A RELIABLE MULTIHOP BROADCASTING IN VANET

By

Nisha Khurana
(09MCES04)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
AHMEDABAD-382481
May 2012

A RELIABLE MULTIHOP BROADCASTING IN VANET

Major Project

Submitted in partial fulfillment of the requirements

For the degree of

Master of Technology in Computer Science & Engineering

By

Nisha Khurana
(09MCES04)

Guide

Dr. S. N. Pradhan

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
AHMEDABAD-382481
May 2012

UNDERTAKING

I, Nisha Khurana, Roll No. 09MCES04, give undertaking that the Major Project entitled ***"A Reliable Multihop Broadcasting in VANET"*** submitted by me, towards the partial fulfillment of the requirements for the degree of Master of Technology in Computer Science and Engineering of Nirma University, Ahmedabad, is the original work carried out by me and I give assurance that no attempt of plagiarism has been made. I understand that in the event of any similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

Signature of Student:

Date:

Place:

CERTIFICATE

This is to certify that Major Project entitled ***"A Reliable Multihop Broadcasting in VANET"*** submitted by **Nisha Khurana (09MCES04)**, towards the partial fulfillment of the requirements for the degree of Master of Technology in Computer Science & Engineering of Nirma University, Ahmedabad is the record of work carried out by her under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this major project, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Dr. S N Pradhan
Guide, Professor
Department of Computer Science & Engg
Institute of Technology
Nirma University
Ahmedabad

Prof. D J patel
Professor and Head
Department of Computer Science & Engg
Institute of Technology
Nirma University
Ahmedabad

Dr. K Kotecha
Director
Institute of Technology
Nirma University
Ahmedabad

ACKNOWLEDGEMENT

On the submission of my project report on ***"A Reliable Multihop Broadcasting in VANET"***, I am deeply indebted to my project guide, **Dr. S N Pradhan**, Professor, CSE department, Institute of Technology, Nirma University, whose help, stimulating suggestions and encouragement helped me during the course of my work. I am heartily thankful to him for his time to time suggestions and the clarity of the concepts of the topic that helped me a lot during this study.

I thank **Prof. D J Patel**, Head, Computer Science & Engineering Department, Institute of Technology, Nirma University, for extending all the support. I am also thankful to **Dr. K kotecha**, Director, Institute of Technology, Nirma University, for his kind support in all respect during my study.

I am thankful to all the **faculties and friends** whose direct and indirect support helped me complete this report.

Last but not the least I am thankful to **almighty** for giving me the strength to overcome all the hurdles, my **parents and family** for their valuable support and sacrifice not only through this study but throughout my life.

ABSTRACT

Broadcasting of messages, especially for safety applications, is one of the fundamental services in vehicular ad-hoc networks (VANETs). These services typically require delivering the information to all vehicles traveling over a geographical area, with high reliability and low delay. At the same time the number of relaying hops must be minimized to avoid the broadcast storm problem. Broadcasting in VANET is very different from routing in mobile ad hoc networks (MANET) due to several reasons such as network topology, mobility patterns, high speed of the nodes, different traffic patterns etc.

These differences imply that conventional ad hoc routing protocols are not appropriate in VANETs for most vehicular broadcast applications. During my M.Tech thesis study, I propose a reliable solution for the broadcasting in VANET designed for the optimum performance of public-safety related applications. Also the broadcasting may be used in many other applications. The protocol is based on slotted p-persistence[17] and RTS/CTS handshake scheme of UMB protocol[16] which guarantees the successful reception of a broadcasting message. At the time of broadcasting it becomes essential to select a node who should be the next relayer otherwise it may lead to broadcast storm problem. While the RTS/CTS scheme makes the protocol reliable and takes care of the hidden-node problem, the next broadcaster is selected using p-persistence algorithm. Simulation results show that the proposed protocol performed better in terms of bytes usage, success rate and propagation time when compared with slotted p-persistence and UMB protocol.

Contents

List of Figures

LIST OF ACRONYMS

ACK	Acknowledge
AU	Application Unit
ACK	Acknowledge
AU	Application Unit
CBR	Constant Bit Rate
C2C-CC	Car to Car Communication Consortium
CBR	Constant Bit Rates
CTB	Clear to Broadcast
CTS	Clear To Send
CW	Congestion Window
DSRC	Dedicated short-range communication
DECA	Density-aware reliable broadcasting protocol
EAEP	Edge-aware epidemic protocol
GIS	Geographic Information System
GPS	Global Position System
IP	Internet Protocol
ITS	Intelligent Transport System
LAN	Local Area Network
LL	Link Layer
MAC	Multiple Access Control
MANET	Mobile Ad-hoc NETwork
MOVE	MObility model generator for VEhicular networks
NS-2	Network Simulator version 2
OBU	On Board Unit
PGB	Preferred Group Broadcast
POCA	Position-aware reliable broadcasting protocol
QoS	Quality of Services
REQ	Request
RTB	Request To Broadcast
RTS	Request To Send
RSU	Road Side Unit
SUMO	Simulator of Urban Mobility
UDP	User Datagram Protocol
UMB	Urban Multihop Broadcast
VANET	Vehicular Ad-hoc NETwork
WAVE	Wireless Access in a Vehicular Environment

1

Introduction

In this global and technological era everything is becoming wireless. Wireless technologies are becoming dominant method of exchanging information. Satellite televisions, cellular phones and wireless internet access are well-known applications of wireless technologies. Wireless research field is growing faster than any other one. It serves a wide range of applications. The researchers are putting their hard efforts in finding the solutions to the challenges of these wireless technologies by finding new protocols for new applications.

In this chapter, an introduction to a wireless technology is presented that is expected to be adopted by both government and manufacturers for future transportation system. It is related to the safety of the vehicles on roads. It is the technology of building a robust network between mobile vehicles; which means vehicles can talk to each other. This promising technology is known as Vehicular Ad-Hoc Networks (VANETs). In this chapter, an introduction to the technology of VANETs will be presented.

1.1 What are Vehicular Networks?

Vehicles connected to each other through an ad hoc formation form a wireless network called "Vehicular Ad Hoc Network".

Vehicular networks are a novel class of wireless networks that have emerged as advances in wireless technologies and the automotive industry. Vehicular networks are spontaneously formed between moving vehicles equipped with wireless interfaces that could be of homogeneous or heterogeneous technologies. These networks, also known as VANETs, are considered as one of the ad hoc network real-life application enabling communication among nearby vehicles as well as between vehicles and nearby fixed equipment, usually described as roadside equipment. Vehicles can be either private, belonging to individuals or private companies, or public transportation means. Fixed equipment can belong to the government or private network operators or service providers.

Vehicular ad-hoc networks (VANETs) are a subgroup of mobile ad hoc networks (MANETs) with the distinguishing property that the nodes are vehicles like cars, trucks, buses and motorcycles. This implies that node movement is restricted by factors like road course, encompassing traffic and traffic regulations. Because of the restricted node movement it is a feasible assumption that the VANET will be supported by some fixed infrastructure that assists with some services and can provide access to stationary networks. The fixed infrastructure will be deployed at critical locations like slip roads, service stations, dangerous intersections or places well-known for hazardous weather conditions.

Nodes are expected to communicate by means of few standards available for wireless communication. To allow communication with participants out of radio range, messages have to be forwarded by other nodes (multi-hop communication). Vehicles are not subject to the strict energy, space and computing capabilities restrictions normally adopted for MANETs. More challenging is the potentially very high speed of the nodes (up to 250 km/h) and the large dimensions of the VANET. One of the VANET goal is to increase road safety. Figure-1 gives the overview of VANET.

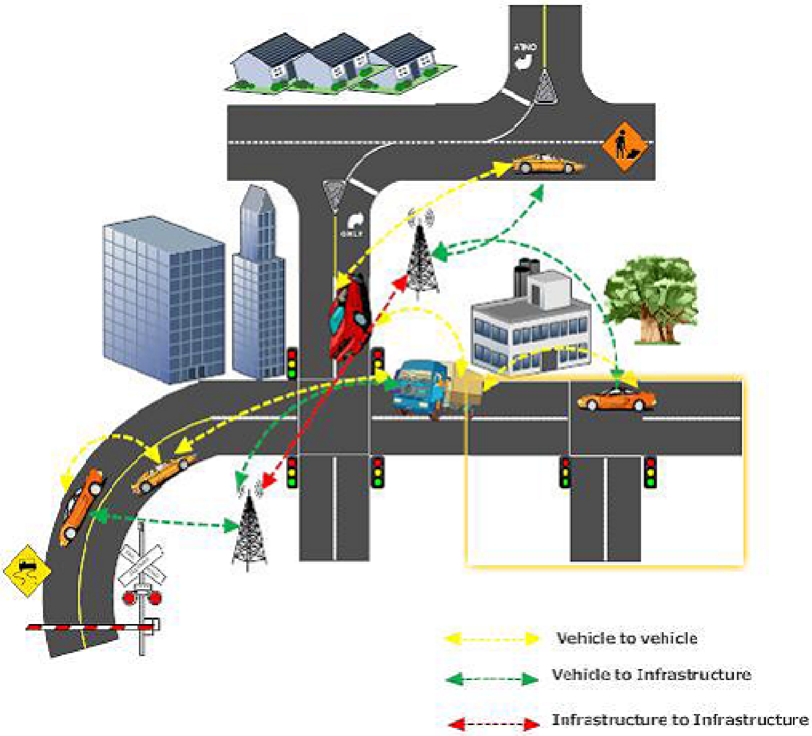

Figure 1.1: Vehicular Ad Hoc Network Overview[6]

1.2 Architecture of VANET

Reference architecture for vehicular network is proposed within the C2C-CC, distinguishing between three domains: in-vehicle, ad hoc, and infrastructure domain [1]. Figure 1.2 illustrates this reference architecture.

The in-vehicle domain refers to a local network inside each vehicle logically composed of two types of units: (i) an on-board unit (OBU) and (ii) one or more application unit(s) (AUs). An OBU is a device in the vehicle having communication capabilities (wireless and/or wired), while an AU is a device executing a single or a set of applications while making use of the OBU's communication capabilities. Indeed, an AU can be an integrated part of a vehicle and be permanently connected to an OBU. It can also be a portable device such as a laptop or PDA that can dynamically attach to (and detach from) an OBU. The AU and OBU are usually connected with a wired connection, while wireless connection is also possible (using, e.g., Bluetooth, WUSB, or UWB). This distinction between AU and OBU is logical, and they can also reside in a single physical unit.

The ad hoc domain is a network composed of vehicles equipped with OBUs and road side units (RSUs) that are stationary along the road. OBUs of different vehicles form a mobile ad hoc network (MANET), where an OBU is equipped with communication devices, including at least a short range wireless communication device dedicated for road safety. OBUs and RSUs can be seen as nodes of an ad hoc network, respectively, mobile and static nodes. An RSU can be attached to an infrastructure network, which in turn can be connected to the Internet. RSUs can also communicate to each other directly or via multihop, and their primary role is the improvement of road safety, by executing special applications and by sending, receiving, or forwarding data in the ad hoc domain.

Two types of infrastructure domain access exist: RSU and hot spot. RSUs may allow OBUs to access the infrastructure, and consequently to be connected to the Internet. OBUs may also communicate with Internet via public, commercial, or private hot spots (Wi-Fi hot spots). In the absence of RSUs and hot spots, OBUs can utilize communication capabilities of cellular radio networks (GSM, GPRS, UMTS, WiMax, and 4G) if they are integrated in the OBU.

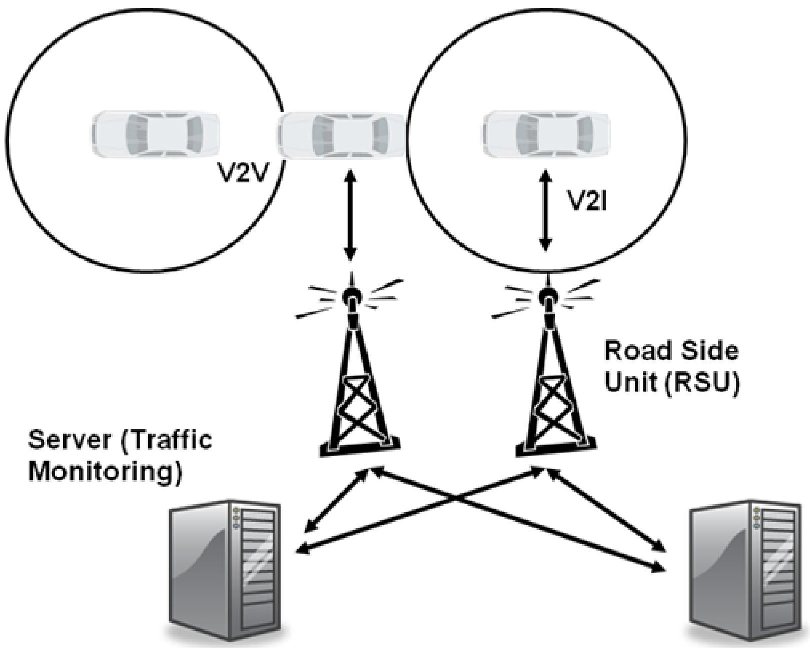

Figure 1.2: VANET Architecture

1.3 Characteristics of Vehicular Networks

Vehicular networks have special behavior and characteristics, distinguishing them from other types of mobile networks[2]. In comparison to other communication networks, vehicular networks come with unique attractive features, as follows:

- Unlimited transmission power
  Mobile device power issues are usually not a significant constraint in vehicular networks as in the case of classical ad hoc or sensor networks, since the node (vehicle) itself can provide continuous power to computing and communication devices. Power can be drawn from on-board batteries and recharged as needed from a gasoline or alternative fuel engine.

- Higher computational capability
  Indeed, operating vehicles can afford significant computing, communication, and sensing capabilities. Vehicles are orders of magnitude larger in size and weight compared to traditional wireless clients and can therefore support significantly heavier computing (and sensorial) components. This, combined with plentiful power, means vehicular computers can be larger, more powerful, and equipped with extremely large storage (up to terabytes of data), as well as powerful wireless transceivers capable of delivering wire-line data rates.

- Predictable mobility
  Unlike classic mobile ad hoc networks, where it is hard to predict the nodes' mobility, vehicles tend to have very predictable movements that are (usually) limited to roadways. Roadway information is often available from positioning systems and map based technologies such as GPS. Given the average speed, current speed, and road trajectory, the future position of a vehicle can be predicted.

1.4 Challenges of VANET

Following are tge challenges of VANET[3]

- Potentially large scale
  Unlike most ad hoc networks studied in the literature that usually assume a limited network size, vehicular networks can in principle extend over the entire road network and so include many participants.

- High mobility
  The environment in which vehicular networks operate is extremely dynamic, and includes extreme configurations: on highways, relative speeds of up to 300 km/h may occur, while density of nodes may be 1-2 vehicles 1 km on low busy roads. On the other hand, in the city, relative speeds can reach up to 60 km/h and nodes' density can be very high, especially during rush hour.

- Partitioned network
  Vehicular networks will be frequently partitioned. The dynamic nature of traffic may result in large inter vehicle gaps in sparsely populated scenarios, and hence in several isolated clusters of nodes.

- Network topology and connectivity
  Vehicular network scenarios are very different from classic ad hoc networks. Since vehicles are moving and changing their position constantly, scenarios are very dynamic. Therefore the network topology changes frequently as the links between nodes connect and disconnect very often. Indeed, the degree to which the network is connected is highly dependent on two factors: the range of wireless links and the fraction of participant vehicles, where only a fraction of vehicles on the road could be equipped with wireless interfaces.

- Broadcasting
  Broadcasting in VANET should be reliable with very low latency. Broadcasting is a common operation in a network to resolve many issues. In a mobile ad hoc network (MANET) in particular, due to host mobility, such operations are expected to be executed more frequently (such as finding a route to a particular host, paging a particular host, and sending an alarm signal). Because radio signals are likely to overlap with others in a geographical area, a straightforward broadcasting by flooding is usually very costly and will result in serious redundancy, contention, and collision, to which is referred as the broadcast storm problem.

- Routing
  High mobile Ad-Hoc network makes the routing difficult in VANET. VANET suffers from several internal and external factors of its dynamic nature. Internal factors include dynamic and highly movement of mobile nodes, frequent changes in network topology etc. External factors include impact of outside environment on network such as roads layout in city and interference of obstacles such as building, railway crossing etc. To overcome these internal and external issues several routing approaches have been proposed.

- Security

  1. Anonymity to prevent car tracking.
  2. Authenticity to prevent false alarms
  3. Integrity to prevent hacking
  4. Accountability to ensure punishment for bad actions

1.5 The Need of Vehicular Communication

1. Avoids accidents by giving the advance warning, like lane change, to the drivers so that they may become alert.

2. Provides basic information like traffic conditions, amenities or the path to be taken from source to destination.

3. Send warning messages and alerts like accidents happened, slippery roads etc.

4. Avoids traffic jams so that the drivers may take a different route.

5. Increases the traveler safety and comfort by providing them the information about the parking spaces, restaurants, stations from where they can download movies, songs, video etc.

6. Decreases traveling time and fuel consumption by guiding the drivers about the different options available of the route to destination.

1.6 Applications of VANET

The three major classes of applications possible in VANET are safety oriented, convenience oriented and commercial oriented. Safety applications will monitor the surrounding road, approaching vehicles, surface of the road, road curves etc. They will exchange messages and co-operate to help other vehicles out under such scenario. Though reliability and latency would be of major concern, it may automate things like emergency braking to avoid potential accidents. Convenience application will be mainly of traffic management type. Their goal would be to enhance traffic efficiency by boosting the degree of convenience for drivers. Commercial applications will provide the driver with the entertainment and services as web access, streaming audio and video. Figure 3 shows the schematic representation of VANET.

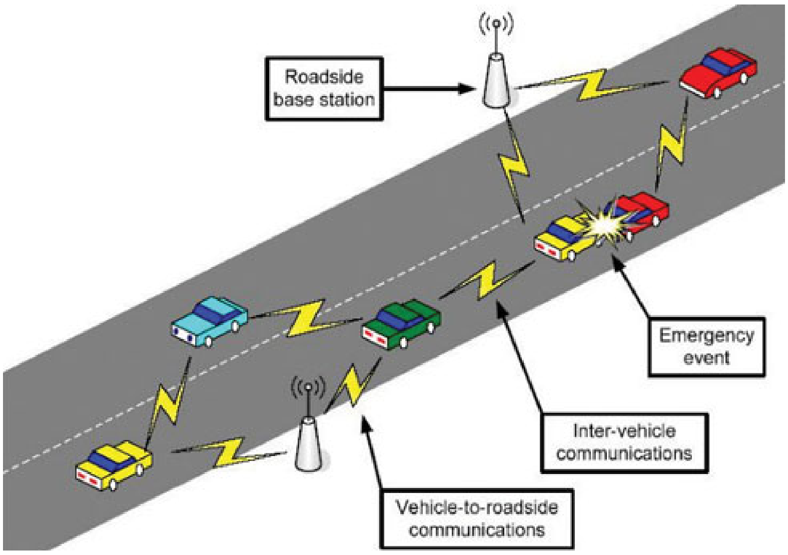

Figure 1.3: Schematic Representation of a Vehicular Adhoc Network

1.7 Classification

1.7.1 Safety Application

Safety applications would be Slow/Stop Vehicle Advisor (SVA) in which a slow or stationary vehicle will broadcast warning message to its neighborhood[4]. Another similar type of application is emergency electronic brake-light (EEBL). In Post Crash Notification (PCN), a vehicle involved in an accident would broadcast warning messages about its position to trailing vehicles so that it can take decision with time in hand as well as to the highway patrol for tow away support. Road Hazard Control Notification (RHCN) deals with cars notifying other cars about road having landslide. Another related application would be road feature notification which deals with notification due to road curve, sudden downhill etc. Cooperative Collision Warning (CCW) alerts two drivers potentially under crash route so that they can mend their ways.

1.7.2 Convenience Application

Congested Road Notification (CRN) detects and notifies about road congestions which can be used for route and trip planning. TOLL is yet another application for vehicle toll collection at the toll booths without stopping the vehicles. Parking Availability Notification (PAN) helps to find the availability of slots in parking lots in a certain geographical area.

1.7.3 Commercial Application

Remote Vehicle Personalization / Diagnostics (RVP/D) helps in downloading of personalized vehicle settings or uploading of vehicle diagnostics from/to infrastructure. Service Announcements (SA) would be of particular interest to roadside business like petrol pumps, highways restaurants to announce their services to the drivers within communication range. Content Map Database Download (CMDD) acts as a portal for getting valuable information from mobile hotspots or home stations. Using Real Time Video Relay (RTVR), on-demand movie experience will not be confined to the constraints of the home and the driver can ask for real time video relay of his favorite movies.

1.8 Wireless Access Methods in VANET

1.8.1 DSRC/WAVE

Dedicated short-range communication (DSRC) is a short- to medium-range communication technology operating in the 5.9 GHz range[5]. The Standards Committee E17.51 endorses a variation of the IEEE 802.11a MAC for the DSRC link. DSRC supports vehicle 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 (wireless access in a vehicular environment). DSRC has two modes of operations: (1) ad hoc mode characterized by distributed multihop networking (vehicle-vehicle), (2) infrastructure mode characterized by a centralized mobile single-hop network (vehicle-gateway).

1.8.2 Cellular Networks

Cellular systems have been evolving rapidly to support the ever increasing demands of mobile networking. 2G systems such as IS-95 and GSM support data communications at the maximum rate of 9.6 kb/sec. To provide 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)[6]. Now 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.

1.8.3 WiMAX/802.16e

802.16e or WiMAX (Worldwide Interoperability for Microwave Access) aims at enabling the delivery of last mile wireless broadband access (¡40 Mb/sec) as an alternative to cable and DSL, thus providing wireless data over long distances. This will fill the gap between 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. It also supports several service levels including guaranteed Quality of Service (QoS) for delay sensitive applications, and an intermediate QoS level for delay tolerant application that requires a minimum guaranteed data rate.

1.8.4 WLAN

WiFi or WLAN can also support broadband wireless services. 802.11 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 to support broadband wireless services. It has long been used as a means of Internet access in vehicles, known as Wardriving. Also, open WiFi mesh networking has received a lot of attention.

2

Literature Survey of Broadcasting in VANET

Due to the high mobility of vehicles, the distribution of nodes within the network changes very rapidly and unexpectedly that wireless links initialize and break down frequently and unpredictably[11]. Taking into consideration that VANET operates in the absence of servers, force OBUs to organize network resources distributive. Thereupon, broadcasting of messages in VANETs plays a crucial rule in almost every application and requires novel solutions that are different from any other form of Ad-Hoc networks. Broadcasting of messages in VANETs is still an open research challenge and needs some efforts to reach an optimum solution.

Broadcasting requirements are: high reliability and high dissemination speed with minimum latency in single-hop as well as multi-hop communications. Problems associated with regular broadcasting algorithms are: the high probability of collision in the broadcasted messages, the lack of feedback and the hidden node problem. In VANETs, there are two types of collisions, collisions of wireless messages in the network domain and the physical collisions of running vehicles. Throughout this work, the default type of collision is the collision between messages in the network domain except what is explicitly said as a vehicular collision[8].

Although broadcasting has a limited usage in Ethernet and MANET (e.g. a DHCP 'Dynamic Host Configuration Protocol' request), it has got a wider range of implementation in VANET applications. Almost all VANET applications depend on sending messages to intended vehicles without explicitly determining their identity, which is a broadcast in its nature. Note that, all signaling techniques that are currently deployed in vehicles (e.g. brake lights and turning right / left lights) are considered a broadcast. With VANET technology, these signals will be exchanged directly between vehicles themselves. This will increase the driver awareness of the road and the traveling luxury as well. In this chapter, I will discuss previous promising contributions in broadcasting protocols in VANET environments.

2.1 Broadcasting Goals

Any broadcasting protocol should satisfy the following goals-

- High reliability: The sender should be acknowledged of delivering the message to the intended vehicle(s).

- Low latency: The time duration from the first attempt of transmission to the end of the broadcasting phase, should be as small as possible.

- Low probability of collision: The protocol should suffer from as minimum collisions as possible, hence, higher reliability and lower latency.

- Hidden node problem: Collisions at receiving nodes due to hidden node problem should be avoided.

- No prior control messaging: In VANET environment, the broadcasting protocol should be highly distributive and does not need any prior information. It is assumed that each node in the network knows nothing about type or exact location of the other nodes.

- Human factors: Mobile nodes in VANET are controlled by human drivers, which mean that node movement is not completely random and it can be expected to some extent.

- High robustness: Location of nodes changes very rapidly and unpredictably (to some extent). Thus, any VANET protocol should be robust at different speeds, different traffic volumes and different environments (rural or urban).

- Different applications: Broadcasting in VANET services many categories of applications. Thus, the deployed protocol should cope with the differences in these applications.

2.2 Categories of Broadcasting Protocols

All of these contributions try to solve just two questions; the first one is "How to deliver the broadcast message to nodes within a single communication range with the highest possible reliability?" which will be designated as reliable protocols. The second one is "How to deliver the broadcast message to the entire network?" which will be designated as dissemination protocols. Although both questions look similar to each other, the first one is used with applications related to direct neighbors (e.g. collision avoidance) and the second is used with applications related to the entire network (e.g. traffic management)[7]. Published reliable protocols use three methods: 'Rebroadcasting' where the transmitter node retransmits the same message for many times, 'Selective ACK' where the transmitter requires ACK from a small set of the neighbors, and 'Changing parameters' where the transmitter changes transmission parameters according to the expected state of the network. Published dissemination protocols use two methods: 'Flooding'[15]

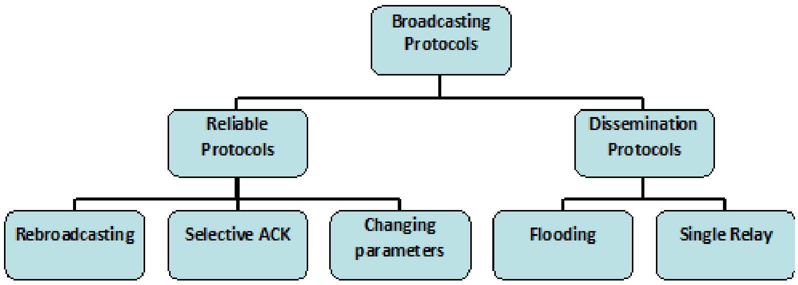

Figure 2.1: Categories of Broadcasting Protocols in VANET

where each node is responsible for determining whether it will rebroadcast the message or not, and 'Single relay' where the transmitter is responsible for determining the next hop node.

Published dissemination protocols use two methods: 'Flooding' where each node is responsible for determining whether it will rebroadcast the message or not, and 'Single relay' where the transmitter is responsible for determining the next hop node.

2.3 Various Strategies

VANET broadcast issues adapt different strategies which fall into various categories including distance based, location based, probability based and topology based.

2.3.1 Distance Based

The idea here is that the node with the greatest distance is the next relaying hop. To implement it, each node has its own timer. Timer will start when any message is received. The duration of the timer is inversely proportional to the relative distance between itself and the broadcaster. When the timer is expired, it will relay the message. To suppress the number of messages, if a node hears the redundant message, meaning that other nodes have already relayed the message, it will stop its attempt. Thus, the node with the furthest distance to the broadcaster shall be the next relaying hop because its transmission waiting time is the shortest. Lowest message propagation delay has been achieved due to the minimum number of hops used but the reception probability is a potential problem because of the large distance.

2.3.2 Location Based

This approach is very similar to distance based. The difference is that: rather than using just the distance, Global Position Devices (GPS) may be used to get the area, map and other useful information as inputs to choose the next relaying hops. The idea here is to divide the road portion into multiple cells. These cells have a basic requirement that vehicle's transmission radius must cover "at least" its adjacent node. When a node hears the message, the index of cell will be used in the calculation instead of the plain distance. Furthermore, if a map is known, a vehicle located at an intersection will have higher priority of retransmission since their transmission covers larger area.

2.3.3 Probability Based

Probability is usually used to reduce the number of transmitted messages when there are too many contenders and hence it reduces the chances of collision. The same methodology has been applied to VANET. While some of the works used the fixed static number, many of which successfully adapted the adaptive number.

Weighted p persistence is the fundamental idea of probability usage in VANET. Upon message reception, instead of using delay timer, each vehicle will retransmit with probability p. However, using equal p to all vehicles is inappropriate since vehicle with further distance should have higher priority. Thus, the retransmit probability is proportional to the distance with respect to the broadcaster. Therefore, there is a higher chance that node whose distance to broadcaster is greater will relay the message.

While using probability requires no overhead and simple to implement, the selection of p can drastically affect overall performance. On one hand, too small p could cause everyone to remain silent and the message cannot be further propagated or will be delayed if there are other backup mechanisms. On the other hand, too large p will cause message collisions and will be a waste of channel resources.

In short, the selection of p must be carefully chosen. The important factor is the vehicle density. Theoretically, when there are n nodes, the probability should be 1/n because, mathematically, there will be only one node that will retransmit. Therefore, according to the theory, if the density is low, then we select large p so that there is someone to relay and if the area is dense with vehicles, small p would be more prefer to avoid unnecessary transmission. The density along with the distance is the input to derive p for each vehicle.

2.3.4 Topology Based

The next idea is graph topology. Topology based approach is the solution adapted widely in wireless sensor network. Since the change of topology in VANET is rapid, most of the time, topology solution is out of question. However, in some scenarios such as freeway where the road is straightforward, it can still be useful.

All nodes create the backbone topology in this case and the criteria of selecting backbone nodes are speed and distance. The backbone nodes will be given responsibility of relaying. Since the topology is created and the relaying path has been established, there will be no contention or collision. However, the process of retaining and creating the backbone is always expensive and sophisticated in high velocity network such as VANET.

2.4 Broadcasting Protocols

There are many reliable broadcasting protocols available. Here I have taken the description of few ones-

2.4.1 Urban Multi-hop Broadcast protocol(UMB)

UMB is designed to address[16] (i) broadcast storm, (ii) hidden node, and (iii) reliability problems in multi-hop broadcast. The UMB protocol is composed of two phases, namely directional broadcast and intersection broadcast. In this protocol a new directional broadcast method is introduced where sender nodes try to select the furthest node in the broadcast direction to assign the duty of forwarding and acknowledging the packet without any apriori topology information i.e., sender selects the furthest node without knowing the ID or position of its neighbors. At the intersections, to disseminate the packets in all directions, it proposes installing repeaters that forward the packet to all road segments.

The most important goals of this protocol are as follows:

1. Avoiding collisions due to hidden nodes: In order to decrease the effect of hidden nodes, a mechanism similar to RTS/CTS handshake in point-to-point communication is employed by our new UMB protocol.

2. Using the channel efficiently: Forwarding duty is assigned to only the furthest vehicle in the transmission range without using the network topology information.

3. Making the broadcast communication as reliable as possible: To achieve the reliability goal, an ACK packet is sent by the vehicle which was selected to forward the packet.

4. Disseminating messages in all directions at an inter section: New directional broadcasts are initiated by the simple repeaters installed at the intersections according to the Intersection Broadcast mechanism.

2.4.2 Edge-aware epidemic protocol(EAEP)

Each of vehicles has its own geographical knowledge which is piggybacked to broadcast messages[9]. By this solution, EAEP operates without beaconing[14]. Upon receipt of a new rebroadcast message, EAEP uses number of transmission from front nodes and back nodes in amount of time to calculate the probability for making decision whether nodes will rebroadcast the message or not. By this mechanism, at the edge of each transmission will be preferred area to rebroadcast messages. But EAEP does not address the intermittent-connectivity issue. Specifically, a node would by no means know whether it has missed some messages its new neighbors have or its neighbors have missed some messages it has. EAEP outperforms the simple flooding in terms of reliability and overhead. However, it provides slow speed of data dissemination. (According to the simulation results shown in, it takes more than 30 seconds to deliver a broadcast message to the majority of vehicles)[9].

2.4.3 Preferred group broadcast(PGB)

PGB is not a reliable broadcasting protocol but it is a solution to prevent broadcast storm problem from route request broadcasting in AODV. Each node in PGB will sense the level of signal strength from neighbor broadcasting. The signal strength is used for waiting timeout calculation. Nodes in the edge of circulated broadcast will set shorter waiting timeout. Only node with shortest timeout will rebroadcast the message. PGB can reduce numbers of RREQ broadcasting. But there exists a problem on low density area[18].

2.4.4 AckPBSM

AckPBSM is a modified version of PBSM, which is a parameter less broadcast in static to highly mobile ad-hoc networks. It uses periodic beacons for exchanging information between nodes. Nodes at the head or tail of each vehicles cluster will set shorter waiting timeout as they are preferred to do the next rebroadcasting. These nodes are in Connected Dominating Set (CDS) or called gateway nodes. Other nodes that are not in CDS will set longer waiting timeout. While the nodes are waiting, they hear from beaconing if their neighbors have already received the messages. If all of their neighbors received the messages, the nodes will not perform any rebroadcast. So AckPBSM need high frequent of beacon to accurately operate. To address intermittent connectivity issue, acknowledgements of broadcast messages are piggybacked in periodic beacons so nodes can rebroadcast only if their neighbors have not received the broadcast messages. It is reported in that AckPBSM outperforms PBSM and DVCAST in terms of reliability and overhead[9].

2.4.5 Density-aware reliable broadcasting protocol(DECA)

It does not require position knowledge. DECA employ only local density information of 1-hop neighbors obtained by beaconing[12]. Before broadcasting, a node selects one neighbor which has the highest local density information to be the next rebroadcast node. Other nodes will randomly set their waiting timeout. If they do not hear anyone rebroadcast the message before the timeout expiration, they will rebroadcast the message. Furthermore, identifiers of the received broadcast messages are included into periodic beacons so that a node can discover its neighbors, which have not received the messages and consequently rebroadcast the messages for those neighbors. The advantage of DECA is it does not require position knowledge to operate so it is more flexible to suit any operating environment.

2.4.6 Position-aware reliable broadcasting protocol (POCA)

It uses adaptive beacon to get neighbors' position and velocity[13]. When nodes want to broadcast messages, they will select the neighbors in preferred distance to rebroadcast the message. The preferred distance is based on the distance between nodes and selector nodes. The selected node will rebroadcast the message immediately. In case the selected nodes do not rebroadcast the message, other nodes which have set waiting timeout since they received message will do this task instead. The waiting timeout is calculated depend on the distance between node and precursor node. So a node that is closest to selected node will rebroadcast the messages. POCA also piggybacks the message identifier to beacon to handle intermittent connectivity. Nodes can know if the neighbors miss some messages and rebroadcast the message to them by set waiting timeout. So a node in the same road section will rebroadcast the messages to neighbors.

3

Proposed Protocol

3.1 Motivation

As quoted from the IEEE 802.11 standard [4], "There is no MAC-level recovery on broadcast or multicast frames. As a result, the reliability of this traffic is reduced, relative to the reliability of directed traffic, due to the increased probability of lost frames from interference, collisions, or time-varying channel properties."

Although the probability of collisions may be dropped down using the RTS/CTS mechanism, the 802.11 standard[19] says that, "The RTS/CTS mechanism cannot be used for messages with broadcast and multicast immediate destination because there are multiple recipients for the RTS, and thus potentially multiple concurrent senders of the CTS in response." As a result, the area of reliable broadcasting is still an open research challenge and needs some new innovations.

3.2 UMB Protocol

This scheme introduces an alternative to the RTS/CTS, the Request to Broadcast (RTB) and the Clear to Broadcast (CTB). Only the transmitter and farthest node from the transmitter exchange the RTB/CTB messages. When a node has a broadcast message to send, it transmits a RTB[16].

The network is being iteratively divided into segments to determine the farthest node from the broadcaster. To determine the farthest node, each node transmits a black burst. When a node receives a RTB, each node computes the length of the black burst based on their distance from the sender.

The length of the black-burst is computed as follows:

$$L = (\frac{d}{range} \times N_{max}) \times SlotTime \tag{3.1}$$

Where
L= Jamming Signal Duration
d= distance between the source and receiver
Range= RTB sender transmission radius

N_{max}= Maximum no. of cells
SlotTime= jamming duration for one slot

Each node will then simultaneously start transmitting a black-burst. If a node finishes transmitting the black-burst and hears no others sending the black burst on the medium, it affirms that it is the farthest node. A CTB is then transmitted to complete the reservation of the channel.

On the other hand, if two or more nodes determine that they are the farthest away, a collision will occur when the CTB is sent. In this case, the RTB is retransmitted to the furthest non-empty segment and that segment is divided into N_{max} sub-segments. This process continues until one of the nodes CTB succeeds. The iterative black-burst is calculated as follows:

$$L_i = \left\lfloor \frac{(d - Llongest_{i-1} * W_{i-1})}{W_{i-1}} * N_{max} \right\rfloor * SlotTime \tag{3.2}$$

$$i = 2, 3, ...D_{max}$$

$$W_i = \frac{Range}{N_{max}}$$

Where
Llongest is the longest black-burst
W_i is the segment width for the i_{th} iteration

After a node is selected to forward the broadcast, the sender transmits the frame to the receiver. Collisions are avoided because the surrounding nodes overhear the RTB/CTB exchange and defer from accessing the channel. The receiver of the broadcast then sends back an ACK to indicate that the frame was successfully received. The receiver continues the process of relaying the broadcast message. The figure on the next page shows the RTB/CTB message scheme of UMB protocol-

3.2.1 Limitations of UMB Protocol

In terms of reliability, RTB/CTB is the only protocol that can guarantee the successful reception. However, RTB/CTB suffers a lot from the slow process due to various reasons which are:

1. The protocol may end up in multiple iterations if the node density is high. Therefore, there will be a waste of time in the process of selecting new relaying hop in just one round.

2. As the longest jamming duration is used in this protocol to find the next broadcaster but the jamming signal duration is of variable length which may result into slow message propagation time.

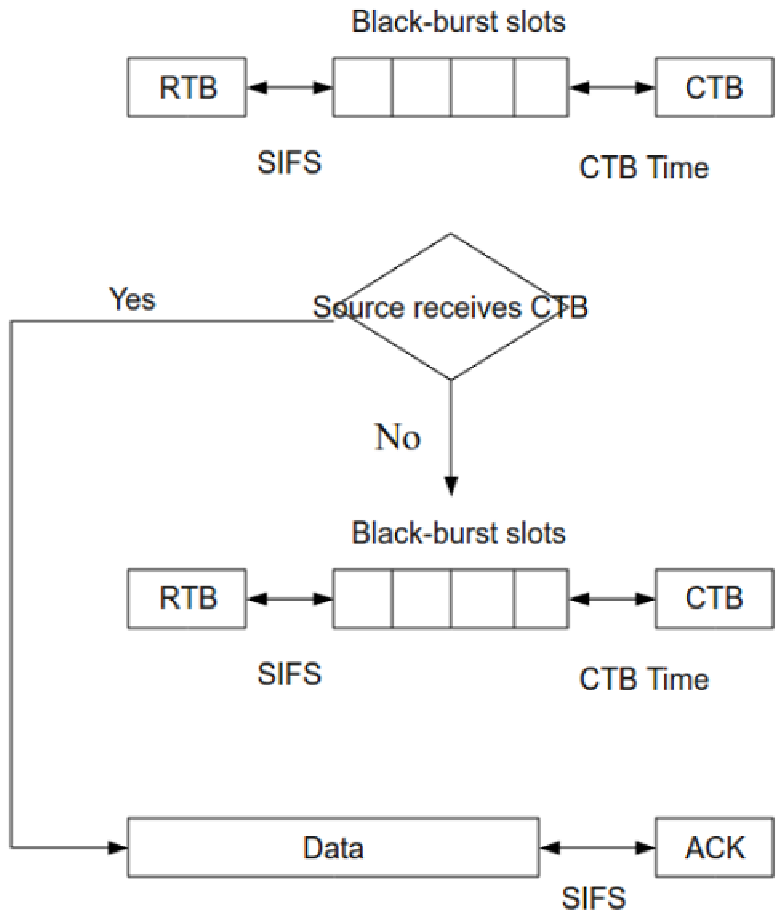

Figure 3.1: RTB/CTB Message Scheme in UMB

3.3 Slotted p-Persistence Broadcasting

Author has introduced three novel distributed broadcast suppression techniques i.e., weighted p-persistence, slotted 1-persistence, and slotted p-persistence schemes[17].

Slotted p-Persistence Broadcasting Rule says that

"Upon receiving a packet, a node checks the packet ID and rebroadcasts with the pre-determined probability p at the assigned time slot TS_{ij}, if it receives the packet for the first time and has not received any duplicates before its assigned time slot; otherwise, it discards the packet."

While UMB uses a black-burst (channel jamming signal) contention approach to determine the farthest vehicle in the transmission range (essentially a MAC-based approach), this protocol employs a similar distance-based suppression technique, aims to reduce the load submitted from the network layer to the data link layer (as opposed to modifying the MAC layer) by combining the probabilistic broadcast technique with timer based suppression. In addition to reducing the overhead, the mechanism we propose also guarantees that all vehicles receive the broadcast message if the network is fully connected.[20]

Basically it is a probability based broadcasting, based on both distance and probability in the selection of next relaying hop. In this model, the road will be divided into multiple cells. Each cell will be allocated with different time slot. The further one will be given lower number of time slot meaning that vehicles located in this cell would have a chance to retransmit message first. To reduce the collision probability and redundant messages, each node will retransmit with fixed probability p.

Each node in this scheme should buffer the message for a certain period of time (e.g., [N, - 1]x WAIT-TIME + d ms) and retransmits with probability 1 if nobody in the neighborhood rebroadcasts in order to prevent the message "die out". Figure below illustrates the concept of the slotted p-persistence with 4 slots. Similar to the p-persistence case, the performance of this scheme also depends on the value chosen for the re-forwarding probability p.

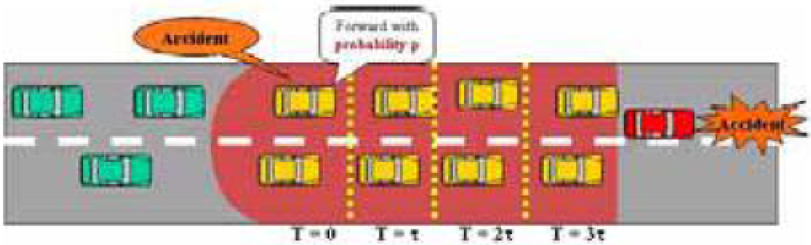

Figure 3.2: Slotted p-persistence Scheme

As seen in figure, the source is on the right side. After it transmits report, each vehicle will calculate its corresponding cell index. The delay timer is set by the following formula.

$$Ts_{ij} = S_{ij} \times D_{ij} \tag{3.3}$$

$$S_{ij} = N_s \times (1 - \frac{D_{ij}}{R}) \tag{3.4}$$

Where
D_{ij}= Distance between node i and j
R= transmission radio of broadcaster
N_s= Maximum no. of slots
S_{ij}= Slot index
Ts_{ij}= Delay time

The vehicles will know which cell they belong to according to the formula. They will set the delay according to the cell index properly. In the figure, vehicles on left most side (cell = 0) which are farthest away from the source rebroadcast immediately with probability p as soon as they get the message. Meanwhile, other vehicles on the other cells would set their timer depending on their respective cell number. If they hear the redundant message while their timers are still counting down, it means that some vehicles have relayed the message already. In response, they will simply cancel the timer stopping their attempt to retransmit. In contrast, if their timers have expired, it means that no one has successfully relayed the message.

Regardless of the reasons, if the delay timer of the vehicles is expired, they will retransmit with probability p. This process will go on for every other cell. In order to assure that there will be some nodes who relay the message, there is a backup plan in slotted p persistence model.

Any node decides not to retransmit must set its delay timer to (cell + 1) slottime meaning that it will try retransmitting again in the next slot time. However, this time, the transmission probability will be set to 1, a 100% probability.

3.3.1 Problems with slotted p-persistence broadcasting

The p-persistence model has the following problems.

- Nodes send large size messages which may lead to collisions and wastage of bandwidth.
- As the network is divided into cells, The nodes in each cell are allotted time slot that are reserved from the farthest one. The time slot will go waste if there is no vehicle in the previous cell.

3.4 Proposed protocol

The proposed protocol is based on UMB and slotted p- persistence broadcasting protocols. I have taken the best features of both the protocols and has proposed a new one. The reason of selecting RTB/CTB as the base protocol is because it is the only protocol that guarantees the successful reception. Whereas the slotted p-persistence scheme has been used to find the next broadcaster in the network.

3.4.1 Features

The proposed protocol has the following features-

1. Reliable transmission in the protocol will be achieved by using RTB/CTB scheme as in UMB protocol.
2. Multiple iterations will not be there when the vehicle density is high. Therefore hidden node problem will be solved only using jamming signal.
3. Jamming duration will be static and short fixed period.
4. Slotted p persistence scheme will be used to find the next relaying hop.
5. Jamming signal will be used to estimate the distance to the closest and furthest node. With this information, there will be no wasted slot since all nodes know a number of slots and a radius that are adjusted.

6. The node in the last cell will set a delay timer to send in the next slot with probability equal to 1.

3.4.2 Protocol Design

1. As in IEEE 802.11 standard after waiting for SIFS, the source transmits RTB when it want to send something. The source also mark a timestamp what the sending time is.

2. Nodes send their black-burst in the shortest possible time (SIFS) after they hear the RTB packet.

3. When the source first hears the jamming signal, it will calculate the estimated distance to the closest node.

4. Once the jamming signal transmission stop, the source will calculate the estimated distance to the furthest node.

5. The source will transmit the report as soon as it hear CTB from all the other nodes.

6. After every node hears the report, the contention period will begin based on slotted p-persistence model. hence the next broadcaster will be based on the probability calculated.

4

Simulation Tools

To simulate the protocols, I have used ns-2.34 and SUMO. SUMO has been used to generate the road topology and mobility patterns. Here is the description of both the simulators-

4.1 Network Simulator - NS

4.1.1 Introduction

NS is Object-oriented Tcl (OTcl) script interpreter that has a simulation event scheduler and network component object libraries, and network setup module libraries.To setup and run a simulation network, a user should write an OTcl script that initiates an event scheduler, sets up the network topology using the network objects and the plumbing functions in the library, and tells traffic sources when to start and stop transmitting packets through the event scheduler.

NS is a discrete event simulator targeted at networking research. NS provides substantial support for simulation of TCP, routing, and multicast protocols over wired and wireless (local and satellite) networks. Another major component of NS beside network objects is the event scheduler. An event in NS is a packet ID that is unique for a packet with scheduled time and the pointer to an object that handles the event. In NS, an event scheduler keeps track of simulation time and fires all the events in the event queue scheduled for the current time by invoking appropriate network components, which usually are the ones who issued the events, and let them do the appropriate action associated with packet pointed by the event.

Figure 4.1 shows the general architecture of NS. In this figure a general user can be thought of standing at the left bottom corner, designing and running simulations in Tcl using the simulator objects in the OTcl library. The event schedulers and most of the network components are implemented in C++ and available to OTcl through an OTcl linkage that is implemented using tclcl. The whole thing together makes NS, which is a OO extended Tcl interpreter with network simulator libraries. NS uses two languages C++ to implement protocols and Object Oriented Tcl (OTcl) to write simulation scripts. NS is an OTcl script interpreter that has a simulation event scheduler and network libraries. So we write an Otcl script, which initiates an event scheduler. In the script we define the topology of the network using objects and functions from the libraries.

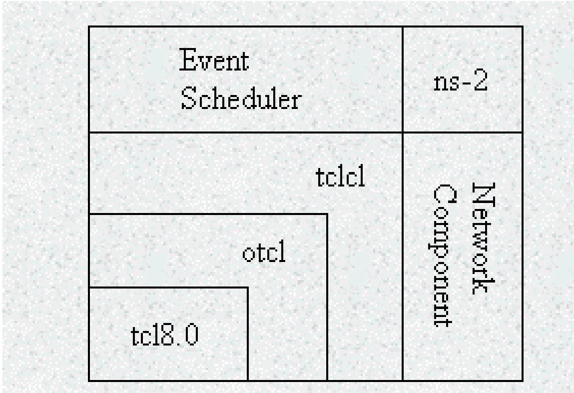

Figure 4.1: Architectural View of NS[21]

Traffic is added to the network and told when to commence and when to end. This is handled by the event scheduler, as is the termination of the simulation. The results from the interpreter can then be analysed or graphically displayed using nam. Without getting to complicated data path implementations are written and compiled by C++ in order to save processing time. The event scheduler and most network components are developed in C++ to give us the NS simulation. The Tclcl is the Tcl/C++ interface between OTcl and C++.

In order to enhance the functionality of NS extensions are added. These extensions are modifications to the C++ or OTcl source code. Alternatively these may be completely new agents or traffic sources etc. As an example the event scheduler and network component object classes are located in the NS-2 directory. In the NS-2 directory are UDP.h and UDP.cc, which are the C++ files used to implement the UDP agent.

Similarly the tcl/lib directory contains the OTcl source code for node, links etc. To patch Ns to work with a new or modified agent means the creation of C++ code, which will involve the creation of a class and possibly a header file. A linkage has to be set up within the C++ class to ensure the OTcl can use instances of the C++ class. Dependent on what is being written, a header file may have to be registered in the packet.h and ns-packet.tcl.

Methods may have to be added to existing classes. Once these modifications are made the makefile will have to be modified to add yourclass.o to the object file list. Then run make clean and make depend before recompiling using make. This will patch the system to use your new or modified class.

4.1.2 Components of NS

NAM It is a Tcl/TK based animation tool for viewing network simulation traces and real world packet traces. It supports topology layout, packet level animation, and various data inspection tools.

Trace Files It records each individual packet as it arrives,departs, or is dropped at a link or queue. Each lines consists of:

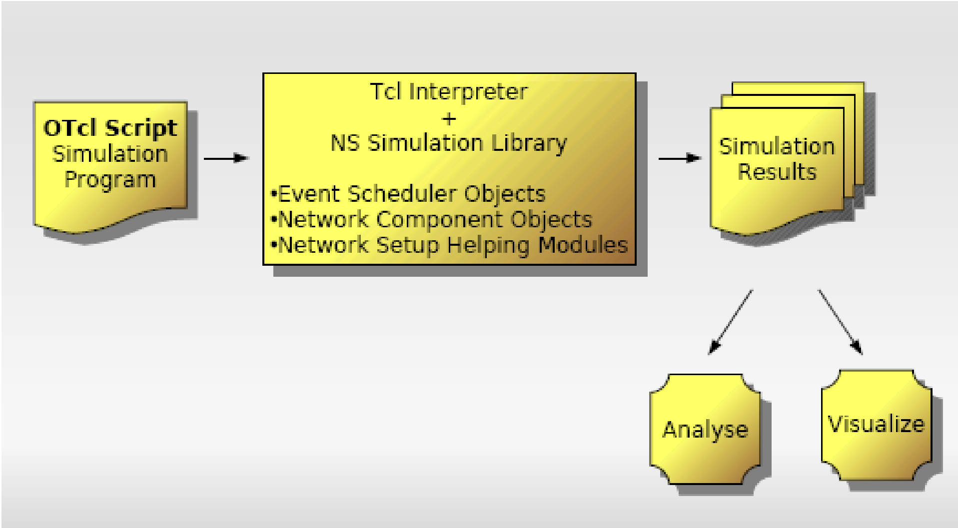

Figure 4.2: Simulation Process[21]

- Event Descriptor (+, -, d, r)
- Simulation time (in seconds) of that event
- From Node
- To Node, which identify the link on which the event occurred
- Packet type
- Packet size
- Flags (appeared as "——" since no flag is set). Currently, NS implements only the Explicit Congestion Notification (ECN) bit, and the remaining bits are not used.
- Source and destination address in forms of "node.port".
- The network layer protocol's packet sequence number.
- Flow id (fid)
- The last field shows the unique id of the packet.

GREP It is a UNIX command to filter a file.The name comes from "search globally for lines matching the regular expression and print them". GREP takes a regular expression on the command line, reads the standard input or list of files, and outputs the lines containing matches for the regular expression

AWK AWK is a general purpose computer language.AWK is designed for processing text-based data, either in files or data streams.The name AWK is derived from the surnames of its authors Alfred Aho, Peter Weinberger, and Brian Kernighan. To run AWK file

1. awk -f file1.awk file2.tr
2. awk f file1.awk file2.tr >out.txt

   file1.awk : is a command file file2.tr : is a primary input file out.txt : is an output file

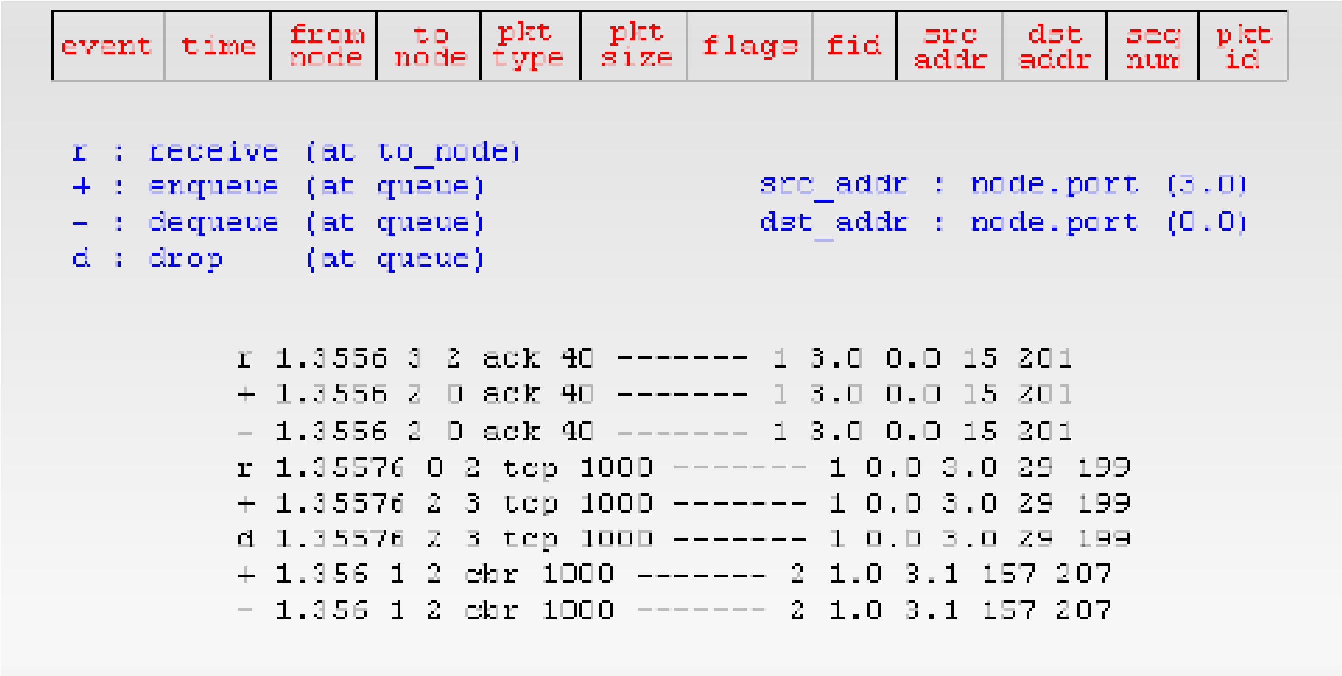

Figure 4.3: Trace File Format

A typical AWK program consists of a series of lines, each of them is on the form /pattern/ action

Pattern is a regular expression.

Action is a command.

Other line forms:

BEGIN { action } Executes action commands at the beginning of the script execution.

{ action } Executes action for each line in the input.

END { action } Executes action commands after the end of input.

4.2 Node Movement and Topology Generation

Instead of specifying and control each nodes' position and movement pattern, we use a CMU tool "setdest" to generate large number of nodes and there movements. The tool use a random waypoint model.

4.2.1 Creating node-movements for wireless scenarios (setdest)

The node-movement generator is available under ns/indep-utils/cmu-scen-gen/setdest directory and consists of setdest.cc,.h and Makefile. Setdest tool is used to generate the positions of nodes and their moving speed and moving directions. The syntax is:

setdest -v 1 -n $numnodes -p $pt -M $maxspeed -t $simtime -x $maxx -y $maxy

For example: setdest -v 1 -n 50 -p 0 -M 20 -t 900 -x 1500 -y 300 will generate a 1500*300 topology with 50 nodes random distributed labeled by a XY(Z) coordinates. After the initial position information, the nodes are specified with their movement destination and speed.

After that, the initial distance (hop counts) information are counted by a GOD. Currently, the god object is used only to store an array of the shortest number of hops required to reach from one node to an other. The god object does not calculate this on the fly during simulation runs, since it can be quite time consuming. The information is loaded into the god object from the movement pattern file.

Then, the nodes are going to move during this 900-second simulation scenario. During this, the distance (hop-counts) information is going to change, thus, the following lines are going to show this recent change. Note that the distance is calculated based on a nominal radio change as "250" meters. The god information should not be available to any of the node. Thus, for a routing protocol, it has to discover the distance by itself with some mechanism.

It's possible that a node reaches its destination before the simulation timer ends. Thus, it needs to re-specify a new direction and speed for it. Also, the average pause time is a parameter to allow a node stop to move in a destination before moving again.[**?**]

4.2.2 Traffic Pattern Generation (cbrgen)

Random traffic connections of TCP and CBR can be setup between mobile nodes using a traffic-scenario generator script. This traffic generator script is available under ns/indep-utils/cmu-scen-gen and is called cbrgen.tcl. It can be used to create CBR and TCP traffics connections between wireless mobile nodes. In order to create a traffic-connection file, we need to define the type of traffic connection (CBR or TCP), the number of nodes and maximum number of connections to be setup between them, a random seed and incase of CBR connections, a rate whose inverse value is used to compute the interval time between the CBR pkts. So the command line looks like the following:

ns cbrgen.tcl [-type cbr—tcp] [-nn nodes] [-seed seed] [-mc connections][-rate rate] >output.txt

The start times for the TCP/CBR connections are randomly generated with a maximum value set at 180.0s, thus the simulation time is at least 180 seconds. And the number of nodes has no relationship to the maximum number of connections (mc), we can have 10 nodes, also 10 connections as one node could have multiple simultaneous connections to other nodes. The parameter "rate" means how many packets per second, thus, for CBR traffic, the packet interval is the reversal of this parameter. And for TCP traffic, we don't have to specify rate, ftp connections are going to be used. the default packet size is 512B. Actually, we can change all this in the "output.txt" file. Note that there is no guarantee that the number of connections are created and the actual nodes involved as source and sink.[**?**]

4.3 Adding New Protocol in NS

First of all, make a directory with the name of your protocol in "$ NS_ROOT/ns-2.33/". In this directory basically we create three files : xyz.cc, xyz.h, xyz_packet.h. In the new header file 'xyz.h' first one have to declare the data structure for the new packet header which is going to carry the relevant data. The header file is used to define all necessary timers (if any) and routing agent which performs protocols functionality. The .cc file actually implements all timers, routing agent and Tcl hooks.[?]

To implement a routing protocol in NS2 you must create an agent by inheriting from Agent class. "Agents represent endpoints where network-layer packets are constructed or consumed, and are used in the implementation of protocols at various layers".This is the main class where to implement our routing protocol. In addition, this class offers a linkage with Tcl interface, to control our routing protocol through simulation scripts written in Tcl.

The Trace class is the base for writing log files with information about what happened during the simulation.

4.3.1 Agent

command method int command(int argc, const char*const* argv)

argv[0] contains the name of the method being invoked, argv[1] is the requested operation, and argv[2..argc-1] are the rest of the arguments which were passed. Within this function we must code some mandatory operations as well as any other operation that we want to make accesible from Tcl. Each case must finish its execution returning either TCL_OK (if everything was fine) or TCL_ERROR (if any error happened).A mandatory command that we always have to implement: "start". The expected behavior of this command is to configure the agent to begin its execution. Another mandatory command to implement is "port-dmux". NS stores a reference to every compiled object (C++ object) in a hash table to provide a fast access to each of them given its name. We make use of that facility to obtain a PortClassifier object given its name. Similarly, there is another mandatory operation called "tracetarget" which simply obtains a Trace object given its name.

recv() recv() is invoked whenever the routing agent receives a packet.

4.3.2 Needed Changes

Then we need to modify following files. Therefore it is better to take backup of these files before ones start adding new protocol, so that one can easily go back.

- $ NS-ROOT/Makefile
- $ NS-ROOT/queue/priqueue.cc
- $ NS-ROOT/common/packet.h
- $ NS-ROOT/trace/cmu-trace.h
- $ NS-ROOT/trace/cmu-trace.cc
- $ NS-ROOT/tcl/lib/ns-packet.tcl
- $ NS-ROOT/tcl/lib/ns-lib.tcl
- $ NS-ROOT/tcl/lib/ns-agent.tcl
- $ NS-ROOT/tcl/lib/ns-mobilenode.tcl

Packet Type declaration

A constant to indicate our new packet type, PT_XYZ is used in .cc file. We will define it inside file common/packet.h. We change PT_NTYPE to 63, and for our protocol PT_XYZ = 62. If you have already installed another routing protocol. Just make sure PT_NTYPE is last, and protocol number is ordered sequentially.

```
common/packet.h
// Protocol packet
static const packet_t PT_XYZ = 62;

// insert new packet types here
static packet_t       PT_NTYPE = 63; // This MUST be the LAST one
```

Just below in same file there is definition of p_info class. Inside constructor we will provide a textual name for our packet type

common/packet.h

```
 p_info() {name_[PT_TCP]= "tcp";
           name_[PT_UDP]= "udp";
           name_[PT_CBR]= "cbr";
           /* ... much more names ... */
           name_[PT_XYZ]= "protoname";  }
```

Tracing Support

Our simulations aim is to get a trace file describing what happened during execution. A Trace object is used to write wanted information of a packet every time it is received, sent or dropped. To log information regarding our packet type we implement the format_protoname() function inside the CMUTrace class. Trace support for wireless simulations is provided by CMUTrace objects.For this we edit trace/cmu-trace.h file.We must add our new function as in the line number 6 of the next example.

trace/cmu-trace.h

```
1: class CMUTrace : public Trace{ 2: /* ... definitions ... */
3:private: 4: /* ... */ 5: void format_aodv(Packet *p, int offset);
6: void format_xyz(Packet *p, int offset); 7: };
```

In cmu-trace.cc, implement format_xyz function. In order to call this recently created function we must change the format() in trace/cmu-trace.cc.

trace/cmu-trace.cc

```
1: void 2: CMUTrace::format(Packet* p, const char *why) 3:{ 4: /*...
*/
5: case PT_PING: 6: break; 7: 8: case PT_XYZ:
9:format_xyz(p,offset); 10: break; 11: 12: default: 13: /* ...*/
14:}
```

TCL Library

Modify tcl files to create routing agent. First define protocol name to use in tcl file. It would done by modifying ~/ns-allinone-2.34/ns-2.34/tcl/lib/ns-packet.tcl

```
1: foreach prot { 2: XYZ 3: AODV 4: ARP 5: # ... 6: NV 7: } {
8:add-packet-header $prot 9: }
```

4.4 Simulation of Urban MObility (SUMO)

"Simulation of Urban MObility", or "SUMO" for short, is an open source, microscopic, multi-model traffic simulation[22]. It allows to simulate how a given traffic demand which consists of single vehicles moves through a given road network. The simulation allows addressing a large set of traffic management topics. It is purely microscopic: each vehicle is modeled explicitly, has an own route, and moves individually through the network.

4.4.1 Features

- Includes all applications needed to prepare and perform a traffic simulation (network and routes import, DUA, simulation)
- Simulation
  - Space-continuous and time-discrete vehicle movement
  - Different vehicle types
  - Multi-lane streets with lane changing
  - Different right-of-way rules, traffic lights
  - A fast openGL graphical user interface
  - Manages networks with several 10.000 edges (streets)
  - Fast execution speed (up to 100.000 vehicle updates/s on a 1GHz machine)
  - Interoperability with other application at run-time
  - Network-wide, edge-based, vehicle-based, and detector-based outputs
- Network Import
  - Imports VISUM, Vissim, Shapefiles, OSM, RoboCup, MATsim, openDRIVE, and XML-Descriptions
  - Missing values are determined via heuristics
- Routing
  - Microscopic routes - each vehicle has an own one
  - Different Dynamic User Assignment algorithms
- High portability
  - Only standard c++ and portable libraries are used
  - Packages for Windows main Linux distributions exist

- High interoperability through usage of XML-data only
- Open source (GPL)

In general, the given information is converted into three files, which will be read in SUMO as input data. Two of the files contain the network information, which will be converted into node and link information in SUMO and is usually named with extension .nod.xml and .edg.xml respectively. The file containing the traffic demand and route information will be named with extension .rou.xml. In addition, two more files with extensions .con.xml and .typ.xml will be included if specification of allowed traffic movements and lane connections at intersections as well as link types are required. A general overview of the required input files for our example is indicated in the following figure

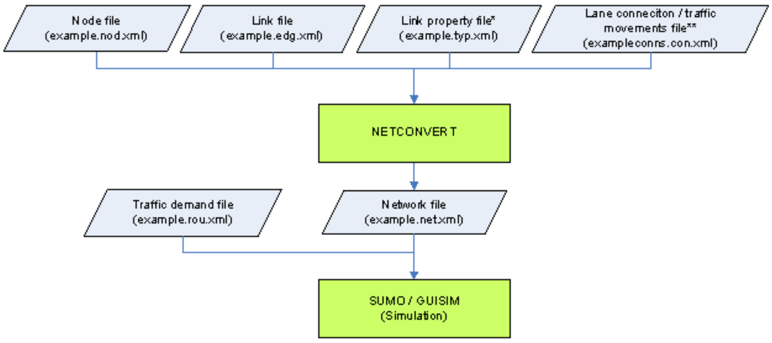

Figure 4.4: The required input files for the example network in SUMO[22]

4.4.2 Traffic Simulation

Traffic simulation in SUMO can be conducted in two ways as described below. The overview of the simulation process is given in the following figure
All file names in the bracket are the file names used in the example.

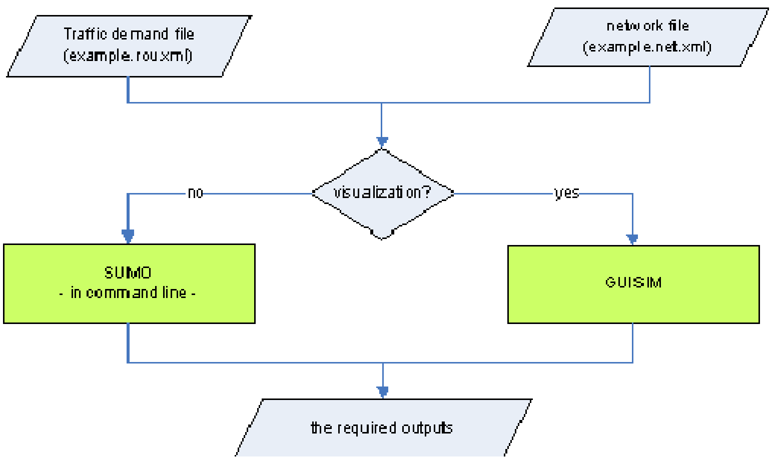

Figure 4.5: Overview of traffic simulation process[22]

4.4.3 Command line

An efficient traffic simulation execution can be achieved with the use of command line, especially when dealing with large and sophisticated traffic networks. To simplify the execution process it is recommended that all the required execution actions, e.g. the path and the name of the input files, the output types, the output directory and the simulation time period is specified in a configuration file. The traffic simulation can be carried out with the use of the following command. sumo -c example.sumo.cfg

4.4.4 SUMO-GUI

The application of SUMO-GUI is the other way to execute the traffic simulation with SUMO. During the execution each vehicular movement and the traffic progression can be observed and the possible bottlenecks can be visually identified. A configuration file for all execution actions is required in SUMO-GUI. Double-click on the Program sumo-gui.exe SUMO-GUI will be activated and a work window will be automatically open. The investigated network can be activated by opening the corresponding configuration file under the File-Menu of the menu bar. Traffic simulation can then be performed by pressing the green triangle button in the main tool bar. The simulation can be stopped any time when the user presses the red squared button. A stopped simulation can be further performed by pressing green triangle, if the simulation time is not up.

5

Simulation Results and Analysis

I have conducted a simulation experiment to measure the performance metrics of all the three protocols. I have used SUMO to generate the different mobility patterns. It also need to be installed JAVA SDK. ns-2.34 (Network Simulator all-in-one) have been used to simulate the network and protocols.

5.1 Simulation Parameters

Default Parameters

Parameter	Value
MAC	UMB (with IEEE 802.11p)
Routing Protocol	AODV
Queue	Priority Queue
Transport Layer	UDP
Antenna	Omni Antenna
Simulation Time	1000 ms
Wireless channel model	Two ray ground
Stream Type	CBR
Link layer queue	50
Range	200 m

Variable Parameters

1. Number of Nodes
2. Maximum speed of the node
3. Background connections
4. Seed Value

5.2 Road Topology

The Network topology has been generated using MOVE (called MOVE - MObility model generator for VEhicular networks). We can execute MOVE by jar file without building the sources. To run it simply run the command at prompt

"java -jar MOVE.jar"

The following steps are performed to create the topology

Mobility Generation

This part of the software generates the mobility model created by SUMO. The following files are generated-

1. map nodes file(mobility.nod.xml)
2. map edge file(mobility.edg.xml)
3. Map configuration file(mobility.netc.cfg)
4. Finally,(mobility.net.xml) will be automatically generated when we select the netc file (mobility.netc.cfg)

Vehicle Movement Generation

1. flow definition (mobility.flow.xml) will specify the groups of vehicle movement flow on simulation.
2. Automatic vehicle movements (mobility.rou.xml) will simply create a number of vehicles at the start of simulation.
3. After the map and movement is complete, we need to specify the configurations of the simulation and save the file as (mobility.sumo.cfg)

Now we can select Visualization to see the actual movement of vehicles.

Now next step is Traffic Model Generation. The traffic model generator consists of two main sections: for NS-2 and Qualnet.

Traffic Model Generator for NS-2 (mobility.tcl)

This editor will generate the traffic simulation file (a tcl file) for NS-2 simulation tool. First import MOVE Trace (i.e. mobility.sumo.tr) and (mobility.net.xml) file for script generator. When the file is saved, (mobility.tcl) is generated. Finally, run the simulation with (mobility.tcl).Finally we can play NAM and see the actual movements.

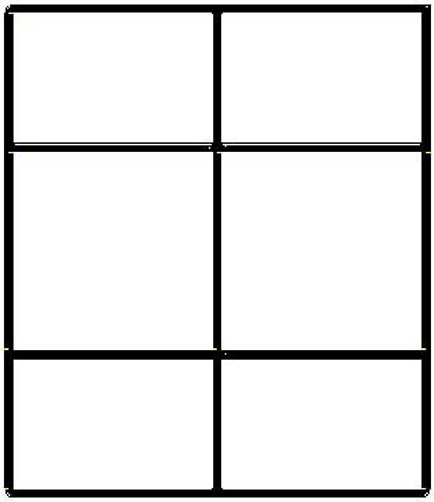

Figure 5.1: Road Topology

Here is the road topology generated and NAM visualization of Traffic.

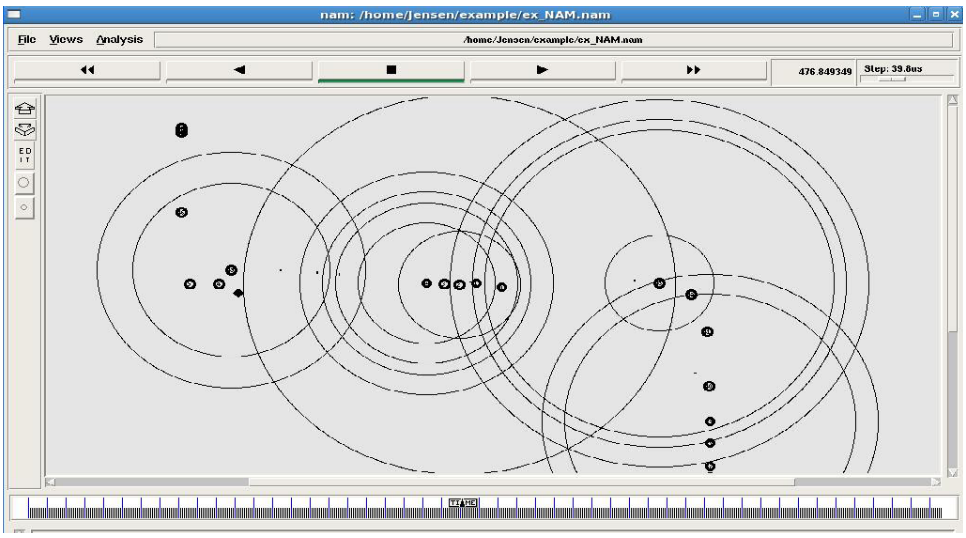

Figure 5.2: NAM Visualization of Traffic

5.3 Performance Metrics

Five metrics have been defined to compare the performance of proposed protocol with p-persistence and UMB protocols.

1. **Success Percentage:** Success Percentage of a packet is the ratio of the cars that receive the broadcast packet to the total number of cars in the simulation. When the average success percentage is lower than 100%, it means that the broadcast packets were not received by all vehicles.

2. **Byte Usage by Nodes:** It indicates the all transmitted bytes used in RTB, CTB, ACK and report.

3. **Dropped Packets (Collisions):** It indicates the total number of packet collisions.

4. **Packet Delivery ratio:** It is the ratio of total received packets to the total sent packets.

5. **Packet Dissemination Time (Delay):** Here the delay refers to the time elapsed between the instance the packet enters the source queue and the reception time of the packet by another node.

5.4 Results

5.4.1 Success Percentage

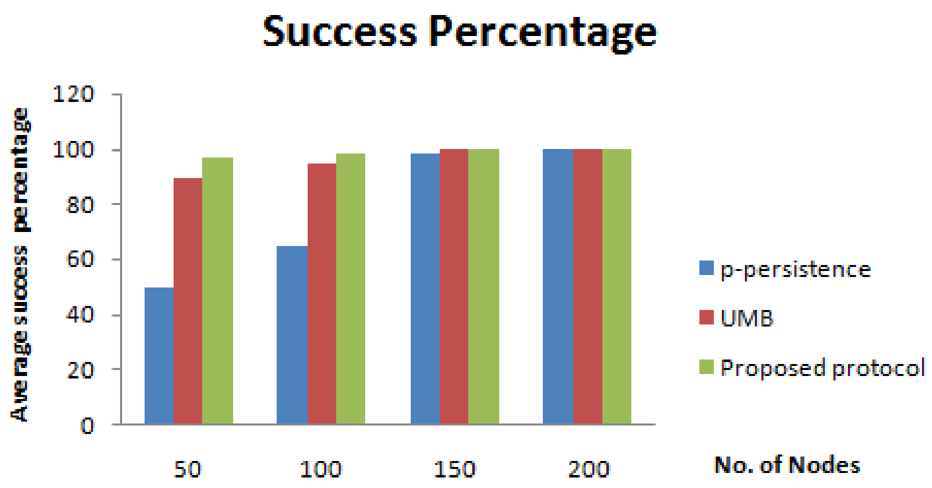

Figure 5.3: Success Percentage

Both RTB/CTB and the proposed protocols are reliable protocols and therefore there is 100% of the report receivers. Using of jamming signal eliminates the hidden/expose node problems. On the other hand, this does not apply to slotted p persistence scheme where report transmission can be interfered with other traffics. The slotted p-persistence broadcasting is good to implement when broadcast storm problem is to be taken care of. But in applications where safety is a concern, UMB and proposed protocol are the best choice so that the message reaches to every node.

5.4.2 Bytes Used by Nodes

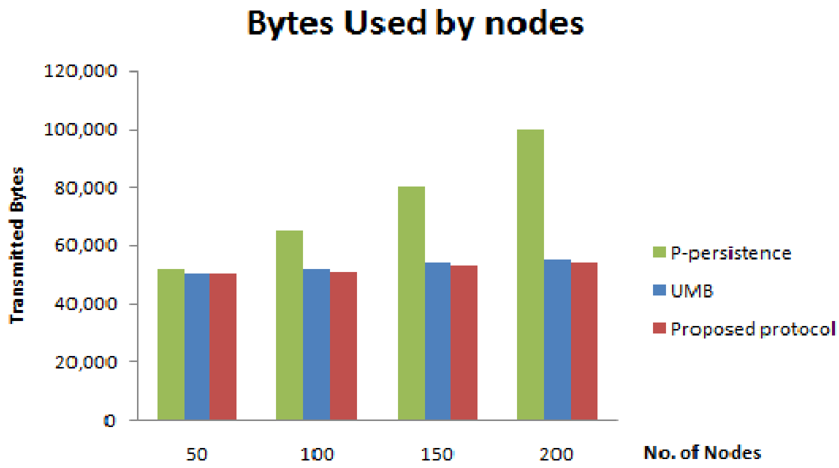

Figure 5.4: Byte Usage by Nodes

According to the result shown in figure, the slotted p-persistence scheme spent the most number of bytes whereas both RTB/CTB and proposed protocol did not spend much number of bytes. This is because nodes contend to transmit a big size report in slotted p-persistance. When there is a transmission redundancy, the number of bytes spent will be rapidly increased. In addition, the byte usage number is proportional to the number of vehicles or vehicle traffic density. This is because there is likeliness that more number of nodes transmit the report because of the increasing number of vehicles.

5.4.3 Dropped Packets (Collisions)

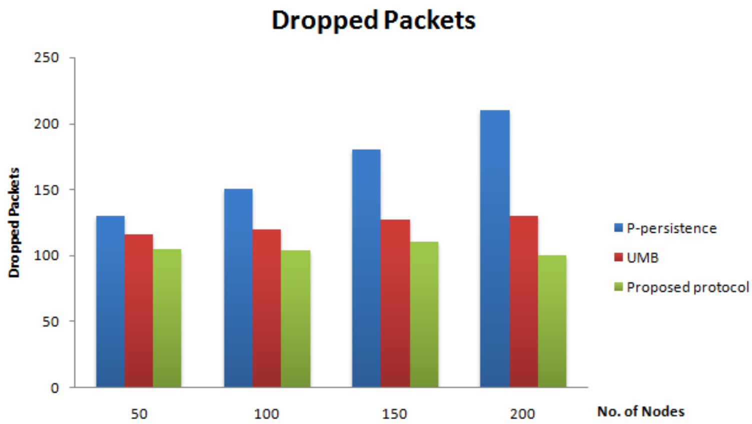

Figure 5.5: Dropped Packets

There are large number of the collisions in Slotted p persistence scheme as the node density increases. On the other hand, both RTB/CTB and proposed scheme have very few collisions because of the their channel access mechanism.

5.4.4 Packet Delivery ratio

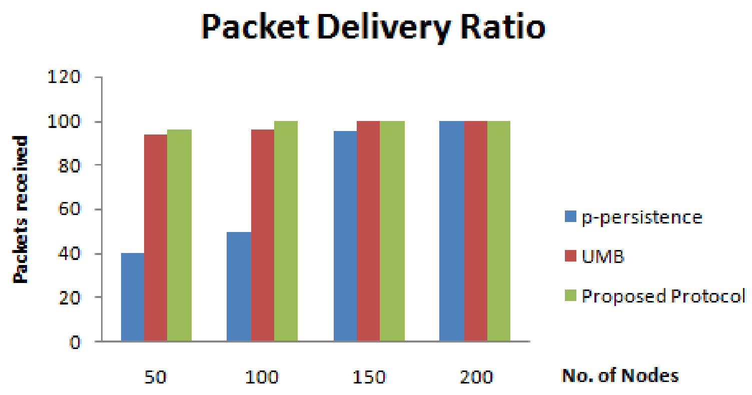

Figure 5.6: Packet Delivery Ratio

It indicates the ratio of the total packets generated to the total packets received. The total packets received in UMB and proposed protocols are almost 100% because of the reliability of the protocol. Whereas there are more number of collisions in slotted p-persistence scheme due to the contention of report transmission, the number grows quickly when the density is increased because of the number of contenders.

5.4.5 Packet Dissemination Time (Delay)

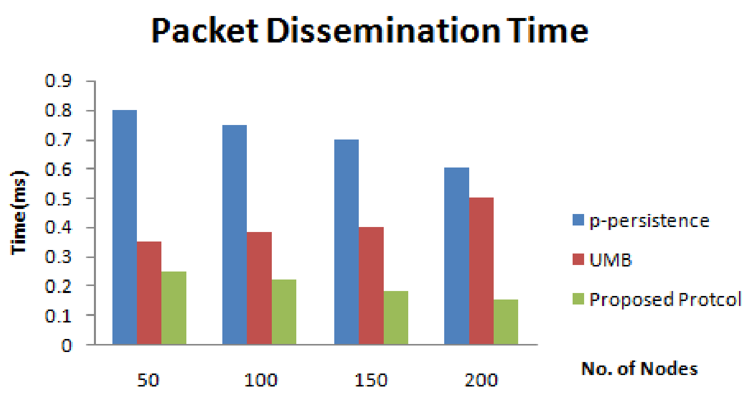

Figure 5.7: Packet Dissemination Time

For slotted p persistence and proposed protocol, the amount of time spent is decreased when the vehicle traffic is more dense. The reason is, when the vehicle density is higher, there are more listeners of a single report transmission. On the other hand, this is opposite for RTB/CTB. In RTB/CTB scheme, more vehicle density leads to more CTB contention rounds which, in turn, results in longer time spent.

In slotted p persistence, each contention slot requires longer time than jamming slot since the slot duration must include the time interval necessary to transmit one packet including DIFS, average CW time and propagational delay. On the other hand, jamming slot requires only the propagational delay. Furthermore, there are also wasted slots utilized by none. All of these reasons contributes to the large amount of time duration spent. For RTB/CTB, the time spent is less than slotted p persistence due to much smaller slot duration.

6

Conclusion & Future Work

UMB, an efficient multi hop broadcast protocol for inter-vehicle communication, obeys 802.11 rules. Therefore it can coexist with other 802.11 modems which do not use this broadcast protocol.

In the proposed protocol, the jamming signal duration has been used to estimate the distance to the closest and the furthest nodes by the propagational delay. However, this might not be feasible in real world implementation because hardware might not be able to process swift enough in the very short duration of propagational delay.

According to the simulation result where there is more vehicle density, speed and background traffic amount, the proposed protocol achieved high reliability and demonstrated better performance than both slotted p persistence and RTB/CTB in every performance matrices including bytes usage, number of collisions and delivered packets.

The future work might include some new ideas such as adaptive probability without using any beacon messages or distance approximation by other methods than jamming signal Or to the best thing we can think of is providing reliability without using jamming signal since the jamming signal will obviously interfere with other traffic.

Bibliography

[1] Car2Car Communication Consortium Manifesto, work in progress, May 2007.

[2] M. Nekovee, "Sensor networks on the road: the promises and challenges of vehicular ad hoc networks and vehicular grids," In Proc. of the Workshop on Ubiquitous Computing and e-Research, Edinburgh, U.K., May 2005.

[3] J. Blum, A. Eskandarian, and L. Hoffmman, "Challenges of intervehicle ad hoc networks," IEEE Trans. Intelligent Transportation Systems 5(4)(December 2004):347-351.

[4] Torrent Moreno, M., Jiang, D., and Hartenstein, H., "Broadcast Reception Rates and Effects of Priority Access in 802.11-based Vehicular Ad-hoc Networks", in ACM VANET, Philadelphia, PA, USA, October 2004.

[5] Xu, Q., Mak, T., Ko, J., and Sengupta, R., "Vehicle-to-Vehicle Safety Messaging in DSRC, in ACM VANET", Philadelphia, PA, USA, October 2004.

[6] Jose Santa a, Antonio F. Gomez-Skarmeta a, Marc Sanchez-Artigas "Computer Architecture and evaluation of a unified V2V and V2I communication system based on cellular networks"

[7] Kulit Na Nakorn Kultida Rojviboonchai "Comparison of Reliable Broadcasting Protocols for Vehicular Ad-Hoc Networks" Chulalongkorn University Bangkok, Thailand

[8] N. Wisitpongphan, F. Bai, P. Mudalige, and O. K. Tonguz, "On the routing problem in disconnected vehicular ad hoc networks," IEEE the 26th INFOCOM'07, Anchorage, Alaska, USA, May 6-12, 2007.

[9] M. Nekovee, and B. Bjarni Bogason, "Reliable and efficient information dissemination in intermittenly connected vehicular ad hoc networks," IEEE the 65th VTC'07-Spring, Dublin, Ireland, April 22-25, 2007.

[10] F. J. Ros, P. M. Ruiz, and I. Stojmenovic, "Reliable and efficient broadcasting in vehicular ad hoc networks," IEEE the 69th VehicularTechnology Conference (VTC'09), Barcelona, Spain, April 26-29, 2009.

[11] O. K. Tonguz, N. Wisitpongphan, F. Bai, P. Mudalige and V. Sadekar, "Broadcasting in VANET," Proc. IEEE INFOCOM MOVE Workshop 2007, Anchorage, USA, May, 2007.

[12] N. Na Nakorn, and K. Rojviboonchai, "DECA: Density-Aware Reliable Broadcasting in Vehicular Ad-Hoc Networks," IEEE the 7th ECTICON2010, Chiang Mai, Thailand, May 19-21, 2010.

[13] K. Na Nakorn and K. Rojviboonchai, "POCA : Position-Aware Reliable Broadcasting in VANET," to be appear in Proc. the 2nd Asia-Pacific Conference of Information Processing APCIP2010, Nanchang, China,September 17-18, 2010.

[14] N. Na Nakorn, and K. Rojviboonchai, "Efficient Beacon Solution for Wireless Ad-Hoc Networks," the 7th JCSSE2010, Bangkok, Thailand, May 12-14, 2010.

[15] Y.-C. Tseng, S.-Y. Ni, Y.-S. Chen, and J.-P. Sheu,"The Broadcast Storm Problem in a mobile ad hoc network," Wireless Networks, vol. 8, pp. 153-167,March-May 2002.

[16] G. Korkmaz, E. Ekici, F. Ozguner, and U. Ozguner, Urban Multi-Hop Broadcast Protocol for Inter-Vehicle Communication Systems, in Proc.of ACM Intl Workshop on Vehicular Ad hoc Networks, Philadelphia, USA, October 2004, pp. 7685.

[17] O.K. Tonguz, N. Wisitpongphan, J.S. Parikht, F. Bait, P. Mudaliget, and V.K. Sadekart, "On the Broadcast Storm Problem in Ad hoc Wireless Networks," BROADNETS 2006. 3rd International Conference, 2006.

[18] V. Naumov, R. Baumann, and T. Gross, An evaluation of inter-vehicle ad hoc networks based on realistic vehicular traces, Proc. ACM the 7th ACM MobiHoc06, Florence, Italy, May 22-25, 2006.

[19] The Institute of Electrical and Electronics Enginneers (IEEE), Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Speci

cations." http:/standards.ieee.org, ANSI/IEEE Std.802.11, 1999. (a.k.a. ISO/IEC 8802-11:1999(E)).

[20] N. Wisitpongphan, O. K. Tonguz, Carnegie Mellon University, J. S. Parikh, P. Mudalige, F. Bai, And V. Sadekar, General Motors Corporation "Broadcast Storm Mitigation Techniques In Vehicular Ad Hoc Networks" IEEE Wireless Communications, December 2007

[21] J. Chung, M. Claypool. NS by Example. Worcester Polytechnic Institute (WPI) http://nile.wpu.edu/NS/purpose.

[22] Daniel Krajzewicz,Christian Roessel SUMO - Simulation of Urban MObility - User Documentation

Appendix A

Books and WebSite References

A. Vehicular Networks from Theory to Practice By Champman and Hall/CRC Computer

B. Vehicular-2-X communication State-of-the-Art and Research in Mobile Vehicular Ad hoc Networks

C. Simulation of Urban MObility (SUMO); http://sumo.sourceforge.net/

D. The Network Simulator (NS-2); http://www.isi.edu/nsnam/ns/

E. http://www.citeseerx.ist.psu.edu

F. http://www.portal.acm.org

G. http://www.ieee.org