POLITECNICO DI TORINO

III Facoltà di INGEGNERIA
Master of Science in Communication Engineering

Master Thesis

Content Downloading in Vehicular Ad-hoc Networks

Advisors:
Prof. Carla-Fabiana Chiasserini
Prof. Claudio Casetti

Candidate:
Zhenhua Zou

September 2009

Summary

The main aim of this thesis is to propose a scheme to download the content in Vehicular Ad-hoc NETwork(VANET) for convenience applications and comfort applications. VANETs will come into reality in the near future because of the widely deployment of Access Points(AP) and the new 802.11p standard. Besides safety critical applications, there is a growing demand for convenience applications and comfort applications. Both these applications do not require timely and reliable delivery of the contents, however they require a higher bandwidth.

Nodes in a VANET have both advantages and disadvantages compared to the nodes in generic Ad-hoc Networks. On one hand, VANET nodes do not have any energy and computational power limit. They know their positions thanks to the GPS receiver and their mobility is highly predictable. On the other hand, high velocity of vehicles results in frequent change of topology. Moreover the communication between AP and vehicle is fleeting due to the limited coverage range of AP and additional time to connect an AP. The communication is also intermittent because ubiquitous deployment of APs is not feasible.

In this thesis, we design a scheme, which utilizes GPS information and solves the problems presented above by using cooperative downloading and Vehicle to Vehicle(V2V) information sharing, to implement the applications in a VANET scenario.

The basic idea of cooperative downloading is that the AP sends sequentially only one copy of the content to the group. More specifically, the header sends the REQUEST message as soon as it enters the coverage range for requesting the content from the AP. After receiving the REQUEST message, the AP sends the TRAFFIC message with a portion of the content to this vehicle header. At the same time, other vehicles, which are also inside the coverage range, can overhear the TRAFFIC message. The AP sends next TRAFFIC message if it receives ACK at the MAC layer. When the AP does not receive ACK for the TRAFFIC message at the MAC layer after maximum retry times, it directly sends the next TRAFFIC message to the new header vehicle without any handshake.

The basic idea of V2V information sharing phase is that the vehicle obtains the missing packets from the other vehicles. The COVERAGE STATE of the vehicle,

which means that the vehicle is still near the coverage range of an AP, is introduced in order not to interfere other vehicles' downloading phase. If the vehicle has not received any TRAFFIC message after traffic_timeout, it goes to COVERAGE STATE. Or if it overhears any TRAFFIC message, it enters COVEARGE STATE. After having not received or overheard any packet for adver_timeout in COVERAGE STATE, it finally starts the V2V information sharing phase.

In V2V sharing phase, the vehicles advertise the received packets by sending broadcast ADVER message. The ADVER message is sent within one-hop. After receiving ADVER message, other vehicles can send back the CTS message about the packets they need. The ADVER sender then decides the forwarder to which the TRANSFER message is sent to. The TRANSFER message contains the payload data. The forwarder is the vehicle which wants the packet and is the farthest vehicle from the source so that the information can be distributed as quick as possible. When either all the vehicles get the contents or the group meets the second AP, the V2V information sharing phase ends.

The scheme is implemented on the Network Simulator 2(NS2). Two application classes are added: *VCDServer* and *VCDClient*. The *GroupManager* class is used to simulate the group management protocol. Moreover, some modifications at lower layers are made to support the overhearing of unicast message and cross layer ACK notifications. Besides those, the AODV routing agent is changed in order to support the correct operation of AP.

Lastly, the performance of the scheme is evaluated. A simple case, which is composed of two vehicles and one AP, is evaluated in the first place. The number of received packets is 16.70% higher and the throughput for the whole group increases a lot by 54.1%. Other simulation results are presented also. Firstly, the size of packet status field with respect to different group size is acceptable and does not change a lot. The size is from 10 bytes to 15 bytes. Secondly, higher sharing timeout can result in more number of received packets but longer content distribution duration time. Thirdly, the throughput per vehicle does not change so much when the file size is different provided that the group can download all the contents from only one AP. Fourthly, the number of received packets and throughput per vehicle increases as the radio range of the nodes becomes larger. Finally, in multiple APs case, the packet receipt status is different for each vehicle because there are not enough time in between two APs to share the contents from the first AP.

Acknowledgements

This thesis was carried out in the TELECOMMUNICATION NETWORKS GROUP, Electronic Department (Dipartimento di Elettronica) in POLITO (Politecnico di Torino), Torino, Italy.

Foremost, I would like to express my sincere gratitude to my advisor, Associate Professor Carla-Fabiana Chiasserini, for her help, encouragement, and useful discussions. Without her tutor, it is impossible to write this thesis and complete the work behind it.

I would also like to thank Marco Fiore for his precious recommendations and help in implementing the protocol in NS2 simulator. It saved me a lot of effort and time. I am also grateful to all the colleagues I have worked together with in the LADISPE lab. The experience with them is enjoyable and worthwhile.

Finally I wish to thank my parents for their long-term support during my study in Italy.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This chapter serves as the background introduction of the thesis. Firstly, this chapter introduces the main characteristics of the Vehicular Ad-hoc Network (VANET) and its differences from the Mobile Ad-hoc Network (MANET). Then the main work of the thesis **Content Distribution in VANET** is presented including its challenges and applications. The third part introduces the simulation tool, The Network Simulator, ns-2 (NS2) for the verification and performance evaluation of the scheme. The last part is the contribution and outline of the thesis.

1.1 VANET

A Vehicular Ad-Hoc Network is a form of Mobile ad-hoc network, to provide communications among nearby On Board Unit (OBU): vehicles and between vehicles and nearby fixed equipment, which is usually described as Road Side Unit (RSU). According to this definition, the communications in VANET can be categorized into two types[1]: Inter-Vehicle Communications (IVC) and Roadside-Vehicle Communications (RVC).

MANET is a form of ad-hoc network, which is defined as an autonomous system of nodes connected by wireless links forming an arbitrary graph (stand-alone fashion), or connected to a fixed infrastructure. Each must forward traffic unrelated to its own use, and therefore be a router. MANET extends the definition of the ad-hoc network by allowing node to freely move independently in any direction. Therefore the nodes will change its links to other devices frequently. The primary challenge in building a MANET is equipping each device to continuously maintain the information required to properly route traffic.[2] VANET is one type of MANET, which may be the most promising application in the near future.

1.1.1 Peculiar Features of VANET

Obviously VANET has its own distinctive features. The differences of VANET compared to MANET are mainly the ability and mobility pattern of the node. These differences can result in both advantages and disadvantages, which are listed in the below.

1. **Advantages**

   - *No Power and Energy Limit*

     In the general case, the MANET node could be personal laptop, cell phone or Personal Digital Assistance (PDA). They are battery-powered with limited power and energy. In the contrary, vehicles are already equipped with many power-consuming electric appliances, such as lights, air conditioner and musical speaker. An extra electronic device with relative low power demand should not be a problem. Furthermore, the vehicles can recharge the battery as it moves.

   - *No Computational Power Limit*

     The computational power is limited too when the node is cell phone or PDA. Due to the restricted product size, the node may have to share the computational power of a not very powerful processor with other key applications in these devices. While in the case of VANET, the user could put an additional, dedicated and potent device to run the application in the vehicle.

   - *Position, Velocity and Time (PVT) information.*

     Equipment of Global Navigation Satellite System (GNSS) can provide PVT information. The Global Positioning System (GPS) is the fully functional GNSS in the world, which can be used freely by anyone, anywhere, and is often used by civilians. It allows the user to determine their current location, the time, and their velocity. Currently many vehicles have the GPS devices installed previously for the navigation purposes in the city streets or traveling. Autonomous civilian GPS horizontal position fixes are typically accurate to about 15 meters. Under a clear view of the sky the accuracy can be as small as about 5 meters horizontally.[3] This is a very good asset when we design the scheme.

   - *Higher predictability in nodes' mobility.*

     Random Waypoint (RWP) model is a commonly used simulation model for mobility, e.g., in Ad Hoc networks. RWP model can not be applied in VANET and is a very poor approximation of the vehicular mobility because the vehicle should follow the pane in the front of it. Therefore it has a high predictability. The vehicles in the same direction could maintain their distances between each other and positions for some time. Combined with the speed and the position information provided by GPS, more advantages can be further explored.

2. **Disadvantages**

   - *Higher topology changes than MANET*

     A vehicle can move at the speed of 40 km/s in urban scenario and up to 100 km/s in highway scenario. There is a high possibility that the topology changes frequently, especially in the crossroad of the urban street or the exit of highway road.

   - *RVC communication is fleeting and intermittent.*

     The connection duration time between them is fleeting, lasting only tens of seconds because the vehicle speed is high and the coverage range of RSU is limited. If we take into account the time for setting up the connection between RSU and OBU, the duration time of the connection could be much less. Moreover a ubiquitous deployment of RSU may require considerable investments and time. It may become true in big cities in the near future. But it can be very difficult along all the highway roads.

1.1.2 Physical and MAC layer

A complete communication protocol should be composed of physical, MAC, routing, network and application layers. In this section, the physical and MAC layer of VANET protocols are presented because these two layers are already standardized and the final scheme is designed upon these two layers.

A new physical and MAC layer is necessary to ensure the operation of VANET because fast and slow fading effects can be expected to cause problems as both the transmitter or/and receiver are moving, potentially toward each other. IEEE 802.11 is currently the most widely used wireless local area network standard in the world. IEEE 802.11p is a draft amendment to the IEEE 802.11 standard to add Wireless Access in Vehicular Environments (WAVE). This includes data exchange between high-speed vehicles and between the vehicles and the roadside infrastructure in the licensed Intelligent Transportation Systems (ITS) band of 5.9 GHz (5.85-5.925 GHz).

- **Physical Layer**

  It is based on the IEEE 802.11a physical layer. However all time parameters are doubled to counteract Inter Symbol Interruption (ISI) due to multi-path and Doppler effect. Therefore the data rate is halved compared to that in IEEE 802.11a. Data rate ranges between 3 (basic rate) and 27 Mb/s. It uses seven 10MHz-wide channels in the 5.9GHz frequency band. One of them is used as control channel with a higher priority.

- **MAC Layer**

  The 802.11p MAC layer is very similar to the previous 802.11 MAC standards. It is also based on Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) and includes some features of the 802.11e standard to support various kinds of services with different QoS levels. The node transmits in control channel only during control channel interval and in service channels only during service channel interval. These two intervals are synchronized among nodes by using time reference provided by GPS.

Due to the limitation of the network simulator, 802.11b standard is used as the underlying physical and MAC layer. Fortunately, there are no physical and MAC layers modifications in the scheme. The scheme can be implemented upon 802.11p standard. But the performance may change due to the different data rate in these two standards.

1.2 Content Distribution in VANET

In this thesis, a scheme is proposed to efficiently distribute contents from the Internet to all the vehicles in VANET. As a matter of fact, RSU is the gateway of the network. It is the only node which can download contents from Internet. In the following, the commonly used content distribution methods on Internet are described firstly. Then the corresponding challenges in VANET and its possible applications are presented.

1.2.1 Content Distribution

The tradition content distribution method on Internet is the Client/Sever Model. A server machine is a high-performance host that is running one or more server programs which share its resources with clients. A client does not share any of its resources, but requests a server's content or service function. Clients therefore initiate communication sessions with servers which await (listen to) incoming requests. It is easy to manage data in Client/Server Model. However it suffers from the problem of single point of failure in server machine. Furthermore, the server can be overloaded when the number of simultaneous client requests to the server increases.[4]

Peer to Peer (P2P) applications such as BitTorrent and Emule have attracted many public attention recently due to their successful deployments. They overcome the above problems in the Client/Server model. This method makes full use of the bandwidth and computational power of each peer because each peer can exchange content among themselves. All involved peers are equally privileged and equipotent participants in the application. That means each host or instance of the program can simultaneously act as both a client and a server.

1.2.2 Challenges

In the VANET scenario, both of the above methods should be used. On one hand, RSU is fixed. It appears as the gateway of the content and the only source of it. It is natural to employ Client/Server model in the content distribution between RSU and vehicles, where vehicles are clients and RSU is the server. On the other hand, in order to fully utilize the bandwidth of the wireless channel, P2P fashion should be used among vehicles. Actually it is the only approach that can be used when all vehicles are outside the coverage of RSU.

Unfortunately, the scenario suffers from both the disadvantages in Client/Server model and P2P sharing.

As regards the Client/Sever model, since the RSU may not be sufficiently deployed along all the roads, the communication in Client/Server manner is fleeting and intermittent. It means that the single point of failure problem is more severe in our case. Nonetheless, the good point is the broadcast property of the wireless transmission. It allows other nodes to overhear the packets in the channel even if it is a unicast packet.

Concerning P2P sharing, it is designed for wired network where the topology of the peers remain stable for a relative long time. Even in wired networks, special attentions should be paid in protocol design to tackle the problem of peer leaving and joining, which is defined as churn. In wireless network, the problem is even worse due to the frequent change of topology, which can be modeled as a higher churn rate.

1.2.3 Applications

The main applications of VANET can be categorized into three types: safety critical applications, convenience applications and comfort applications.[1]

- *Safety critical applications*

  Safety critical applications are geared primarily toward avoiding accidents by transmitting warning messages such as collision messages. They have a very strong time delay constraint and a high reliability requirement of the transmission.

- *Convenience applications*

  Convenience applications include traffic management application, traffic coordination application and parking availability et.c. Usually they do not require timely and reliable delivery of the messages.

- *Comfort applications*

  The main focus of comfort applications is to make travel more pleasant by providing audio and video streaming, web-browsing and email et,c. They need a higher bandwidth than the above two applications.

In our scenario, the safety criticial applications are not considered. In order to transmit safety critical messages, some modifications are needed at the lower physical and MAC layer. 802.11p standard supports this functionality by introducing QoS and using control channel. Therefore the proposed scheme can be mainly used in convenience and comfort applications, which do not have strong time constraint.

1.3 NS2

In the thesis, NS2 is used to implement the scheme and evaluate the performance. NS2 [5] is a discrete event simulator targeted at networking research. It is a packet-level simulator which is essentially a centralized discrete event scheduler to schedule the events such as packet and timer expiration. NS2 provides substantial support for simulation of Transmission Control Protocol (TCP), routing, and multicast protocols over wired and wireless (local and satellite) networks.

NS2[6] is based on two languages: C++ and OTcl. The compiled C++ hierarchy allows us to achieve efficiency in the simulation and faster execution times. The detailed definition and operation of the protocols are defined in C++ language. In the OTcl script provided by the user, we can define a particular network topology, the movement of the vehicles, the transmission range and data rate of the nodes. It allows us a quicker way to generate simulation results than using C++ language because in C++ language we have to recompile the source code every time we modify the parameters of the simulation.

Therefore the final submitted work is composed of two parts of C++ source codes and OTcl scripts. The scheme's implementation is written in C++ code and the performance evaluation is written in OTCL scripts, which setup the network scenario and configure different parameters in the scheme, such as the group size, the mobility pattern, the location of the nodes, the timeout value, the transmission range, the file size, etc.

Although NS2 is open source and free for academic use, it is not very well-structured and there are some approximations. Moreover some modules are missing in NS2. More descriptions regarding the implementation are in *chapter 4*.

1.4 Outline and Contributions

1.4.1 Terminology

It is necessary to normalize the terms used in this thesis in order to make it more consistent and clear because in literature there are many different terms used to describe the same thing.

Terms In This Thesis	Terms In Other Literature
Access Point(AP)	Road Side Unit(RSU)
Vehicle	On Board Unit(OBU)
Inter-Vehicle Comm.(IVC)	Vehicle to Vehicle Comm.(V2V)
Roadside-Vehicle Comm.(RVC)	RSU to Vehicles Comm.(I2V)
	Vehicles to RSU Comm.(V2I)

Table 1.1. Terminology Table

1.4.2 Contributions

The main contribution of the thesis is that a scheme is proposed in VANET, which aims to download as much as possible and as fast as possible the content for each vehicle. The scheme is composed of two phases. The first one is the cooperative downloading of the content from AP. The second one is the Vehicle To Vehicle (V2V) content sharing among the vehicles. Furthermore this scheme utilizes the advantages and tries to solve the problems presented in the above sections. Its main ideas can be summarized in the following items:

1. All vehicles cooperatively download the content from AP given the fact that they are interested in one content and they are aware of each other's packet receipt status and location information. The AP sends sequentially only one copy of the content to the group.

2. The sharing of the contents among vehicles uses the idea of geographical multicasting.

3. All vehicles work in overhear mode in order to maximize the data traffic received from AP or other vehicles.

4. The ACK message in application layer is eliminated since the node can use the ACK message in MAC layer if the communication is within one-hop range.

5. The scheme tries to receive more contents from AP by suppressing unnecessary transmissions when the vehicles are still under or near the coverage of AP.

1.4.3 Outline

The contents of each chapter are as follows:

- *chapter1* gives a general introduction of the thesis.
- *chapter2* is the literature survey of the current techniques related with our work.
- *chapter3* details the final proposed scheme after several design iterations including the message format, the states and the state flows.
- *chapter4* describes the implementation details about the code in NS2.
- *chapter5* discusses the simulation results of the scheme with different parameters.
- *chapter6* concludes the thesis and points out the future work.

Chapter 2

State of the Art

2.1 Introduction

This chapter introduces the state of the art, which deals with the content distribution in MANET, VANET and any other networks that are relevant with the thesis objective. The papers are classified by protocol layers from the MAC layer to the application layer.

2.2 MAC Layer

The vehicles exchange their content in P2P fashion when they are not under the coverage of the RSU. Since most of the packets would be the same in the channel, it could save a lot of bandwidth when we employ multicast techniques. One approach is to use multicast in MAC layer. However there are some problems of wireless multicast in MAC layer, which are not considered in the traditional MAC multicast protocols:

1. Due to the broadcast property of the wireless transmission, packets suffer from increased instances of hidden terminal problem.

2. The ACKs transmitted by every receiver to the sender greatly increase the overhead.

3. Transmission rate optimization is another problem too. In IEEE 802.11 protocols, the broadcast transmission is done at the minimum rate such that the possibility that all the receivers have correctly received the packets is high. However the overall throughput is not the highest. Then we need to choose the transmission rate to maximize the overall throughput.

First of all, there are some debates on whether the protocol should use the ACK or not. As in the case of wired network, the broadcast/multicast message is sent without ACKs. However in wireless network, the transmission of ACKs to ensure the receipts of the message seems necessary. In the tree-based protocols, the forwarding failure in the nodes near the multicast root results in the whole unsuccessful transmission in that branch of tree. Moreover we need to implement the ACKs in the MAC layer rather than in application layer because MAC layer multicast error control schemes lead to shorter delays than application layer schemes. In addition, previous CSMA/CA in IEEE 802.11 does not have any MAC layer recovery on multicast/broadcast frames. On the other hand, the introduction of ACK greatly increases the complexity of the protocol. After all the proposed mechanisms can be categorized into two main methods: Leader-Based Protocol (LBP) and ACK-based.

Secondly, a separate VC-MAC multicast protocol, which uses distributed manner to select the optimal relay set, is briefly reviewed.

2.2.1 LBP-based

LBP was first proposed in Kuri's[7] paper. A leader is set to be the only node which responds to the sender's Request To Send (RTS) frame and to reply with ACKs. Other non-leader can notify their incorrect receipt by jamming the ACK transmitted by leader. However the original LBP method suffers from the effective and efficient problems. Firstly, It is a negative feedback algorithm. If a node cannot receive the packet or the Negative ACKnowledgement (NACK) is lost, the multicast sender can't be notified the unsuccessful transmission. Secondly, only at the time when all members in the group receive the data at the same time, the multicast transaction is finished. In the case of high channel error rate, the possibility of all multicast receivers being received the packet is small. Furthermore, combined with the high mobility of the node, the efficiency of LBP is low. It has an even worse performance in VANET. Thus this method is suitable for error-tolerant data transmission.

In order to solve the above LBP problems, a Beacon-driven Leader-Based Protocol (BLBP) is proposed by Li[8]. A beacon frame is sent before the the actual data frame telling all the receivers about the packet duration and the sequence number. In this case non-leader will not miss the transmission of packets and the sequence number can eliminate the requirements of simultaneous successful receipts. However it has a strong assumption that the beacon frames are more reliable due to its smaller size and the fact that they are transmitted at a lower rate. It does not solve the efficient problems either.

2.2.2 ACK-based

The basic idea of ACK-based protocol was explained in Gossain[9]. In this paper, the author proposes a Multicast-aware MAC Protocol (MMP) for MANET. It attaches an Extended Multicast Header (EMH) by the multicast agent, which provides the address of the next hop nodes that are supposed to receive the multicast packet. The MAC layer in MMP uses the EMH field to support an ACK based data delivery. After sending the data packet, the transmitter waits for the ACK from each of its destinations in a strictly sequential order to avoid the collision of ACKs. A retransmission of the multicast packet is performed if the ACK from any of the nodes in EMH is missing. In order to cope with the hidden terminal problems, after the first failed trial, a multicast RTS packet is sent to distinguish between hidden terminal case and the case when the nodes move outside the range.

Kim[10] uses one Orthogonal Frequency Division Multiplex (OFDM) symbol for the ACKs from all STAtion (STA), and each member STA indicates its packet reception status by utilizing a sub-carrier within the OFDM symbol. It greatly reduces the overhead and the transmission time for ACKs. This paper only concerns error recovery. Another paper[11] enhances it to resolve the hidden node problem over wireless multicast using the same method. However it is a cross-layer design with physical layer and Orthogonal Frequency-Division Multiple Access (OFDMA) is not available in current 802.11 protocols' physical layer.

There are some other supplementary approaches to increase the performance in ACK-based methods. In [12], the author describes a location aware multicast MAC protocol. It uses location information to determine a subset of all the receivers. The recipient of the ACK from the nodes inside this subset can guarantee the correct receive of multicast message to all possible destinations thus reducing the overhead in transmitting ACKs. In [13], the author reduces the overhead by using cumulative ACKs as in the TCP protocol. The main idea is that in a successive packet transmission, the Clear To Send (CTS) packet sent by the receiver can also serve as the ACK for the previous sent packet. It includes a field SEQ and the number of the latest successfully received DATA frame at a group member. The last sent packet then needs an explicit ACK. It is very like the cumulative ACKs in TCP protocol.

2.2.3 VC-MAC

This paper[14] proposes a novel protocol called VC-MAC. It theoretically analyzes the selection of optimal relay set using Weighted Independent Set (WIS) model and designs a back-off mechanism to select the concurrent relays in a distributed manner. The users who have already got a copy are called potential relays. The users who want the copy but failed to get it during the gateway's broadcast are called potential destinations. By the principle of cooperative communication, performance enhancement is achieved by using suitable relays to forward packets to the receivers. After analyzing the problem of selecting the best optimal relay nodes, the paper models it as the WIS problem. Then a greedy iteration algorithm is used to solve the problem. It selects a set of nodes with higher value of an equation. In actual implementation, the backoff time is inverse proportional to this value. Thus the nodes with higher value have the chance to transmit and the nodes in its interference range can not transmit. It achieves the goal of selecting best relays in distributed manner. However it assumes the channel condition is symmetric and it needs the measurement of Signal-to-Noise Ratio (SNR) value and determination the nodes pair whose simultaneous transmissions interfere with each other. Both of them may be unrealistic in high topology change scenarios such as VANET.

2.3 Routing Layers

In this section, we discuss the routing layer. Luckily, there are several survey papers on this topic. Although they are not current and do not include everything, it provides a good classification of the protocols. A special interest is given to the location based methods because the vehicles are equipped with GNSS receiver. Small Group Multicast (SGM) is discussed as well. In the last subsection, we introduce the scheme of Preferred Group Broadcast (PGB), whose idea is used in my final protocol.

2.3.1 Introduction

Another approach to efficiently disseminate content is multicasting in routing layer. Conventional multicast routing protocols were designed for wired networks which have a stable network topology. High topology changes usually come along the low bandwidth. However wireless multicast protocols have some advantages compared to wired network too. Due to the broadcast nature of wireless medium, multicasting can improve the efficiency of the wireless link when multiple mobile nodes are located within the transmission range of a node.

There are already some survey papers on MANET. Since VANET are a subset of MANET, it is worthwhile to have a look at the protocols in MANET. Both of these two papers[15, 16] category the protocols as the followings:

- Flooding

  Although it is very simple, it suffers from the problem of inefficiency. Many duplicates are received in each node and it subsequently results in a higher probability of collisions.

- Tree-based

  It is more efficient than flooding. However the distribution tree must be rebuilt frequently due to the constant topology change in VANET. Moreover the root node is much more important than other nodes. The failure of it could destroy the whole distribution tree.

- Mesh-based

  It tries to solve the robustness problem of tree-based protocols. It provides redundancy by using alternative paths and allows multicast datagrams to be delivered to the receiver if one of the paths fails.

- Overlay-based

  The nodes move frequently and the control message should be exchanged frequently as well to make the distribution tree up-to-date. In order to achieve scalability in the terms of control overhead, overlay-based protocols keep state information in group members. A virtual network is built upon the VANET topology among group members. It remains static for some time even if the real topology is changed. Nevertheless the cost is the inefficient path among the virtual network.

- Backbone-based

  It is another approach to reduce control overhead by using a hierarchical method. A simple and stable backbone is built. Within the backbone, multicast routing may be flooding or mesh-based. As regards the non-core multicast nodes, a tree-based distribution is applied.

- Stateless

  Stateless multicast protocols reduce the non-members' control overhead by explicitly listing destinations in every packet. The non-members use underlying unicast protocol to select the next hop. Its main drawback is that it introduces high overhead in its listing destinations information when the group size is huge. Fortunately, the group size usually is small in VANET in a given small area.

However much of the effort has been put on designing routing protocols to support effective and efficient communications between multicast group members. The building of the group is ignored. Additionally, as can be seen from the survey paper, there is no a "one-for-all" scheme that works well in scenarios with different network sizes, traffic overloads and node mobility patterns. There does not exist any protocol that outperforms others in all aspects.

Moreover this paper[16] addresses three important factors that none of the above methods have considered and that can be used in VANET scenarios.

1. The movements of the vehicle are predictable despite of their high speed. They move along streets and according to traffic modes.
2. Geographical routing could provide additional information.
3. The use of static road side unit can benefit a lot.

2.3.2 Location-Based Routing

Since vehicles are generally equipped with GNSS receiver which can provide position, velocity and time information, some particular location-based approaches are described.

In[17], the author proposes the Position-Based Multicast (PBM). The forwarding nodes are selected based on information about the positions of all one hop neighbors and the positions of all individual destinations which are carried in every packet header. This may not be suitable for highly mobile and dense V2V networks. Information about the positions of vehicles becomes invalid time to time due to mobility of vehicles, and the size of a packet header would be significantly increased for carrying the position information of many recipients. Accordingly, delay for packet dissemination would increase.

Scalable Position-Based Multicast (SPBM)[18] introduced hierarchical group membership management in order to cope with the drawback of PBM caused by many recipients. The network is subdivided by hierarchical levels: A geographical region in the network can be identified by a particular combination of hierarchical levels. The multicast members in geographical regions are aggregated into hierarchical levels. The hierarchy information is carried in the packet header instead of the list of position information about all destinations.

Robust and Scalable Geographic Multicast (RSGM)[19] is similar to SPBM in the sense that the network is divided into geographical zones and multicast members are maintained through regional group membership management, but it applies position based unicasting to forward multicast packets.

2.3.3 Small Group Multicast

Generally speaking, there seems to be two kinds of multicast application models: a broadcast-like multicast that sends data to a very large number of destinations and a "narrowcast" multicast that sends data to a fairly small group. An example of the broadcast model is the audio and video multicasting of an Internet Engineering Task Force (IETF) working group session to sites around the world. A video conference involving three or four parties is an example of the narrow cast model. For reasons discussed below, it seems prudent to use different mechanisms for these two cases.

The following SGM[20] solves the scalability problems and administrative complexity problems. SGM is a solution for supporting a very large number of small multicast groups in Internet. The basic idea is that the network should not be burdened with running multicast routing protocols to build and maintain the distribution trees for very large numbers of small multicast groups. If a group has a small number of members, the list of destinations can be carried in the packets and the network doesn't need to build and maintain a distribution tree for that group. In wireless network, the overhead of running routing protocols may be significant comparing to the available bandwidth.

In SGM, the source node keeps track of the destinations to which it wants to send packets. The source encodes the list of destinations in an SGM header and then sends the packet to a router. Each router along the way parses the SGM header, partitions the destinations based on each destinations next hop, and forwards an appropriate SGM packet to each of the next hops. Each final hop removes the SGM encoding and forwards the data to its destination as a standard unicast packet. In effect, SGM is unburdening the network of multicast state by carrying the state in each data packet and processing it in real time along the forwarding path.

Actually the above stateless routing protocols are a kind of SGM. Tree- and mesh-based approaches have an overhead of creating and maintaining the delivery tree/mesh with time. In a MANET environment, frequent movement of mobile nodes considerably increases the overhead in maintaining the delivery tree/mesh. To minimize the effect of such a problem, stateless multicast is proposed wherein a source explicitly mentions the list of destinations in the packet header. Stateless multicast focuses on small group multicast and assumes the underlying routing protocol to take care of forwarding the packet to the respective destinations based on the addresses contained in the header.

We summary the advantages and disadvantages in the following items:

- Advantages:

    1. It eliminates the need for multicast routing protocols or multicast state maintenance in the network. In the case of Internet, it avoids IP multicast artifacts such as group addressing and Internet Group Management Protocol (IGMP).
    2. It is scalable.
    3. It ensures that data packets flow on the optimal path between source and destination.
    4. It complements current IP multicast.

- Disadvantages:

  1. Extra bytes must be sent in multicast packets for the list of destinations;
  2. it is not suitable for huge broadcast-like multicasts; it is targeted for small conferences.

We would also like to have a look at the combination of the above two techniques, which is called **Small Group Multicast With Geographical information**.

The authors extend the Xcast protocols with geographical information in this paper[21]. Xcast explicitly encodes the destination addresses list in data packets instead of using a multicast address. It is actually a kind of stateless(or SGM) multicast protocol.

The authors have made certain assumptions in solving the small group multicast problem. These assumptions include:

- The multicast group is small, so that the list of group members addresses can be included in each data packet without incurring much overhead.
- Every member of a multicast group is aware of other members of the group. This membership control is explicit and can be done by application level mechanisms such as querying a well-known server or flooding queries in the network.
- The underlying unicast routing protocol is able to forward packets from source to destination along or close to the shortest path.
- Each group node is aware of the geometric location of its own, by using some type of location positioning system.
- Each node in a multicast group has the location information of all other nodes in the same group, hence a location update mechanism is needed.

The packet distribution tree is constructed explicitly with the goal of minimizing the overall bandwidth cost of the tree. The tree construction algorithms include a Location-Guided K-ary (LGK) tree and a Location-Guided Steiner (LGS) tree. Both of them utilize the geometric locations of the destination nodes as heuristics to compute the trees, and are accompanied by a hybrid location update mechanism to disseminate location information among a group of nodes. Our methods are similar to the location-guided Steiner tree but with a much less computation burden due to the predictability of the vehicle's mobility. Moreover the paper addresses the problems of the outdated and up-dated geographical information. They should also be discussed in our scheme as a system parameter.

2.3.4 PGB

PGB[22] is a broadcasting mechanisms that aims at:

1. reducing control messages overhead, by eliminating redundant transmissions;
2. obtaining stable routes with the ability to auto-correct. While a minimization of a routing load is very desirable in any of the possible ad hoc scenarios, the stability of a chosen route becomes especially important in an environment where fast moving vehicles are used as wireless hosts.

PGB solves all of the possible problems by allowing only certain nodes to rebroadcast a route request. This restriction significantly limits the set of possible intermediate nodes and in turn automatically reduces the routing load. PGB works in both cases: when the information about nodes coordinates is available and when not.

PGB classifies each node that receives a broadcast packet (e.g., route request) into one of the three groups, based on the sensed signal level:

- Preferred Group (PG) – the preferred set of nodes
- IN group – nodes with a signal stronger than in PG
- OUT group – nodes with a signal weaker than in PG

Combined with the position information, the node can determine whether it is in the PG or not by calculating the distance between the sender and the receiver or the received signal power. Only the nodes in the preferred group are allowed to rebroadcast the packet again.

2.4 Transport Layer

Cabernet is the only paper that deals with efficient downloading of data from the Access Point (AP). The proposed technique lies in the transport layer and the association of the AP.

In this paper[23], it states that network connectivity are both fleeting, intermittent and suffers from high packet loss rate. Their experimental results suggest that Cabernet is well suited to a large class of non-interactive applications without the modification and control of the AP. Two new components for improving open Wireless Fidelity (IEEE 802.11b wireless networking) (WiFi) data delivery are introduced: QuickWiFi and Cabernet Transport Protocol (CTP).

A streamlined client-side process to establish end-to-end connectivity reducing connection time from 10 seconds to 400 milliseconds. It combines all the different protocols involved in obtaining connectivity (across all layers) into a single process, including a new optimal channel scanning policy. Some basic ideas are:

1. Automatically associates the first open AP it encounters;

2. Reduces the timeout value to 100ms for the trials;

3. No need to authenticate, sends association request immediately after authentication;

4. Ping-through is used to test the connectivity to the Internet of the AP;

5. Reduces the time of determining whether the vehicle is out of the radio range to 500ms;

6. An optimized channel scanning sequences to find a new AP is used. It is based on the channel distribution of the AP which is acquired during the experiments.

On the other hand, to improve end-to-end throughput over lossy wireless links, it develops the CTP, which outperforms TCP over opportunistic WiFi networks by not confusing WiFi losses for network congestion. Unlike most previous work on efficient wireless transport protocols, CTP does not require modifications to AP; instead, it uses a lightweight probing scheme to determine the loss rate from Internet hosts to an AP. The main difference between CTP and a normal TCP socket is that a CTP session does not break when the underlying IP address changes or path disappears. Furthermore, from their experiments the packet loss rate is in excess of 20%. Therefore some of the methods in literature are not suitable in this high and busty losses. Finally, communication with legacy Internet hosts is done using a proxy that serves two functionalities: firstly, it hides intermittent connectivity from protocols running on fixed Internet hosts and shields them from the changed IP address; and secondly, it translates between CTP and popular protocols like Hypertext Transfer Protocol (HTTP) and TCP. It also tests connectivity of AP.

As in the same case of TCP, CTP has its own congestion control mechanism:

1. Data flow from a car to an Internet host: The sender receives immediate feedback from the absence of link layer ACKs; propagating this information to the CTP layer allows the sender to retransmit data without reducing the CTP transmission rate.

2. Data flow from an Internet host to a car: A special probing strategy is used to estimate congestion loss. Moreover the corresponding rate decrease rule is related only to the congestion loss.

On short duration of encounters, CTP performance twice better than TCP. However on high duration, it has the same performance as TCP. It can co-work with TCP.

2.5 Application Layer

Some of the interesting schemes lie in the application layer due to its simplicity. MobTorrent uses Wireless Wide-Area Network (WWAN) as the control channel to increase throughput. The cooperative downloading mechanism is introduced in two papers. SPAWN borrows the idea of Bittorrent into the wireless network.

2.5.1 MobTorrent

In this paper[24], the authors present MobTorrent, an on-demand, user-driven framework designed for vehicles which have intermittent high speed access to roadside WiFi. Therefore, instead of limiting high speed data transfer to the short contact periods between APs and mobile clients, high speed transfers among vehicles are opportunistically exploited. It has two main characteristics: pre-fetching on AP and scheduling algorithm of packets replication.

Mobile nodes in MobTorrent use the WWAN network as a control channel. When a mobile client wants to initiate a download, instead of waiting for contact with the AP, it informs one (or multiple) selected AP to pre-fetch the content. In the NS2 simulation, this part can be ignored and it may not be easily implemented using only the WiFi connection.

The scheduling algorithm in MobTorrent replicates the pre-fetched data on the mobile helpers so that the total amount of data transferred and the average transfer rate to the mobile clients are maximized. It addresses the problem of how to relay the data to a client via mobile helpers.

Two kinds of nodes are used to deliver the data to clients. One is the direct relay in the opposite driving direction. It downloads information from the AP and drives it to the client. Since it is in the opposite direction, the transmission may be failed. Then once having driven past the client, the information stored in direct relay is not available anymore. On the other hand, the other one is the forerunner nodes, which are before the client and remains in that position for a long time. The basic idea is that the relay nodes should replicate the data to the forerunner as soon as possible because the forerunner can hold that information longer than the relay nodes. The forerunner replicates the data to relays or indirect relays to transfer information to the client. Moreover the blocks with the lowest replication level are chosen to fully utilize the contact capacity.

The difference between MobTorrent and my scheme is the usage of the control channel. The MobTorrent uses the WWAN. My scheme uses 802.11b itself and we can not explore the direct relay's capability.

2.5.2 Cooperative Download for Mobile Phones

The concept of cooperative downloading from server can be used in my case to efficiently download content from the AP. Both of the following two papers use this idea.

In this paper, Perrucci[25] proposes an architecture where mobile phones are grouped together working cooperatively to download the web site contents to decrease the users' web browsing cycle. The basic idea is that they separately ask for different part of the content. After retrieving them, they exchange locally the data with each other by any short range communication. The paper argues that when the downloading phases of the two nodes are overlapped, the virtual rate will be increased by cooperative browsing.

In the case of using Bluetooth as the short range communication, the master in the cluster initiate two sublists of all the needed components and send one of them to the slave. Then both phones start to download and exchange information afterwards. The paper also states that the slave may not cooperate and a proxy server may be used to give better performance in terms of speed and energy consumption. By simulation the average duration of the downloading phase was reduce of the 47% if compared to the original one.

In the other paper, Militano[26] describes the same idea of cooperative downloading by two mobile phones. But it uses a different approach. It distinguish two cases: static server-aware & dynamic server-unaware. In the former case, the server prepares two different part and sends them to the corresponding nodes. In the latter case, the two nodes negotiate at first and they fetch different part of the contents using HTTP/1.1 Range Request.

Both of these two papers only assume the optimum case, where all nodes are cooperative. Besides they only show that the cooperation benefits a lot and give a simple scenario. However they do not propose a scheme to force cooperation or to increase the benefits of cooperation.

2.5.3 SPAWN

The paper[27] proposes SPAWN, a cooperative strategy for content delivery and sharing in VANET. It uses a gossip mechanism that leverages the inherent broadcast nature of the wireless medium, and a piece-selection strategy that takes proximity into account in decisions to exchange pieces. The main benefit of this scheme is that it restrains the message exchange within few hops. The main contribution of this paper is extending the current P2P protocols into VANET.

The Ad Hoc network extends and complements the Internet(the application field of Bittorrent) in the following items:

- Wireless communication is broadcast in nature.
- Every nodes acts as a router in multi-hop scenario. The data can be cached for later use.
- TCP throughput degrades a lot over multi-hop wireless connection.
- The churn degree is high comparing to the wired case because of the high mobility of the nodes.

The procedure is composed of the following two phases:

1. Peer Discovery
    - centralized approach: the gateway provides the peer list like Bittorrent.
    - gossip message: Due to the high churn rate, the node broadcasts in MAC layer the gossip message which contains its identification and the data it has. It also contains a list of nodes which possess the data. Actually three mechanisms of dealing with gossip messages are addressed: 1. probabilistic spawn. 2. rate-limited-recent spawn. 3. rate-limited-random spawn.
2. Peer Selection while BT uses tit-for-tat strategy, it uses proximity-driven piece selection strategy. For example, it selects the rarest piece among all closest peers.

2.6 Miscellaneous

In this section, we introduce some general papers that are relevant in my thsis topic.

2.6.1 Network Coding

Networking coding is the method of information packing. Network coding is a field of information theory and coding theory and is a method of attaining maximum information flow in a network. Therefore it can be used in P2P message exchange to maximize the throughput.

Lee[28] realizes for the first time the utility of random liner code of network coding for P2P file sharing system in mobile networks. It restricts the logical peers of P2P protocol to physical neighbors and assumes high randomness of the cars.

The paper designs an entirely new protocol to address all previous problems at once. In the protocol, the nodes exchange coded frames instead of file pieces. A coded frame is a liner combination of file piece. An encoding vector is used to generate code frame and stored in the header of frames for later decoding. The node keeps requesting neighbors to send coded frames until it collects N coded frames carrying encoding vectors that are linearly independent of each other. Due to the fact that the scheme only limits the transmission in one-hop. The overhead of the possible underlying routing protocol is minimized. Moreover higher mobility results in higher possibility of receiving independent coded frames. Thus the average download delay decreases as mobility increases.

Another paper[29] addresses the delivery of the safety warning messages, which should be disseminated in a reliable and timely way. Messages are targeted to the vehicles behind. Vehicles in the opposite direction on a highway are exploited as data mules, mobile nodes physically delivering data to destination, to overcome intermittent network connectivity cause by sparse vehicle traffic. It compares network coding, erasure coding and the simple repetition based strategies. It shows that the network coding based strategy outperforms erasure coding and repetition based strategies in the highway scenario.

2.6.2 LINGER

This paper[30] proposes a framework called LINGER. The main motivation is to dispatch and confine information in a localized area, a Geocast case. The information bearers are responsible for driving the information to the target area and keeping the information linger within the area. The selection of information bearers is performed in a distributed and cooperative way. A single metric referred to as lingering index is used to select the most suitable information bearer.

Three basic steps of the framework are:

1. The source generates the message with the geographical coordinate and the target area defined as the circular region of radius R.

2. The moment for transferring information to new information bearer is when lingering index is negative in order to achieve low overhead with minimum number of object handovers.

3. The destination of the information is always set as the node with the highest lingering index. When a node finds its lingering index is negative, it broadcasts one-hop POLL message to find the best candidate. All neighbor nodes which have received the POLL message respond with BID message at the time decreasing with the difference between local lingering index and the received lingering index. Therefore the node with the highest lingering index

will transmit BID message first. Then the object is transferred to this node.

As regards as the ingering index, the paper outlines several requirements and purposes that the lingering index should serve, identifying appropriate maths functions of localization parameters that realize those requirements. The paper points out three contributions which should be considered:

1. the distance away from the target
2. the heading direction of the car
3. the speed of the car

The paper then defines three functions each of which identifies the above one item. Finally by adding up the three contributions, the lingering index function is formulated.

2.6.3 To Cache or Not To Cache

This paper[31] addresses cooperative caching strategy in MANET where information is changed in P2P fashion. When a node receives some information from its counterparts, it will decide whether to store the information or not and for how long independently from other nodes just considering only content query and information reply messages that a node sees on the channel. Therefore no any additional control messages are needed. The objective is to create a content diversity in a fully distributed scheme so that users likely find a copy of the different information items nearby and avoid flooding the network with query messages.

This paper has a very important assumption. That is: A cache-all-you-see approach is unfeasible. The capacity of users' storage is limited. Otherwise, no caching is needed if there is an umlimited size of memory.

The feature of the content retrieval system is:

1. Upon a request generation, the node broadcasts a query message for the chunks.
2. If the node which receives the query message has the information, the information is sent back to the source of the request in a unicast fashion. Otherwise it can rebroadcast a query for the missing chunks. Notice that only the source of the request can be entitled to cache the copy.
3. From the previous overheard message in the network, the node can determine the *information presence* in its proximity. Obviously the lower information presence is, the higher the cache drop time is and vice versa.

The main part of the paper then describes how to estimate the presence of information chunks in its proximity and how to compute the cache drop time.

1. Information Presence estimation is calculated mainly by the hopping distance between the nodes. Two counters are stated:

   - Provider Counter: It accounts for the case when the node acts as the provider. In that case, the information is stored at the receiving node.
   - Transit Counter: It accounts for the case when the node overhears or receives transiting message. In that case, the information is stored both at the transmitting and receiving node. Both contributions should be considered. The paper also addresses the case when the responding message is overheard.

   Finally, all of the above contributions are added as the information presence estimation which has a higher bound of value one.

2. Cache Drop Time. The presence index is only valid till one of the nodes who has the information drops it after the cache drop time expires. The paper assumes that the cache drop time of the neighbors is not too different from its own estimation. Thus it uses local estimation as the validity of the presence index. In order not to lead to an on/off behavior and yields discontinuities, a filter is used to sum and smooth the contributions.

2.7 Conclusion

This chapter summarizes the relevant papers about content distribution in VANET. For one thing, not all the papers are directly related with the scheme. The ideas behind them are interesting and worth to mention because they still target at the same topic of the thesis. For another, we borrow some ideas. MobTorrent [24] is very similar to our objective. However it uses the control channel to pre-fetch the content, which we don't have in our proposed scheme. Besides it evaluates the performance in real scenario. From [25] and [26], the idea of cooperative downloading is used in the proposed scheme. Finally, the method to disseminate the contents in V2V sharing phases comes from PGB [22] and LINGER [30].

Chapter 3

Scheme

This chapter presents the scheme. Some statements and assumptions are stated firstly followed by the scheme outline. Then the state and message type summaries are provided to facilitate the explanation of the scheme. The scheme is composed of two stages: the cooperative downloading of content from the AP and the V2V sharing among the vehicles. Then these two parts are separately described.

3.1 Statements and Assumptions

Before describing the scheme, some statements should be made, which are important to the description of the scheme. They are:

- All vehicles work in **overhear** mode by default. They can overhear the packets that are not destined to themselves in order to achieve maximum data throughput in the wireless channel.

- All the packets are transmitted in **unicast** manner by default. If the packets are broadcast messages, the scheme explicitly states that.

- Both the vehicle and the AP can use **MAC layer ACK message** notification to eliminate the unnecessary ACK messages in the application layer because a correct transmission in IEEE 802.11 MAC layer standard includes an ACK message sent by the receiver. However only when the sender and receiver are in one-hop communication range, this cross layer optimization can be used.

- The vehicle can determine whether it is inside the **AP's coverage range** or not. Usually AP broadcasts the BEACON message to advertise its existence. The vehicle can find itself inside an AP's coverage range when it receives the BEACON message from the AP. On the contrary it knows that it is outside the coverage range when it has not received the BEACON message for some time.

Additionally the group management protocol is ignored because it is an independent part of the overall scheme. However it is an important **assumption** when describing the scheme. It is mainly due to the limited time and effort of the thesis work.

Later work can be devoted to add the group management protocol which takes before and during the proposed scheme. Some detailed assumptions are listed below:

- *First of all*, all vehicles know the group membership ID. That means the vehicles know that how many group members there are in the group, their identifications and the header of the group.

- *Secondly*, all vehicles know the position and velocity of other group members. The position and velocity are provided by GPS receiver installed in each vehicle and are exchanged among the vehicles. Moreover, the position and velocity update are done every half or one second. There is an extra delay to disseminate these messages among the vehicles. This delay is dependent on the group size, the group topology and the message dissemination approach.

- *Thirdly*, it is assumed that the group is fixed during the scheme operation. Although it may be true that new vehicles can join the group during the content distribution, this simplification does not have much impact on the performance of the scheme. However when some vehicles leave the group, the performance would decrease.

- *Lastly*, all the members in the group are interested in one content. Actually, if vehicles are interested in more contents, they can form multiple groups.

3.2 Scheme Outline

The main aim of the thesis is to design a scheme that can download as much as possible and as quick as possible the contents for the vehicles. The vehicles should fully utilize the chance when they can download the contents from the AP because it is the only source of the content. Thus the basic idea is to cooperatively download the contents from the AP. After this cooperative downloading phase, the vehicles try to share among themselves all the contents they have downloaded from the APs. The vehicles ask for the missing contents from the other vehicles in the group.

Therefore, the scheme is composed of two separate phases. The first one is the cooperative downloading phase of the vehicles from the AP. Actually since all the vehicles can communicate with each other, only one copy of the packet is necessary for the group. Thus the idea of the cooperative downloading from the AP is to send sequentially all the contents to the group. AP directly sends the next packet to the new vehicle header without any handshake provided that the previous vehicle header can not receive message anymore. In this way, exactly at least one copy of the contents is sent to the group in the quickest way. There may be multiple copies of messages in the group since the vehicles can overhear packets.

The second one is the V2V information sharing phase among the vehicles. The vehicles send advertisement messages to tell other vehicles the packets they have. After negotiations, the sender unicasts the messages which contain the payload data to the receiver. All other vehicles can overhear the unicast message and possibly get the contents they are lack of. We use PGB [22] similar methods to distribute the contents inside the group as quick as possible.

The last problem is the transitions between these two phases. On one hand, the vehicles have to wait some time to begin the V2V sharing phase. It is reasonable that the vehicles should start to share the contents as soon as they cannot download packets from the AP. However in order not to interfere other vehicles' cooperative downloading phases, they have to remain silent. On the other hand, the vehicles should directly go to cooperative downloading phase provided that they need more packets from the AP.

3.3 State Summary

The following words summarize all the states to make the scheme description sections more clear.

IDLE_STATE

It is the starting state of all nodes.

DOWNLOAD_STATE

The vehicle downloads content from one AP or the AP sends contents to one vehicle.

COVERAGE_STATE

The vehicle is near the coverage of one AP.

TRANSFER_WAIT_STATE

The vehicle is outside the range of all APs and it waits for the transmission of ADVER message, which starts the V2V sharing phase.

CTS_WAIT_STATE

The vehicle has successfully sent out the ADVER message. Now it waits for reply of CTS message from other vehicles.

TRANSFER_CTS_STATE

The vehicle has received one CTS message and it still waits for possible subsequent CTS messages.

TRANSFER_SEND_STATE

The vehicle serves as the source in V2V sharing phase. It sends out the data packet.

TRANSFER_RECEIVE_STATE

The vehicle serves as the sink in V2V sharing phase. It receives the data packet.

FINISH_STATE

It is the end state of all nodes. It means that the nodes either have sent out (in the case of AP) or have downloaded (in the case of vehicle) all the contents.

3.4 Message Type & Timeout Policy

3.4.1 Message Type

There are six application layer messages in the scheme. They carry different fields according to their roles in the scheme. Their size differs too.[1] The formation of the messages is summarized in the following tables.

REQUEST

REQUEST(Table 3.1)is used for requesting content from AP. Since it represents the whole group and there is no overlapping of different APs, the sender and receiver information are unnecessary. However it is the only message type which needs the *Group Information*.[2] This message is needed for AP to select the next header in the group and the packet number from which AP should send to the group.

Sender: Header Vehicle in the Group

[1]Their size is the concern of the implementation and performance evaluation, which are addressed in the chapter 4 and chapter 5.

[2]It is described in the next section.

Type	Tx ID	Rx ID	Payload	Seq No	Pkt Status	Group Info.
√						√

Table 3.1. REQUEST Message Formation

Receiver: AP

TRAFFIC

TRAFFIC(Table 3.2) has payload and the sequence No. fields because the vehicle can overhear packets. Furthermore it is a unicast message with sender and receiver information.

Type	Tx ID	Rx ID	Payload	Seq No	Pkt Status	Group Info.
√	√	√	√	√		

Table 3.2. TRAFFIC Message Formation

Sender: AP
Receiver: Header Vehicle in the Group

TRANSFER

TRANSFER(Table 3.3) is the same as the above TRAFFIC message except the type field.

Type	Tx ID	Rx ID	Payload	Seq No	Pkt Status	Group Info.
√	√	√	√	√		

Table 3.3. TRANSFER Message Formation

Sender: Vehicle Sharing Sender
Receiver: Vehicle Sharing Receiver

ADVER

ADVER(Table 3.4) is used for advertising the packets that the sender has. Firstly, it is a broadcast message so it does not have the receiver ID. Secondly, the sender is needed for the possible receivers to respond with the CTS message. Lastly, the important field in this message type is the packet status field. It specifies the packet that the sender has. Obviously, the size of this field is variable. Different compression techniques could be applied here to reduce the size.[3]

Type	Tx ID	Rx ID	Payload	Seq No	Pkt Status	Group Info.
√	√				√	

Table 3.4. ADVER Message Formation

Sender: Vehicle
Receiver: Any Vehicle[4]

CTS

CTS(Table 3.5) is used as the reply for the broadcast ADVER message. It is a unicast message so that the receiver ID is in the message type. Although it also has packet status, its meaning is different. Here it means the packets that the sender needs.

Type	Tx ID	Rx ID	Payload	Seq No	Pkt Status	Group Info.
√	√	√			√	

Table 3.5. CTS Message Formation

Sender: Vehicle Sharing Receiver
Receiver: Vehicle Sharing Sender

FINISH

FINISH(Table 3.6) ends the transfer transaction among the vehicles. Since there maybe multiple receivers in one transaction. This message is a broadcast message too. Only sender field is needed.

Type	Tx ID	Rx ID	Payload	Seq No	Pkt Status	Group Info.
√	√					

Table 3.6. FINISH Message Formation

Sender: Vehicle Sharing Sender
Receiver: Vehicle Sharing Receiver

[3]Two methods are discussed in the following section.
[4]It is a broadcast message, any vehicle can receive it.

3.4.2 Packet Status Field

The packet status information is used in two message types: CTS and ADVER. In CTS message, it means the packets that the sender needs. While in ADVER message, it denotes the packets that the sender has.

Bit Notation Method

One bit in the packet payload data can be used to imply the status. When this bit is one, it means that the corresponding packet is set. When this bit is zero, it means that the corresponding packet is not set. Therefore the size of the packet status follows the equation: $size(bytes) = \frac{TotalFileSize}{PacketSize*8}$. However if the file size is very large, the packet status field can be large too. For example, for a file with size 10 Mega bytes and packet size of 512 bytes, the packet status size is $\frac{1024*1024*10}{512*8} = 2560Bytes$.

Rung Length Encoding

Run-length encoding is a possible candidate to reduce the size. Run-length encoding (RLE) [32] is a very simple form of data compression in which runs of data (that is, sequences in which the same data value occurs in many consecutive data elements) are stored as a single data value and count, rather than as the original run.

In our scenario, the packet receipts are consecutive either from AP or from other vehicles. Moreover there are only two possible data values: 0 or 1. We only have to indicate the first data value and then follows the counts. The data compression ratio could be very high. For example, if the vehicle has the packet from No.50 to No.100 and the total packets number is 1000, then the coding is:

```
(0),(50,50,900)
```

The value in the first parenthesis is the initial value. The initial value is zero because the node does not have packet No.1. Then the first 50 means that the zero value lasts for 50 times. Then it follows 50 ones from packet No.50 to No.100. Finally, it ends with 900 zeros.

To sum up, we need only 16 bytes using RLE method [5] in this case instead of 125 bytes when we use bit notation method.

3.4.3 Group Information Size

The field and its corresponding size is in the following Table 3.7.

The main part is the GPS information, which includes the coordinates of the nodes and its velocity vector. As can be seen, the size of individual nodes is 17 bytes. Therefore the total size is 17 bytes multiplying the group size.

[5] If each field is represented by Integer type

name	ID	x	y	speed modules	speed angle
size	1	4	4	4	4

Table 3.7. Group Information Size

3.4.4 Timeout Update Policy

AdverTimer

The timeout value for AdverTimer is randomly chosen between an interval from 0 to T. The timeout value is chosen randomly because the node stays some time in the CTS_WAIT_STATE to wait for CTS replies. If the timeout value is a fix value, there would be a dead lock situation. For example, node A transmits ADVER message when node B is in CTS_WAIT_STATE; node B transmits ADVER message when node A is in CTS_WAIT_STATE. In this case, no transfer would ever happen because node does not respond with the ADVER message.[6]

Furthermore, the value of T is doubled every time it hasn't received any CTS reply. It behaves like the mechanisms in 802.11 CSMA_CA.

CtsTxTimer

The timeout value of this timer is inversely proportional to the distance between the two vehicles so that the farthest vehicle transmits the CTS message firstly.

3.5 Cooperative Downloading Phase

This section presents the cooperative downloading of the content from the AP.

Before arriving at the AP, the vehicles exchange their position and velocity information. They are all interested in one content. Therefore they form a group. According to the group management protocol, a group header is selected. As what have been said above, the group management protocol is not considered.

The header is chosen such that the group can download maximum data from the AP. The header of the group could simply be the vehicle which will firstly enter the coverage range of the AP. However other mechanisms can be considered also, such as the channel condition between the header and the AP, the stay time of the header in the AP's coverage, etc. However all of these are irrelevant with this scheme and are considered in the group management protocol.

[6] Actually, we can use two threads. One is sending thread and another one is receiving thread. This should be the future work of the thesis.

There are two type of nodes: AP and vehicle. The rationale and the working details are described in the following subsections.

3.5.1 AP State Flow

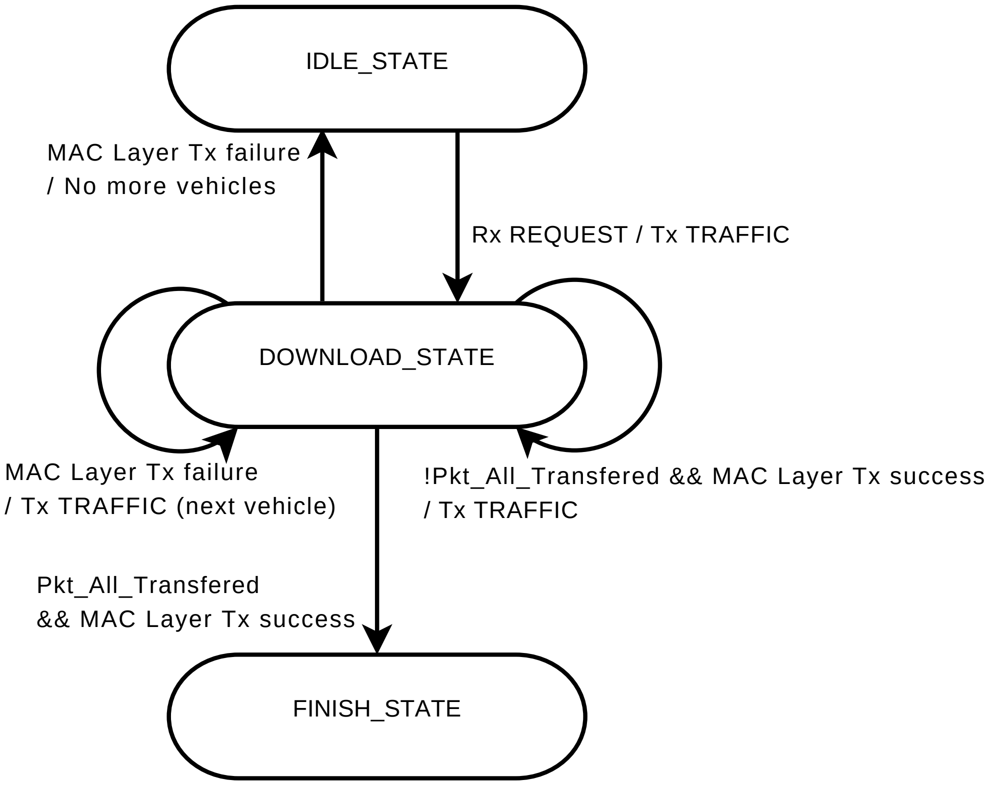

Figure 3.1. AP State Flow

The main role of AP is the data gateway of the network. AP only transmits data packets to the vehicles and does not receive any data packet from the vehicles.

Moreover, every data exchange transaction needs initial handshake procedure. It can save a lot of time if this initial handshake procedure is optimized. Fortunately, thanks to the group management protocol, only one initial handshake procedure is needed for the whole group.

The state transition is in Figure 3.1 and the detail of the scheme is presented below:

- When the header enters the coverage of the AP, it initiates the downloading from the AP by sending the REQUEST message. After receiving the REQUEST message, the AP sends the TRAFFIC message with the payload data to the header of the group.

- Then the AP starts the transmission of next TRAFFIC payload data if it correctly receives the ACK message at the MAC layer. In contrast, if the AP does not receive ACK message at the MAC layer after maximum retry times, the AP starts the transmission of the next payload data directly to the next vehicle header in the group.[7]

- When all data packets that the group asks for have been successfully downloaded to the whole group, the AP enters the FINISH_STATE.

3.5.2 Vehicle

In this subsection the state diagraph of the scheme from the viewpoint of the vehicle is introduced. Although the vehicle could or could not be the header, they are the same in the diagraph.

First of all, the reason to include COVERAGE_STATE in this scheme is to suppress the data transmission when one vehicle just leaves the coverage range of the AP. When the vehicle leaves the range, its broadcast transmission could arrive at the vehicle which is still inside the coverage range of the AP. If the vehicle is inside the transmission, it may be receiving the traffic packet from the AP. Therefore a collision occurs in this node. If we want high number of received packets, the vehicle should remain silent for some time.

However the introduction of this state could decrease the distribution time of the content. So it is a pair of trade-offs between timely distribution and high number of received packets. In this scheme, the timeout value, which determines resident time in this state, is configurable. This timeout value can be decreased when timely distribution is more important than high number of received packets.

The state transition details are as follows in the Figure 3.2:

- When the header enters the coverage range of the AP [8], it transmits the REQUEST message, goes to DOWNLOAD_STATE and the TrafficTimer is started. In this state, the vehicle resets the TrafficTimer every time it receives TRAFFIC message from the AP. If it does not receive any TRAFFIC message after TrafficTimeout, it means that it leaves the coverage range. Then it goes to COVERAGE_STATE and starts the SharingTimer.

- The V2V sharing among the vehicles may collide with the transmission between the AP and the vehicle when the V2V sharing phase happens near the coverage range of the AP. From my initial simulation results, the bottleneck

[7]The choice of the next vehicle header to send the TRAFFIC payload data is also the issue of the group management protocol. The AP can choose the next vehicle which is the nearest to it. However other measurements could be used as well.

[8]When it receives a period BEACON message from that AP.

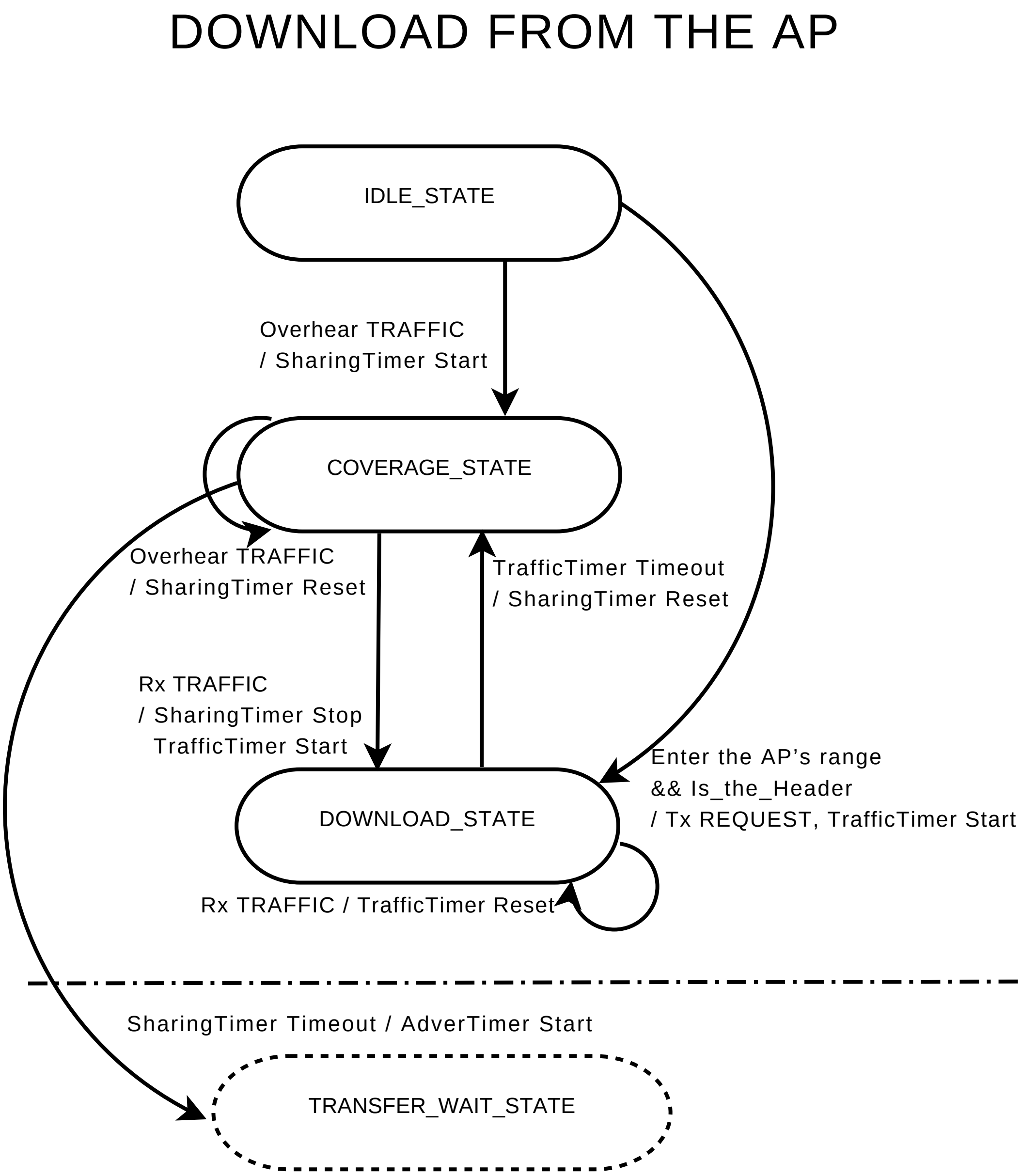

Figure 3.2. Vehicle Downloading

of the system could be the data downloaded from the AP because AP is not densely deployed. So the COVERAGE_STATE is added to increase the number of received packets. When entering this state, the SharingTimer is started or reset. The vehicle enters COVERAGE_STATE when it overhears TRAFFIC message and resets the timer if it overhears TRAFFIC message again. If it does not receive any TRAFFIC message after SharingTimeout, it starts the

V2V sharing phase which is detailed in the following section.

- The vehicle enters the DOWNLOAD_STATE from COVERAGE_STATE when it receives TRAFFIC message[9]. Then the TrafficTimer is started and the SharingTimer is stopped.

3.6 V2V Sharing Phase

3.6.1 Rationale

The main problem in the P2P sharing is the peer selection and the content advertisement. Which peer should be chosen as the receiver and what part of content does each peer have? We have to solve the same problem in our case as well.

1. This V2V sharing scenario is a pure P2P topology **without any central point for coordination**. The first question to be answered is the advertisement of the packets. Message flooding seems to be the only approach because of the lack of the central point. Furthermore, due to quick changes of topology, forwarding out-of-date advertisement messages is unnecessary. If the vehicle find the content it is interested in, the routing path from the source has probably changed during that time. Therefore, broadcast message is used but only in **one-hop** range for the advertisement.

2. We can explore the broadcast property of wireless transmission. All vehicles within the one-hop transmission range of the transmitter can receive the packet. As what has been discussed in the PGB[22] method, the vehicle which is in the preferred group should rebroadcast the packet because in this way the packets can be disseminated fast and efficiently. Only the vehicle in the preferred group can actually initiate the transfer of the packets and act as the receiver of the packet while other nodes inside the one-hop transmission range can overhear the packets.[10]

The following subsections detail the operation in two parts: V2V sharing send part and V2V sharing receive part.

3.6.2 V2V Sharing Send Part

This is the sharing send part where the vehicle acts as the source of the contents.

[9] It is selected as the next vehicle by the AP.

[10] There is a problem when only the node that is not inside the preferred group needs the packet. However the possibility is not high due to the cooperative downloading scheme of the content from the AP.

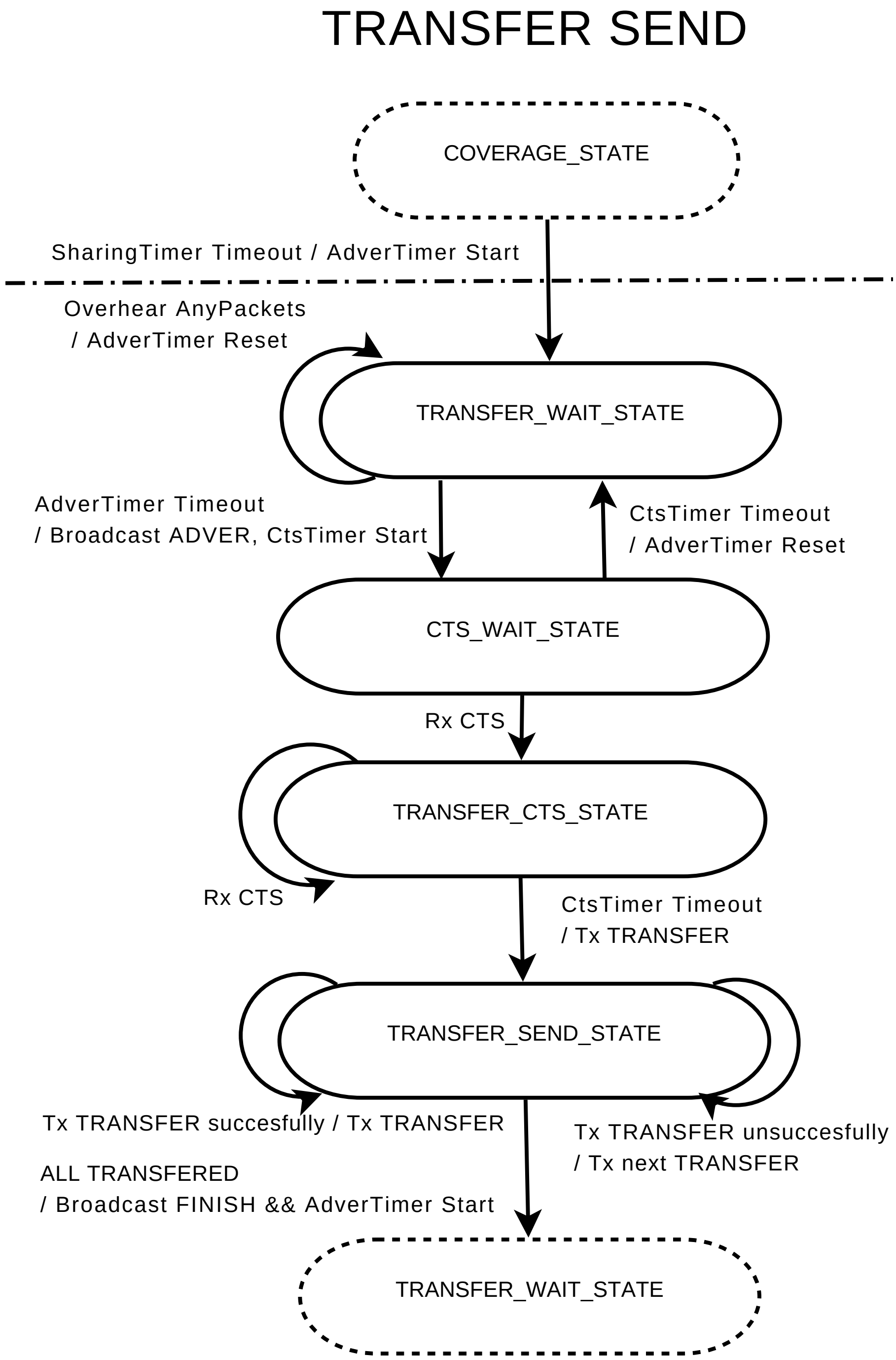

Figure 3.3. Vehicle Sharing Send Part

The principle of the cooperative communication is selecting suitable relays to forward packets to the receivers. If the PGB similar mechanism is used, there may be multiple forwarders. One case is that one forwarder is in the front of the sender and the other one is behind the sender. Both forwarders do not know the existence of each other. So the sender waits to collect all the possible replies. As regards other nodes within the transmission range, because they are nearer to the transmitter the possibility that the packets are not correctly received in these nodes are extremely low.

The details are as the follows and shown in the Figure 3.3.

- After sensing the channel free for SharingTime in COVERAGE_STATE, the vehicle is sure that it is relatively far way from the coverage range of the AP. Then it enters the TRANSFER_WAIT_STATE and starts AdverTimer.

- When the vehicle senses that the channel is free for AdverTime in TRANSFER_WAIT_STATE, it broadcasts ADVER message. The ADVER message contains the information of the packets the sender has. Then it goes to CTS_WAIT_STATE waiting for the CTS message from the possible forwarders. However if it overhears any packet in TRANSFER_WAIT_STATE, it resets the AdverTimer again.

- If no CTS message is received in CtsTimeout time in CTS_WAIT_STATE, the vehicle goes back to TRANSFER_WAIT_STATE and waits for the next time interval to send again the ADVER message.

- If CTS message is received, the vehicle goes to TRANSFER_CTS_STATE. It can also receive CTS message in this state because more CTS messages could be sent from the possible forwarders. After CtsTimeout time, the vehicle enters the TRANSFER_SEND_STATE.

- TRANSFER_SEND_STATE is the actual data transfer state. By analyzing all the CTSs from possible receivers, the sender can determine the packets it should send. It is possible that both the receivers need the same packet. In this case, only one unicast message is sent to one of the two receivers because the other receiver can receive the packet for sure. The vehicle only records the first received CTS message. Therefore the vehicle which firstly sent the CTS message is chosen as the receiver.[11]

- After all the data have been sent, the vehicle broadcasts FINISH message to explicitly notify the end of the transfer.

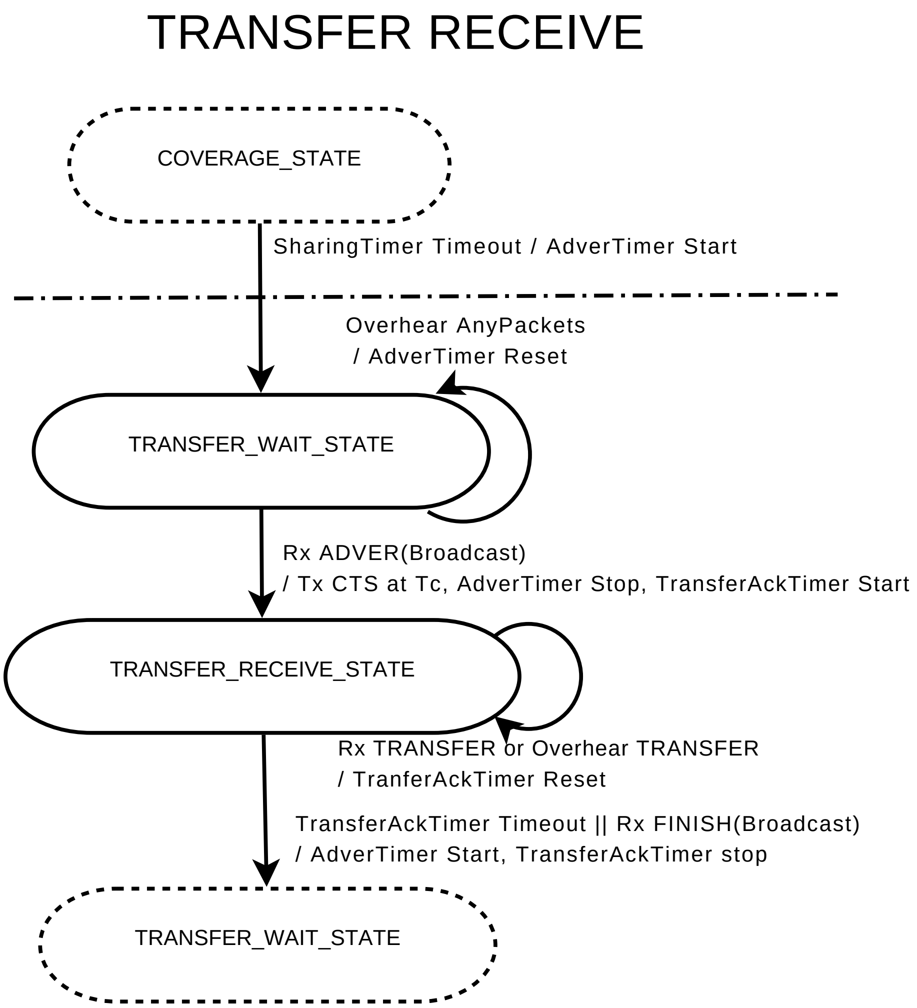

Figure 3.4. Vehicle Sharing Receive Part

3.6.3 V2V Sharing Receive Part

This is the V2V sharing receive part where the vehicle acts as the destination of the packets. The state transition is in Figure 3.4. There are some duplications on the state flow compared to the send part. Therefore only the differences are described in the following.

- When the vehicle receives ADVER message in TRANSFER_WAIT_STATE, it stops AdverTimer and starts TransferAckTimer. Then the forwarder enters the TRANSFER_RECEIVE_STATE by sending CTS message at Tc. Moreover

[11]Notice that, no matter whether the message is received or not, the sender would send the next packet.

Tc is a random value. Otherwise multiple CTS message could arrive at the transfer sender simultaneously. Tc is inversely proportional to the distance between between the vehicles so that the farthest vehicle is chosen as the receiver. In this case, more vehicles can overhear the unicast message.

- In TRANSFER_RECEIVE_STATE, the vehicle receives the TRANSFER message. No ACK message in application layer is sent. Therefore, when either the vehicle receives FINISH message or TransferAckTimer is expired, it goes back to TRANSFER_WAIT_STATE.

3.7 Transition From Sharing to Downloading

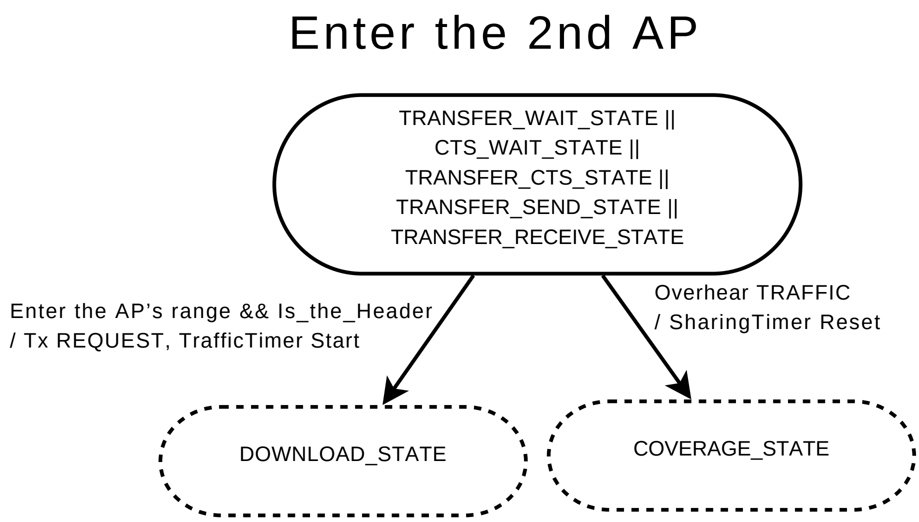

Figure 3.5. Transition From Sharing to Downloading

The transition from downloading to sharing is included in the above two sections. This section deals with the transition from sharing to downloading.

We give a higher priority to downloading in our design. The vehicles return to their states in cooperative downloading phase as stated in the Figure 3.5. Due to the higher priority of the cooperative downloading phase, if the header of the group enters again the AP's coverage and it still needs data to download, it jumps to DOWNLOAD_STATE directly no matter what state it is currently in. The vehicle header sends the REQUEST message and does the normal downloading operation. Moreover, TrafficTimer is stared in DOWNLOAD_STATE

All other vehicles whose transmissions may collide with the downloading phase enter COVERAGE_STATE. If it overhears any Traffic packets that are related with the cooperative downloading phase, it enters the COVERAGE_STATE. Furthermore, SharingTimer is started in COVERAGE_STATE.

Chapter 4

Implementation

In this chapter, the implementation details of the scheme are demonstrated. Every simulator has its own merits and weaknesses, let alone NS2. This chapter firstly introduces some limitations of NS2. Then the details of the scheme implementation are elaborated. Some lower layer modifications are mentioned as well because NS2 is a rather messy open-source software. The last part is about the simulation scenario setup.

4.1 Issues on NS2

Simulators always need assumptions to make their calculations viable. There is a maintained list of limitations in the NS2 website [33]. However the actual limitations found[1] during the simulation are far more beyond that list.

4.1.1 Simplifications

Notably, in NS2 the wireless extension, derived from CMU Monarch Project [34], has two assumptions simplifying the physical world, which are significant in the VANET scenario:

1. Nodes do not move significantly over the length of time they transmit or receive a packet. This assumption holds only for mobile nodes of high-rate and low-speed. In the VANET scenario, the transmission rate could be as high as 11Mb/s. But the nodes move at a high-speed.

2. Node velocity is insignificant compared to the speed of light. In particular, none of the provided propagation models include Doppler effects, although they could.

[1] Mainly by surfing on the Internet and using Google

These assumptions are in the physical layer and the problems become more severe when the speed is much higher. If we want to evaluate the performance enhancement when we modify some parameters of physical or MAC layer, we should take into account these simplifications. Fortunately, although the actual throughput may be different from the simulation throughput, we are more interested in how many more packets can be downloaded in the new scheme compared to that in the standard scheme.

4.1.2 Limitations

This subsection talks about the limitations which could be implemented but not at the moment in NS2 simulator. Of course these limitations are relevant with our scenario.

1. Currently IEEE 802.11b standard is used instead of IEEE 802.11p standard due to the limitations in NS2. WirelessPhyExt and 802 11Ext in ns-2.33 support 802.11p parameter settings. However there are some problems of the underlying routing protocols. The routing is built after a long time.[2]

2. NS2 supports infrastructure mode for IEEE 802.11 in the current version 2.33. Beacon frame, Scanning, Authentication and Association functions have been implemented. But AP is mainly used for forwarding data in the BSS. The messages sent to the AP are not delivered to the upper layer.

As regards the first limitation, at the moment we can not have any better solution. The only way is to use the newest release of NS2.

Concerning the second limitation, the AP is simulated as a "mobile node" in NS2 which remains static in the whole simulation time. Since the AP is the mobile node. Three further issues should be addressed as well.

1. The AP should use routing agent as well to establish the connection with the other vehicles. In our implementation, AODV routing protocol is used. Thus some possible modifications are needed to make the routing agent work correctly in this scenario. [3]

2. The beacon frame is not sent by the AP. Consequently, the vehicle can't sense the existence of the nearby AP. In my implementation, the solution is to use an independent C++ class to update the distance between the vehicles and the AP. Once the distance is smaller than the coverage range of an AP, this C++ class can inform the vehicle the existence of the AP.

[2]Actually it is a bug in ns-2.33 version. It is fixed in the newest ns-2.34 version when this thesis is almost finished.

[3]More details are in the lower layer modifications section of this chapter.

3. The scanning, authentication and association functions are not simulated if the AP is a mobile node. Therefore, in order to make the simulated throughput be closer to the actual throughput. We have to make an assumption to consider the time spent in these three functions. However the time can only be estimated as an approximation. We can reduce the radio range of the AP or use an extra timer.[4]

4.1.3 Packet Size

In general, a packet consists of a packet header and data payload. Packet header stores packet attributes (e.g., source and destination IP addresses) necessary for packet delivery, while data payload contains user information. Although this concept is typical in practice, NS2 models packets differently.[35]

In most cases, NS2 extracts information from data payload and stores the information into packet header. This idea removes the necessity to process data payload at runtime. For example, instead of counting the number of bits in a packet, NS2 stores packet size in variable *hdr_cmn::size_*, and accesses this variable at runtime.

Given a pointer to a Packet object p, the following statement sets the packet size to be "my_size"

```
hdr_cmn* hdr = hdr_cmn::access(p);
hdr->size() = my_size;
```

We can simplify some implementations in NS2 because of the above mechanisms.

1. We can add any specific field in the Packet class as long as the `size_` field is not changed. We use this method to pass the pointer of the application instance to the lower layer to achieve cross layer operation.

2. The data fields of the packet header are the same for all packet types. But the size is different for different packet type because they only needs one portion of all the fields. So we only have to set the size to different value. No individual message struct definition is needed for every type of message.

3. As regards the compression, the actual data can be directly passed to the receiver. What we have to do is to set the size of the data to the size after compression.

[4] However the biggest problem is to set the value of the timeout and the range. In some literature, it says that it could take tens of seconds to setup the connection with the AP.

4.2 Application Layer

4.2.1 Overall Architecture

The whole system is named as Vehicle Content Downloading (VCD). The vehicle is implemented as *VCDClient*; the AP is implemented as *VCDServer*. All of them are in application layer. They are both inherited from the *VCDBase* class. There are only two virtual functions which are used for cross layer operation. The class hierarchy is in the Figure 4.1.

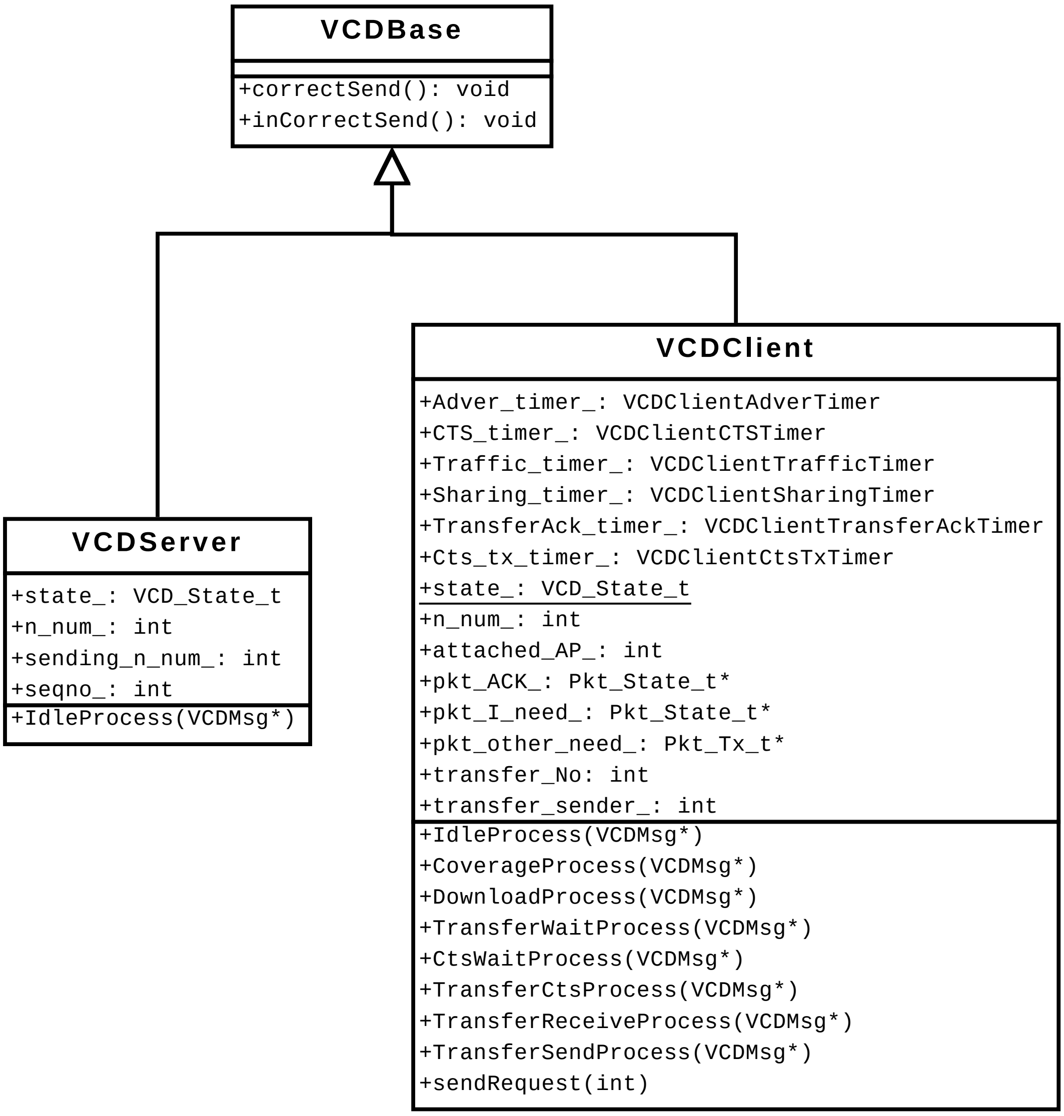

Figure 4.1. Class Hierarchy

The `correctSend()` and `inCorrectSend()` functions are called according to the packets' sending status.

Every class has a varialbe called `n_num_` which is the identification number. `state_` is used to specify the current state of the node.

The most important function is the *STATEProcess* function, where *STATE* means the current state of the node.

Generate Structure Inside *process_data* Function

The *process_data* function is called by the lower agent to pass the message to the application layer. This is the place where the scheme mainly handles the received messages. Then the different process function is called with respect to the current state. For example, if the current state of the class is DOWNLOAD_STATE, then the corresponding `DownloadProcess(VCDMsg*)` function is called. Inside every *STATEProcess* function, the software flow is as follows:

StateHandler

```
//Main State Transition Structure
if (data->getReceiver() == BROADCAST) { // BROADCAST MSG
  case MESSAGE_TYPE_1:
  case MESSAGE_TYPE_2:
  case MESSAGE_TYPE_3:
  ... ...
  default:
}
else if (n_num_ != data->getReceiver()) { // OVERHEAR MSG
  case MESSAGE_TYPE_1:
  case MESSAGE_TYPE_2:
  case MESSAGE_TYPE_3:
  ... ...
  default:
}
else { // MSG to this node
  case MESSAGE_TYPE_1:
  case MESSAGE_TYPE_2:
  case MESSAGE_TYPE_3:
  ... ...
  default:
}
```

4.2.2 VCDServer

There is only one *STATEProcess* function in *VCDServer* class, which is the `IdleProcess(VCDMsg*)` function. In this function, only the message type of REQUEST is acceptable.

On one hand, in the `correctSend()` function, the next packet is sent to the same vehicle. On the other hand, in the `inCorrectSend()` function, the next header of the group is selected by the AP. Here the nearest vehicle is chosen as the next header in the group.

`sending_n_num_` is the identification of that vehicle because AP can only send to one vehicle at a time.

4.2.3 VCDClient

The structure of *VCDClient* is much more complex than that of *VCDServer*. Besides the `IdleProcess(VCDMsg*)` function, there are other similar state transition functions.

Furthermore, six additional timers are present in the *VCDClient*. Their functionalities can refer to the scheme description in chapter 3. After timeout, the corresponding timeout is called. The following is the definition of an example timer.

Timer
```
class VCDClientCtsTxTimer : public TimerHandler
{
public:
    VCDClientCtsTxTimer(VCDClient *app) {
        app_ = app;
    }

    void expire(Event *);

protected:
    VCDClient *app_;
};
```

The pointer to the application is assigned when the timer is created. Then the `void expire(Event *)` function is called by the simulator when the timer reaches the timeout time. So inside the expire function, we call the corresponding function to handle the timeout event. There is also a similar structure of handling timeout event. The code does different operations according to the current state.

`sendRequest(int)` function is called by the GroupManager Class, which is detailed in the following section. GroupManager Class calls this function through the

OTcl domain. Therefore the VCDClient class should override the `command` function to accept the sendRequest command from the OTcl script, which is called by GroupManager, as follows:

Command Acceptance
```
int VCDClient::command(int argc, const char*const* argv)
{
    if (argc == 3) {
        if (strcmp(argv[1], "sendRequest") == 0) {
            sendRequest(atoi(argv[2]));
            return(TCL_OK);
        }
    }
    return (Application::command(argc, argv));
}
```

Three packet status related pointers are defined. `pkt_ACK_` is the packet receipt status of this *VCDClient*. `pkt_I_need_` is used for the building of the CTS message in the case of receiver in the V2V sharing phase. `pkt_Other_need_` is updated when new CTS is received in the case of sender in the V2V sharing phase.

4.2.4 Input Parameters

A specific class *InputParameter* is used solely for the input of general simulation parameters such as the timeout value, each packet size and total file size. Although these parameters can be moved to the individual *VCDClient* and *VCDServer* class, some of them are the same, such as the file size and the group size.

The parameters are defined as the static member variables of the class such that other class can access these variables without referring to one specific instance of the class. *InputParameter* class should be derived from the basic *TclObject* class in order to be accessed from OTcl domain. The definition of this class is as follows:[5]

InputParameter
```
1 class InputParameter : public TclObject
2 {
3 public:
4     InputParameter();
5 
6     static double gps_interval_;
7                 .... .... ....
8 };
```

[5] Other parameters are declared from line 7.

The mechanisms about how to bind static C++ class member variables is provided in the manual [6]. Two extra methods `bind()` and `method()` are added in the derived class;

```
static class InputParameterClass : public TclClass
{
public:
    InputParameterClass() : TclClass("InputParameter") {}
      .... .... ....
    virtual void bind();
    virtual int method(int argc, const char*const* argv);
} class_Input_Parameter;
```

Create the binding methods in the implementation of `bind()` with `add_method()`, then implement the handler in `method()` in a similar way as one would do in `TclObject::command()`. Notice that the number of arguments passed to `TclClass::method()` are different from those passed to `TclObject::command()`. The former has two more arguments in the front. The example of one parameter `gps_interval_` is shown in the following.[6]

gps_interval_ example

```
void InputParameterClass::bind()
{
    TclClass::bind();
    add_method("gps_interval_");
}

int InputParameterClass::method(int ac, const char*const* av)
{
    Tcl& tcl = Tcl::instance();
    int argc = ac - 2;
    const char*const* argv = av + 2;

    if (argc == 2) {
        if (strcmp(argv[1], "gps_interval_") == 0) {
            tcl.resultf("%f", InputParameter::gps_interval_);
            return (TCL_OK);
        } else if (argc == 3) {
            if (strcmp(argv[1],"gps_interval_") == 0) {
                InputParameter::gps_interval_ = atof(argv[2]);
                return (TCL_OK);
```

[6]Similar methods are used for other input parameters. Just to remember to the type of the parameter could be Integer instead of Double.

```
        }
}
```

After this, we can then use the following OTcl command to access and change values of InputParameter::gps_interval_:

```
InputParameter gps_interval_ 1
set i [InputParameter gps_interval_]
```

The list of all configurable parameters are in the Table 4.1. As a matter fact, the performance of scheme under different parameter setting is fully analyzed in the chapter 5.

Name	Type	Default	Meaning
gps_interval_	Double	1s	The update interval of PVT
adver_timeout_	Double	0.15s	Adver Timer
traffic_timeout_	Double	0.1s	Traffic Timer
sharing_timeout_	Double	30s	Sharing Timer
cts_timeout_	Double	0.2s	CTS Timer
transfer_ack_timeout_	Double	0.3s	TransferAck Timer
cts_tx_timeout_	Double	0.09s	CTSTx Timer
packet_size_	Int	512	payload size in each message
file_size_	Int	NO	file size
ap_num_	Int	1-3	the number of APs
vehicle_num_	Int	1-10	the number of vehicles
radio_range_	Int	150m	the radio range of nodes

Table 4.1. Configurable Parameters

4.2.5 Group Management Protocols

A global group management class *GroupManager* in NS2 is used to simulate the exchange of the information. The main information that should be exchanged among these vehicles are PVT information. In order to achieve modularity in programming, basically all the PVT information are stored in this Class so that it should provide the following two functions:

```
static double calDistance(int, int);
```

It calculates the distance of two nodes.[7]

```
static void updateLoc(MobileNode*, int);
```

Since all the PVT information is inside the GroupManager, this function can be called by individual node to update its location.[8]

Therefore three static linked lists are maintained for reference to *VCDClients* in GroupManager Class shown in the below.

```
static VCDClient** vcd_list_;
static Node_Info_t* pos_info_;
static MobileNode** node_list_;
```

Furthermore, a list is used to specify the header of the group.

The flowchart of GroupManager is in the Figure 4.2 :

4.3 Lower Layers

4.3.1 Overhearing of Unicast Message

In order to support promiscuous mode, two parts of the underlying protocol codes are modified including the MAC layer and the routing layer.

Overhear Routing Layer

```
1  if (ch->ptype() != PT_AODV && ch->direction() == hdr_cmn::UP
2      && ((u_int32_t)ih->daddr() == IP_BROADCAST)
3      || (ih->daddr() == here_.addr_)
4      || (ih->daddr() != here_.addr_
5          && (u_int32_t)ih->daddr() != IP_BROADCAST
6          && ch->direction() == hdr_cmn::UP
7          && ch->ptype() == PT_MESSAGE)) {
8      dmux_->recv(p,0);
9      return;
10  }
```

The above code part from line four to seven is added to support overhearing in routing layer.[9] The last two lines test whether the message is the VCD message or

[7]Note that all nodes inside the whole simulation time has different ID numbers which are input from the OTcl Script.

[8]Mainly used for *VCDServer* which remains stable during the whole simulation time and does not need to be updated.

[9]The modifications are only made in aodv.cc file because AP can only send to one vehicle at a time.

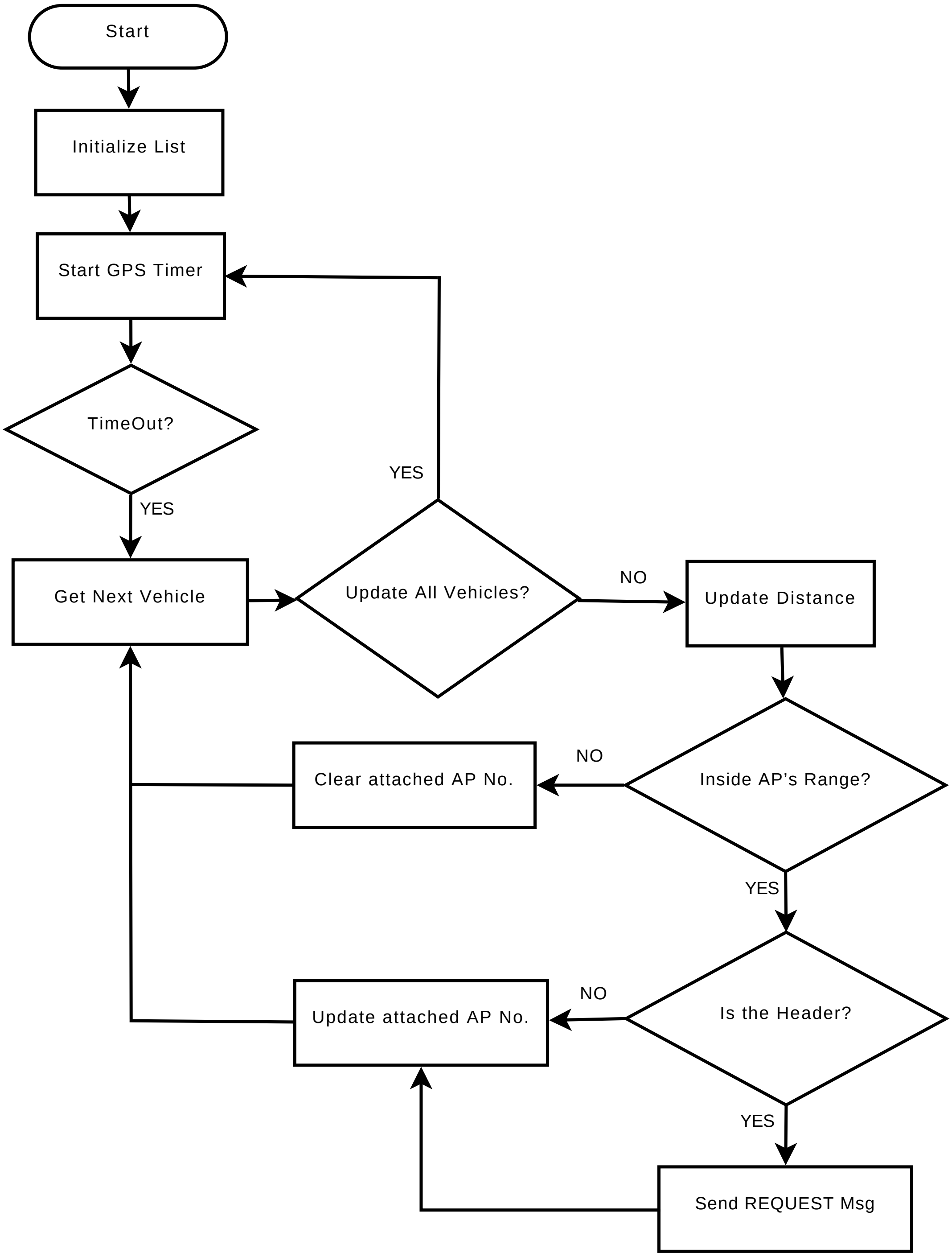

Figure 4.2. Flow Chart Group Manager

not.[10] The first two lines are added to make the condition more clear. It eliminates the possibility that the message is unicast to this node and the message is a broadcast message.

The following code part is used to support overhearing in MAC layer. If the message the type is `PT_MESSAGE`, then the message is passed to the `uptarget_`. In this case, `uptarget_` is the routing AODV agent.

Overhear MAC Layer

```
// the application needs 802.11 promiscuous mode
// to be turned on, thus we pass the packet to the
// upper layer even if it is destined to a different
// mobile node
if (type == MAC_Type_Data &&
    subtype == MAC_Subtype_Data &&
    ch->ptype() == PT_MESSAGE) {
    // adjust the MAC packet size
    // i.e. strip off the mac header
    ch->size() -= phymib_.getHdrLen11();
    ch->num_forwards() += 1;

    uptarget_->recv(pktRx_, (Handler*) 0);
    goto done;
}
```

4.3.2 Cross Layer Implementation

A pointer to the base class *VCDBase* is added in the definition of *Packet* class.

```
VCDBase* ackHandler;
```

Therefore upon the correct receipt, the *Packet* class can call the corresponding `correctSend()` function. Upon the incorrect receipt, the *Packet* class can call the corresponding `CorrectSend()` function. Therefore in the packet building of *MessagePassingAgent*, which is the transport agent, the pointer to the application layer is assigned. So the following function is declared in *Agent*, *MessageAgent* Class, which actually assigns the pointer.

```
virtual void sendmsgWithAck(int, AppData*, VCDBase*);
```

Now the problem falls into how to find the correct and incorrect transmission of the packet. The function `void Mac802_11::RetransmitDATA()` retransmits the packet again if it fails to receive the ack message. When the retransmission times is larger than the threshold, a call to the callback function before discarding the packet is as follows:

[10] MessagePassingAgent is used. Therefore the message type is PT_MESSAGE.

```
pktTx_->ackHandler->inCorrectSend();
```

As regards the correct receipt, the packet is successfully sent if the sender receives the ack message. Therefore in `void Mac802_11::recvACK(Packet *p)`, the callback function is invoked. Furthermore, only the VCD_DATA message needs callback operation. Thus the code is as follows:

```
if (pktTx_->userdata() && pktTx_->userdata()->type() == VCD_DATA) {
   pktTx_->ackHandler->correctSend();
}
```

4.3.3 Modifications of AODV Protocol

There are two modifications to be made in order to let the scheme work properly.

RERR Messages

In the line 477 of the *aodv.cc* file, the following code section is added:

Elimination of RERR Message

```
/*
 * I am trying to forward a packet for someone else to which
 * I don't have a route.
 */
else {
       assert(rt->rt_hops != INFINITY2);
       forward(rt, p, NO_DELAY);
       return;
       ... ...
}
```

The original code here is to build the RERR message and transmit it to rebuild the route. The packet will be stored in the local cache. It will be sent after the route has been rebuilt. It does not work as what we want because we can not suffer a long transmission delay of the packets.

We make these changes because the communications are all within one-hop range. There is no need for sending RERR message. Furthermore if the agent does not have the route for the message, the added two lines will also transmit it as the broadcast message. Otherwise, some application messages will not be transmitted due to reason that the route has not been built.

RREQ Retries

In the *aodv.h* file, we change the following two parameters. The original ones are commented. The values are all decreased.

RREQ RETRIES and TIMEOUT

```
// No. of times to do network-wide search before timing out for
// MAX_RREQ_TIMEOUT sec.
//#define RREQ_RETRIES                3
#define RREQ_RETRIES                1
// timeout after doing network-wide search RREQ_RETRIES times
//#define MAX_RREQ_TIMEOUT        10.0 //sec
#define MAX_RREQ_TIMEOUT        0.1  //sec
```

The changes made here seems to be more vague. Actually it is a compromise.

If the AP finds out it can not send the message to the vehicle, it will try to rebuild the route by sending RREQ_RETRIES and the message will be stored in the sending queue cache. The waiting timeout value could be large. However in our scenario, if the AP can not send the message, it should simply try to send to the next vehicle. So the value is decreased. The final solution should be modifying the protocol to let the AP send the message immediately instead of trying to rebuild the route.

4.4 Application Data

No special VCD packet header is added, as what is usually done when adding a new protocol. But an application data type `VCD_DATA` is added to denote the different packet type. A *VCDMsg* class, which is a sub-class of *AppData*, is declared. The name is resisted in the *ns-process.h* file.

It has the following fields:

```
Pkt_State_t* pkt_;
VCD_Header_t type_;
int sender_;
int receiver_;
int payload_size_;
int seqno_;
```

It is different from the fields described in chapter 3. `pkt_` is used to describe the packet status because we directly pass this pointer to the receiver thanks to the packet size definition, which is described in the earlier part of this chapter. Furthermore, we do not have any group information field here because of GroupManager class. However we have to add the group information size when determining the packet size.

Comparing to the tables in chapter 3, the following two tables(Table 4.2 and Table 4.3) add the size of individual field and summarize them together. [11] [12]

Field	Size	**REQUEST**	**TRAFFIC**	**TRANSFER**
Type	4	√	√	√
Tx ID	4		√	√
Rx ID	4		√	√
Payload	512		√	√
Seq No	4		√	√
Pkt Status	15			
Group Info.	20*n	√		
Total Size		4+20*n	532	532

Table 4.2. Message Formation Part I

Field	Size(Bytes)	**ADVER**	**CTS**	**FINISH**
Type	4	√	√	√
Tx ID	4	√	√	√
Rx ID	4		√	
Payload	512			
Seq No	4			
Pkt Status	15	√	√	
Group Info.	20*n	√		
Total Size		23+20*n	27	8

Table 4.3. Message Formation Part II

4.5 Simulation Setup

Scripts are used to reduce the simulation time without compiling every time when one parameter is changed. In this section, OTcl is firstly introduced. Then the radio range setting is detailed.

[11] n is the number of the vehicles.

[12] The pkt status size is got from the simulation results in chapter 5.

4.5.1 OTcl Scripts

OTcl, short for MIT Object Tcl, is an extension to Tcl/Tk for object-oriented programming [36]. Here it is used to configure the simulation scenario.

Besides the standard OTcl language syntax, some special requirements are needed to setup the correct simulation. All nodes including AP and vehicles are identified by a unique number starting from zero, which is configured in the OTcl scripts for individual node. The lower numbers are assigned for vehicles while higher numbers are assigned for APs. Then the AP number and Vehicle number are also configured to let the program know the boundary of these numbers. For example, in the case of three nodes with one AP. `$val(MN)` is two and `$val(AP)` is one. Node 0 and node 1 should be the vehicle; Node 2 is the AP.

4.5.2 Radio Propagation Models

We use two ray ground reflection model which takes into account the ground reflection. However, The two-ray model does not give a good result for a short distance due to the oscillation caused by the constructive and destructive combination of the two rays. Instead, the free space model is still used when d is small. Therefore, a cross-over distance d_c is calculated in this model. When $d < d_c$, free space model is used. When $d > d_c$, two ray model is used.

The radio propagation models are used to predict the received signal power of each packet. At the physical layer of each wireless node, there is a receiving threshold. When a packet is received, if its signal power is below the receiving threshold, it is marked as error and dropped by the MAC layer. In some applications, a user may want to specify the communication range of wireless nodes. This can be done by set an appropriate value of the receiving threshold in the network interface.

There are three related threshold values: RXThresh_, CSThresh_, PThresh_. *RXThresh_* is the reception threshold. If the received signal strength is greater than this threshold, the packet can be successfully received. The other two have their meanings but they are irrelevant in our setting.

A separate C program is provided at *ns/indep-utils/propagation/threshold.cc* to compute the receiving threshold.

The command to set the transmission range to 150 meters is `./threshold -m TwoRayGround 150`. The resulting receiving threshold RXThresh_ is: 2.81838e-09. Other parameters are in the following Table 4.4:

Recall that the *GroupManager* class needs explicitly the radio range of the AP to determine whether it is inside the coverage of one AP or not. Therefore, the radio range is also configurable by using *InputParameter* class. Moreover, the range should be consistent with the RXThreash_ value.

name	**value**
transmit power	0.281838
frequency	9.14e+08
transmit antenna gain	1
receive antenna gain	1
system loss	1
transmit antenna height	1.5
receive antenna height	1.5

Table 4.4. Propagation Model Parameter

Chapter 5

Simulation Results

This chapter is arranged as both the functional test and the performance evaluation. The tunable parameters of scheme including group size, file size, timeout value and radio range are described in the beginning of this chapter. Then a simple case with only two vehicles, which is used as the functional test in the development of the scheme code, is presented. The number of received packets per vehicle and the throughput of the group are evaluated.

Then the rest of the chapter describes the performances with respect to different values of the tunable parameters.

5.1 Tunable Parameters

We firstly introduce some parameters that are configurable in the scheme. These parameters are input from the OTcl script. We would like to compare the performances when these parameters are different.

- *Group Size:*
  As the group size grows, more contents can be downloaded from the AP by using cooperative downloading. However more vehicles in the group result in a higher possibility of transmission collision. The throughput per vehicle may decrease as the group size becomes larger.

- *File Size:*
  The file size can be as high as several mega bytes because we consider convenience applications as well. The communication between the AP and the vehicles can last for tens of seconds. Therefore the group could download all the contents from just only one AP provided that the data rate of IEEE 802.11b standard is 11Mbit/s. We probably need a second AP to download all the contents if the file size is larger.

- *Timeout Value:*
  There are six timers in the scheme design. There may be some constraints of the values. Furthermore, the performances are of course different when we use various timeout values.

- *Radio Range:*
  The radio range of the nodes can result in different performances. On one hand, increasing radio range extends the communication duration time between the vehicles and the AP, which brings more contents to the group. On the other hand, larger radio range causes more collisions from other nearby vehicles or groups.

5.2 A Simple Case

In this section, a simple case with two vehicles and one AP is considered. As shown in the Figure 5.1, the two vehicles travel along the square route with 975 meters long on each side. They enter at the bottom-left corner with the coordinate (25,25), go to the bottom-right corner and so on. Finally they leave at the bottom-left corner. In addition, the AP is located at the bottom-right corner with the coordinate (975,25).

To better understand the scheme, the five basic stages in this case are shown in the Figure 5.1.

1. The two vehicles (Vehicle A and Vehicle B) form a group and have a common interest in the same content. They communicate with each other to exchange their positions and velocities.[1]

2. Vehicle A is selected as the vehicle header by the group manager protocol since it is nearer to the AP than vehicle B is. Then the AP sends the contents sequentially to vehicle A. Vehicle B can overhear the packets, which carry the contents, since it is inside the coverage range of AP too.

3. Vehicle B is selected as the new vehicle header by the AP when vehicle A moves out the coverage range. The AP sends the contents sequentially to vehicle B.

4. The two vehicles exchange the contents they have downloaded from the AP after handshake procedure.

5. All the contents have been shared between them.

[1]The exchange is simulated in the *GroupManager* class.

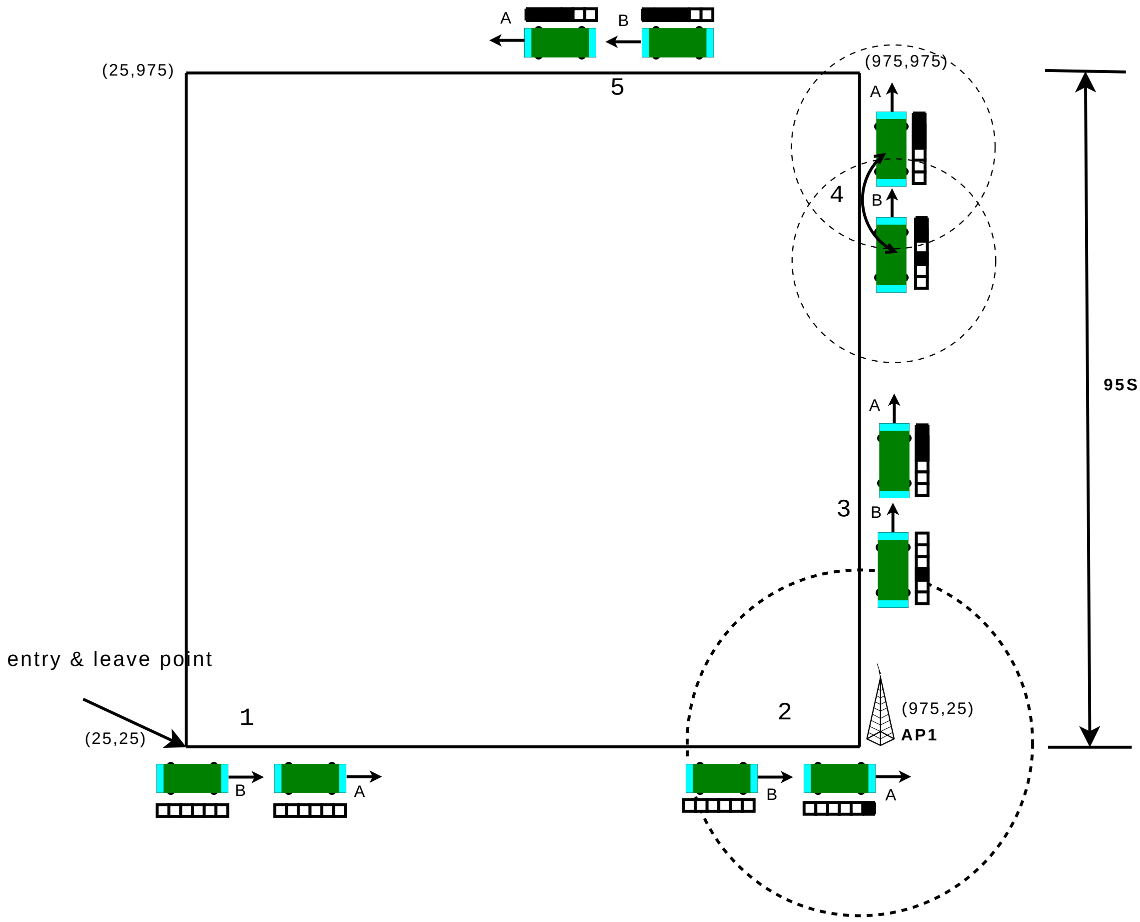

Figure 5.1. The Two Vehicles Case

Mobility Pattern

The trace file, which describes the mobility pattern of the nodes in NS2, can be generated by VanetMobiSim [37]. But it can not generate the mobility pattern we want when the vehicles' number is high because the initial positions of vehicles are random. In consequence the trace file is input manually.

The two vehicles move at the constant speed of 10m/s. [2]. The distance of the two vehicles is 50 meters. The trace file is:

Trace

```
$node_(0) set X_ 50
$node_(0) set Y_ 25
$node_(0) set Z_ 0
$ns_ at 0.0   "$node_(0) setdest 975 25 10"
$ns_ at 92.5  "$node_(0) setdest 975 975 10"
$ns_ at 187.5 "$node_(0) setdest 25 975 10"
$ns_ at 282.5 "$node_(0) setdest 25 25 10"

$node_(1) set X_ 100
$node_(1) set Y_ 25
$node_(1) set Z_ 0
$ns_ at 0.0   "$node_(1) setdest 975 25 10"
$ns_ at 87.5  "$node_(1) setdest 975 975 10"
$ns_ at 182.5 "$node_(1) setdest 25 975 10"
$ns_ at 277.5 "$node_(1) setdest 25 25 10"

$node_(2) set X_ 975
$node_(2) set Y_ 25
$node_(2) set Z_ 0
```

There are three nodes in the above script. Node 0 and node 1 are the vehicles; Node 2 is the AP. It specifies the initial positions by setting X axes, Y axes and Z axes.[3] The remaining lines set the moving destination and the speed of the vehicles at that specific time.

Scenario Size and Simulation Time

The simulation area is a square with 950 meters long on each side. Thus the vehicle needs 95 seconds to move along one side if the speed is 10m/s. Then the simulation time can be calculated as approximately 380 seconds.

[2]It is equal to 36km/s
[3]The height is Z axes and useless.

Timeout Value

All the timers use the default timeout value in the Table 4.1. There are reasons why these default values are set.

The `traffic_timeout_` is 0.1 second, which is the basic timeout value.

The `adver_timeout_` is 0.15 second because there may be many vehicles waiting for transmitting the ADVER message. Extending the duration can decrease the possibility that the two vehicles are transmitting at the same moment.

The `sharing_timeout_` is set to 30 seconds because the transmission range of the nodes is 150 meters. When the speed is 10m/s, setting the timeout to 30 seconds can make sure that the transmissions in the V2V sharing phases do not collide with the transmissions which happen within the AP's coverage range.

The `cts_timeout_` is set to 0.2 second, the double of the basic timeout value, since during this time interval the vehicle waits for possibly two replies.

Consequently the `transfer_ack_timeout_` should be the sum of `cts_timeout_` and basic timeout value, which is 0.3 second.

The `cts_tx_timeout_` is 0.09 second which is 90% of the basic timeout value since the transmission delay from the CTS message sender to the CTS message receiver should be considered.

File Size

The file size is unlimited in this case. We would like to know the maximum number of packets that can be downloaded from the AP.

Results

The simulation result is shown in the Figure 5.2. The performance changes for each simulation because the AdverTimer's timeout value is randomly chosen. Thus the mean value is plotted in the graph. In the following, we enumerate the five stages related with the content distribution.

1. Node 1 is selected as the vehicle header at time 73.00 because it is nearer to the AP. In this stage, the AP transmits the packets from No.0 to No.23041 to node 1. At time 102.54, node 1 moves out of the coverage range and no more packets are received from the AP after that time.

2. Node 0 enters the coverage range of the AP at time 77.50. From that time till 102.54, it overhears the packets from No.3511 to No.23041.

3. Node 0 is now the header because node 1 moves out of the coverage range of the AP at time 102.54. It downloads packets from No.23042 to No.26891. The cooperative downloading phase ends at time 107.54.

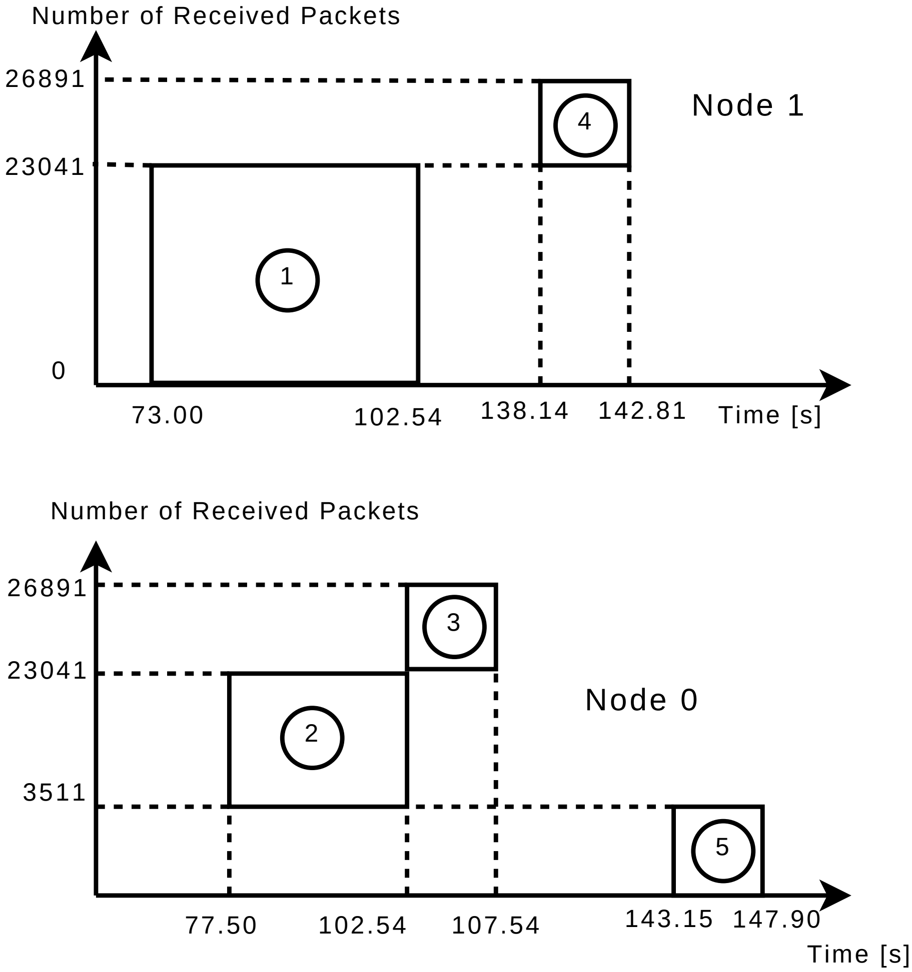

Figure 5.2. A Simple Case

4. The V2V sharing begins at time 138.14, the handshake procedure of which takes around half second.[4] Node 1 receives packets from No.23042 to No.26891.

5. After another handshake procedure, the transfer from node 1 to node 0 of packets from No.0 to No.3511 takes place at time 143.15 and ends at time 147.90.

Analysis

As can be seen from the Figure 5.2, the number of the packets that can be downloaded for a single node without any cooperation is 23042. By using cooperative downloading, both vehicles can download 26892 packets now. Therefore the number of received packets per vehicle is 16.70% higher.[5]

The throughput of a single node is 3119.7 Kb/s.[6] [7] The throughput of node 1 after using the scheme is 2702.0 Kb/s.[8] The throughput is lower because there is an extra time for setting up the V2V sharing phase. The throughput of node 0 is even lower because the transfer from node 1 to node 0 happens after the transfer from node 0 to node 1. The throughput of it is 2395.7 kb/s.[9]

Although the throughput of individual nodes decreases, the throughput of the group increases a lot by 54.1%.[10] Moreover, the group throughput could increase a lot if the group size grows.

5.3 Group Size

In this section, an important observation is introduced in the first part. Then we compare the performances when the group size is increased from two to seven. Other parameters do not change with respect to the simple case.

5.3.1 The Span of the Fleet

The number of the received packets when group size is different depends actually on the distance span of the fleet. If the span is very large, it can download more contents even if there is smaller number of vehicles in the group. As plotted in the Figure 5.3, obviously more contents can be downloaded in case a) than those in case b).

[4]The 30 seconds timeout of SharingTimer is not considered.

[5]$(26892 - 23042)/23042 = 16.70\%$

[6]Each packet size is 512 bytes.

[7]$512 * 23042/(102.544 - 73)/1024 * 8 = 3119.7Kb/s$

[8]$512 * 26892/(142.81 - 30 - 73)/1024 * 8 = 2702.0Kb/s$

[9]$512 * 26892/(147.90 - 30 - 73)/1024 * 8 = 2395.7Kb/s$

[10]$(2395.7 * 2 - 3119.7)/3119.7 = 54.1\%$

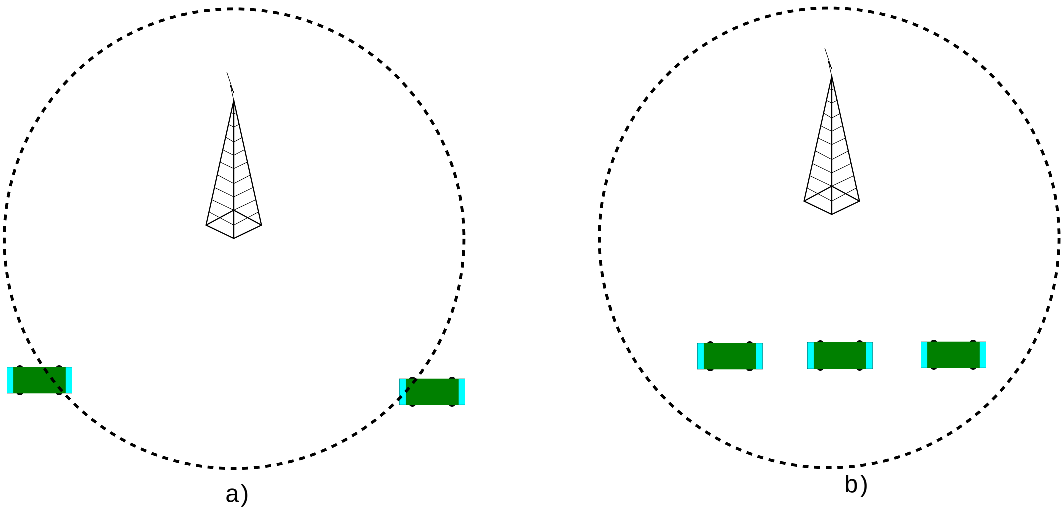

Figure 5.3. Distance Span Example

Unfortunately, the distance span of the fleet is not controllable if we use Vanet-Mobisim [37] to generate the trace file. It is another reason why all the trace files are input manually. They are 50 meters away from each other.

5.3.2 Comparison of Different Group Size

X axes	**50**	**100**	**150**	**200**	**250**	**300**	**350**
3 Vehicles	√	√	√				
4 Vehicles	√	√	√	√			
5 Vehicles	√	√	√	√	√		
6 Vehicles	√	√	√	√	√	√	
7 Vehicles	√	√	√	√	√	√	√

Table 5.1. Initial Positions

First of all, the initial positions of the nodes are shown in the Table 5.1. All the vehicles are separated 50 meters from each other and have the same Y axes of 25. Z axes is useless because we do not consider height in the simulation scenario.

Secondly, the simulation is performed ten times for the reason that the timeout value of AdverTimer is random. The minimum, maximum and average values are recorded.

The simulation results and their analysis are stated in the following.

Increase of Received Packets

Figure 5.4 shows the increase of the received packets when the group size grows from one to seven.

Group Size	2	3	4	5	6	7
Increase Percentage	16.74%	33.75%	50.69%	64.47%	84.35%	101.45%

Table 5.2. Increase of Received Packets

The increase percentage compared to one vehicle scenario is shown in the Table 5.2. Adding one vehicle means that the span of the fleet is increased by 50 meters in consequence of the fact that all the vehicles are 50 meters away from each other. The distance that the group travel in the coverage range of the AP is the sum of the distance span and the double of the coverage range. This distance is 300 meters in the case of one vehicle. As a result, the increase percentage is around 16.7% when one vehicle is added in the group.

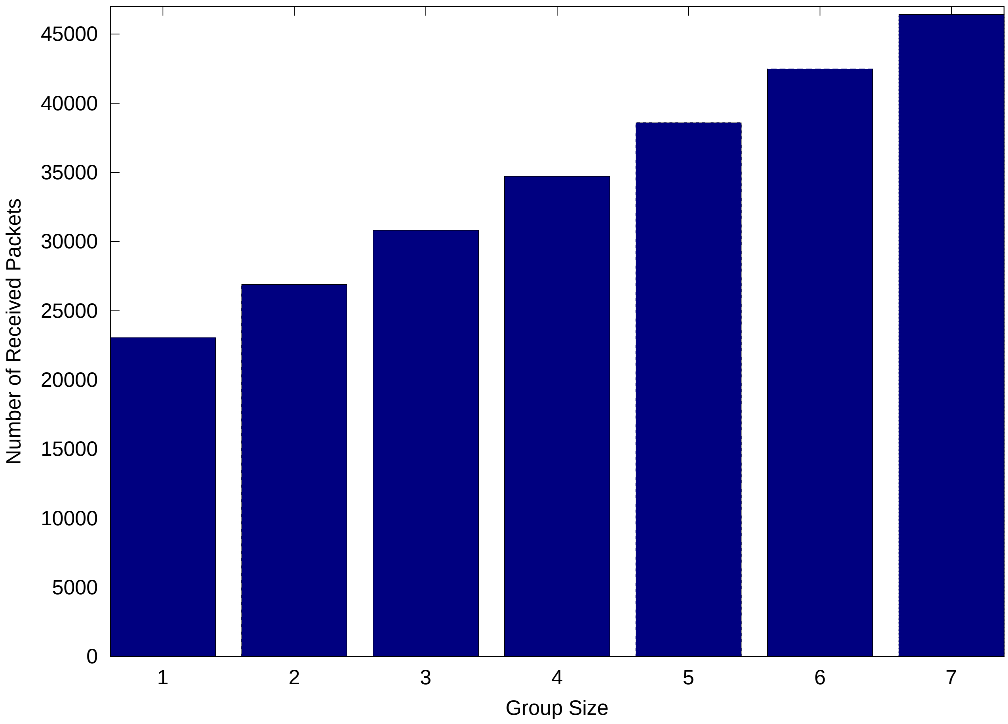

Figure 5.4. Number of Received Packets Per Vehicle vs Group Size

Time Spent in Sharing Phase

The time spent in the V2V sharing phase is another interesting aspect. As shown in the Figure 5.4, more contents are downloaded from the AP when the group size is larger. Consequently the group needs more time to share the contents among their members. The result is plotted in the Figure 5.5.

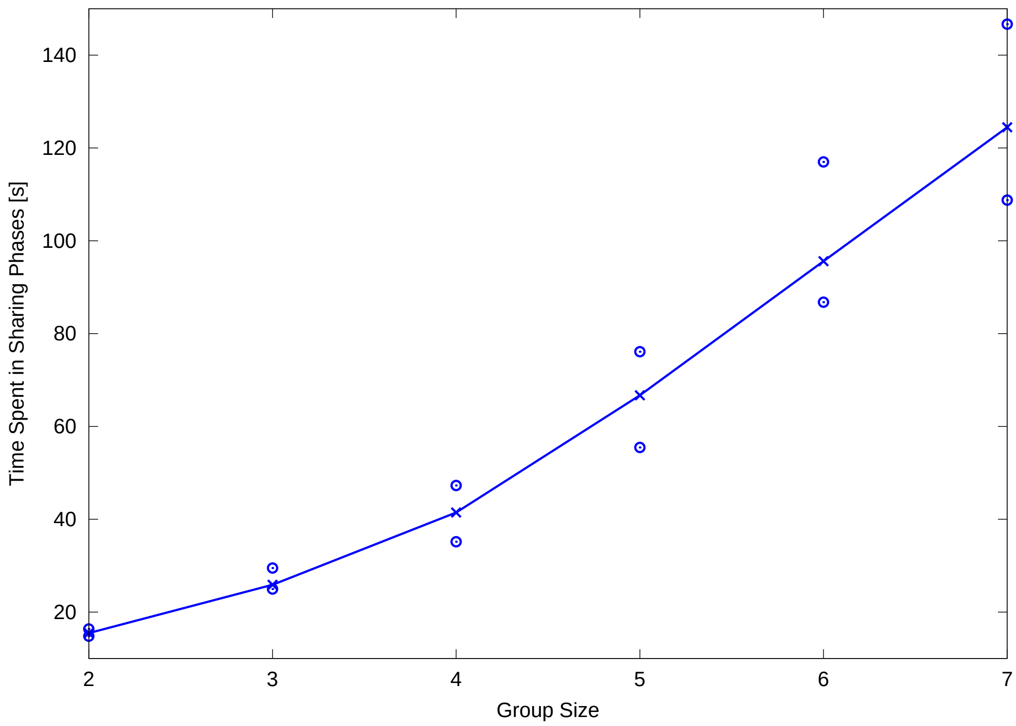

Figure 5.5. Time in Sharing Phase vs Group Size

The average value is drawn as the `x` point and connected by the line. The minimum and maximum values are plotted in the graph too, which are represented as circles.

The time spent in the V2V sharing phase increases as the group size grows. But it is not linearly dependent on the group size and it increases a bit quicker. This result is due to two reasons. The first one is that more collisions occur when sending ADVER messages. So more time is needed to set up the V2V sharing phase. The second one is the fact that there are more contents waiting to be shared.

Furthermore the variance is higher when there are more vehicles because random timeout value is used to coordinate the sending moment of ADVER message. The variance is enlarged when more vehicles are involved in this procedure.

Individual Throughput

In this section, we compare the throughput of individual members in the group. The start time of the scheme is the moment when the vehicle header sends the REQUEST message. The end time of the scheme is the V2V sharing end time of the group. Consequently the content distribution duration time of the scheme is calculated as the difference of these two moments.

As shown in the Figure 5.6, the throughput per vehicle decreases as the group size grows. The throughput per vehicle is the division of the number of the received packets and the content distribution duration time. Although the received packets increases, the time used for content distribution increase much more when the group size increases. As a result, the throughput per vehicle decreases.

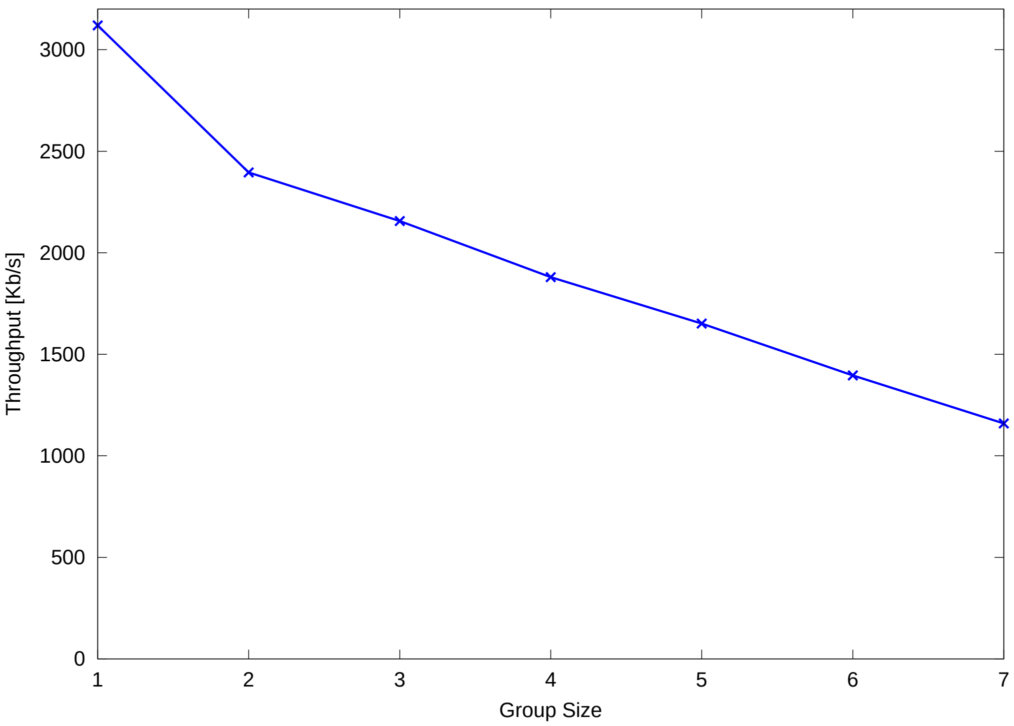

Figure 5.6. Individual Throughput vs Group Size

Fortunately, the good point is that the number of received packets per vehicle in high number of vehicles is larger than that in small number of vehicles. Consequently, if the file size is small, increasing the group size does not help. Additionally, the throughput per vehicle does not fall dramatically when the group size is large.

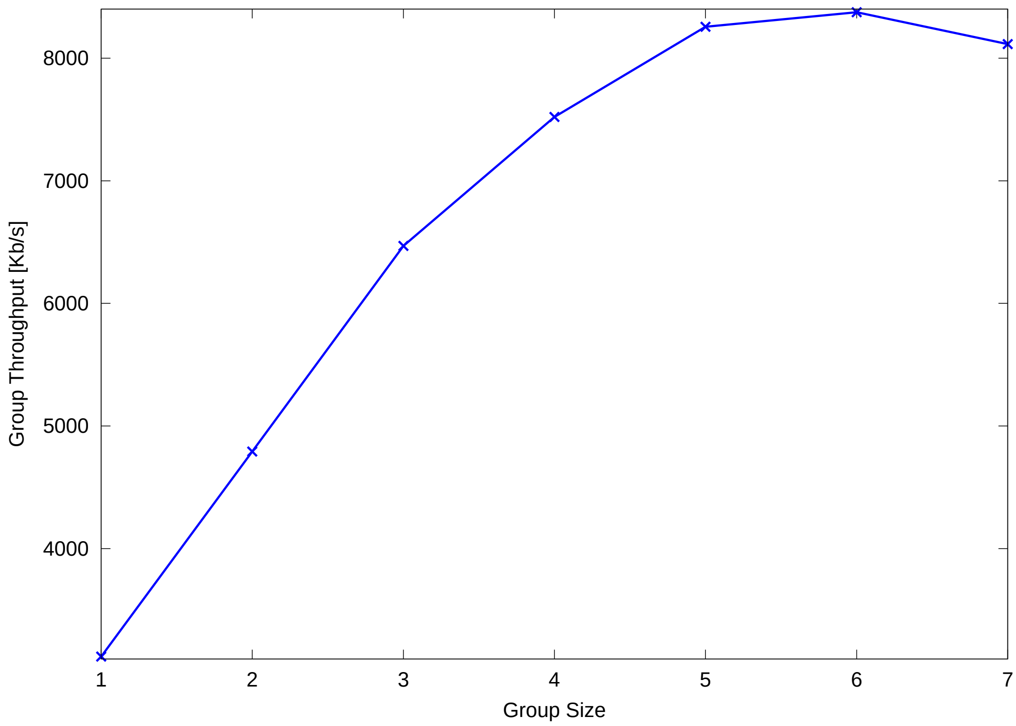

Figure 5.7. Group Throughput vs Group Size

Group Throughput

The group throughput is also plotted in the Figure 5.7. It is calculated as the total number of received packets in the group divided by the distribution duration time. We can expect that the group throughput increase significantly when more vehicles are in the group. It is true when the group size is small. However as more vehicles are involved, the increase of the group throughput goes down and finally the group throughput reaches the flex point because the throughput per vehicle reduces shown in the Figure 5.6.

5.4 Packet Status Size

Group Size	**2**	**3**	**4**	**5**	**6**	**7**
Appearance Times	4	9	12	27	40	59
Average Size(Bytes)	10.40	11.00	12.30	12.60	15.45	14.07

Table 5.3. Packet Status Size

In chapter 4, Run Length Coding is introduced to advertise the status of the packets that one vehicle has or needs. The results are listed in the Table 5.3 including the average size and the appearance times.

The size of packet status field raises from 10 bytes to 15 bytes as the group size increases. It is acceptable and the total message size does not increase a lot when this field is added. Moreover it does not change a lot.

5.5 Sharing Timer Timeout

We would also like to compare the performances when different timeout values are used.

The most important timer is the SharingTimer, which determines the time spent in the transition from the cooperative downloading phase to the V2V sharing phase. The default value is 30 seconds which ensures the separation of these two phases. In the following simulation scenario, the timeout value is chosen from 1 second to 30 seconds in an interval of 3 seconds and the group size is seven. The simulation is run ten times at each timeout value and the mean value is plotted.

Actually this timeout value is a trade-off between the number of received packets per vehicle and content distribution duration time. On one hand, the number of received packets per vehicle decreases when `sharing_timeout_` is smaller. It is due to the reason that if the vehicles enter the V2V sharing phase as soon as they leave the coverage range of AP, their transmissions cause collisions in the vehicles which are inside the coverage range of one AP and are receiving contents from that AP. The collision happens when the TRANSFER message and TRAFFIC message are sent to the vehicle simultaneously. On the other hand, if the sharing timer's timeout value is high, the upcoming V2V sharing phase does not interfere with the remaining cooperative downloading phase in the rear of the vehicle group. Therefore the number of received packets are larger. However more time is consumed before all the contents are shared among the vehicles in the group.

Number of Received Packets

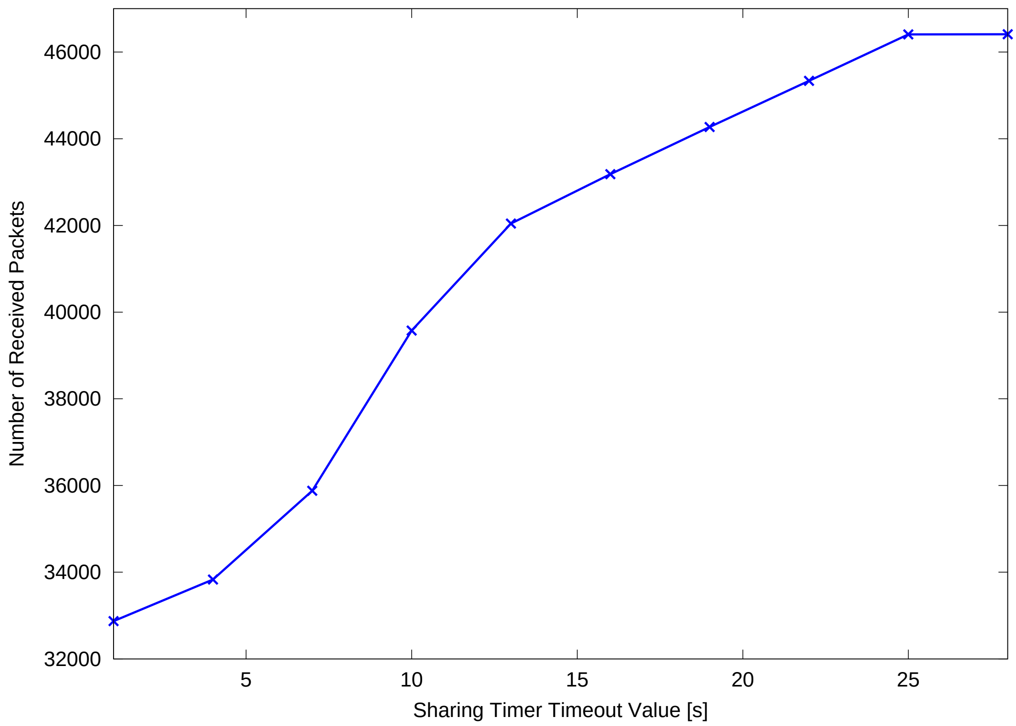

Figure 5.8. Number of Received Packets vs Sharing Timeout Value

Figure 5.8 denotes the number of received packets per vehicle. It increases as the timeout value becomes larger. But the growth rate decreases. When the timeout value is around thirty seconds, it almost remains stable because the two phases are totally separated.

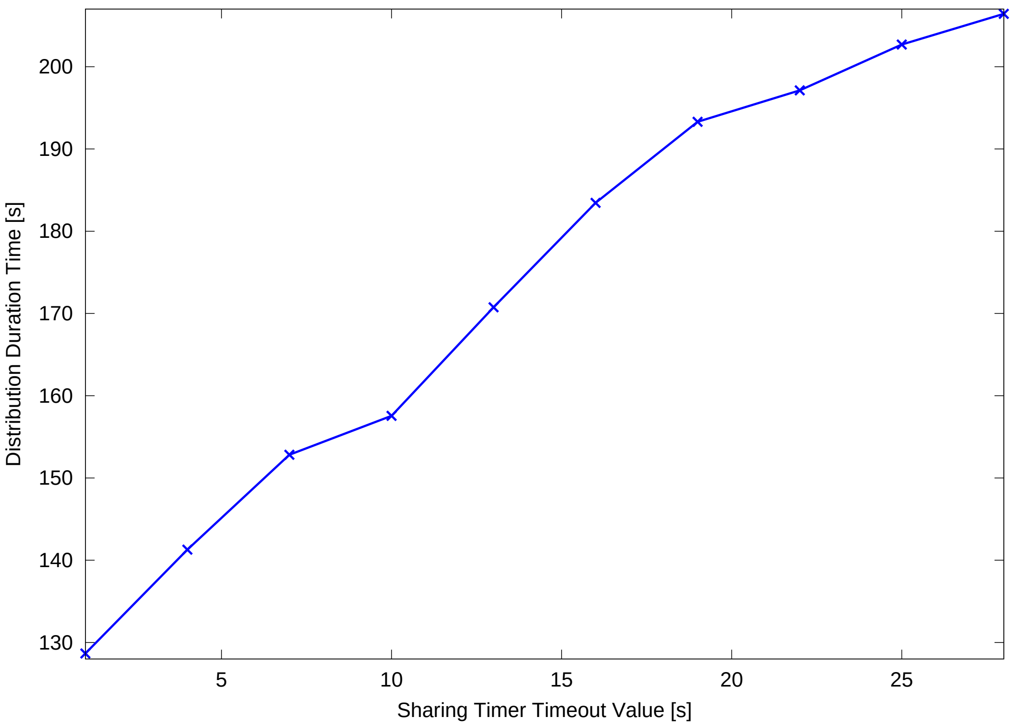

Figure 5.9. Distribution Duration Time vs Sharing Timeout Value

Distribution Duration Time

Figure 5.9 shows the distribution duration time, which is calculated from the moment when the vehicle header sends the REQUEST message to the moment when the last V2V sharing phase ends in the group.

The time used to distribute the packets increases because of the addition time of the SharingTimer and more contents in the group. The time growth in large timeout value is not as much as that in small timeout value because the increase of the number of received packets per vehicle is smaller.

Throughput Per Vehicle

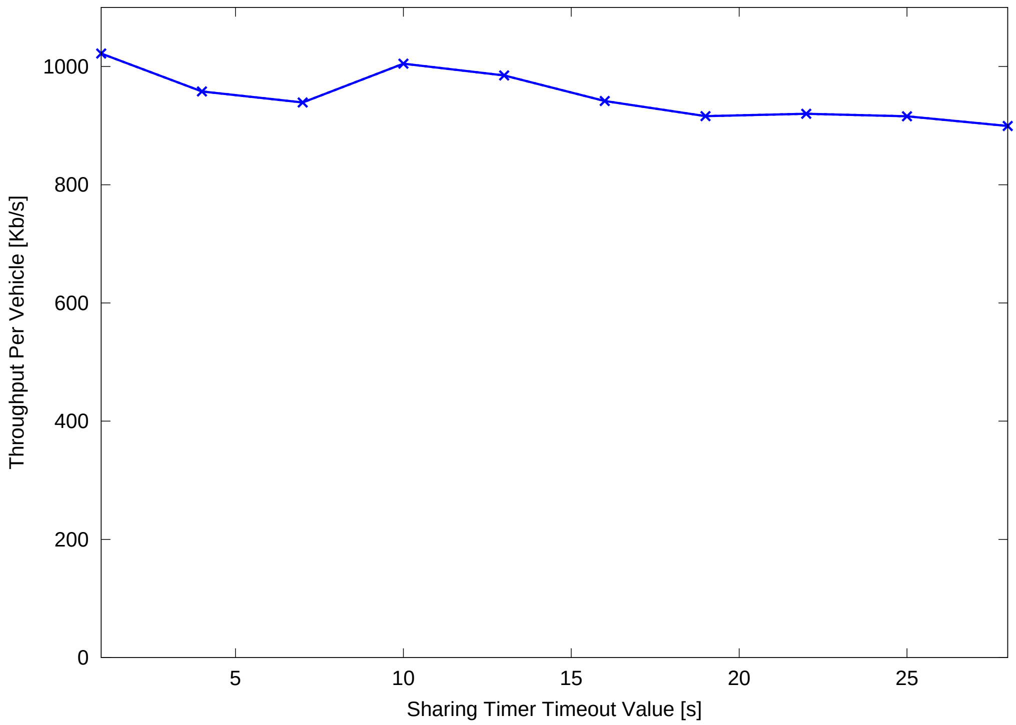

Figure 5.10. Throughput Per Vehicle vs Sharing Timeout Value

We can then compute the throughput per vehicle as the number of received packets per vehicle divided by the distribution duration time in the Figure 5.10.

The throughput per vehicle has a trend of decrease. The smallest throughput per vehicle is around 70% of the largest throughput per vehicle. Although the throughput per vehicle is small in high timeout value, we can not deduce it is useless since the vehicle can download more contents. When the file size is very large, it is necessary to increase the timeout value in order to download as much as possible the contents from the AP.

5.6 Fixed File Size

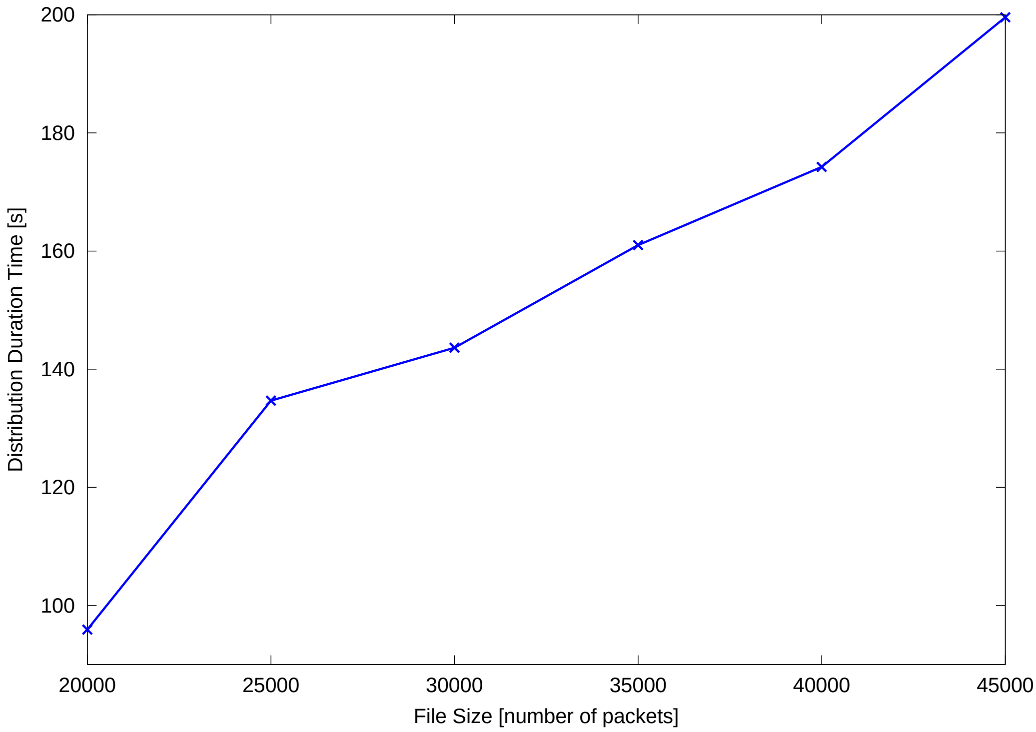

Figure 5.11. Distribution Duration Time vs File Size

In this section, the file is changed from unlimited size to limited size and the group size is seven. All other parameters are set to the default values with respect to the simple case and ten simulations are run again to generate the average result. We can get from the previous simulation that the number of received packets per vehicle is 46,413 when group size is seven. Therefore in this case, the file size in terms of the number of packets is set from 20,000 to 45,000 with a separation of 5000 from each other.

In the Figure 5.11, it is obvious that the distribution duration time increases when the file size grows because more contents have to be downloaded from the AP and shared among the vehicles. What is more important is shown in the Figure 5.12. The throughput per vehicle does not change so much when the file size is different provided that the group can download all the contents from only one AP.

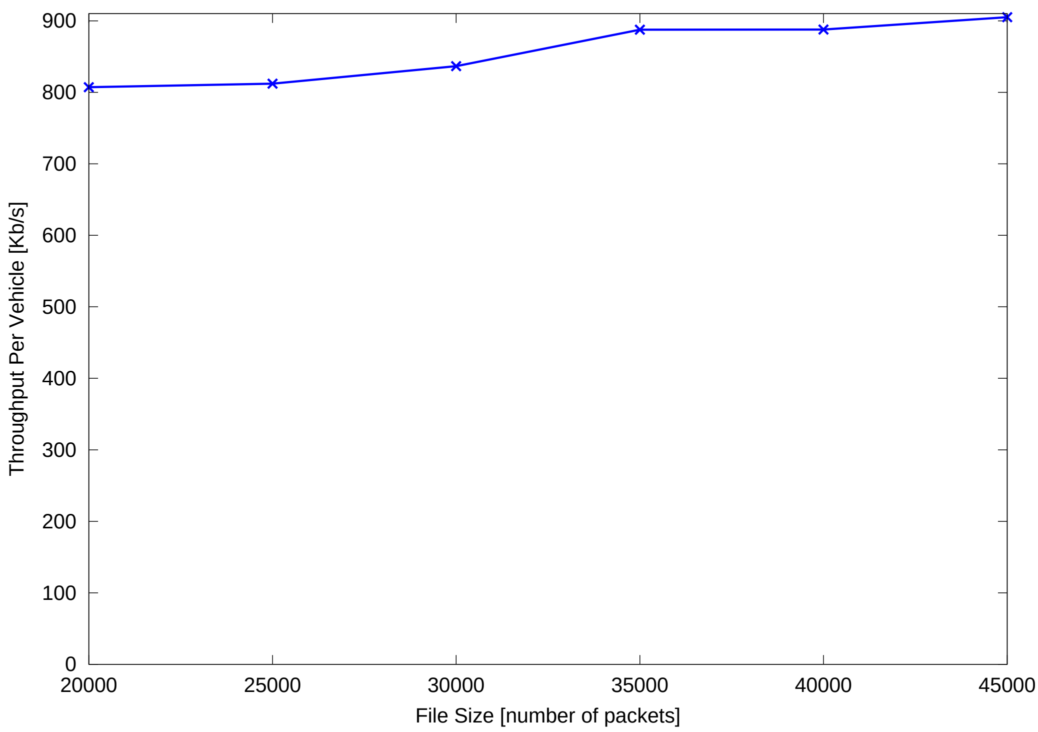

Figure 5.12. Throughput Per Vehicle vs File Size

5.7 Radio Range

In this section, the performance is evaluated when the radio range is different.

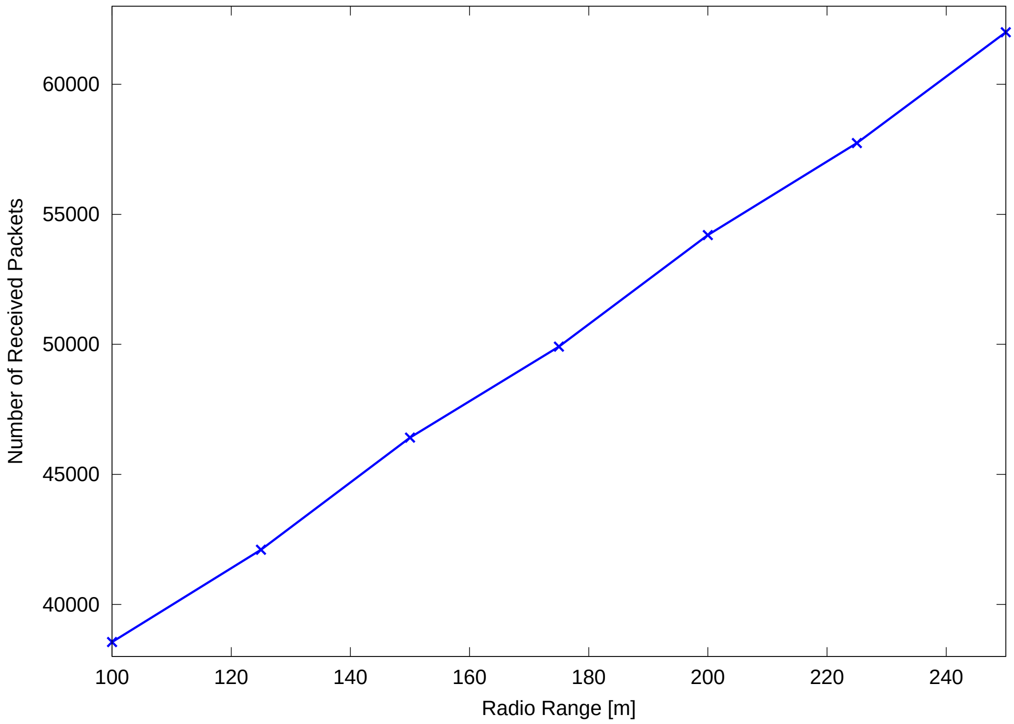

Figure 5.13. Number of Received Packets vs Radio Range

One thing to be stated is that the transmission range of the nodes including AP and vehicles is the same. Although they could be different and normally AP's range is larger than vehicle's range, most of the transmissions in our scenario are bi-directional unicast transmissions and the remaining broadcast transmissions in the scheme are preceded or followed by other bi-directional unicast messages. Therefore the setting of asymmetric transmission range is unnecessary.

In order to set different radio ranges, we have to change the receive threshold value as described in chapter 4. The thresholds with respect to different radio range are reported in the Table 5.4.

As regards the tunable parameters, the group is still composed of seven vehicles and the others are the same as those in the simple case. We run the simulation ten times and the average value is calculated.

The results are shown in the Figure 5.13 and the Figure 5.14. As can be easily derived, if we raise the radio range, the number of the received packets and the

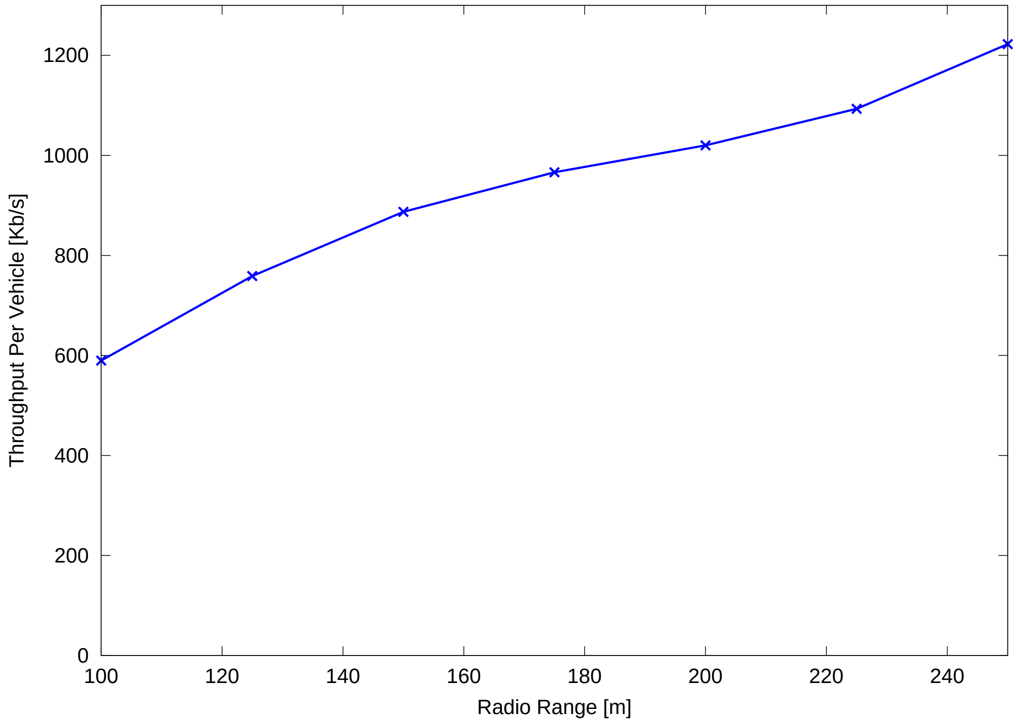

Figure 5.14. Throughput Per Vehicle vs Radio Range

Radio Range(m)	100	125	150	175
RXThresh_	1.42681e-08	5.8442e-09	2.81838e-09	1.52129e-09
Radio Range(m)	200	225	250	
RXThresh_	8.91754e-10	5.56717e-10	3.65262e-10	

Table 5.4. Receive Threshold

throughput per vehicle increase proportionally as well. However it is an ideal situation because we only consider one group in the simulation scenario. Actually if we increase the radio range, there could be more interferences from other vehicles or groups. As a result, the number of the received packets and the throughput per vehicle would decrease.

5.8 Multiple APs

In this section, we evaluate the performance when there are multiple APs. The second AP is added at the left-top corner with the coordinate (975, 975). All other

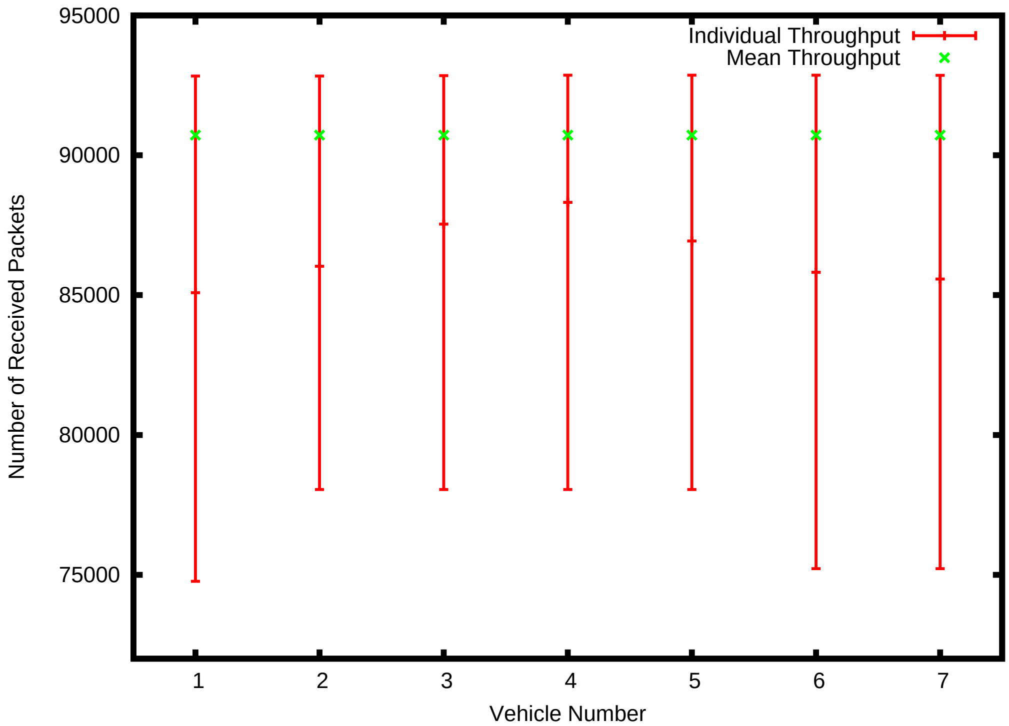

Figure 5.15. Packets Receipts Ratio Vs Two APs

parameters remain the same. The simulation time is set to 350 seconds because it only needs this time duration to travel along the square. Thus there may not be enough time for the group to successfully share all the contents when they leave the second AP.

Four Vehicles

We firstly simulate the case when the group size is four. All the vehicles in the group can download 69441 packets and successfully share them before the end of the simulation because the travel duration time between the two APs is around 160 seconds[11] and the time spent in the V2V sharing phase is less than 160 seconds. All the contents downloaded from the first AP can be shared among all the vehicles before they enter the coverage range of the second AP. Furthermore the number of received packets per vehicle is almost twice as that in one AP's case.

Seven Vehicles

However if we increase the group size, the number of received packets per vehicle is different. The simulation has been run for thirty times and the minimum, maximum and average number of received packets per vehicle is plotted in the Figure 5.15 when the group size is seven.

The number of received packets per vehicle is almost the same. However it does not remain stable for each simulation. The distribution duration time in seven vehicles' case is longer than 160 seconds. Therefore there is a possibility that the still running V2V sharing phase in the rear of the group would interfere the cooperative downloading phase in the beginning of the group which happens in the coverage range of the second AP. Due to the same collision reason, the number of the received packets is less than twice of that in one AP's case.

5.9 Conclusion

In this chapter, the performance of the scheme is evaluated. The tunable parameters of the simulation scenario include group size, file size, timeout value and radio range of the node.

A simple example is shown which is composed of two vehicles and one AP. In this scenario, the number of received packets per vehicle is 16.70% higher and the throughput of the group increases by 54.1%. The number of received packets increases linearly and throughput per vehicle decreases as the group size increases

[11] (950*2-150*2)/10=160s. 150 is the coverage range and 10 is the speed of the vehicle

because more time are spent in setting up the V2V sharing phase and more transmission collisions occur.

Other simulation results are presented also. Firstly, the size of packet status field with respect to different group size is acceptable and does not change a lot. The size is from 10 bytes to 15 bytes. Secondly, higher sharing timeout can result in more number of received packets but longer content distribution duration time. Thirdly, the throughput per vehicle does not change so much when the file size is different provided that the group can download all the contents from only one AP. Fourthly, the number of received packets per vehicle and throughput increases as the radio range of the nodes is larger when. Finally, in multiple APs case, the packet receipt status is different for each vehicle because there are not enough time in between two APs to share the contents from the first AP.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this thesis, we design a scheme for the vehicle equipped with IEEE 802.11 wireless card to download as much as possible and as quick as possible the data from the AP in the VANET scenario.

First of all, a rather complete survey of the thesis background is written in chapter 1 and chapter 2. Some general characteristics that are related with VANET scenario and content distribution methods are introduced in chapter 1. More specific approaches, some of which are used in the thesis, are presented in chapter 2.

The scheme is composed of cooperative downloading and V2V sharing phases. In the cooperative downloading phase, a header is selected for the whole group to send the initial downloading request. The subsequent downloading is smoothly transferred from one vehicle in the group to the other. In addition, the packet sequences sent from the AP is consistent.

In the V2V sharing phases, due to the lack of central point of coordination and the high speed of node, the advertisement message is sent within one-hop range. Furthermore, we borrow the idea from recent methods in the literature such as Preferred Group Broadcast(PGB) and Hamlet to distribute the packets within the group.

The scheme is implemented on the NS2 simulator. The general structure of the source codes including two application classes: *VCDClient* and *VCDServer* is described. The *GroupManager* class is used to simulate the group management protocol. Some lower layer modifications to fulfill the scheme are mentioned too. Furthermore, OTcl scripts are used to setup the simulation scenario.

In the last chapter, the functional test and performance evaluation are performed.

6.2 Future Work

The work in this thesis is not complete due to the limited time and effort. The simulation is not exhaustive as well. Therefore we can not make sure that the proposed scheme is the best choice. Actually as can be seen in chapter 5, there is no one-for-all scheme for various scenarios. However some parameters may need to be changed when more scenarios are considered. We summarize the planned works that could be done to enhance the thesis work in the future.

1. The scheme is not complete because the group management protocol is ignored. Future work should be devoted to design a management protocol.

2. A more sufficient mechanism to switch between the downloading phases and the sharing phases should be used. In the initial scheme, the ACK message is sent by the vehicle as the reply to the TRAFFIC message from the AP. However the ACK message at the application layer is deleted in the final scheme in order to save bandwidth because there is already an ACK message at the MAC layer. A possible solution is to use CTS/RTS like mechanism. The vehicle can send SUPPRESS message periodically if it is chosen as the header of the group.

3. The performance needs to be evaluated when the scheme uses 802.11p standard as lower protocol layers. These differences include bandwidth, the ability to counteract Inter Symbol Interferences and Doppler shift and so on. WirelessPhyExt and 802_11Ext in NS-2.33 support 802.11p parameter settings. However it has some bugs. The bugs during my initial implementation of the scheme is fixed in the latest version NS-2.34, which is released when the thesis is almost finished.

4. We should also implement the infrastructure mode. It is not currently implemented because of the insufficiency of the NS-2 code. The throughput could be different because of the initial association procedure between the vehicle with the AP.

5. The evaluation of different routing protocols should be done. Now we only evaluated the performance when AODV routing protocol is used. Therefore only the corresponding codes in the AODV protocol are modified. If we want to use DSDR routing protocol, we have to modify the DSDR routing protocol as well. Actually although we can use the current routing protocols like AODV and DSDR, another better choice is to use a complete new routing protocol. The new routing protocol does not need to handle the multi-hop communications. Some time parameters should be changed too.

6. The scheme is implemented at the application layer. Some of the functionalities can be moved to the lower layer like the exchange of the PVT information.

7. A more realistic mobility pattern model should be used. The crossroad of the urban street and the exit of highway road are not considered in the simulation scenario. Furthermore, the mobility pattern of the vehicles in the performance evaluation is very limited.

8. The implementation in V2V sharing phases uses only one thread. The vehicle can only be in send or receive state. In the future implementation, we can use two threads. One is send thread and another one is receive thread.

Bibliography

[1] M.L. Sichitiu and M. Kihl. Inter-vehicle communication systems: a survey. *Communications Surveys and Tutorials, IEEE*, 10(2):88–105, Quarter 2008.

[2] Mobile ad hoc network. http://en.wikipedia.org/wiki/manet.

[3] Global Positioning System. http://en.wikipedia.org/wiki/global_positioning_system.

[4] Client/Sever Model. http://en.wikipedia.org/wiki/client-server.

[5] Network Simulator ns 2. http://www.isi.edu/nsnam/ns/.

[6] The ns Manual. http://www.isi.edu/nsnam/ns/doc/.

[7] J Kuri and SK Kasera. Reliable multicast in multi-access wireless lans. In *Infocom*, 1999.

[8] Zhao Li and Th. Herfet. Beacon-driven leader based protocol over a ge channel for mac layer multicast error control. In *International Journal of Communications, Network and System Science (IJCNS)*, 2008.

[9] H. Gossain, N. Nandiraju, K. Anand, and D.P. Agrawal. Supporting mac layer multicast in ieee 802.11 based manets: issues and solutions. In *Local Computer Networks, 2004. 29th Annual IEEE International Conference on*, pages 172–179, Nov. 2004.

[10] Byung-Seo Kim, Sung Won Kim, and R.L. Ekl. Ofdma-based reliable multicasting mac protocol for wlans. *Vehicular Technology, IEEE Transactions on*, 57(5):3136–3145, Sept. 2008.

[11] S. W. Kim and B. Kim. Ofdma-based reliable multicast mac protocol for wireless ad-hoc networks. In *ETRI Journal*, volume 31, pages 83–85, February 2009.

[12] Min-Te Sun, Lifei Huang, A. Arora, and Ten-Hwang Lai. Reliable mac layer multicast in ieee 802.11 wireless networks. In *Parallel Processing, 2002. Proceedings. International Conference on*, pages 527–536, Aug. 2002.

[13] Xiaoli Wang, Lan Wang, Yingjie Wang, and Daqing Gu. Reliable multicast mechanism in wlan with extended implicit mac acknowledgment. In *Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE*, pages 2695–2699, May 2008.

[14] Jin Zhang, Qian Zhang, and Weijia Jia. A novel mac protocol for cooperative downloading in vehicular networks. In *Global Telecommunications Conference,*

2007. GLOBECOM '07. IEEE, pages 4974–4978, Nov. 2007.

[15] Carlos De Morais Cordeiro, Hrishikesh Gossain, and Dharma P. Agrawal. Multicast over wireless mobile ad hoc networks: Present and future directions. *IEEE Network*, 17:52–59, 2003.

[16] Alberto Gordillo Munoz. Multicast over vehicle ad hoc networks. A surey paper in 2009.

[17] Martin Mauve, Holger Fussler, Jorg Widmer, and Thomas Lang. Position-based multicast routing for mobile ad-hoc networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 7(3):53–55, 2003.

[18] Matthias Transier, Holger Fler, Jrg Widmer, Martin Mauve, and Wolfgang Effelsberg. A hierarchical approach to position-based multicast for mobile ad-hoc networks. In *Springers Wireless Networks - The Journal of Mobile Communication, Computation and Information*, 2004.

[19] Xiaojing Xiang, Zehua Zhou, and Xin Wang. Robust and scalable geographic multicast protocol for mobile ad-hoc networks. In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pages 2301–2305, May 2007.

[20] R. Boivie, N. Feldman, and C. Metz. Small group multicast: a new solution for multicasting on the internet. In *Internet Computing, IEEE*, volume 4, pages 75 – 79, 2000.

[21] K. Chen and K. Nahrstedt. Effective location-guided tree construction algorithms for small group multicast in manet. In *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1180–1189 vol.3, 2002.

[22] Valery Naumov, Rainer Baumann, and Thomas Gross. An evaluation of inter-vehicle ad hoc networks based on realistic vehicular traces. In *MobiHoc '06: Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing*, pages 108–119, New York, NY, USA, 2006. ACM.

[23] Jakob Eriksson, Hari Balakrishnan, and Samuel Madden. Cabernet: vehicular content delivery using wifi. In *MobiCom '08: Proceedings of the 14th ACM international conference on Mobile computing and networking*, pages 199–210, New York, NY, USA, 2008. ACM.

[24] Bin Bin Chen and Mun Choon Chan. Mobtorrent: A framework for mobile internet access from vehicles. In *INFOCOM 2009. The 28th Conference on Computer Communications. IEEE*, 2009.

[25] Gian Paolo Perrucci, Frank H.P. Fitzek, Amine Boudali, Maria Canovas Mateos, Peter Nejsum, and Sune Studstrup. Cooperative web browsing for mobile phones. In *International Symposium on Wireless Personal Multimedia Communications(WPMC'07)*, 2007.

[26] L. Militano, F.H.P. Fitzek, A. Iera, and A. Molinaro. On the benecial effects of cooperative wireless peer to peer networking. In *Tyrrhenian International*

Workshop on Digital Communications, TIWDC, 2007.

[27] A. Nandan, S. Das, G. Pau, M. Gerla, and M.Y. Sanadidi. Co-operative downloading in vehicular ad-hoc wireless networks. In *Wireless On-demand Network Systems and Services, 2005. WONS 2005. Second Annual Conference on*, pages 32–41, Jan. 2005.

[28] Uichin Lee, Joon-Sang Park, Joseph Yeh, Giovanni Pau, and Mario Gerla. Code torrent: content distribution using network coding in vanet. In *MobiShare '06: Proceedings of the 1st international workshop on Decentralized resource sharing in mobile computing and networking*, pages 1–5, New York, NY, USA, 2006. ACM Press.

[29] Joon-Sang Park, Uichin Lee, Soon Y. Oh, Mario Gerla, Desmond S. Lun, Won W. Ro, and Joonseok Park. Delay analysis of car-to-car reliable data delivery strategies based on data mulling with network coding. *IEICE transaction on information and system*, E91-D(10):2524–2527, 2008.

[30] Marco Fiore, Diego Borsetti, Claudio Casetti, and Carla-Fabiana Chiasserini. Cooperative support for localized services in vanets. Technical report, Dipartimento di Elettronica, Politecnico di Torino.

[31] Marco Fiore, Francesco Mininni, Claudio Casetti, and Carla-Fabiana Chiasserini. To cache or not to cache? In *INFOCOM*, 2009.

[32] Run length encoding. http://en.wikipedia.org/wiki/run-length_encoding.

[33] Ns Limitations. Introduction to network simulator ns2.

[34] The CMU Monarch Projects Wireless and Mobility Extensions to ns. http://www.monarch.cs.cmu.edu/.

[35] T. Issaraiyakul and E. Hossain. http://www.isi.edu/nsnam/ns/ns-limitations.html.

[36] OTcl Object Tcl Extensions. http://bmrc.berkeley.edu/research/cmt/cmtdoc/otcl/.

[37] VanetMobiSim. http://vanet.eurecom.fr/.

Acronyms

AP Access Point. 17–20

BLBP Beacon-driven Leader-Based Protocol. 10

CSMA/CA Carrier Sense Multiple Access with Collision Avoidance. 4, 10

CTP Cabernet Transport Protocol. 17–19

CTS Clear To Send. 11, 38, 39

EMH Extended Multicast Header. 11

GNSS Global Navigation Satellite System. 2, 12, 14

GPS Global Positioning System. 2–4, 26

HTTP Hypertext Transfer Protocol. 18, 20

IETF Internet Engineering Task Force. 14

IGMP Internet Group Management Protocol. 15

ISI Inter Symbol Interruption. 3

ITS Intelligent Transportation Systems. 3

IVC Inter-Vehicle Communications. 1

LBP Leader-Based Protocol. 10

LGK Location-Guided K-ary. 16

LGS Location-Guided Steiner. 16

MANET Mobile Ad-hoc Network. 1, 2, 9, 11, 12, 15, 23

VANET Vehicular Ad-hoc Network. 1–5, 9, 10, 12–14, 20, 21, 24

VCD Vehicle Content Downloading. 46

WAVE Wireless Access in Vehicular Environments. 3

WiFi Wireless Fidelity (IEEE 802.11b wireless networking). 17–19

WIS Weighted Independent Set. 11, 12

WWAN Wireless Wide-Area Network. 19, 20