the shortest path algorithm does not take into consideration wheather a certain path is populated with vehicles or junctions while forwarding data packets [9].

2.5.4.2. Greedy Perimeter Coordinator Routing (GPCR). As discussed in previous sections nodes are highly mobile in VANETs, node planarization can become a clumsy, inaccurate, and continuous process. In Greedy Perimeter Coordinator Routing (GPCR), they have observed that an urban street map naturally forms a planar graph such that node planarization can be completely eliminated. In this new presentation of the planar graph they are using the topology of the roads and the nodes. Source nodes

forward as far as they can reach along roads in both greedy and perimeter mode. It stops at junctions where a decision about which next road segment to turn into can be determined.

GPCR improves routing performance as packets travel shorter hops in the perimeter mode eliminating the inaccuracy of node planarization. The improved routing decision keeps packets from being routed to the wrong direction which often leads to higher delay. GPCR does not use a map to determine whether a node is located at a junction. It uses two different heuristics to determine whether a node is in a junction. The first heuristic uses beacon messages to determine a node is located at a junction or not. The second heuristic is correlation coefficient calculation that shows a node related to its neighbors. A correlation coefficient close to 0 shows there is no linear relationship between the positions of the neighbors. This indicates the node is located at a junction. Their evaluation, based on a dedicated vehicular traffic simulator, has shown that packet delivery rate does increase over GPSR [7].

2.5.4.3. Greedy Border Superiority Routing (GBSR). The GBSR protocol is a position-based routing protocol which enhances greedy forwarding of GPSR. Greedy mode and recovery mode are the two modes like GPSR algorithm. In the greedy mode, packets are forwarded to nodes which are closer to the destination node. When a packet achieves a local maximum in GPSR, the node switches to the perimeter mode for that packet. In the perimeter mode, it is not effective to forward backward in short distance through planarized links in the vehicular network scenarios. The best way to recover from the local maximum is to make the packet achieve a conversion point as fast as possible [19, 26]. In the recovery mode of GBSR, a packet can be hopped up to a radio boundary node. For creating a border superior sub-graph, the GBSR protocol eliminates the overlaid links from the neighbor list if only both of nodes are on the same road.

On the other hand, because of high mobility in vehicular networks, the coherency of location information on the neighbor list goes down. With shorter period of HELLO

beacons the coherence of neighbor list can come strong. However, this heavily increases the network loads. Regardless of how the period is reduced, neighbor lists may still include some stale information in the VANET. In the greedy forwarding mode, the possibility of staleness can be relatively high to the farthest node, and the probability of routing failure will increase [19]. In order to overcome this shortcoming, the GBSR protocol manages its neighbor list adaptively. When a node receives a HELLO beacon from a neighbor node, the node compares the current location of the neighbor node with the previous location of that. The node can obtain some useful information during the procedure. Every node takes advantage of this information and manages its neighbor list more effectively [19].

Like GPSR protocol, in the GBSR protocol, the next hop is selected with right-hand rule. A node extracts a border superior sub-graph from its neighbor list and node should select the closest node with a minimum angle and toward which it has to move counter-clockwise to reach the destination node. The selected node becomes the next hop in this mode. With the sub-graph, packets in the local maximum are greedily transmitted through the route. Once arrival to the conversion point, the packets are forwarded with greedy mode.

2.5.5. Performance Evaluation of Routing Protocols

Different metrics is going to be used for different type of application requirements. Below, the most common evaluation metrics for routing protocols are listed [27].

- Packet delivery ratio with/without retransmission (PDR): The fraction of data packets that are successfully transmitted and sent from source to destination.
- Routing overhead (RO): The number of routing packets required by the protocol to construct and maintain the routes.
- Delay: Average end-to-end transmission delay of successfully transferred packets.
- Number of hops: A measure of expected route length per source-destination.
- Average delay of the first packet (Latency): Average time difference between the first data packet originated and received by its destination.

- Connectivity: Number of reachable/unreachable node pairs.
- Average queue length: Average MAC buffer length.

Depending on the network parameters, obtained results differ in the simulation environment. Parameters which effect above metrics are shown below.

- Topology: Real map, grid, block size, roads and partial roads (if starting or ending points of a street are not in the simulation area).
- Number of nodes/vehicles: The number of nodes can be fixed or varied.
- Density of nodes: The number of nodes per area (nodes/m²) or per road.
- Heterogeneity: Percentage of vehicles in the network that are not equipped with a radio, some vehicles transmitting, some not.
- Wireless transmission range: Transmission range determines the number of reachable nodes.
- Duration of simulation
- Routing protocol
- Mobility models
- Packet Size
- Path loss models
- Placement parameters: Distance between cars, lane change, waiting time at an intersection, average node speed, speed distribution, and step size (direction change, speed update).
- Data traffic rates

3. PROPOSED ROUTING ALGORITHM: ARAV

Various geographic routing protocols for VC system have been proposed in the literature. We analyze that the current configurations of some routing algorithms only focus on specific requirements to resolve the current situation. The biggest common drawback of these protocols is that they cater to only one or two road characteristic problems in their side and may not be applied to various applications. For example, GBSR [19] worked for the problem about maximum number of hops in perimeter mode of GPSR [10] and created stale free Neighbor List Management schemes. Conversely, writers of GPCR [18] tried to solve the packet delivery ratio problem in GPSR [10] with junction advertisement algorithm. We have realized that one proposed routing solution does not offer a solution for many different kinds of applications. We suggest an adaptive routing algorithm ARAV that it takes into consideration street topology and vehicle mobilities. So that, it provides the most appropriate solution under certain constraints. Figure 3.5 summarized this approach.

We propose Adaptive Routing Algorithm For Vehicular Networks in City Environment (ARAV), a solution that improves the packet delivery ratio of GPSR and also improves the GPCR with road vehicle density information. ARAV differs from GPCR and GBSR, in that it uses routing algorithms adaptively in two routing modes. In ARAV greedy mode, we address the shortcomings of GPCR by considering the density of the paths. If the source and destination are different direction and roads, packets are routed from one junction to another and the direction of hops are determined by taking into account the density of the road. In ARAV perimeter mode, we use the same assumptions with GBSR in order to generate a border superior sub-graph. ARAV protocol eliminates the overlaid links from the neighbor list until both nodes are on the same road. Our simulation results show that the proposed ARAV outperforms the GPSR in terms of packet delivery ratio, throughput and delay. Moreover, it reduces the hop count used in the perimeter compared to GPSR.

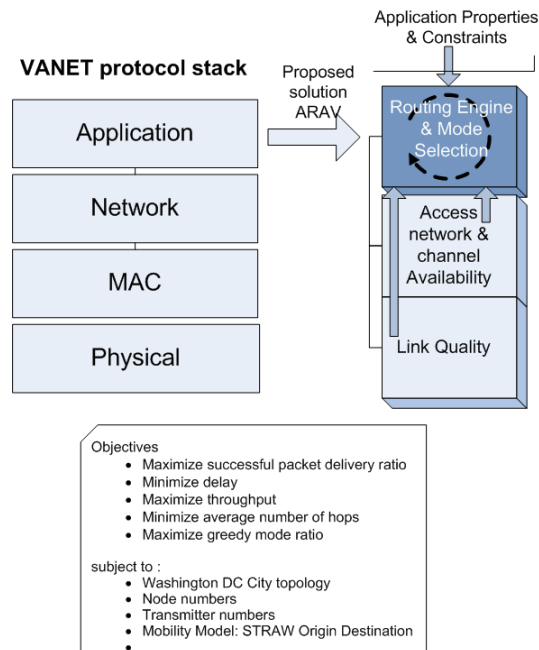


Figure 3.1. Proposed scheme.

As discussed in Section 2.5.4, the GPSR [10] uses the greedy forwarding method hops that are chosen based on nodes which are geographically closer to the destination node. To select the next relay node among the neighbor nodes, each node makes use of location information on the neighbor list. For that matter, we assume that each node is aware of its own position, its neighbors and also the position of the destination node [22]. A source node acquires the location of the destination node from a global location service which provides lookup service that maps the addresses of the nodes to locations. Each node periodically broadcasts a HELLO beacon which contains its ID and position information to interchange the location information between nodes. As can be seen, the role of HELLO beacon is crucial for most of the position based routing protocols. Therefore, these protocols use the location of neighboring nodes for route decisions.

Because of high mobility in vehicular networks, the integrity of location information on the neighbor list deteriorates. By assigning a shorter period of HELLO beacons the coherence of neighbor list can be firmer. However, this approach increases the network load significantly. In VANET environment, regardless of how the period is reduced, neighbor lists may still contain some stale nodes [19]. In the greedy for-

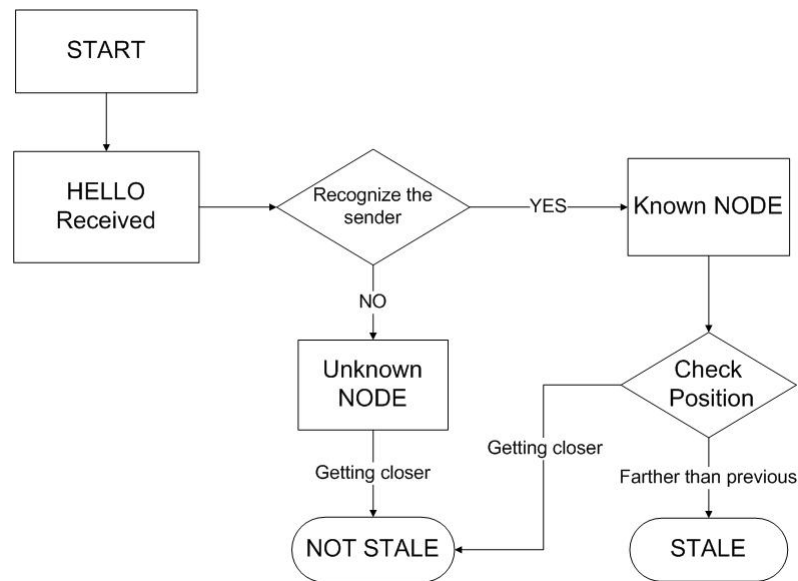


Figure 3.2. Flowchart of STALE flag algorithm.

warding mode, the possibility of staleness can be relatively high to the farthest node, and then the probability of routing failure will increase [19].

In order to handle this failure, the ARAV algorithm manages its neighbor list adaptively with the same assumptions in the GBSR algorithm. When a node receives a HELLO beacon from a neighbor node, it uses a comparison algorithm for the current location of the neighbor node with the previous location. Every node can achieve useful information by using this algorithm and manages its neighbor list more effectively. When a node receives a HELLO beacon from an unknown neighbor, the node considers that the neighbor node is getting closer. As a result algorithm sets node information stale value as false. On the other hand, the node receives a HELLO beacon from a known neighbor node and the node which receives HELLO is aware that its current position is farther than the previous position of the neighbor node. So this information indicates that the neighbor will be stale. Then, STALE flag is set to true. If a node receives a beacon coming from a known neighbor and they are coming closer, then this node only updates that node information. The STALE flag of the entry is set by

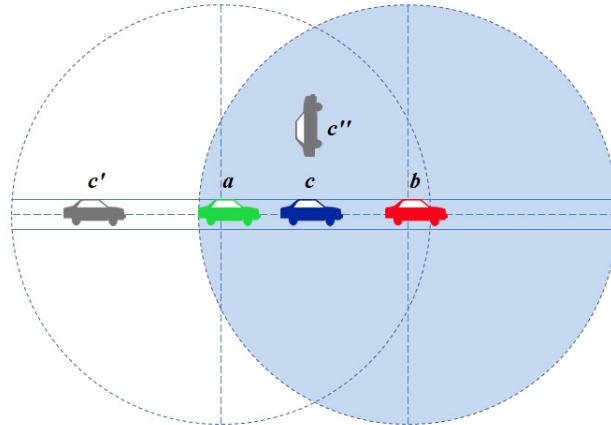


Figure 3.3. Repair strategy of the ARAV algorithm.

false [19]. The flowchart of STALE flag comparison algorithm is shown in Figure 3.2.

3.1. Repair Strategy of the ARAV Algorithm

When a packet achieves a local maximum in GPSR, the node switches to the perimeter mode for that packet. In the perimeter mode, it is not effective to transmit backward in short distance through planarized links in VANET environment. As described in the previous section, the best way to recover from the local maximum is to make the packet achieve a conversion point as fast as possible [19, 26]. To solve the above mentioned issues, ARAV eliminates overlaid links from the neighbor list with the same assumptions in the GBSR algorithm. However, ARAV does not eliminate all the links. It eliminates the overlaid link from the neighbor list only if both nodes are on the same road for generating a border superior sub-graph. In Figure 3.3, each circle shows the radio coverage of the node a and b . As can be seen from the figure, c , c' , c'' and b are the neighbor nodes of node a . The link(a,c) can be eliminated by link(a,b). However, the link(a,c'') can not be eliminated by the link(a,b) because of different directions.

In this way, for the recovery mode of ARAV protocol, packets are forwarded by the right-hand rule like the GPSR protocol. When a VANET node firstly enters into

the recovery mode in ARAV algorithm, it extracts a border superior sub-graph instead of generating a planar graph from its neighbor list, and the node should select the closest node with a minimum angle and toward which it has to move counter-clockwise to reach the destination node [19].

3.2. Greedy Mode of the ARAV Algorithm

ARAV performs routing reactively based on the high mobility of the VANET environment. This is done by considering the density of traveling nodes and junction information besides other network conditions. The critical point of ARAV algorithm is that it takes the advantage of streets and junction without using any external information such as static street maps.

Because of the high rise buildings, radio signals are blocked and data packets should be routed along the streets in urban environments. Actual routing decisions are taken on junction points. Therefore, packets should be transmitted to a node on a junction instead of being transmitted across a junction [26].

Figure 3.4 depicts an example for selecting suitable junction node algorithm. In GPSR greedy forwarding scheme, source node would forward the packet beyond the junction to node F1 if regular greedy forwarding is used. An alternative path to the destination node can be found without getting stuck in a local optimum by forwarding the packet to node A1. The packet is forwarded along the street towards the next junction if the forwarding node is located on a street and not on a junction [18]. We use the same assumptions in GPCR [18] to select junction points.

In ARAV, each node calculates the correlation coefficient (ρ_{xy}) with respect to the position of its neighbors. A correlation coefficient near to 1 shows linear coherence. This means it is located in the middle of the street. A correlation coefficient close to 0 shows that the node is located on a junction therefore no linear coherence. By adjusting a threshold ϵ and comparing (ρ_{xy}) with (ϵ) we predict the location of the nodes. The case $\rho_{xy} \geq \epsilon$ shows that it is located on a street, while $\rho_{xy} < \epsilon$ shows that it

is located within the area of a junction. Apart from GPCR, threshold value $\epsilon = 0.5$ is used to fit city topology in experiments. In [18], developers of GPCR select threshold value as 0.9 for more linear coherence.

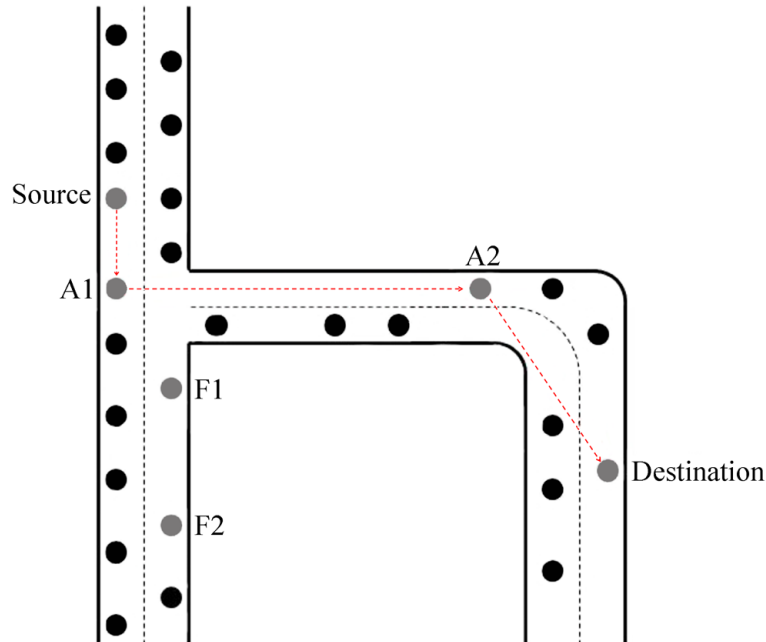


Figure 3.4. Greedy Routing vs. ARAV Routing in the junction.

As shown in Figure 3.4, packets are routed from a junction to the most appropriate path and thereafter continue to be forwarded until they reach another junction. At this next junction, routing decisions are made in order to select the next optimal path to take. In addition, the next best hops for the packets are selected reactively. If there are more than one node in the junction, the ARAV algorithm selects the highest calculated weighted score node in its neighbor list in the junction. In contrast, GPCR [18] algorithm selects any random node in the junction if there are many nodes.

In the ARAV, a VANET source node creates its neighbor list which is in the junction. Afterward in that junction neighbor list, the ARAV calculates each node's velocity, direction of travel, current position and vehicle density and selects the best one for destination. The hop that follows and the paths are chosen based on the highest calculated weighted score. This reactive multi-hop routing process will continue until the packet reaches its destination [9]. Algorithm parameters for calculating vehicle

Table 3.1. System parameters.

Symbol	Explanation
D_i	Distance from current node to the target node
D_j	Distance from next hop to the target node
D_k	D_j / D_i
V_i	Velocity vector
α	Weight factor for position of node
β	Weight factor for direction of travel
W_p	Calculated weighted score for a node on a path

densities are given in Table 3.1.

The proximity of a node to the targeted node is determined by using the value of D_k . The ratio of α and β can be adjusted based on the estimated density of the network and the values of α and β adds up to 1. The weight factor of the direction and also the velocity vector influences the node to be chosen. With the help of the position and direction of the node an optimal next hop can be obtained. As a result, the node with the highest calculated score is chosen to forward packets based on the score W_p calculated according to the following equation. The general structure of ARAV algorithm is shown in Figure 3.5.

The proximity of a node to the targeted node is determined by using the value of D_k . The ratio of α and β can be adjusted based on the estimated density of the network and the values of α and β adds up to 1. The weight factor of the direction and also the velocity vector influences the node to be chosen. With the help of the position and direction of the node an optimal next hop can be obtained. As a result, the node with the highest calculated score is chosen to forward packets based on the score W_p calculated according to the following equation:

$$W_p = \alpha(1 - D_k) + \beta(V_i) \quad (3.1)$$

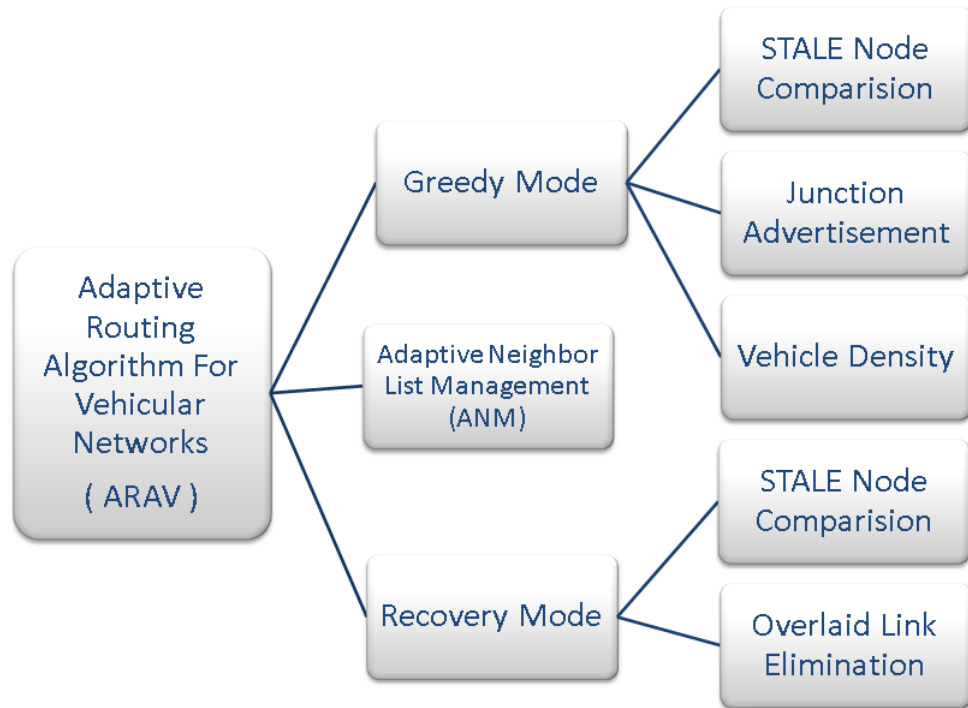


Figure 3.5. The General Structure of ARAV algorithm.

4. PERFORMANCE EVALUATION

4.1. Simulator Design

Routing and transmission of data packets efficiently over ad hoc networks is difficult as they grow. Analytically quantifying the performance and complex behavior of even simple protocols in their entirety is often inaccurate. Improvements in this area of research depends on the performance of simulation tools, more specifically, on the scalability of wireless network simulators [12].

In addition, implementation of actual experiments is burdensome: acquisition hundreds of devices, managing the physical space available for such an experiment and aggregation of data for moving the vehicles are just some of the difficulties in making the empirical efforts discouraging. Therefore, the vast majority of research in this area solely on simulation based a fact that underscores the critical role of efficient simulators [12].

The simulation engine organizes the execution of code parts referring simulated logic. Most existing simulators for ad hoc networks, such as ns-2, OPNET Modeler, OMNeT++ and also JiST/SWANS are based on discrete events in simulation time. With simulation time, the simulation engine, a concept of time, decoupled from the real execution time of the simulation [13].

Discrete-event simulators are important scientific instruments and the focus of a vast body of computer science research, which is addressed to their efficient planning and execution. The JiST system, which stands for Java in simulation time is followed by a long series of simulation frameworks, languages and systems. JiST is a new Java-based discrete event simulation engine with various unique design features [28]. JiST, combines the advantages of traditional systems-based such as ns-2 and language-based GloMoSim approaches to simulation structure [11].

Figure 4.1 presents the taxonomy of VANET simulation software. In [29] all existing VANET simulation software are categorized into three different parts. They are (i) vehicular mobility generators, (ii) network simulators, and (iii) VANET simulators.

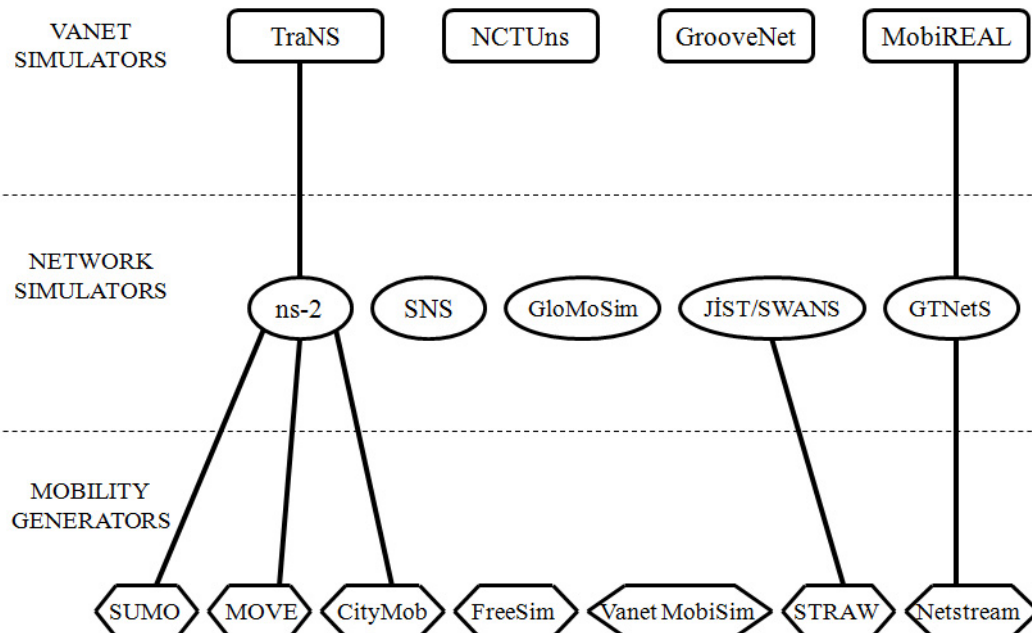


Figure 4.1. A taxonomy of VANET simulation software.

Scalable Wireless Ad hoc Network Simulator (SWANS) is a component-based virtual machine simulator on top of the JiST platform. SWANS outperforms NS2 and GloMoSim significantly, both in time and space. Unlike any other existing network simulator, SWANS efficiently embed existing network applications and run them through simulated networks. SWANS was created primarily because existing wireless network simulation tools are not sufficient for the needs of current research [12].

4.1.1. JiST/SWANS

JiST simulation entities are implemented and compiled as normal Java classes. For that, all of the advantages in Java apply also in JiST. Platform independence, library support are some of advantages that we can say. For presenting simulation time semantics, JiST partitions simulations “entities” which are actually Java objects as well, but whose public methods are called by the scheduler. As the scheduler is not a standard JAVA behavior, this call methods are redirected to the simulation engine.

The redirection process is presented by the application byte-code rewriting before the start of a simulation. During simulation, when a method of an “entity” is called, the call returns immediately, a new entry in the scheduling queue is inserted and executed later [13].

Below, significant benefits of the described JiST approach is listed:

- *Type of Security:* ”Delivery of events” is simply a method call. The virtual machine always checks the compatibility of the source and destination. When receiving events, no additional type testing at the receiving entity needs to be done.
- *No marshaling/demmarshaling:* Since there are no explicit event data structures, there is no overhead for marshaling/demmarshaling data. Instead, the entities only pass references within the VM. With RMI, references can also be transferred across VM boundaries [13].
- *Java:* With Java, provides a wealth of advantages: robustness due to type safety, large library, ease of use, well known to many developers, platform independence, garbage collection, good IDE support, etc [13].
- *Byte-code rewriting:* Because class files are converted and source access is not required. Existing libraries and protocol implementations can be used directly for simulations.

SWANS is a complete library for the simulation of MANETs, based on the JiST engine. MANET simulations require a model for environment and for the node. In swans, the ”Field” creates node mobility and radio propagation. Nodes are a set of entities for implementation of various protocol layers, where the radio entity is connected to the global field entity. Packets traverse the protocol stack entities usually as simple references, virtually for free. Reproduction is possible only if necessary, for example, if a packet is transmitted and must be amended by the forwarders [13].

With regard to radio propagation, researches who uses JiST/SWANS, can choose a free space or two ray path loss model for their simulations with Rayleigh fading,

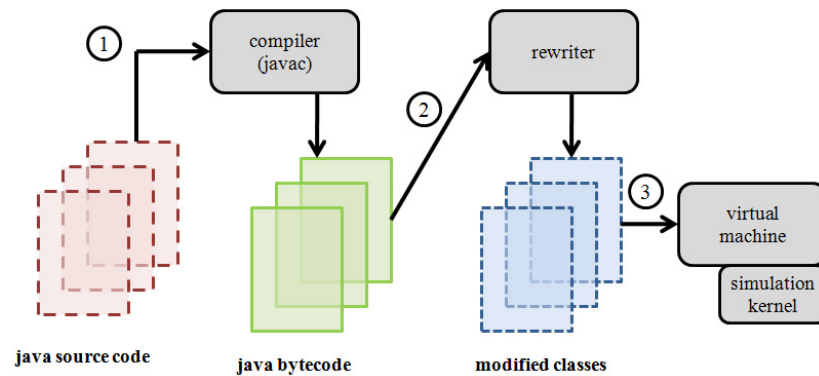


Figure 4.2. The JiST system architecture.

Rician fading or without fading. In addition, a statistics package fall drop strategy is eligible. As node mobility, the standard distribution provides teleporting, random walk and random waypoint models. Moreover, in our thesis we are using STRAW mobility model that we will discuss it in Section 4.2.1.

SWANS also come with basic radio noise models, an implementation of 802.11b MAC, IPv4, AODV, DSR and ZRP MANET routing, as well as TCP and UDP transport and several applications are included. For our thesis, we also develop and add geographic routing protocol GPSR into SWANS. We use the benefits of GPSR methods in Ns-2 and SWANS++ for our thesis.

4.1.2. Architecture of JiST/SWANS

The JiST system architecture, depicted in Figure 4.2, consists of four distinct components: a compiler, a bytecode rewriter, a simulation kernel and a virtual machine. When someone create a plain JiST simulation programs in plain and compiles it to bytecode, then these compiled classes are modified with a bytecode-level rewriter which runs over a simulation kernel. The simulation program, the rewriter and the JiST kernel are all written in pure Java. In this way, this entire process occurs within a standard, unmodified Java virtual machine (JVM) [11].

The integration of simulation semantics in the Java programming language allows us a great deal of work, with Java language, its standard libraries and existing compiler.

JiST also uses the benefits of automatic garbage collection, type safety, reflection and other different features of Java. With the help of this approach, the learning curve for users is reduced and facilitates reuse of code for building simulations. With a standard virtual machine, it provides highly optimized, efficient and also portable execution platform [11, 28].

In summary, a key benefit of the JiST is that it allows for efficient execution of simulation programs within the context of a modern and popular language. JiST integrates simulation semantics, found in a simulation languages and simulation libraries with modern language features. This structure leads to a system that is easy to use, robust and efficient [11, 28].

4.1.2.1. Architectural Differences in JiST. The most plain difference between ns-2 and JiST/SWANS is the general approach of implementing simulation logic. While ns-2 provides interesting possibilities with scripting capabilities, we can say that the approach of creating an object oriented simulation core in C++ is a standard approach. Alternatives to this approach would be simulation environments or domain-specific programming language which requires a timed schedule, by taking over system services. Virtual machine simulator is the given name by the developers of JiST, with rewriting Java byte code corresponding to during class loading, it is possible to record and schedule simulation events [13].

Although the virtual machine itself does not know the schedule of the execution in simulation events, writers of [13] see that JiST which is implemented at a low level keeps the overhead for the maintenance of simulation logic relatively small. They also add that, the core logic of JiST is implemented only with a scheduler component, bytecode rewriter and some marker interfaces and utility functions. There is no need for special protocols or event serialization to run simulations with Java in Simulation Time.

Memory management is reduced, JiST benefits not only things like parallel gen-

erational garbage collection but also it reduces the resources needed by simulations. So, timeless objects, securely can be exchanged between entities without the need to clone them. The amount of memory saved in JIST is very high. On the other hand, simulators like ns-2 clone objects very aggressively for guarantee temporal integrity between entities [11,13,28].

4.1.2.2. Architectural Differences in SWANS. There are characteristic advantages in SWANS compared to ns-2. Smart data structures for faster wireless signal transmission and the implementation of movement are two critical main critical issues that differentiate SWANS from ns-2. The mobility of nodes in a field is implemented in SWANS is time-stepped paradigm not a strictly event based subsystem. This is implemented by the developers and SWANS can update the position of nodes on the field with the selected approach. This is a deviation from a strict event model, as experienced in ns-2. Writers of the [13] also argue that, with large numbers of nodes it is clear that there is limited frequency of position updates in a time-stepped model which scales better than the strict event based model. In strict event based model, it is needed to update positions of numerous nodes for every event [12].

SWANS calculates the position updates not only based on precomputed trace files but it also computes mobility on demand during the simulation. This lets more interactive and flexible mobility models. An application where sharing and using congestion warnings on vehicle networks can be easily simulated in SWANS. Because the change in the direction of a node on the simulation data is readily available in the Swans simulation environment [13].

When a node is sending, wireless nodes in ns-2 register themselves on a channel for receiving radio signals. Then, each node is in charge to compute the signal strength respect to the sender position and has to decide if a signal can be received or not. SWANS works on a model named hierarchical binning which limits the number of nodes computing the signal strength at their receiver which are actually within the signal range of a sender [12]. SWANS uses a data structure which keeps track of node

positions in the field. With its help, SWANS easily addresses nodes in a given signal transmission within a maximum range. For calculating actual reception success, only nodes in this area have to use full signal propagation algorithm. We don't need to work with any position update in such sending operations. Because, time-stepped mobility subsystems keep node positions updated [13].

4.1.3. Performance Analysis of JiST/SWANS Compared to ns-2

Because of the different implementations of movement in the simulators it is not easy to directly compare the raw execution times. As JiST/SWANS computes the movements of all nodes during the runtime of the simulation, writers of [13] mentioned that there is no way to subtract the time it takes to calculate these movements from the other parts of the simulation. This is also applies for ns-2. In ns-2, the subprogram that computes the movements is not part of the main simulation. This is because of the ns-2 structure. Actually, movements are reused repeatedly in different simulation which are previously been rendered into a trace file.

In [13] two simulation environments are compared in a cumulative manner. In Figure 4.3, the cumulative execution times of all subprograms that compose a simulation of CGGC routing are compared. ns-2 is listed two times, one is, excludes the time required to generate movements for ns-2, in order to describe movement reuses across different simulations. The concernancy of this practice in relation to performance characteristics of ns-2 is also clearly visible. When they compare JiST/SWANS against ns-2 without the time required to precompute movements, saw that CGGC routing in JiST/SWANS scales almost linearly with the number of nodes on the field. After they add the time required to compute the movements in ns-2, as is done in JiST/SWANS, ns-2 shows significantly worse scalability compared to JiST/SWANS. This graph is drawn on a logarithmic scale for better readability [13].

Architectural advantages in JiST/SWANS (efficient memory management, run-time profiling and reducing the overhead) creates better performance than ns-2. The efficiency of the time-stepped mobility in JiST/SWANS and the hierarchical binning

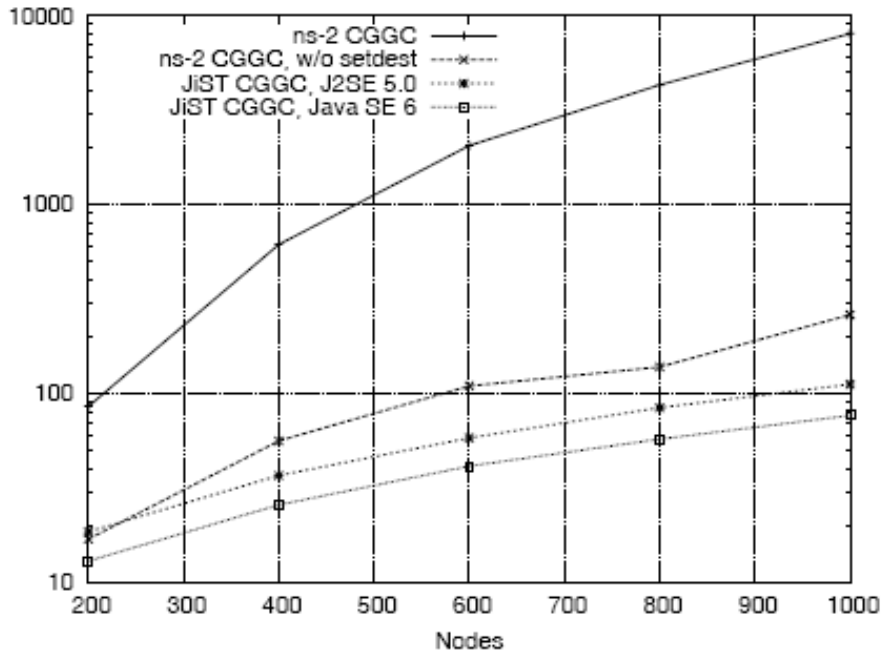


Figure 4.3. Processing time of CGGC routing.

strategies are the explanations for the significant differences in performance.

4.1.4. Used Machines For Simulations

For the simulations carried out for this thesis we used the following machines:

- Intel Dual Processor
- 2x Intel Core 2 Duo T7700 CPU 2.40GHz
- 2x L2 Cache 512kb
- Memory 2.00 GB
- Windows 7 Enterprise

During execution of simulations we often experienced the limitation of these machines. Due to this, we had to reduce the number of simulations resulting in a bit weaker statistics.

4.2. Mobility Model Used for ARAV

A realistic mobility model for vehicular ad-hoc networks (VANETs) is critical for accurate results in simulation performance. In this part of our thesis the implementation of STRAW (STreetRANDOM Waypoint) is discussed which is a reformer mobility model for VANETs. With STRAW, nodes move according to a realistic vehicular traffic model on roads defined by real street map data. In our thesis we used STRAW mobility model with Origin Destination scheme which will be explained in further sections with details.

As discussed in previous sections, vehicular nodes move on streets often separated by buildings, trees or other barriers. Because of this, with increasing the average distance between nodes mostly reduces the overall signal strength received at each node. Connectivity in this environment is a critical requirement for a wide range of different kind of important applications such as real-time traffic monitoring, traffic coordination, targeted vehicular communications, etc. Because of the high cost and also the time constraints of creating ad-hoc network protocols in the real-world, most studies work on simulators for experimental results.

Writers of [14] bring out a realistic mobility model for accurate network simulation results. The Random Waypoint (RWP) [30] mobility model can cause inaccurate results when measure the performance of wireless ad-hoc network protocols and applications. STRAW provides more accurate results with a vehicular mobility model on real US cities, based on the operation of real vehicular traffic. STRAW's current implementation is written for the highly efficient JiST/SWANS discrete-event simulator which is explained in previous section. Writers of [14] presented STRAW which constrains node movement to streets defined by map data for real cities. STRAW limits the node mobility according to vehicular congestion and simplified traffic control mechanisms [31].

A small number of researchers have worked on street-constraints with using real road plans in their VANET simulations. In Saha and Johnson [32], they state that a

random waypoint model is sufficiently similar to the street mobility in terms of network connectivity. They reach this conclusion using a 500m transmission range without path loss model. Moreover, the mobility model that they present does not account for realistic vehicular traffic phenomena such as car-following and traffic control at intersections.

4.2.1. STRAW Mobility Model

In wireless network simulators, each node is used for sending and receiving messages. Moreover, simulators reposition the node on a field according to their mobility models. Wireless network performance cannot reach accurate wireless network simulation results unless the underlying mobility model is sufficiently accurate.

Unluckily, in many individual vehicular traffic simulators each vehicle is used when it enters or leaves a segment; when inside a segment, all vehicles are the same with each other. Studies conducted with this design choice, made it necessary to develop a new alternative traffic model to guarantee accurate wireless network simulation results.

STRAW combines a simple car-following model with traffic control to present vehicular congestion, which models real traffic conditions. STRAW mobility model also combines the idea of an enabled vehicular penetration which shows the cars equipped with radios and actively communicating.

STRAW base on real street plans to build a road map for the specified targets. For each road segment, STRAW uses information such as the road class, residential roads or divided highways, the start and end points of the segment, the name of the street and a list of points along the segment if it is not a straight line. Moreover, STRAW provides at least one lane for each direction that vehicles can move. Street placement model is used which places a vehicle in a lane of random street just before an intersection to determine the initial positions of vehicles on the field. All vehicles are initially stopped position. The new vehicle is placed after the existing one, if there is an another vehicle on that lane [14]. STRAW sets the vehicle speed relative to

the front vehicle. STRAW also uses additional rules to deal with intersections and traffic controls, such as stop signs and traffic lights. Illustration of vehicular mobility components and their interactions in STRAW is shown in Figure 4.4.

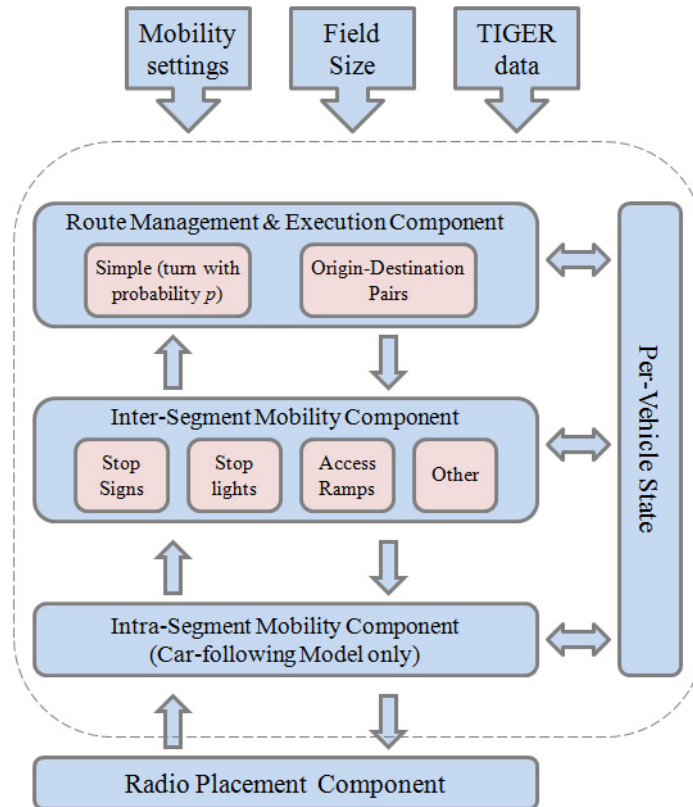


Figure 4.4. Illustration of vehicular mobility components and their interactions in STRAW.

4.2.1.1. Intra-segment Mobility in STRAW. When the simulation starts, nodes move according to the car-following model. Constraints for adjusting the car flow is specified with the following rules :

- The car which comes to the intersection point and the road segment which it wants to drive is full. In this case, the car stops before the intersection, it will wait until there is a place in the next road segment.
- If there is a car in front of the current car, the current car will decelerate its speed to maintain speed-based following distance. STRAW uses the formula which is cited in [33].

$$S = \alpha + \beta V + \gamma V^2 \quad (4.1)$$

where,

S is the following distance.

V is the current vehicle's speed.

α is the vehicle length.

β is the reaction time (we use 0.75 seconds).

γ is the reciprocal of twice the maximum average deceleration of the following vehicle (we use the empirically-derived value [33]).

- If the car in front goes faster than the current car, no speed adjustment is needed.
- If the car meets with a traffic control, it will decelerate before an intersection with a stop sign; if the stoplight turns into green it increases its speed.
- If the car turns onto a new street, it slows down before the intersection to 5mph. This is a reasonable speed for return of crossroads. After returning to the junction it begins to accelerate.

4.2.1.2. Route Management and Execution in STRAW. During simulation, the Route Management and Execution (RME) determines the path which is taken by each vehicle. There are two models for this management. Simple intersegment mobility (Simple STRAW) and mobility with origin-destination (OD) pairs (STRAW OD). In simple STRAW mobility model, it maintains a single value, shows the probability that it will turn at any given intersection, for selecting next road segments. However in the STRAW OD mobility model, the decision is based on the precalculated shortest path between the vehicle's specified origin and destination.

In our thesis we use the origin destination STRAW mobility model for more accurate results. It brings more runtime overhead and it requires knowledge of the driving patterns. However, STRAW OD model is more an exact representation of vehicular

travel [14]. The STRAW OD model requires a significantly longer execution time, which is due to the high cost of computing shortest paths. The STRAW OD model is much more sensitive to region size, because of the shortest path calculation’s worst-case execution time is $O(b^d)$, where b is the number of segments at each intersection and d is the length of the route returned by the algorithm. Thus, as the size of the map grows, the maximum distance between two waypoints increases, which makes d larger.

4.2.2. Implementation of STRAW in JiST/SWANS

STRAW is currently implemented as an extension to JiST/SWANS which is detailed in Section 4.1.

Vehicular mobility model implementation extends interfaces which provided by the SWANS simulator in the `jist.swans.field` package, including the `Field` interface, which encapsulates functionality for mapping radios to locations, the `Mobility` interface, which provides interfaces for implementing the mobility model and the `Spatial` interface, which provides interfaces for locating nodes in the `Field`. The classes that implement vehicular mobility model is consist in the `jist.swans.field.streets` package [31]. Map data is loaded into the simulator by the `StreetMobility` abstract class, which implements the `Mobility` interface and is extended by the RME components to decide the next road. Upon initialization, the `StreetMobility` class loads street information from files containing the road segment information, road segment shape and street name [31].

In our thesis, the US Census Bureau’s TIGER [15] data files, Washington DC Map is used for the source of street plans. The TIGER data is supplied as packages, organized with state county. Files in the packages provide the information about various geographic features. For example, locations and dimensions of schools, parks, roads and other landmarks. We use Washington DC MAP files to extract the names, locations and shapes of roads, road “classes”, which can be used for estimation of the speed limit and the capacity of each road. The detailed of TIGER data is discussed in Section 4.3.

4.3. City Topology: The Census Bureau's TIGER System

In urban centers, buildings and similar obstacles limit the propagation of the communication signal and split the network topology [8]. Factors such as road configuration, traffic laws, safety limits, and physical limits affect the mobility of vehicles. Drivers' behavior and interaction with each other also contribute to the vehicle mobility pattern [8]. Vehicle mobility creates a highly dynamic topology. In heavy traffic congestion, vehicles may be within a couple of meters proximity of one another, whereas on a sparsely populated road the distance may be hundreds of meters. Traffic condition may change rapidly between congested and sparse due to traffic jams, accidents, and road constraints.

We presented ARAV routing approach, to overcome the challenges of city scenarios. In our thesis we used STRAW OD mobility model for exact representation of vehicular travel. We also had used real city maps for more accurate experimental results. We used TIGER digital Washington DC map for real city situations. The name TIGER comes from the Topologically Integrated Geographic Encoding and Referencing. The TIGER system which is developed at the U.S Census Bureau creates system and digital map databases to support Decennial Census and other Bureau programs. The geographic structure of the United States and its territories is built upon the design of the TIGER database which follows the graph theory and related fields of mathematics to provide disciplined, mathematical description. Built on the location and relationship of streets, rivers, railroads, and other features to each other and to the numerous geographic entities the TIGER database defines the topological structure.

Automated map scanning, manual map "digitizing", standard data keying, and sophisticated computer file matching is build up in TIGER database with an integrated variety of encoding techniques. The aim was to supply automated retrieval of geographic information about the United States and its territories [15]. Creation and management of the digital geographic database covers complete coverage of the United States, Puerto Rico, the U.S. Virgin Islands, American Samoa, Guam, the Commonwealth of the Northern Mariana Islands, and the Midway Islands [15].

The TIGER/Line files consists of information such as location in latitude and longitude, the name, the type of feature, address ranges for most streets, the geographic relationship to other features, and other related information. TIGER/Line files are digital data describing geographic features are not graphic images of maps. With the appropriate software, we are using JIST/SWANS to produce maps ranging in details. Till now, many local governments have used the TIGER/Line data in applications which requires digital street maps. Also, the private sector has used TIGER/Line data to create products that produce maps for government, business and the general public [15].

For the simulations in our thesis, we used a real city topology map from TIGER Washington DC [15] street map as shown in Figure 4.5. The movement of the nodes was generated with a dedicated vehicular traffic simulator STRAW.

4.4. Experimental Results

The well known network simulator JIST/SWANS [11–13] was used to perform the simulations. The ARAV protocol was compared with the existing routing protocol GPSR. STRAW (STreetRANdom Waypoint) [14] mobility model was used to simulate the movement pattern of moving vehicles on streets or roads. STRAW provides accurate simulation results by using a vehicular mobility model based on the operation of real vehicular traffic for real US city maps. STRAW is also written for the highly efficient JiST/SWANS discrete-event simulator.

For the simulations, we used a real city topology map from Topologically Integrated Geographic Encoding and Referencing system (TIGER) which is a part of Washington DC [15] street map as shown in Figure 4.5. The movement of the nodes was generated with a dedicated vehicular traffic simulator and represents a real world movement pattern for this given scenario [16]. The nodes communicate using the IEEE 802.11 wireless LAN standard with a maximum range of 750 m. Real world tests in [10] with cars have shown that this is a reasonable value when using external antennas [18]. For each simulation, a different number of sender-receiver pairs was randomly selected. The nodes move at an average velocity of up to 15 m/s.

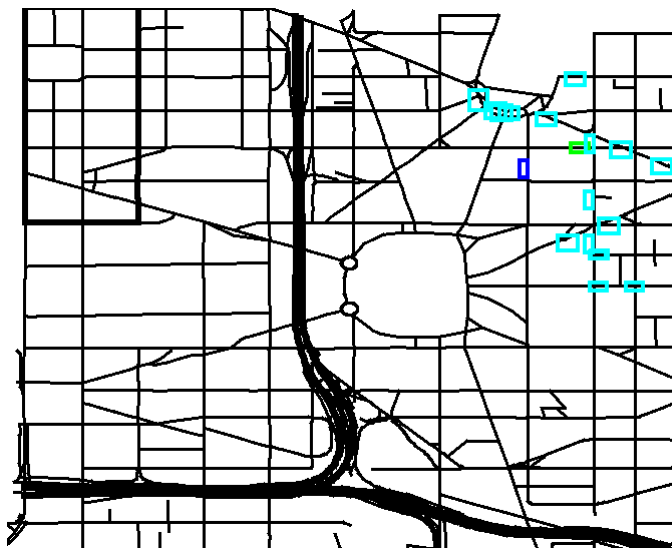


Figure 4.5. Map 1: Washington DC, USA street map.

Table 4.1. Simulation parameters.

Parameter	Value
Number of Vehicles	100–400
Average speed of vehicles (m/s)	15
Number of packet Senders	5–20
Transmission Range (meter)	750
Constant Bit Rate (Packets/Second)	4
Packet Size (Bytes)	1084
Vehicle beacon interval (Seconds)	0.5
MAC Protocol	IEEE 802.11
α	0.3
β	0.7

The initial beaconing rate was set to 0.5 s. For these simulations, a simple traffic model was used to generate communications among nodes. Data packets were transmitted among nodes using Constant Bit Rate (CBR) where packets are sent continuously until the end of the transmission. The connection rate was fixed at 0.25 s, that is, 4 packets per second, with packet sizes of 1084 bytes.

Performance metrics are critical to show the effectiveness of any routing protocol. The achieved packet delivery rate (PDR) and greedy perimeter were measured ratio in this paper and defined as:

$$PDR = D_p/D_s \quad (4.2)$$

where D_p is the total packet received in destination and D_s is the total packet sent to the destination.

We denote ARAV with junction scheme as ARAV1 and without junction scheme

as ARAV2. We also devised an adaptive algorithm ARAV3 which uses junction scheme adaptively when it is needed. When the current forwarder node and the destination nodes orthogonal projection velocity vectors are on the same direction and the angle between them is lower than 7° in ARAV3, we assume that they are on the same road. For this situation we did not use junction scheme. In these situation, hop that follows are chosen based on only the highest calculated weighted scores.

In the ARAV1 algorithm, a VANET source node creates its neighbor list which is in the junction. Afterward in that junction neighbor list, the ARAV1 algorithm calculates each node's velocity, direction of travel, current position and vehicle density and selects the best one for destination. The hop that follows and the paths are chosen based on the highest calculated weighted scores. This reactive multi-hop routing process will continue until the packet reaches its destination [9]. On the algorithms, these factors will have different weights which is related on network conditions and location of the vehicles.

On the other hand, in ARAV2 algorithm, a VANET source node creates its neighborlist and the path is chosen based on the highest calculated weighted scores. The computation process in ARAV2 is less than ARAV1. Because ARAV2 doesn't uses junction advertisement algorithm, however it calculates weighted score algorithm in nodes. With varying algorithm in different conditions, a packet has to switch between different forwarding modes. The different modes relate to the current location of a vehicle and affect the path and forwarding process. Because of these situations and the changing traffic conditions, the weights are not constant.

Figure 4.6 shows the PDR results obtained in Map 1. As shown in Figure 4.6 with different number of nodes and transmitters in city environment, ARAV adaptive algorithm outperforms GPSR. ARAV3, with 5 transmitters and 100 nodes, maintains a PDR above 0.85 whereas for the GPSR, the ratio has dropped below 0.4. The poor results shown by the GPSR in sparse densities are because of its insufficiency to handle disconnected networks. The GPSR uses the perimeter mode as its recovery strategy that often fails since most nodes are not connected.

With ANM in ARAV algorithm each node can distinguish the stale nodes on its neighbor list. ARAV protocols accomplish nearly a stale-free neighbor list in combination with the ANM scheme. Moreover, in the greedy part of the ARAV algorithm, packets are routed from one junction to another, and both the direction and the position of the hops that follow are taken into account by considering the density of the path. Nonetheless, in the perimeter part of ARAV, overlaid links are eliminated from the neighbor list until both of the nodes are on the same road. Because of the above mentioned reasons all ARAV algorithms outperform the GPSR.

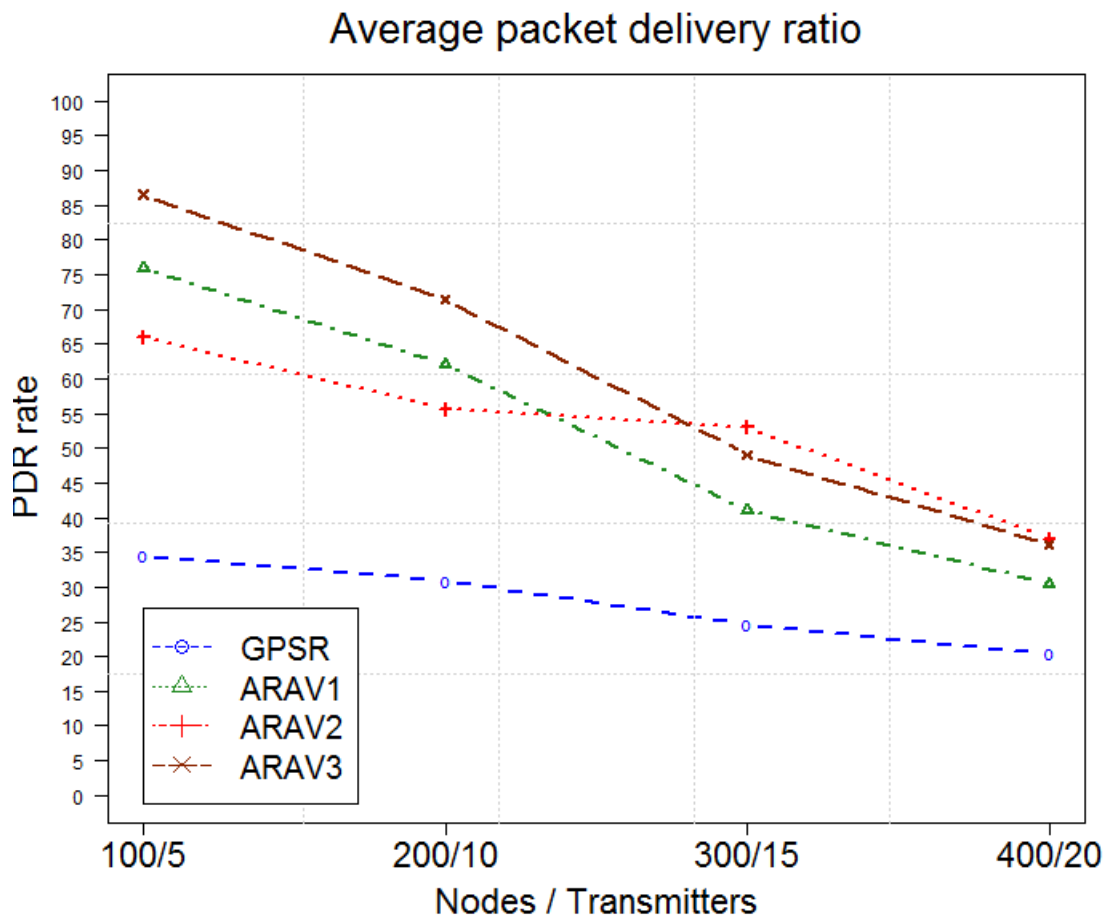


Figure 4.6. Average packet delivery ratio for different number of nodes and transmitters.

However, for high density networks, the results for PDR go down in ARAV1 because of the correlation calculation in neighbor table for junctions. As a result, it creates a slight increase in latency and a higher average number of hops. This increase

in hop counts and latency may bring the local maximum and converts the routing to perimeter mode. So a packet which meets up with the local maximum in correlation calculation causes an increase in the percentage of the perimeter mode in ARAV1 as shown in Figure 4.10. However, the results are still better than GPSR for high density networks.

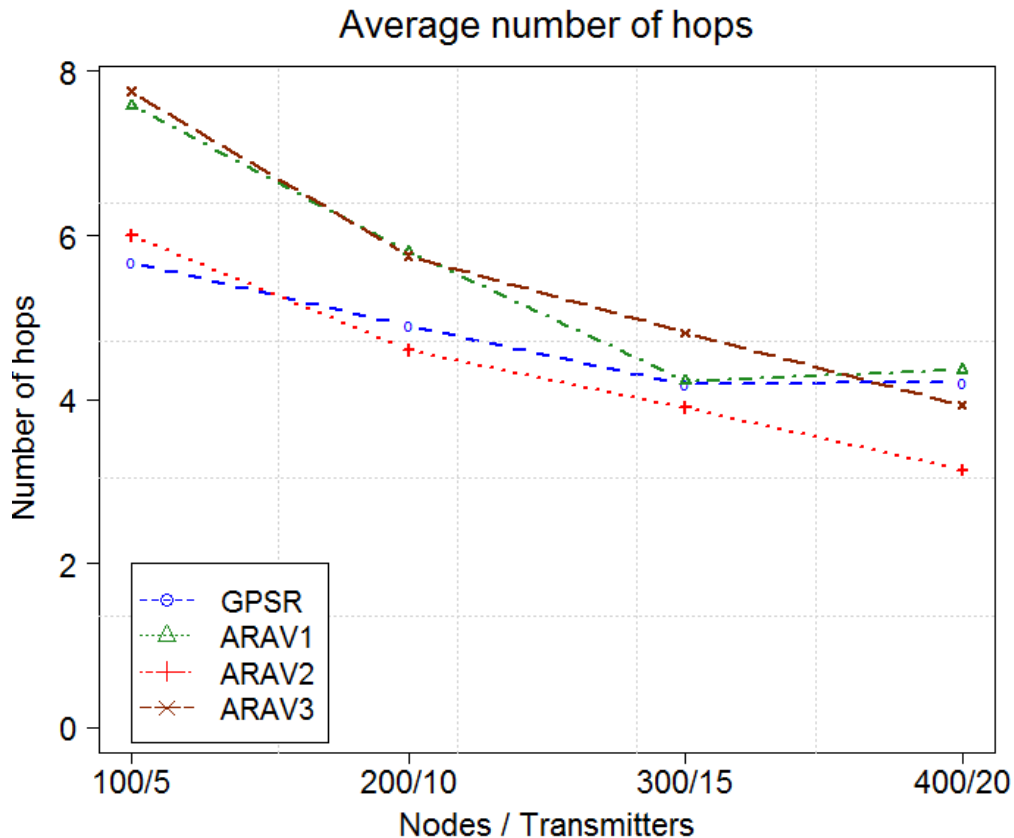


Figure 4.7. Average number of hops for different number of nodes and transmitters.

As shown in Figure 4.6, ARAV algorithms were also run without the junction assumptions for the same node and transmitter numbers. ARAV2 algorithm was used without junction scheme but with vehicle density algorithm. This scheme in high density networks outperforms both ARAV1 and GPSR. Results are similar with ARAV3. Similarly, Figure 4.7 shows the average number of hops for each received packet in destination. In ARAV1, the hop count is higher than ARAV2 and GPSR. This is because ARAV1 selects junction nodes rather than forwarding to nodes which are geographically closer to the destination. This is the main drawback of the ARAV1 algorithm. However this is addressed in ARAV3 algorithm. ARAV3 algorithm uses adaptive mech-

anism for selecting junction nodes or nodes near to destination. Because of this, for higher densities ARAV3 hop count is lower than ARAV1. For high densities, the average number of hops in perimeter mode decreases. In the perimeter part of ARAV algorithms, overlaid links are eliminated from the neighbor list until both of nodes are on the same road. This results in lower hop counts and therefore lower forwarding delays in the recovery mode.

In Figure 4.8, we compute the average delay in one successful transmission. Figure show that with increasing number of nodes and transmitters in the network, the delay of ARAV2 is still decreasing. This is because ARAV2 works with greedy weight algorithm and does not use junction advertisement algorithm. On the other side, there is an decrease/increase in ARAV1. This is because of the correlation calculation in neighbor table for junctions. This calculation causes an increase in delay when there are many nodes in junction points. For ARAV3 algorithm, delay is decreasing for higher densities. This is because of its adaptive scheme for selecting junction points. Final results also show that, for average delay our algorithm ARAV for both modes, outperforms GPSR. In figure 4.9 the average throughput of the algorithms is shown.

The ratio of greedy/perimeter modes is also another important performance indicator. Figure 4.10 shows greedy/perimeter ratio for different algorithms and also the node and transmitter numbers. As indicated in the graph, the greedy ratio of ARAV is higher than GPSR for all node transmitter pairs. With minimum number of transmitters, ARAV1 results are better than ARAV2 scheme. On the other hand, when the number of nodes increases, the performance of ARAV1 suffers compared to ARAV2 algorithm.

Figure 4.10 shows greedy/perimeter ratio for different algorithms and also the node and transmitter numbers. As indicated in the graph, the greedy ratio of ARAV is higher than GPSR for all node transmitter pairs. With minimum number of transmitters, ARAV1 results are better than ARAV2 scheme. On the other hand, when the number of nodes increases, the performance of ARAV1 suffers compared to ARAV2 algorithm.

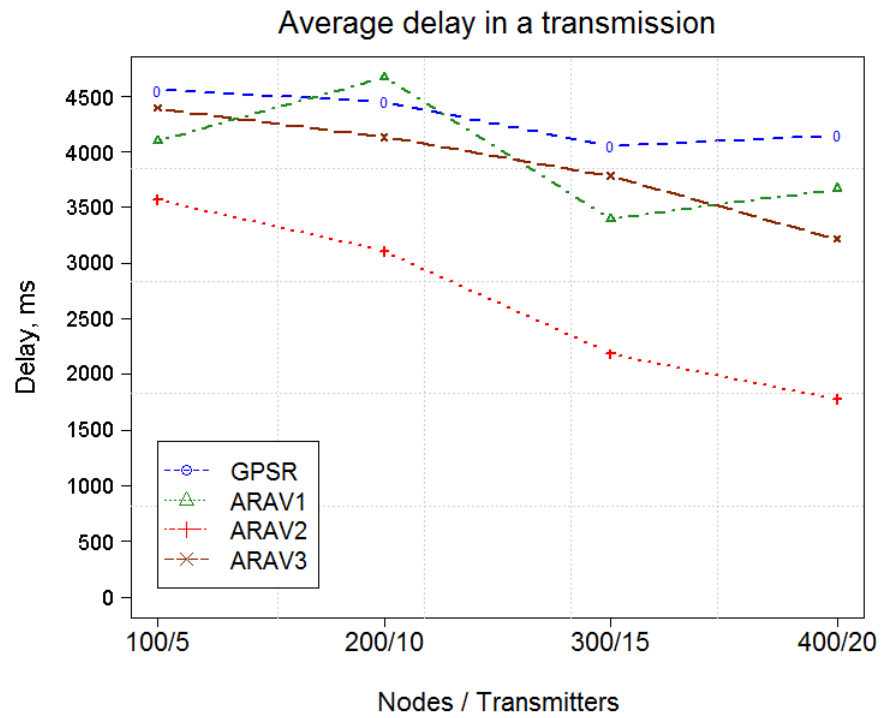


Figure 4.8. Average delay in one transmission for different number of nodes and transmitters.

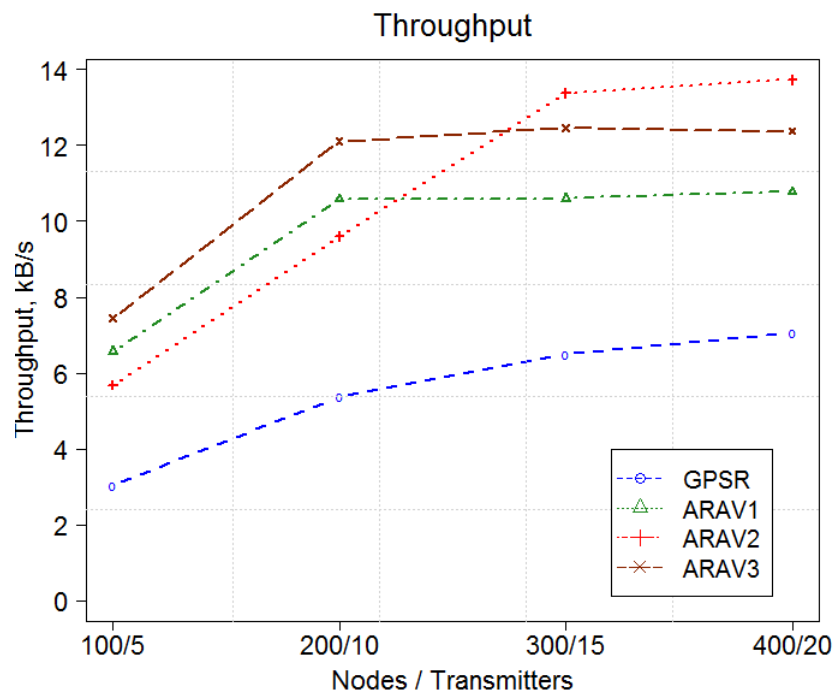
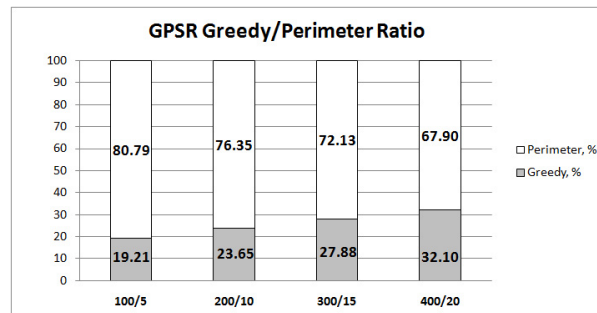
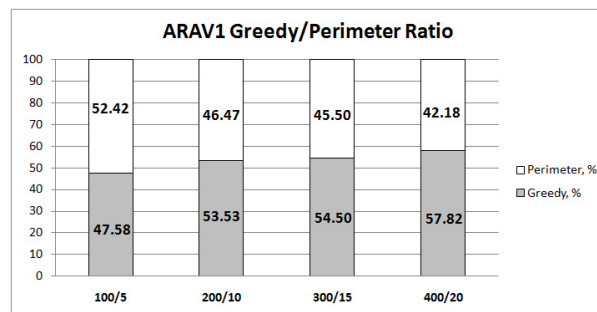


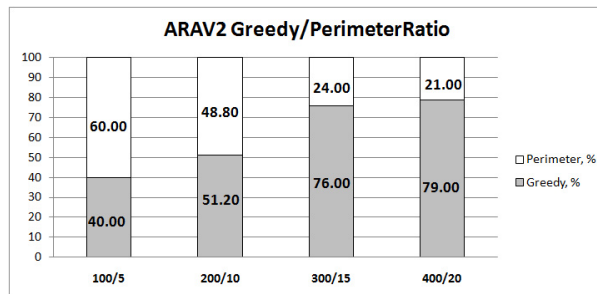
Figure 4.9. Average throughput in one transmission.



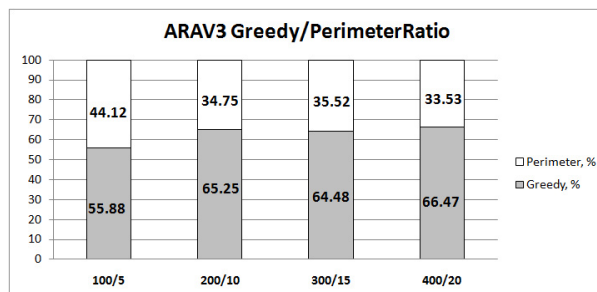
(a) GPSR



(b) ARAV with Junctions (ARAV1)



(c) ARAV without Junctions (ARAV2)



(d) ARAV with Adaptive Junction Algorithm (ARAV3)

Figure 4.10. Greedy perimeter ratio for different algorithms.

5. CONCLUSION and FUTURE WORK

Fueled by the need for the new solutions, many geographic routing protocols for VC system have been proposed in the literature. When we analyze current configurations of some routing algorithms we see that they only focus on specific requirements to resolve the current situation. One big common drawback of these protocols is that they cater only one or two road characteristic problems in their side and can not be applied to various applications. For example, GBSR [19] worked for the problem about maximum number of hops in perimeter mode of GPSR [10] and created stale free Neighbor List Management schemes. On the other hand, writers of GPCR [18] worked to solve the packet delivery ratio problem in GPSR [10] with junction advertisement algorithm. We have seen that one proposed routing solution does not create a solution for many different kinds of applications. We propose an adaptive routing algorithm ARAV that it takes into consideration street topology and vehicle mobilities. Then, it provides the most appropriate solution under certain constraints. Detailed ARAV approach is summarized in Figure 3.1 for more information.

We used well known network simulator JIST/SWANS [11–13] with Street Random Waypoint Origin destination (STRAW OD) [14] mobility model to perform ARAV algorithm. The movement of the nodes was generated with a dedicated vehicular traffic simulator and represents a real world movement pattern. For the simulations, we used a TIGER real city topology map, Washington DC [15] street map. For each simulation, a different number of sender-receiver pairs was randomly selected. As seen in the graphs in Section 4.4, ARAV has produced promising results when implemented in a VANET city environment. ARAV, with 5 transmitters and 100 nodes, maintains a PDR above 0.75 whereas for the GPSR, the ratio has dropped below 0.4 in DC city map. The analysis of the results with different performance metrics show that complete ARAV algorithm gives much better results among various network densities.

We describe Adaptive Routing Algorithm For Vehicular Networks (ARAV): a new position-based routing approach which modifies GPCR and GBSR is presented to

make use of the advantages of these algorithms in an adaptive manner. For the greedy part of ARAV, GPCR assumptions are used with some modifications. However, for the perimeter part of ARAV, GBSR is used. For both modes of the routing algorithm adaptive neighborlist management (ANM) with stale node detection algorithm is used. ARAV differs from these algorithms, in that it uses these routing algorithms adaptively in two routing modes. Simulations with realistic traffic movements are performed in a real map to compare ARAV1, ARAV2 and ARAV3 to GPSR scheme. As seen in the graphs, ARAV has produced promising results when implemented in a VANET city environment. The analysis of the results with different performance metrics show that complete ARAV algorithm performs better results among various network densities. As a future work, ARAV should be improved to communicate with Road Side infrastructures. For this issue, sparse RVC system can be deployed by stages, therefore not requiring substantial investments before any available benefits. In this way, ARAV can be applied to work with various different kind of applications.

REFERENCES

1. Qi, L., “Research on Intelligent Transportation System Technologies and Applications”, *Workshop on Power Electronics and Intelligent Transportation System (PEITS)*, pp. 529–531, IEEE Computer Society, 2008.
2. Kargl, F., E. Schoch, and M. Weber, “Security for Inter-Vehicular Communication Mechanisms: What is Next?”, in *P. Papadimitratos and J.-P. Hubaux, “Report on the ‘Secure Vehicular Communications: Results and Challenges Ahead’ Workshop” ACM Mobile Computing and Communications Review*, Vol. 12, No. 2, 2008.
3. Sichitiu, M. L. and M. Kihl, “Inter-vehicle Communication Systems: a Survey”, *Communications Surveys & Tutorials*, Vol. 10, pp. 88–105, 2008.
4. Olariu, S. and M. C. Weigle, *Vehicular Networks: From Theory to Practice*, Chapman & Hall/CRC, 2009.
5. Luo, J. and J.-P. Hubaux, “A Survey of Inter-Vehicle Communication”, *EPFL Lausanne Switzerland Tech Rep*, Vol. 11, No. 2, pp. 1–12, 2004.
6. Jiang, D. and L. Delgrossi, *IEEE 802.11p: Towards an International Standard for Wireless Access in Vehicular Environments*, pp. 2036–2040, IEEE, 2008.
7. Gongjun, Y., S. El-Tawab, and D. B. Rawat, “Survey of Routing Protocols in Vehicular Ad Hoc Networks”, Watfa, M. K. (editor), *Advances in Vehicular Ad-Hoc Networks: Developments and Challenges*, pp. 149–170, IGI Global, 2010.
8. Artimy, M. M., W. Robertson, and W. J. Phillips, “Vehicular Ad Hoc Networks: An Emerging Technology Toward Safe and Efficient Transportation”, *Algorithms and Protocols for Wireless and Mobile Ad Hoc Networks*, pp. 405–432, John Wiley & Sons, Inc., 2008.

9. Tee, C. and A. Lee, “Adaptive Reactive Routing for VANET in City Environments”, *Pervasive Systems, Algorithms, and Networks (ISPAN)*, pp. 610–614, IEEE Computer Society, 2009.
10. Karp, B. and H. T. Kung, “GPSR: Greedy Perimeter Stateless Routing for Wireless Networks”, *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, pp. 243–254, ACM, 2000.
11. Barr, R., Z. J. Haas, and R. van Renesse, “JiST: n Efficient Approach to Simulation Using Virtual Machines”, *Software: Practice and Experience*, Vol. 35, No. 6, pp. 539–576, 2005.
12. Rimon Barr, Z. J. H. and R. van Renesse, “Scalable Wireless Ad Hoc Network Simulation”, *Handbook On Theoretical And Algorithmic Aspects Of Sensor, Ad Hoc Wireless, and Peer-to-Peer Networks*, Auerbach Publications, 2005.
13. Schoch, E., M. Feiri, F. Kargl, and M. Weber, “Simulation of Ad Hoc Networks: Ns-2 Compared to JiST/SWANS”, *Simutools '08: Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops*, pp. 1–8, Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering (ICST), 2008.
14. Choffnes, D. R. and F. E. Bustamante, “An Integrated Mobility and Traffic Model for Vehicular Wireless Networks”, *Proceedings of the 2nd ACM International Workshop on Vehicular Ad Hoc Networks*, VANET '05, pp. 69–78, ACM, 2005.
15. “Topologically Integrated Geographic Encoding and Referencing system”, 2006 Second Edition TIGER Line Files, DC Map <ftp://ftp2.census.gov/geo/tiger/tiger2006se/DC/>.
16. Lochert, C., H. Hartenstein, J. Tian, H. Fussler, D. Hermann, and M. Mauve, “A Routing Strategy for Vehicular Ad Hoc Networks in City Environments”, *Proceedings of Intelligent Vehicles Symposium*, pp. 156–161, IEEE, 2003.

17. Zeadally, S., R. Hunt, Y.-S. Chen, A. Irwin, and A. Hassan, “Vehicular Ad Hoc Networks (VANETS): Status, Results and Challenges”, *Telecommunication Systems*, pp. 1–25, 2010.
18. Lochert, C., M. Mauve, H. Fussler, and H. Hartenstein, “Geographic Routing in City Scenarios”, *SIGMOBILE Mobile Computing Communication Review*, Vol. 9, pp. 69–72, 2005.
19. Yu, S., H. Choi, and G. Cho, “A New Recovery Method for Greedy Routing Protocols in High Mobile Vehicular Communications”, *Vehicular Electronics and Safety*, pp. 45–50, IEEE International Conference on Vehicular Electronics and Safety (ICVES), 2008.
20. Kayis, O., *Cluster Design For Inter-Vehicle Communication*, MS Thesis, Galatasaray University, 2008.
21. Taha, M. M. I., *Broadcasting Protocols in Vehicular Ad-Hoc Networks*, MS Thesis, Assiut University, 2008.
22. Camp, T., J. Boleng, and L. Wilcox, “Location Information Services in Mobile Ad Hoc Networks”, Vol. 5, pp. 3318–3324, IEEE International Conference on Communications, 2002.
23. Toussaint, G. T., “The Relative Neighbourhood Graph of a Finite Planar Set”, *Pattern Recognition*, Vol. 12, No. 4, pp. 261–268, 1980.
24. Gabriel, R. K. and R. R. Sokal, “A New Statistical Approach to Geographic Variation Analysis”, *Systematic Zoology*, Vol. 18, No. 3, pp. 259–278, 1969.
25. Dijkstra, E. W., “A Note on Two Problems in Connexion with Graphs”, *Numerische Mathematik*, Vol. 1, pp. 269–271, 1959.
26. Lee, K., J. Haerri, U. Lee, and M. Gerla, “Enhanced Perimeter Routing for Ge-

- ographic Forwarding Protocols in Urban Vehicular Scenarios”, *IEEE Globecom Workshops*, pp. 1–10, 2007.
27. Akay, H., *Realistic Mobility Modeling For Vehicular Ad Hoc Networks*, MS Thesis, Boğaziçi University, 2008.
 28. Barr, R., *JiST, Java in Simulation Time User Guide*, Cornell University, 2005.
 29. Martinez, F. J., C. K. Toh, J.-C. Cano, C. T. Calafate, and P. Manzoni, “A Survey and Comparative Study of Simulators for Vehicular Ad Hoc Networks (VANETs)”, *Wireless Communications and Mobile Computing*, Vol. 11, No. 7, pp. 813–828, 2009.
 30. Yoon, J., M. Liu, and B. Noble, “Sound Mobility Models”, *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking*, The Annual International Conference on Mobile Computing and Networking (MobiCom), pp. 205–216, ACM, 2003.
 31. Choffnes, D. R. and F. E. Bustamante, “Modeling Vehicular Traffic and Mobility for Vehicular Wireless Networks”, Technical report, Northwestern University, 2005.
 32. Saha, A. K. and D. B. Johnson, “Modeling Mobility for Vehicular Ad-hoc Networks”, *Proceedings of the 1st Acm International Workshop on Vehicular Ad Hoc Networks*, pp. 91–92, ACM, 2004.
 33. Choffnes, D. R. and F. E. Bustamante, “Car following models In Trac Flow Theory”, *Transportation Research Board*, 1992.