A Study on Efficient and Reliable Data Transfer Protocols for Vehicular Ad Hoc Networks

by

Celimuge Wu

A Thesis Submitted in
Partial Fulfilment of the Requirements
for the Degree of
DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF ELECTRO-COMMUNICATIONS
June 2010

A Study on Efficient and Reliable Data Transfer Protocols for Vehicular Ad Hoc Networks

by

Celimuge Wu

APPROVED by SUPERVISORY COMMITTEE :

Chairperson : Prof. Hiroyoshi Morita

Member : Prof. Hiroshi Nagaoka

Member : Prof. Akihiko Ohsuga

Member : Prof. Tsutomu Yoshinaga

Member : Assoc. Prof. Hisashi Koga

Member : Assoc. Prof. Satoshi Ohzahata

車車間アドホックネットワークにおける高効率高信頼性データ転送プロトコルについての研究

策力木格

概要

車両アドホックネットワーク（Vehicular Ad hoc Network, VANET）は近隣の車車間や車と固定路側装置の間でコミュニケーションを提供するモバイルアドホックネットワークの一種類である。車車間アドホックネットワークにおいていろいろなアプリケーションが検討されているが、それらアプリケーションのルーティング要件に従って本論文ではポイントツーポイントデータ転送アプリケーションとブロードキャストデータ転送アプリケーションの二種類に分類する。ポイントツーポイントデータ転送を使うことにより、ユーザが車で移動時に音楽をダウンロードする、メールを送信する、後部座席の乗客がゲームをすることが可能になる。ブロードキャストデータ転送は交通警告メッセージや、近くのサービス情報やリアルタイムのルート情報などのメッセージ配布に使用される。

ポイントツーポイントコミュニケーションの場合には、車車間アドホックネットワークトポロジーの頻繁な変化により、AODV などの汎用ルーティング・プロトコルが適応できない。したがって、一連の中間ノードを通してソースノードから目的ノードまでの信頼できる経路を見つけることは特に重要である。ブロードキャストデータ通信の場合、データ転送プロトコルが、様々な交通状況において高信頼性、低遅延と低オーバヘッドを提供するべきである。

車車間アドホックネットワークにおける効率的なポイントツーポイント通信を実現するため、本論文では高い移動状況において効率よく機能するルーティング・プロトコル QLAODV (Q-Learning AODV)を提案する。QLAODV は分散的強化学習ルーティング・プロトコルである。QLAODV は Q-Learning を利用してネットワークのリンクステータス情報を学習する。リンクステータス情報として、ホップ数、帯域幅、移動性を考慮する。また、QLAODV はユニーキャストパケットを使って経路の有用性をチェックすることにより、Q-Learning がダイナミックネットワーク環境において効率よく働くことをサポートする。QLAODV では従来のルート・メンテナンスと異なるダイナミックルート変更メカニズムを使って、ルートが切断される前によりよいルートに変更する。これにより、ルート・エラーによるパケットロスを削減できる。また常によりよい経路を使うことによってネットワーク全体の効率を向上させることが可能になる。ダイナミックルート変更メカニズムを使うことにより QLAODV がダイナミックな車車間アドホックネットワークのポイントツウポイントデータ転送に適応できる。

また、車車間アドホックネットワークにおいて、ブロードキャストメッセージを配布するために、本論文では高信頼かつ高効率なマルチホップ・ブロードキャスト・プロトコルを提案する。提案プロトコルは様々な交通状況において厳密な信頼性を提供する。提案プロトコルでは効率的なメッセージ受信状況確認メカニズムを使った。メッセージの受信が確認できない場合には、メッセージを再送することにより、厳密な信頼性を保証する。また、提案プロトコルはブロードキャストメッセージ数を削減することにより、高密度のネットワーク環境における冗長なブロードキャストを回避してオーバヘッドを最低限に抑える。 また、本論文は車の移動性を考えた中継ノード選択アルゴリズムを提案して、それを提案ブロードキャストプロトコルに利用した。この提案中継ノード選択アルゴリズムを使うことにより、提案ブロードキャストプロトコルが車車間アドホックネットワークのダイナミックモバイル特性に適応できる。信頼性、効率性そして移動性を考慮した結果、提案マルチホップ・ブロードキャスト・プロトコルが車車間アドホックネットワークでのブロードキャストデータ転送に適応できる。

A Study on Efficient and Reliable Data Transfer Protocols for Vehicular Ad Hoc Networks

Celimuge Wu

ABSTRACT

A Vehicular Ad hoc Network (VANET) is a form of mobile ad hoc network providing communications between vehicles in close proximity, and between vehicles and nearby fixed roadside equipment. Vehicular ad hoc networks attract a variety of applications which can be classified into two categories of point to point data transfer applications and broadcast data transfer applications according to their routing requirements. Point to point data transfer applications can provide point-to-point connectivity to vehicular nodes while on the move, so the users can download music, send emails, or play back-seat passenger games. On the other hands, broadcast applications can be used to disseminate messages such as traffic alert messages, nearby service information and real-time routes information.

In case of point-to-point communications, general-purpose ad hoc routing protocols such as AODV cannot work efficiently due to frequent changes of network topology caused by vehicles' movement. Thus, the routing problem of finding reliable paths from a traffic source to a traffic destination through a series of intermediate forwarding nodes is particularly challenging. As far as broadcast communications are concerned, the routing protocols should provide strict reliability, lower delay and should be lightweight and suitable for different traffic conditions.

To provide efficient point-to-point data transfers in vehicular ad hoc networks, this thesis first proposes a VANET routing protocol QLAODV (Q-Learning AODV) which fits for unicast applications in high mobility scenarios. QLAODV is a distributed reinforcement learning routing protocol, which uses a Q-Learning algorithm to infer network state information and uses unicast control packets to check the path availability in a real time manner in order to allow Q-Learning to work efficiently in highly dynamic network environment. QLAODV is favored by its dynamic route change mechanism and therefore is capable of reacting quickly to network topology changes.

To disseminate broadcast messages in vehicular ad hoc networks, this thesis also proposes a reliable and efficient multi-hop broadcast protocol for vehicular ad hoc networks. The proposed protocol provides the strict reliability in various traffic conditions. This protocol also performs low overhead by means of reducing rebroadcast redundancy in a high-density network environment. This thesis also proposes an enhanced multipoint relay (MPR) selection algorithm that considers vehicles' mobility and then uses it for relay node selection.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1. Mobile Ad Hoc Networks and Vehicular Ad Hoc Networks

1.1.1. Mobile Ad Hoc Networks

A mobile ad hoc network (MANET) is a self-configuring network of mobile devices connected by wireless links. The vision of mobile ad hoc networking is to support robust and efficient operation in mobile wireless networks by incorporating routing functionality into mobile nodes. Due to its high flexibility and low cost, mobile ad hoc technologies can be inexpensive alternatives or enhancements to fixed network and cell-based mobile network infrastructures. Despite the mentioned advantages and potential application possibilities, ad-hoc networks are yet far from being deployed on large-scale commercial basis. Some fundamental ad-hoc networking problems remain unsolved or need optimized solutions.

A MANET consists of mobile nodes, which are free to move arbitrarily. The nodes may be located in or on airplanes, ships, trucks, cars, perhaps even on people or very small devices. A MANET is an autonomous system of mobile nodes. The system may operate in isolation, or may have interfaces with a fixed network. In general, MANETs have several characteristics as follows.

1. Dynamic topologies: Nodes are free to move arbitrarily. Thus, the network topology may change unpredictably and rapidly.
2. Bandwidth-constrained wireless links: Wireless links will continue to have significantly lower capacity than wired links. In addition, due to the effects of multiple access, fading, noise, and interference conditions, the effective throughput of wireless communications is often much less than a radio's maximum transmission rate.
3. Energy-constrained device: Some or all of the nodes in a MANET may rely on batteries or other exhaustible means for their energy. For these nodes, the most important system design criteria for optimization may be energy conservation.
4. Limited physical security: Mobile wireless networks are generally more prone to physical security threats than fixed-cable nets are. The increased possibility of eavesdropping, spoofing, and denial-of-service attacks should be carefully considered.
5. MANET nodes are equipped with wireless transmitters and receivers using antennas that can be omnidirectional or unidirectional.

As above mentioned, mobile ad hoc networks have many unique characteristics compared to wired networks. As a result, they arise many different researches from different aspects. These researches include the issues on connectivity, capacity, medium access, routing, Quality of Service, transport, application, cross layer design and so on.

1.1.2. Vehicular Ad Hoc Networks

A Vehicular Ad hoc Network (VANET) is a form of mobile ad hoc network providing communications between vehicles in close proximity, and between vehicles and nearby fixed roadside equipment. Figure 1.1 shows an example of vehicular ad hoc networks. A benefit of using vehicular ad hoc networks is to be possible to deploy these networks in areas where it isn't feasible to install the needed infrastructure. It would be expensive and unrealistic to install access points to cover all of the roads in the world. Another benefit of using vehicular ad hoc networks is they can be quickly deployed with no administrator involvement.

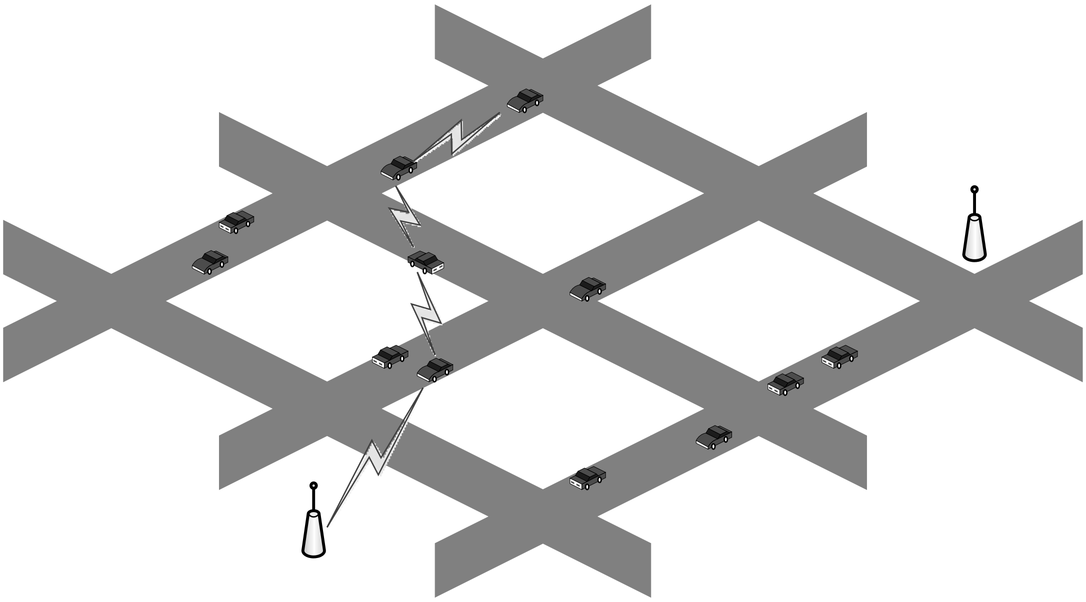


Figure 1.1 – An example of vehicular ad hoc networks.

Vehicular Ad Hoc Networks have several properties that distinguish them from other mobile ad hoc networks.

1. High mobility: Nodes (vehicles) in vehicular ad hoc networks are highly mobile. This feature makes multi-hop communications in vehicular ad hoc networks particularly challenging.

2. Restricted vehicle mobility: The mobility of vehicles in the ad hoc networks is restricted to the roads on which the vehicles travel. Therefore, the mobility of vehicles are relatively predictable.

3. Enough power supply: Battery power is not an issue for vehicular ad hoc net works because vehicles usually have enough power supply by its equipped generator.

4. Unpredictable node density: Node density in vehicular ad hoc networks is time variable. The node density can be very high when a traffic jam occurs and can be very low in the late night hours.

1.2. Vehicular Ad Hoc Network Applications

The opportunities for vehicular ad networks are growing rapidly. Chaabouni *et al.* [1] have given an overview of some inter-vehicular applications and their main characteristics. According to [1], inter-vehicular applications can be classified into four main application groups; traffic safety, floating car data, Internet access and group communication.

Toor *et al.* [2] have reviewed the possible applications used in VANETs, namely, safety applications and user applications by identifying their requirements. Applications that increase vehicle safety on the roads are called safety applications. Applications that provide value added services, for example, entertainment, are called user applications.

However, in this thesis, VANET routing protocols are classified into two categories according to their intended receivers: point-to-point data transfer applications and broadcast data transfer applications. Point-to-point data transfer applications are the applications in which one sender sends data to another receiver. In contrast, broadcast data transfer applications are the applications in which one sender disseminates data to multiple intended receivers. The design requirement of point-to-point data transfer protocols is totally different to the requirement of broadcast data transfer protocols. With this classification, protocol designers can easily understand the design fundamentals.

1.2.1. Point-to-point Data Transfer Applications

Plenty of interesting applications for vehicular ad hoc networks are expected to be available in the near future. Point to point data transfer applications in vehicular ad hoc networks can be in the following form but not limited to.

1. Internet access: Although a vehicle does not have the Internet access, the vehicle can access the Internet using multi-hop communication with other vehicles which have been already connected with the Internet. The vehicle also can use roadside access point to connect to the Internet at the help of other vehicles.

2. Information Collection: A roadside access point can gather the nearby vehicles' information using multi hop communications. This can be achieved by vehicles use point-to-point communications to send its own data to the access point. This kind of information collection can be very interesting. For example, the access point can easily gather the road traffic information using point-to-point applications and can use these information to help other vehicles to avoid traffic jam.

3. Request for Information: Service information can be disseminated by using multi-hop broadcast methods. However, the broadcast is a bandwidth consuming approach, especially when many vehicles do not need the information. Because of limited wireless resource, comfort application should reserve enough bandwidth to safety applications (In this thesis, applications that increase vehicle safety on the roads are called safety applications. Applications that provide value added services, for example, entertainment, are called comfort applications). Therefore, to reduce redundant broadcast, point-to-point applications are used to provide information in a reactive manner. When a vehicle needs the information, it can send a request to the information holder. The information holder, a vehicle or access point, then uses point-to-point communications to provide the information to the vehicle.

1.2.2. Broadcast Data Transfer Applications

Broadcast data transfer applications in vehicular ad hoc networks can be in the following form but not limited to.

1. Accident warning: Vehicles travel at a high speed on major roads. This gives drivers very little time to react to the vehicle in front of them. As a result, when an accident occurs, the approaching vehicles crash before they stop. To avoid this crash, emergency information such as collision or emergency braking can be propagated along the road to notify drivers ahead of time so that necessary action can be taken to avoid accidents [3]. These applications also can be used to warn cars of an accident that occurred further along the road, thus preventing a pile-up from occurring [2].

2. Traffic alert system: Roadside equipment can broadcast warning information, i.e. “ Slippery Road ” and “Right Lane Closed Ahead,” to alert the upcoming vehicles to decelerate. A vehicle that senses a warning situation also can disseminate these information in order to help the following vehicles to do a preventive action.

3. Service information dissemination: Gas stations and parking lots can disseminate service information to multiple hops away. Roadside equipment can also disseminate recent traffic information to the upcoming vehicles to make the driving more efficient.

1.3. Challenges in Vehicular Ad Hoc Networks

Due to the unique features of vehicular ad hoc networks (Section 1.1.2), data transfer issues in vehicular ad hoc networks become very challenging.

1.3.1. Challenges in Point-to-point Data Transfer

1. High degree of mobility: The main challenges of point-to-point data transfer in vehicular ad hoc networks are high mobility and frequent link changes. Thus, the routing problem of finding reliable paths from a traffic source to a traffic destination through a series of intermediate forwarding nodes is particularly challenging. (The terms node and vehicle are used interchangeable in this thesis.) Therefore, to design an efficient routing protocol for VANETs is very crucial. Plenty of routing protocols have been proposed to handle point-to-point communication in mobile ad hoc networks. However, existing MANET routing protocols are not suitable in VANETs because of frequent link changes and route reconstructions. Although there have been several routing protocols which are designed for VANETs, they all have their limitations [4–11]. Some are only designed for highway scenarios or other particular scenarios [4, 7, 11], while others rely on the existence of positioning devices or other auxiliary devices [4–10]. Thus, they are not general solutions to VANET routing problems and therefore can not be used in various situations of inter vehicular point-to-point applications.

2. Efficiency: Efficiency is also a critical issue in point-to-point data transfer protocols. The efficiency includes high delivery ratio, low overhead, low end-to-end delay etc. In order to perform efficient data transfers in vehicular ad hoc networks, the best route that indicates the highest performance should be chosen. A good point-to-point data transfer protocol also should be able to efficiently utilize the limited wireless resources.

1.3.2. Challenges in Broadcast Data Transfer

1. Reliability: The main purpose of broadcast data transfer in vehicular ad hoc networks is to disseminate accident warnings and traffic alert messages. Broadcast data transfers are expected to be able to significantly reduce the number of road

accidents. However, the precondition is that the related messages can be successfully delivered to all desired receivers. Therefore, reliability is the most important issue for broadcast data transfer in vehicular ad hoc networks. Since wireless communication is unreliable, reliability issue should be considered in the protocol design.

2. Efficiency: Like other mobile ad hoc networks, wireless resources in vehicular ad hoc networks are limited. Therefore, broadcast messages should be delivered to intended receivers without consuming too much wireless resources. A short dissemination delay also should be guaranteed because a long delay can make a warning message meaningless. However, when the node density is high, it becomes difficult to provide high level of reliability and efficiency because of collision and contention. Therefore, the protocol design for broadcast data transfer applications should consider high-density environment.

3. Mobility handling: There is no reliability and efficiency without considering mobility in vehicular ad hoc networks. Topology changes can happen any time in vehicular ad hoc networks. This results ensuring reliability and efficiency becomes difficult. Many broadcast protocols, which aim to provide efficiency and mobility, only consider low level of node movement or do not consider mobility at all. In the case of highly mobile vehicular ad hoc networks, these protocols should be reconsidered.

1.4. Research Contributions

The main objective of this thesis is to discuss efficient data transfer schemes in both point-to-point and broadcast data transfer in vehicular ad hoc networks. The main contributions of the thesis are as follows.

1.4.1. A Point-to-point Routing Protocol

Based on original AODV, an enhanced routing protocol called QLAODV (Q-Learning AODV) is proposed. QLAODV uses a Q-Learning algorithm [12, 13] to estimate whole network link status information from local communication and to change routes preemptively using learned information. A route change request/reply mechanism is also proposed to check the availability of a newly learned route. Simulations are conducted in random waypoint model, freeway model, Manhattan model and real street map based mobility model. Through the performance comparison with AODV and two extensions of AODV protocol on different mobility model, QLAODV is confirmed to be able to discover better routes in dynamically changing network without having to know the network topology and traffic patterns in advance, and without the need for any centralized routing control system, therefore can adjust quickly to topology changes.

1.4.2. A Multi Hop Broadcast protocol

To perform efficient multi hop relays, a relay node selection algorithm (enhanced MPR selection algorithm), which considers node mobility, is proposed. As a result of including mobility prediction in MPR selection procedure [14], the enhanced MPR algorithm selects relatively stable nodes and therefore can improve data dissemination ratio regardless of node velocity and hello interval.

Based on the enhanced MPR selection algorithm, a reliable and efficient broadcast protocol is proposed. The proposed protocol can work well in various traffic conditions. The proposed protocol uses a hop-by-hop retransmission scheme in the data flooding to provide the strict reliability in various traffic conditions. This protocol also performs low overhead in a high-density network environment by means of introducing boundary nodes which are in charge of rebroadcasting. The proposed protocol also works well in a sparse network. The effectiveness of the proposed protocol is confirmed through simulations using the network simulator ns-2.

1.5. Thesis Outline

The remainder of the thesis is organized as follows.

Chapter 2 gives a brief description of the data transfers in vehicular ad hoc networks. VANET routing protocols are classified into point-to-point data transfer protocols and broadcast data transfer protocols according to their intended receivers.

In Chapter 3, QLAODV (Q-Learning AODV), an enhanced routing protocol which extends AODV, is proposed. QLAODV uses a Q-Learning algorithm [13] to achieve whole network link status information from local communication and to change routes preemptively using the information so learned. In order to make the Q-Learning algorithm work efficiently in highly dynamic networks, a route change request/reply mechanism is proposed to check the usability of a newly learned route.

Chapter 4 first proposes an enhanced MPR selection algorithm which takes into account the node mobility. Based on the enhanced MPR selection algorithm, a multi-hop broadcast protocol which can deliver messages to all desired receivers is proposed in the same chapter. The proposed protocol uses selected boundary nodes to relay data for avoiding broadcast storm problems in high-density networks. This mechanism substantially reduces the message overhead as compared to a simple flooding mechanism. The proposed protocol is robust to mobility and channel error by use of a strict retransmission mechanism in case of packet losses.

Finally, Chapter 5 draws the conclusions and directions for future works.

Chapter 2

Overviews of data transfers in vehicular ad hoc networks and reinforcement learning

In this thesis, VANET routing protocols are classified into point-to-point data transfer protocols and broadcast data transfer protocols according to their intended receivers. Point-to-point data transfers are typically executed by connecting two nodes together over other intermediate nodes. Multi-hop broadcast transfers can be used when a sender node has to disseminate data to a large number of destination nodes.

2.1. Point-to-point Data Transfers

The main features of vehicular ad hoc networks are high mobility and frequent link changes. Thus, the routing problem of finding reliable paths from a traffic source to a traffic destination through a series of intermediate forwarding nodes is particularly challenging. Therefore, to design an efficient routing protocol for VANETs is very crucial. Li and Wang [15] have discussed the research challenge of routing in VANETs and surveyed recent routing protocols. Generally, VANET routing protocols in point-to-point applications can be classified into two different approaches; position based routing and topology based routing.

In the position based routing, the routing decision at each node is based on the destination's position contained in the packet and the position of the forwarding node's neighbors [16,17]. Thus, position based routing protocols require availability of participating nodes' physical position. Each node determines its own position through the use of GPS or other type of positioning service. However, to acquire the destination node's position, a location service is required. Therefore, the performance of position based routing in highly dynamic networks usually depends on what kind of location service has been used.

In contrast, topology based routing protocols use the information about the links that exist in the network to perform packet forwarding. A large number of topology based routing protocols have been proposed for mobile ad hoc networks (MANETs). Therefore, extending mobile ad hoc network routing protocols for vehicular ad hoc networks is considered to be a solution to point-to-point routing problem.

2.1.1. Position Based Routing Protocols

GPSR

Karp and Kung [18] have proposed Greedy Perimeter Stateless Routing, GPSR, which is a routing algorithm that uses geographical information to achieve robust packet delivery on densely deployed wireless networks. GPSR makes greedy forwarding decisions using only information about a router's immediate neighbors in the network topology. When a packet reaches a region where greedy forwarding is impossible, the algorithm recovers by routing around the perimeter of the region.

The algorithm consists of two methods for forwarding packets: greedy forwarding, which is used wherever possible, and perimeter forwarding, which is used in the regions greedy forwarding cannot be.

1. Greedy forwarding: Under GPSR, packets are marked by their originator with

their destinations' locations. As a result, a forwarding node can make a locally optimal, greedy choice in choosing a packet's next hop. Specifically, if a node knows its neighbors' positions, the locally optimal choice of next hop is the neighbor geographically closest to the packet's destination. Forwarding in this regime follows successively closer geographic hops, until the destination is reached.

2. Perimeter forwarding: Greedy forwarding uses only neighbor nodes' positions to route packets. As a result, greedy forwarding fails in some topologies, in which the only route to a destination requires a packet move temporarily farther in geometric distance from the destination. In that case, GPSR uses perimeter forwarding. In a densely deployed network, greedy forwarding usually result a good performance. However, when the network is sparsely connected, greedy forwarding may fail frequently.

GPSR's benefits all stem from geographic routing's use of only immediate-neighbor information in forwarding decisions. GPSR uses position information to achieve small per-node routing state, small routing protocol message complexity, and robust packet delivery on densely deployed wireless networks. GPSR is expected to be more powerful than topology based routing protocol in a large-scale network because of its scalability. However, GPSR does not take into account that how to acquire the destination node's position information. Generally, position based routing protocols use a location service to get other nodes' position information. When the network topology changes frequently, a lot of control packet is required to get recent position information of other nodes.

2.1.2. Topology Based Routing Protocols

Topology based routing protocols can be further divided into proactive, reactive, and hybrid approaches. Proactive algorithms employ classical routing strategies such as distance vector routing or link state routing. They maintain routing information about the available paths in the network even if these paths are not currently used. The main drawback of these approaches is that the maintenance of unused paths may occupy a significant part of the available bandwidth if the topology of the network changes frequently [19]. In comparison, reactive routing protocols maintain only the routes that are currently in use, thereby reducing the burden on the network. This type of protocols find a route on demand by flooding the network with route request packets. On the other hand, hybrid routing protocols combined both the proactive and reactive approaches. Even though hybrid routing approach presents an efficient and scalable routing strategy for large scale environments, it is not suitable for VANET applications where communication partners of each node are expected to be in a short distance.

OLSR

The Optimized Link State Routing Protocol (OLSR) [20] is a well-known proactive routing protocol which is developed for mobile ad hoc networks. It operates as a table driven, proactive protocol, i.e., exchanges topology information with other nodes of the network regularly. Each node selects a set of its neighbor nodes as “multipoint relays” (*MPR*). In OLSR, only nodes, selected as such *MPRs*, are responsible for forwarding control traffic, intended for diffusion into the entire network. *MPRs* provide an efficient mechanism for flooding control traffic by reducing the number of transmissions required. This technique significantly reduces the number of retransmissions required to flood a message to all nodes in the network. OLSR only requires partial link state information to be flooded in order to provide shortest path routes. The minimal set of link state information required is, that all nodes, selected as *MPRs*, declare the links to their *MPR* selectors. Additional topological information, if present, may be utilized e.g., for redundancy purposes.

OLSR can optimize the reactivity to topological changes by reducing the maximum time interval for periodic control message transmission. Furthermore, as OLSR continuously maintains routes to all destinations in the network, the protocol is beneficial for traffic patterns where a large subset of nodes are communicating with another large subset of nodes, and where the [source, destination] pairs are changing over time. The protocol is particularly suited for large and dense networks, as the optimization done using $MPRs$ works well in this context. The larger and more dense a network, the more optimization can be achieved as compared to the classic link state algorithm.

OLSR is designed to work in a completely distributed manner and does not depend on any central entity. The protocol does not require reliable transmission of control messages: each node sends control messages periodically, and can therefore sustain a reasonable loss of some such messages. Such losses occur frequently in radio networks due to collisions or other transmission problems.

AODV

The Ad hoc On-Demand Distance Vector (AODV) [21] is known as a good performer in MANET routing protocols. AODV enables dynamic, self-starting, multihop routing between participating mobile nodes wishing to establish and maintain an ad hoc network. AODV allows mobile nodes to obtain routes quickly for new destinations, and does not require the nodes to maintain routes to destinations that are not in active communication. The operation of AODV is loop-free. By avoiding the Bellman-Ford "counting to infinity" problem, AODV offers quick convergence when the ad hoc network topology changes. When links break, AODV causes the affected set of nodes to be notified so that they are able to invalidate the routes using the lost link.

Route Requests (RREQs), Route Replies (RREPs), and Route Errors (RERRs) are the message types defined by AODV. When a route to a new destination is needed, a node broadcasts a RREQ to find a route to the destination. The route can be determined when the RREQ reaches either the destination itself, or an intermediate node with a "fresh enough" route to the destination. A "fresh enough" route is a valid route entry for the destination whose associated sequence number is at least as great as that contained in the RREQ. The route is made available by unicasting a RREP back to the origination of the RREQ. Note that upon receiving the route request, each node caches a route back to the originator of the route request, so that the RREP can be unicast from the destination along a path to that originator, or likewise from any intermediate node that is able to satisfy the request.

Nodes monitor the link status of next hops in active routes. When a link break in an active route is detected, a RERR message is used to notify the originator of the route that the loss of that link has occurred. The RERR message indicates those destinations (possibly subnets) which are no longer reachable by way of the broken link. When a link break occurs in an active route, the node upstream of that break may choose to repair the link locally if the destination is no farther than MAX_REPAIR_TTL hops away. To repair the link break, the node increments the sequence number for the destination and then broadcasts a RREQ for that destination. The node initiating the repair then waits the discovery period to receive RREPs in response to the RREQ. Data packets are buffered during the local repair. If, at the end of the discovery period, the repairing node has not received a RREP (or other control message creating the route or updating the route) to the destination, the node sends a RERR message to the source node and all the buffered packets will be dropped. If the local repair option is disabled, the node upstream of that link break will send a RERR to the source node and all packets to the destination node will be dropped.

ZRP

Hybrid routing protocols combine both the proactive and reactive approaches. Zone Routing Protocol [22] is a hybrid routing framework suitable for a wide variety of mobile ad-hoc networks, especially those with large network spans and diverse mobility patterns. Each node proactively maintains routes within a local region (referred to as the routing zone). ZRP uses a proactive approach in the Intra-zone Routing and uses a reactive approach in the Inter-zone Routing.

In ZRP, when the destination is beyond the routing zone, the source node uses a globally reactive route query/reply mechanism to discover a route. The route discovery process in ZRP can be made much more efficient than proactive protocols in terms of wireless resources, at the expense of longer latency. The proactive maintenance of routing zones helps improve the quality of discovered routes, by making them more robust to changes in network topology. The ZRP can be configured for a particular network by proper selection of a single parameter, the routing zone radius.

Although hybrid routing approach presents an efficient and scalable routing strategy for large scale environments, it is not suitable for VANET applications where communication partners of each node are expected to be in short distance.

2.1.3. Comparison and Solution

For position based routing protocols, in order to learn the current position of a specific node, help of location service which mobile nodes register their current positions, is required. When a node does not know the position of a communication partner, the node contacts the location service and requests that information. However, nodes's positions become unstable with increasing mobility. Maintaining position information needs additional control packets which lead to bandwidth wastage. Therefore, it is expected that the performance of position based routing will be limited by high control overhead.

In this thesis, topology based routing approach is employed in VANET point-to-point applications. There are several reasons for employing the topology based approach. First, the topology based approach does not depend on particular instruments. Although GPS like positioning service is considered to be possible in VANETs, it may not be affordable for every vehicle. Secondly, when compared with topology based routing protocols, position based routing protocols could not offer enough performance. Many performance comparisons between position based routing protocols and topology based routing protocols assume that nodes can determine the location of their neighbors and destinations (e.g., GPSR [18], MURU [23]). Usually, in order to determine the position of other node, location service is necessary in position based routing. In highly dynamic networks, to get precise position information of other nodes, large numbers of signaling packets are needed. Apparently, these controls incur more traffic overhead and lead to performance deterioration. Third, the position based approach is likely to fail if obstacles influence the transmission ranges of vehicles.

Topology based routing in VANETs has been studied recently and many protocols have been proposed [4–8]. Namboodiri and Gao [4] introduce a prediction-based routing (PBR) protocol which takes advantage of the predictable mobility pattern of vehicles on highways. Since the prediction-based routing protocol [4] only focuses on free-flowing highway traffic, this protocol is not applicable to urban vehicular ad hoc networks. Taleb *et al.* [5] introduce a scheme which groups vehicles according to their moving directions. However, in case of winding roads (e.g., mountainous areas), the approach of grouping vehicle on the basis of their velocity vector is inadequate. Yang *et al.* [6] present the connectivity aware routing protocol, which selects routes with the highest probability of connectivity and thus avoids network disconnections in VANETs. The connectivity aware routing protocol [6] assumes vehicles are installed with a pre-loaded digital map. Ducourthial *et al.* [7] present a novel approach for routing in highly dynamic networks, relying on condition-based communication. Instead of transporting addresses (or positions), a message is sent with some conditions used for retransmission or reception. Owing to the dynamic evaluation of the conditions, this conditional-transmission technique can efficiently support the high dynamic of vehicular networks. However, the main drawback of the conditional-transmission technique is its application dependency feature. Considering the unique character of VANETs, Lu *et al.* [8] present a thorough discussion on the feasibility of enhancing the network performance by introduction of buses, road lamps and traffic lights as the bridge nodes in the city area. However, this thesis aims to propose a general VANET routing protocol that considers the main feature of VANETs but does not rely on the bridge nodes.

It appears to be more promising to modify existing routing protocol than to design a new protocol from scratch. AODV is known as a good performer protocol in reactive routing protocols. There have been several research attempts which try to extend use of the AODV protocol to VANETs [9–11]. Menouar *et al.* [9] improve the AODV routing process by selecting the most stable route with respect to the movement of the vehicles. However, the movement prediction-based routing proposed by Menouar *et al.* [9] is only interested in route discovery process, therefore can not adapt quickly to frequent topology changes. Moreover, the protocol generates more overhead than AODV. Wang *et al.* [10] introduce a Two-Phase routing protocol (TOPO) that incorporates map information in routing. TOPO defines two phases in routing, namely routing in access and overlay. While overlay is a graph of high vehicular density roads, (e.g. state roads, highways), access is the rest of the areas/roads connecting to the overlay. The protocol defines an overlay graph with roads of high vehicular density and forwards packet along the pre-calculated path in the overlay. Since mainly consider the large scale VANETs, TOPO does not work well in small scale scenarios. This is because, in a small scale vehicular ad hoc network, using overlay in routing results longer hops and consequent performance degradation. TOPO utilizes the road and traffic information on overlay and delivers message along overlay to the access area of destination. Therefore, TOPO is not suitable for high data rate traffic because it faces the problem of wireless channel congestion in the overlay. Abedi *et al.* [11] propose DAODV protocol that uses two parameters, direction of movement and vehicle position, to select the next hop during the route discovery phase. Although DAODV protocol can establish more stable route than AODV, it only considers situations that the source node and the destination node are moving in the same direction or in the opposite direction. The situation of source node and destination node are orthogonal is not discussed. Additionally, Refs. [5, 6, 9–11] assume that every node knows its own position and Wang *et al.* [10] assume that map information is also available.

Through the above examination, it is easy to know that although many protocols have been proposed in VANETs, there are still a lot of work needs to be done. A general routing protocol in point-to-point applications should satisfy following requirements.

1. The protocol should acquire current network topology information without incurring much signaling load and without centralized routing control system.
2. The protocol should be reactive in nature and can construct new routes before existing routes fail.
3. The protocol should consider the main feature of VANETs, and should not depend

on particular instrument or particular situation.

Considering above requirements, this thesis adopts a reinforcement learning algorithm to acquire network topology information in a distributed manner. The possible routes are evaluated based on the stability, bandwidth usage and route length using Q-Learning algorithm [13]. A dynamic route change mechanism is used to change to a new route before the current route is disconnected. Without loss of generality, a topology-based approach, which is independent to GPS like localization service and roadside instruments, is employed.

2.2. Broadcast Data Transfers

Broadcast is a frequently used method in vehicular ad hoc networks, such as sharing road traffic, weather, emergency, road condition among vehicles, and delivering advertisements and announcements [15]. When the message needs to be disseminated to the vehicles beyond the transmission range, multi-hop data transfers are used.

2.2.1. Flooding

The simplest way to implement a broadcast service is flooding. In Flooding, each node rebroadcasts messages to all of its neighbors except the one it got this message from. Flooding is simple and easy to implement. Flooding gives a high probability that each node, which is not isolated from the network, will receive the broadcast message. However, in a highly dynamic network where collisions and node movement may happen, flooding cannot provide enough reliability. Its non-support of retransmission degrades the data delivery ratio and its redundant rebroadcasts described below causes many collisions resulting in making the delivery ratio worse. In the heavy traffic condition where the VANET communication is likely to be exploit, many vehicles exist in a dense manner within a radio transmission range. In such a high-density network environment, flooding introduces redundant rebroadcast, that is, many vehicles within a radio transmission range try to rebroadcast a received message, and it causes high overhead in the data dissemination.

Flooding intends to guarantee a message will eventually reach all nodes in the network. However, flooding is not enough for disseminating vehicular safety messages which require high level of reliability (in this thesis, messages that increase vehicle safety on the roads are called vehicle safety messages). Even in a sparsely connected vehicular ad hoc network, flooding gives poor performance when the packet generation rate is high. This is because the collision frequently occurs and flooding does not consider the retransmission.

2.2.2. *MPR*

The Optimized Link State Routing Protocol (OLSR) [20] employs multipoint relay (*MPR*) technique [14] to substantially reduce the message overhead as compared to a classical flooding mechanism, where every node retransmits each message when it receives the first copy of the message. In OLSR, each node selects a set of its neighbor nodes as "multipoint relays" (*MPRs*). Only nodes, selected as *MPRs*, are responsible for forwarding the control traffic, intended for diffusion into the entire network. *MPRs* provide an efficient mechanism for flooding control traffic by reducing the number of transmissions required.

The idea of multipoint relay is to minimize the overhead of flooding messages in the network by reducing redundant rebroadcasts in the same region. Each node in the network selects a set of nodes in its one-hop neighborhood which may retransmit the broadcast messages. This set of selected neighbor nodes is called the "Multipoint Relay" (*MPR*) set of the node. The neighbors of node N which are not in its *MPR* set, receive and process broadcast messages but do not retransmit broadcast messages received from node N.

Each node selects its *MPR* set from its one-hop neighbors. This set is selected such that these nodes cover (in terms of radio range) all two-hop nodes. The *MPR* set of N, denoted as $MPR(N)$, is then an arbitrary subset of the one-hop neighborhood of

N. $MPR(N)$ satisfies the following condition: every node in the symmetric strict 2-hop neighborhood of N must have a link towards $MPR(N)$. The smaller a MPR set (in term of the number of nodes in the set), the less control traffic overhead results from forwarding control messages.

In OLSR, each node maintains information about the set of neighbors that have selected them as MPR. This set is called the "Multipoint Relay Selector set" (MPR selector set) of a node (Note that the MPR selector set is different from the MPR set). A node obtains this information (MPR selector set) from periodic HELLO messages received from its neighbors.

A broadcast message, intended to be diffused in the whole network, coming from any of the MPR selectors of node N is assumed to be retransmitted by node N, if N has not received the broadcast message yet. The MPR-set of a node can be changed over time (i.e., when a node selects another MPR-set) and is indicated by the node in their HELLO messages.

Qayyum *et al.* [14] have proposed a heuristic for the selection of multipoint relays as follows.

1. Start with an empty multipoint relay set $MPR(x)$.
2. First select those one-hop neighbor nodes in $N(x)$ as multipoint relays which are the only neighbor of some node in $N^2(x)$, and add these one-hop neighbor nodes to the multipoint relay set $MPR(x)$.
3. While there still exist some node in $N^2(x)$ which is not covered by the multipoint relay set $MPR(x)$:
   (a) For each node in $N(x)$ which is not in $MPR(x)$, compute the number of nodes that the node covers among the uncovered nodes in the set $N^2(x)$.
   (b) Add that node of $N(x)$ in $MPR(x)$ for which this number is maximum.

MPR can optimize the message dissemination by minimizing the number of messages flooded in the network. The technique is particularly suitable for large and dense networks. However, MPR cannot be used in vehicular ad hoc networks without enhancement because MPR does not consider node mobility at all. In vehicular ad hoc networks, because of node movement, the neighborhood information can become imprecise. Therefore, the relay node selection mechanism in vehicular ad hoc networks should consider the node mobility.

2.2.3. Weighted p-Persistence, Slotted 1-Persistence and Slotted p-Persistence Scheme

Wisitpongphan and Tonguz [3] have proposed three probabilistic and timer-based broadcast suppression techniques: weighted p-persistence, slotted 1-persistence and slotted p-persistence Scheme.

In the weighted p-persistence scheme, upon receiving a packet from node i, node j checks the packet ID and rebroadcasts with probability p_{ij} if the node receives the packet for the first time; otherwise, the node discards the packet. Denoting the relative distance between nodes i and j by D_{ij} and the average transmission range by R, the forwarding probability, p_{ij}, is calculated on a per packet basis using

$$p_{ij} = \frac{D_{ij}}{R}. \tag{2.1}$$

In the slotted 1-persistence scheme, upon receiving a packet, a node checks the packet ID and rebroadcasts with probability 1 at the assigned time slot $T_{S_{ij}}$ if the node receives the packet for the first time and has not received any duplicates before its assigned time slot; otherwise, the node discards the packet. Given the relative distance between nodes i and j, D_{ij}, the average transmission range, R, and the predetermined number of slots N_s, $T_{S_{ij}}$ is calculated as

$$T_{S_{ij}} = S_{ij} \times \tau \tag{2.2}$$

where τ is a estimated one-hop delay, which includes the medium access delay and propagation delay, and S_{ij} is the assigned slot number, which can be expressed as

$$S_{ij} = N_s(1 - \lceil \frac{min(D_{ij}, R)}{R} \rceil). \tag{2.3}$$

In the slotted p-persistence scheme, upon receiving a packet, a node checks the packet ID. If the node receives the packet for the first time and has not received any duplicates before its assigned time slot, the node rebroadcasts with the pre-determined probability p at the assigned time slot $T_{S_{ij}}$, as expressed by Eq. 2.2. Otherwise, it discards the packet.

2.2.4. Main Issues

Plenty of broadcast protocols have been proposed to to perform multi hop broadcasts in vehicular ad hoc networks. Tonguz, *et al.* [24] propose a distributed vehicular broadcasting protocol which is designed for safety and transport efficiency applications in VANETs. Liu, *et al.* [25] analyze and evaluate techniques for achieving reliable broadcast in error-prone multi-hop wireless networks, and propose an overall algorithm encompassing a combination of the investigated techniques as an efficient solution for reliable broadcasting in multi-hop wireless networks. Jiang, *et al.* [26] propose an alarm message broadcast routing protocol REAR, which has higher reliability than a location-based algorithm with fewer broadcast packets. Khakbaz, *et al.* [27] present a method that improves the delivery ratio of broadcast messages by overcoming problem of connectivity gaps by sending small messages periodically. They [27] consider fragmentation problem in vehicular ad hoc network and study its effect on broadcasting process. In the protocol [27], when face a gap, every forwarder of a packet sends small messages periodically. These methods are used to understand entrance of a new vehicle that can be selected as the next forwarder of the packet. However, Refs. [25–27] do not consider high-density network environments at all. Besides, the proposals in Ref. [24] and Ref. [25] do not consider topology changes caused by vehicles' movements. Ref. [26] suffers from a higher dissemination latency.

Other researchers have focused on efficient broadcast methods for vehicular ad hoc networks in high-density environments. As mentioned before, Wisitpongphan and Tonguz [3] propose three probabilistic and timer-based broadcast suppression techniques. Blaszczyszyn *et al.* [28] present an opportunistic routing protocol that uses a modified 802.11 MAC protocol using active signaling to select the best relay from all the vehicles that have correctly received the packet. Since all of these proposals do not introduce reliable data delivery schemes, these methods may not be able to work well in sparse network environments or under medium or low traffic load conditions.

Designing a reliable and efficient multi-hop broadcast data transfer protocol for vehicular ad hoc networks is challenging. A good protocol should provide high reliability and efficiency in various traffic conditions. In a high-density network, to provide reliable and efficient data transfers, a novel data relay scheme should be considered. To achieve high reliability, an efficient mechanism should be considered to check the reception status of all receivers without increasing much overhead. Due to highly dynamic feature of vehicular ad hoc networks, the protocol also should be robust to node mobility. Retransmissions should be issued when a data loss occurred.

2.3. Reinforcement Learning

Reinforcement learning is a sub-area of machine learning concerned with how an agent ought to take actions in an environment so as to maximize some notion of long-term reward. Reinforcement learning algorithms attempt to find a policy that maps states of a system to the actions the agent ought to take in those states. Reinforcement learning is the problem faced by an agent that must learn behavior through trial-and-error interactions with a dynamic environment [12, 13].

Reinforcement learning differs from the more widely studied problem of supervised learning in several ways. The most important difference is that there is no presentation of input/output pairs. Instead, after choosing an action the agent is told the immediate reward and the subsequent state, but is not told which action would have been in its best long-term interests. It is necessary for the agent to gather useful experience about the possible system states, actions, transitions and rewards actively to act optimally.

In reinforcement learning, there is a focus on on-line performance, which involves finding a balance between exploration and exploitation. This is another difference between reinforcement learning and supervised learning. In reinforcement learning, the evaluation of the system is often concurrent with learning.

2.3.1. Reinforcement Learning Model

Formally, the reinforcement learning model consists of: (a) a discrete set of environment states, S; (b) a discrete set of agent actions, A; and (c) a set of scalar reinforcement rewards, R. In the standard reinforcement-learning model, an agent is connected to its environment via perception and action, as depicted in Figure 2.1. The agent perceives own state and then chooses an action. The action changes the state of the environment, and the value of this state transition is communicated to the agent through a scalar reinforcement signal (Reward). The agent should choose actions that tend to increase the long-run sum of values of the reinforcement signal. The agent can learn to do this over time by systematic trial and error.

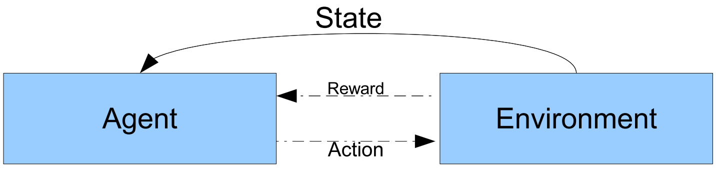


Figure 2.1 – The standard reinforcement learning model.

2.3.2. Q-Learning

There is a wide variety of reinforcement learning algorithms that guide the agent to increase the long-run sum of values of the reinforcement signal. There have been several research which employ actor-critic method to solve reinforcement learning problem [29–31]. In actor-critic method, the learning model consists of two parts: the critic, which maintains the state value estimate, and the actor, which is responsible for choosing the appropriate actions at each state. Actor-critic has the ability to respond to smoothly varying states with smoothly varying actions. Actor-critic systems can form a continuous mapping from state to action and update this policy based on the local reward signal from the critic.

In this thesis, Q-Learning [13] is used. Q-Learning is a recent form of reinforcement learning algorithm that does not need a model of its environment and works by estimating the values of state-action pairs. Q-Learning is generally considered in the case that states and actions are both discrete. This is not a problem because in the reinforcement model which is used in this thesis, both states and actions are discrete. Q-Learning is easy to implement. The main advantage of Q-learning over actor-critic learning is exploration insensitivity [32]. This means that the details of the exploration strategy will not affect the convergence of the learning algorithm. For these reasons, Q-learning has been attracting increasing interest in the machine learning communities.

Q-learning is a reinforcement learning technique that works by learning an action-value function that gives the expected utility of taking a given action in a given state and following a fixed policy thereafter. One of the strengths of Q-learning is that it is able to compare the expected utility of the available actions without requiring a model of the environment [12, 13].

Let $Q(s_t, a_t)$ be the expected discounted reinforcement of taking action a_t in state s_t, the Q-learning rule is as follows.

$$\begin{aligned} Q(s_t, a_t) \quad \leftarrow \quad & (1-\alpha)Q(s_t, a_t) + \\ & \alpha \left\{ R + \gamma \max_a Q(s_{t+1}, a) \right\} \end{aligned} \tag{2.4}$$

In Eq. 2.4, s_{t+1} is the next state of the agent after choosing action a_t in the state s_t. R is the reward after choosing action a_t in the state s_t. α is the learning rate, and γ is the discount factor.

The learning rate determines to what extent the newly acquired information overrides the old information. A factor of 0 makes the agent not learn anything, while a factor of 1 makes the agent consider only the most recent information.

The discount factor determines the importance of future rewards. A factor of 0 makes the agent only consider the current reward, while a factor approaching 1 makes the agent strive for a long-term high reward. If the discount factor meets or exceeds 1, the Q-values will not converge to optimal values.

In the Q-Learning algorithm (Eq. 2.4), each time the agent is given a reward, new values are calculated for the combination of the state s_t and the action a_t. The core of the algorithm is a simple value iteration update. By learning which action is optimal for each state, the agent maximizes long-run sum of values of the reinforcement signal.

When the Q-values are nearly converged to their optimal values, it is appropriate for the agent to use greedy strategy to take the action with the highest Q-value. However, during learning, the agent has to take exploration. It is a difficult problem to make a trade-off between exploitation and exploration.

The simplest exploration strategy is to take the action with the largest Q-value by default, but with probability p, choose an action at random. Some versions of this strategy start with a large value of p to encourage initial exploration, and then decrease p slowly [12].

Chapter 3

Q-Learning AODV

3.1. Introduction

The main distinctive features of vehicular ad hoc networks are high mobility and frequent link changes. Thus, the routing problem of finding reliable paths from a traffic source to a traffic destination through a series of intermediate forwarding nodes is particularly challenging. It is therefore crucial to design an efficient routing protocol for VANETs. Li and Wang [15] have discussed the research challenge of routing in VANETs and surveyed recent routing protocols. Generally, VANET routing protocols in inter-vehicular unicast applications can be classified into two different approaches: position-based and topology-based routing. In position-based routing, the routing decision at each node is based on the destination's position and the position of the forwarding node's neighbors [16, 17]. Maintaining position information needs additional control packets which lead to bandwidth wastage. Therefore, the performance of position-based routing is limited by high control overheads. In contrast, topology-based routing protocols use the information about the links that exist in the network to perform packet forwarding. Although there have been several topology-based routing protocols which are designed for VANET, they all have their limitations.

This chapter discusses the design of general inter vehicular point-to-point routing protocol whose purpose is to react quickly to node mobility and topology changes. This chapter considers making use of the main feature of VANETs without relying on particular instrument or particular situation. Based on original AODV, an enhanced routing protocol called QLAODV is proposed. QLAODV uses a Q-Learning algorithm [13] to estimate network link status information from local communication and change route preemptively using the information so learned. In order to make Q-Learning work efficiently in highly dynamic networks, a route change request/reply mechanism is proposed to check the availability of a newly learned route. Through exhaustive simulation, QLAODV is confirmed to be able to discover better route in dynamically changing networks without knowing the network topology and traffic patterns in advance, therefore can adjust quickly to topology changes.

3.2. Related Work and the Contribution of this Study

3.2.1. Routing Protocols in VANETs

As mentioned above, VANET unicast routing protocols can be classified into two different approaches: position-based and topology-based routing. This chapter proposes a topology-based routing protocol in VANET unicast applications. There are several reasons for employing a topology based approach. Firstly, the topology-based approach does not depend on particular instruments such as GPS positioning devices, which are not affordable for use in every vehicle. Secondly, position-based routing protocols have not been able to produce fully satisfied results. Many performance comparisons between position-based routing protocols and topology-based routing protocols assume that nodes can determine the location of their neighbors and of the destination (e.g., GPSR [18], MURU [23]). In highly dynamic networks, to get precise position information about other nodes, large numbers of signaling packets are needed. This incurs a greater traffic overhead and leads to performance deterioration.

Topology based routing in vehicular ad hoc networks has been studied recently and many protocols have been proposed [4–11]. Some are only designed for highway scenarios or other particular scenarios [4, 7, 11], while others rely on the existence of positioning devices or other auxiliary devices [4–10]. In short, these are not general solutions to VANET point-to-point data transfer problems and therefore cannot be used in various situations of point-to-point applications.

It appears to be more promising to modify an existing routing protocol than to design a new protocol from scratch. However, existing MANET routing protocols are not suitable for VANETs because of frequent link changes and route reconstructions. AODV [21] is known as a good performer in MANET routing protocols. There have been several research attempts which try to extend use of the AODV protocols to vehicular ad hoc networks [9–11]. However, these protocols do not consider the route error handling issue which is a main draw back of AODV because of high overhead in route request flooding.

3.2.2. Route Errors and Link Breakage Processing

AODV

In AODV, when a link break occurs in an active route, the node upstream of that break may try to perform a local repair or send back a route error (RERR) packet to the source node, depending on whether or not local repair is enabled. If local repair is disabled, all the packets that are transmitted between the instant of link failure and the reception of RERR at the source are dropped. If local repair is used, the upstream intermediate node tries to establish a new route segment from itself to the destination. However, the local repair mechanism has some limitations. First, the condition for invoking local repair is that the destination should be no farther than a preset number of hops away from the broken link. Second, the local repair mechanism introduces route non-optimality, and suffers from frequent link breaks and heavy control overheads in networks with high node mobility.

AODV-HPDF

A scheme, which improves the data delivery fraction of AODV (AODV-HPDF) by utilizing local repair at the upstream intermediate node without the hop-distance condition, has been presented by Liang and Wang [33]. In AODV-HPDF, the node that detected the link break will send an RERR packet to the source node. When the source node has received the RREP packet, it will initiate a route-rediscovery process if the data transmission still necessary. Also, the node that detected the link break will be treated as a new source node and a route discovery process will be initiated on that node with a limited time-to-live (TTL) RREQ packet and a limited timeout. Once the new temporary primary route has been built successfully by the new source node within the timeout of the RREQ packet, the buffered data packets will be sent to the destination node through the new route. While providing slightly better performance, AODV-HPDF suffers from a high control overhead in high mobility scenarios.

Neighborhood Route Diffusion

A novel technique called Neighborhood Route Diffusion (NRD) has been proposed by Quwaider *et al.* [34]. The key idea is to perform the local diffusion of selective route information to neighbor nodes, in order to create a temporary envelope of emergency route information to a destination around all nodes that are actively forwarding packets to that specific destination.

The route diffusion information across neighbors is piggybacked over hello packets that are usually used in neighbor discovery by the underlying routing protocol. Using such hello packets, a node disseminates its routing table entries (i.e. destination, next hop etc.) for its active routes. Upon receiving a hello packet, a node updates its neighbor table with the information about its neighbors' routing table entries for their active routes. Thus the efficiency of the packet salvaging mechanism depends on the number of hops that the routing information is diffused up to. However, route lengthens for the salvaged packets increase with an increasing number of hops. NRD uses two-hop route diffusion in through the hello packets.

When a link on a route fails due to mobility, the upstream intermediate node on the failed link can forward packets to one of its neighbors, which has already been provided with route information for the corresponding destination. This can salvage packets without relying on slow and control-heavy end-to-end and local repair mechanisms. However, the advantage of NRD decreases with an increasing number of destinations because in that case the likelihood of finding routing information for a destination will be lower.

3.2.3. Reinforcement Learning Approaches in Routing Protocols

In recent years, reinforcement learning [12] has been attracting increasing interest in the machine learning and artificial intelligence communities. Boyan and Littman [35] describe the Q-routing algorithm for packet routing, in which a reinforcement learning module is embedded into each node of a switching network. Since Q-routing is designed for wired networks, it is not suitable to VANETs. Chang *et al.* [36] use reinforcement learning methods to control both packet routing decisions and node mobility to improve the connectivity of a network. However, it is impossible to control node movement in vehicular ad hoc networks. Dowling *et al.* [37] have proposed collaborative reinforcement learning (CRL), which enables groups of reinforcement learning agents to solve system optimization problems online in dynamic, decentralized networks. They evaluate an implementation of CRL in a routing protocol for MANETs, which is called SAMPLE. However, CRL has the problem of convergence to suboptimal solutions. What is more important is that SAMPLE does not consider link breakage due to node mobility which is the main feature of VANETs. Although SAMPLE performs well in high packet error rate scenarios, it has worse packet delivery ratios than AODV in cases where the packet loss due to radio interference is low. In SAMPLE, routing information is advertised in the network by attaching it to data packets. As a result, it increases the data packet size and so introduces a large overhead in high data rate applications.

Usaha and Barria [30] have proposed a path discovery scheme based on reinforcement learning. The scheme supports QoS routing in mobile ad hoc networks (MANETs) in the presence of imprecise information. The scheme increases the probability of success in finding feasible paths and reduces average path cost of a previously proposed ticket based probing (TBP) path discovery scheme. The scheme employs a reinforcement learning method called the on-policy first visit Monte Carlo (ONMC) method. The experimental results show that reinforcement learning techniques can play an important role in controlling search messages overhead in environments in which the outcome of a decision is only partially observable. Usaha and Barria [31] also have applied a reinforcement learning (RL) method, namely, the actor-critic method with belief state concept (ACBS) to support QoS routing at the network level in a MANET. The aim of the scheme is to maximize the probability of success in finding feasible paths while maintaining communication overhead under control in presence of information uncertainty. The simulation results show that the TBP schemes based on the ACBS method can achieve good ticket-issuing policies, in terms of the accumulated reward per episode, when compared to the original heuristic TBP scheme and the flooding-based TBP scheme. This thesis considers using reinforcement learning to infer network state information in a highly mobile environment.

3.2.4. The Contribution of this Study

Based on the original AODV, QLAODV, an enhanced routing protocol, is proposed. QLAODV uses a Q-Learning algorithm [13] to estimate whole network link status information from local communication and to change routes preemptively using the information so learned. In order to make Q-Learning work efficiently in highly dynamic networks, a route change request/reply mechanism is proposed to check the availability of a newly learned route. Through exhaustive simulation, it is confirmed that QLAODV is able to discover better routes in a dynamically changing network without knowing the network topology and traffic patterns in advance, and therefore can adjust quickly to topology changes.

3.3. QLAODV Protocol Design

This section presents a detailed description of the proposed protocol QLAODV (Q-Learning AODV). QLAODV is an enhanced topology-based routing protocol based on AODV.

3.3.1. Design Principles

AODV is known as a good performer in MANET routing protocols. However, AODV faces many problems when is used in vehicular ad hoc networks. AODV only considers hop count in route selection and therefore the selected route may fail shortly after the construction. AODV takes action after the link failure, and the route discovery approach of AODV is very costly in terms of overhead and delay. Therefore, QLAODV is proposed here to extend original AODV to make it suitable for vehicular ad hoc networks.

The main advantages of QLAODV over AODV are as follows. First, QLAODV employs a dynamic route change mechanism to switch routes before they fail. Next, QLAODV uses a reinforcement learning algorithm to infer network link state information in a distributed manner. Last, QLAODV considers the hop count, stability and bandwidth efficiency in route selection.

In QLAODV, when a source node needs to communicate with a destination node, it checks its routing table for a route. If no route exists, QLAODV uses the normal route discovery approach of AODV to create a route to the destination. The normal route discovery of AODV is used because reactively finding a route can be novel in terms of control overhead. To avoid the frequent route discovery of AODV protocol in highly dynamic networks, a dynamic route change mechanism is used to switch routes preemptively and therefore reduce the number of route request broadcasts. AODV-like reactive routing protocols take action after the link failure, and the route discovery approach of AODV is very costly in terms of overhead and delay. When a link fails, all packets in the intermediate nodes will be dropped if AODV fails to the local repair. The dynamic route change approach is more efficient than taking action after occurrence of link failures. Using the dynamic route change mechanism, QLAODV not only can reduce the number of route errors but also can optimize the routes concurrently to improve the network throughput.

In order to discover better routes, Q-Learning, a recent form of reinforcement learning algorithm, is used to infer network link state information in a distributed manner. In vehicular ad hoc networks, since car movement is restricted by the road, stable neighbor nodes are likely to exist. Also, whether a route is good or bad is determined by all nodes on the road. Here, Q-Learning is used to learn the link information. Every network node acts as a learning agent and gathers network link state information while interacting with its local environment. A mechanism is also proposed to check the availability of a newly learned route. The mechanism supplements the Q-Learning algorithm in order to work efficiently in highly dynamic networks.

In vehicular ad hoc networks, network topology changes frequently. It is therefore difficult to estimate link quality. Estimating link quality is a major task to develop an efficient routing protocol. Obviously, only considering a hop count as a routing metric is far from the optimal solution. In order to meet the requirements of inter-vehicular applications, QLAODV considers the hop count, stability and bandwidth efficiency in route selection.

In QLAODV, when a source node needs to communicate to a destination node, it checks its routing table for a route. If none exists, QLAODV uses original route discovery approach of AODV to create a route to the destination. To avoid frequent route discovery of AODV in highly dynamic networks, the proposed protocol uses dynamic route change mechanism to switch route preemptively and therefore reduce the number of route request broadcasting. This is efficient than conventional route breakage handling mechanisms which take action after link failure.

3.3.2. Modeling VANET Routing Problem as a Reinforcement Learning Task

It is difficult to use simple rule to determine the packet forwarding policy in highly dynamic networks because of frequent link changes. Wang *et al.* [38] have presented a supervised learning framework which can be used to produce useful information automatically and to help make informed decisions in sensor networks. Caleffi and Paura [39] have proposed to improve the estimation of the link quality resorting to a bio-inspired estimator based on the neural network paradigm. However, such proposals do not consider increment of link failures by node movement.

In vehicular ad hoc networks, when a source node needs to communicate with a destination node in multiple hops away, the source node first has to select a next hop node which performs forwarding. The next hop node then selects its next hop node. The above steps continue until the destination node is reached. Upon selecting a next hop node, a node does not know whether the packets can be delivered to the destination or not. This is because the result also depends on actions of other forwarding nodes. Therefore, the mathematical model of the problem may be too expensive to derive.

AODV uses a broadcast route discovery method to establish a route to the destination. However, the route can easily become a sub-optimal route due to topology changes. The frequent topology changes make it necessary to change forwarding policy concurrently.

Fortunately, the use of reinforcement learning [12] can handle these problems. Reinforcement learning is the problem faced by an agent who must learn behavior through trial-and-error interactions with a dynamic environment. Formally, the reinforcement learning model consists of: (a) a discrete set of environment states, S; (b) a discrete set of agent actions, A; and (c) a set of scalar reinforcement rewards, R.

Reinforcement learning algorithms attempt to find a policy that maps states of a system to the actions the agent ought to take in those states. In reinforcement learning, correct input/output pairs are never presented and the evaluation of the system is often concurrent with learning. After choosing an action the agent is told the immediate reward and the subsequent state, but is not told which action is the best. Therefore, the agent has to gather useful experience about the possible states, actions, transitions and rewards to act optimally. The agent also has to be able to learn from delayed reward because it may go through a series of states with insignificant reward, and then finally arrive at a state with high reward. In the case of delivering a packet in vehicular ad hoc networks, upon choosing a next hop, a node does not know whether the choice is good or not until the packet reaches the destination node.

In this work, the network routing problem in vehicular ad hoc networks is modeled as follows. The entire vehicular ad hoc network is the environment. Its components include the mobile nodes, the links between the nodes and packets. Each packet $P(o, d)$, indexed by its originator node o and destination node d is an agent. Each node in the network is considered a state of the agent. The set of all nodes in the network is the state space. A node selects the next hop that the node should forward a packet to (or delivers the packet to the upper layer if the current node is the destination node). Hence the possible set of actions allowed at the node is nothing but the set of neighbors. The state transitions are equivalent to a packet being delivered from one node to its neighbor.

Since it is impossible to have a global view on network state transitions, reinforcement-learning tasks are distributed to each node. Nodes exchange their knowledge through hello messages. Each node only needs to select its best next hop. Upon selecting the next hop, the node should immediately receive back the next hop node's estimate of the best route. However, considering the control overhead and implementation complexity, the proposed protocol uses periodic hello messages to help nodes to revise their estimates. In QLAODV, the agent might receive a negative reward if the route change attempt fails (this will be explained in 3.3.8).

3.3.3. Distributed Q-Learning in QLAODV

For VANETs, as a packet is routed, there is no way to determine the reward until the packet reaches the destination node. Hence using the model-based approach is not possible. Therefore, Q-Learning [13], which is able to compare the expected utility of the available actions without requiring a model of the environment, is used.

Q-Learning is a recent form of reinforcement learning algorithm that does not need a model of its environment and works by estimating the values of state-action pairs. The Q-value $Q(s,a)$ $(s \in S, a \in A)$ in Q-learning is an estimate of the value of future rewards if the agent takes a particular action a when in a particular state s. By exploring the environment, the agents build a table of Q-values for each environment state and each possible action. Except when making an exploratory action, the agents select the action with the highest Q-value. The learning rate and the discount factor are important parameters of the Q-learning algorithm. The learning rate parameter limits how quickly learning can occur. It governs how quickly the Q-values can change with each state/action change. The discount factor controls the value placed on future rewards. If the value is low, immediate rewards are optimized, while higher values of the discount factor cause the learning algorithm to count future rewards more strongly.

The Q-Learning algorithm that is used in QLAODV is defined as follows. Every node maintains a Q-Table which consists of Q-values $Q(d,x)$ whose values range from 0 to 1, where d is the destination node and x is the next hop to the destination. QLAODV uses a dynamic Q-Table, such that the size of the Q-Table of a node is determined by the number of destination nodes and neighbor nodes. The Q-Table and learning tasks are distributed among the different nodes (states). In QLAODV, exploration can be achieved by updating the Q-values when the agent receives a hello message. Therefore, when choosing a next hop, QLAODV lets the agent act greedily, taking the action with the highest Q-value in each situation. If a packet is able reach its destination node through the action x, the reward R will be 1, and otherwise R will be 0. More specifically, when a node receives a hello from the destination node, the reward R will 1 and otherwise R

will be 0.

The discount factor is an important parameter of the Q-learning algorithm. QLAODV uses a variable discount factor, which is determined by the hop count, link stability and available bandwidth of nodes on the route. The information will be discounted when it passes through the node and will also be discounted according to link stability and bandwidth usage. In this way, QLAODV ensures that the selected route is the shorter, more stable route with enough bandwidth. The local used bandwidth BW is estimated as

$$BW(\text{bps}) = \frac{n \times S_B \times 8}{T}, \tag{3.1}$$

as defined by Renesse *et al.* [40]. In Eq. 3.1, n is the number of packet sent and received by a node. S_B is the size of a packet in bytes while T is the time period. In QLAODV, T is set to 0.5 s. It is assumed that all nodes have the same maximum bandwidth. The Available Bandwidth can be calculated by subtracting the local used bandwidth from the Maximum Bandwidth using the following equation.

$$\text{Available Bandwidth} = \text{Maximum Bandwidth} - \text{Used Bandwidth} \tag{3.2}$$

Vehicular ad hoc networks are distributed networks in which every node works independent to each other. Therefore, the Q-Learning algorithm used in QLAODV has its own characteristic compared to conventional Q-Learning algorithm. In QLAODV, each node acts as an agent and executes the Q-Learning algorithm independently. There are two types of actions as shown in Figure 3.1.

First, choosing a next hop node is a type of action, exploitation action, which the agent wants to optimize its long-term reward. For this type of action, the action states are the destination of the packet should be delivered. The actions are the neighbors of the agent.

Second, reception of a hello message is a type of action, exploration action, which is used to explore a new path. For this type of action, both the action states and the actions are the neighbors of the current agent. Each agent uses exploration actions to

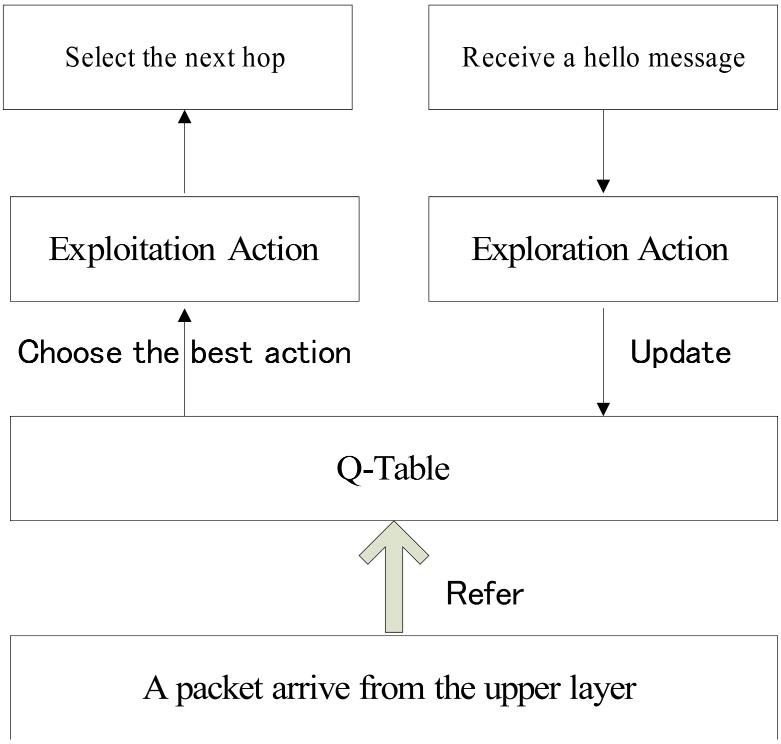


Figure 3.1 – Exploitation action and exploration action in QLAODV.

infer link state information.

When there is a need to deliver a data to a destination node, the agent uses the exploitation action based on the information which have been collected by exploration actions. Because each node uses hello messages to periodically update the link state information, when selecting a next hop node, the agent can easily choose the node which has maximal Q-value. In this way, the agent can choose the best node as the next hop.

3.3.4. Maintenance of Q-Table

In QLAODV, every node uses hello messages to exchange link information with its neighbors. This link information includes a part of the Q-Table (MaxQValues), the mobility factor of the node and the bandwidth factor of the node. In this chapter, $Q_s(d, x)$ is defined as the Q-Metric of node s bound to destination node d through neighbor x.

In QLAODV, when the hello timer expires, every node first calculates an array (MaxQValues) which contains maximum Q-Metrics for each destination node in the network.

Every node x then calculates a mobility factor MF_x as

$$MF_x = \begin{cases} \sqrt{\frac{|N_x \cap N_x^p|}{|N_x \cup N_x^p|}}, & \text{if } N_x \cup N_x^p \neq \phi \\ 0, & \text{otherwise} \end{cases} \tag{3.3}$$

where N_x is the current neighbor set of node x, and N_x^p denotes the neighbor set of node x at the time that the previous hello was sent. $|A|$ is used to denote the number of elements in set A. Every node needs to maintain a N_x^p. When the hello timer expires a node uses this value and the current neighbor set to calculate MF_x. The MF_x will reflect a higher value for a relatively stable node. In case of a static network, MF_x will be 1 for every node.

Every node x also needs to calculate a bandwidth factor BF_x as

$$BF_x = \frac{\text{Available Bandwidth } of\ x}{\text{Maximum Bandwidth } of\ x.} \tag{3.4}$$

Every node then attaches the MaxQValues, MF_x and BF_x to the hello message.

It is assumed that agents know nothing about the rest of the network at the start of communication. This means that all elements of Q-Table (Q-values) are initialized to 0. $Q_s(d, x)$ is the value that node s estimates as the practicability of delivery of a packet bound for node d by way of neighbor node x. This estimation represents the whole network performance because it considers multiple metrics of hop count, stability and available bandwidth. Upon receiving a hello packet from the neighbor x, a node first calculates a discount factor γ_x as

$$\gamma_x = \gamma \times \sqrt{MF_x \times BF_x} \tag{3.5}$$

where γ is a predefined value. γ_x should satisfy $0 < \gamma_x < 1$ to consider the hop count. Since the mobility factor (MF_x) and bandwidth factor (BF_x) are considered in

γ_x's calculation, γ is set to the relatively large value of 0.9.

The node s then revises its estimate as

$$\begin{aligned} Q_s(d,x) \;&\leftarrow\; (1-\alpha)Q_s(d,x) + \\ &\alpha\left\{R + \gamma_x \max_{y\in N_x} Q_x(d,y)\right\} \end{aligned} \tag{3.6}$$

where N_x denotes the set of neighbors of node x and R denotes the reward. R is defined as

$$R = \begin{cases} 1, & \text{if } s \in N_d \\ 0, & \text{otherwise} \end{cases} \tag{3.7}$$

where N_d is the set of neighbors of d. This means that if a node receives a hello from the destination, the reward will be 1 and otherwise 0. In Eq. 3.6, $\max_{y\in N_x} Q_x(d,y)$, actually an element of MaxQValues, is calculated by the hello sender node and sent with its hello message. In this way, a hello sender node does not need to send the whole Q-Table and hence can reduce the hello overhead.

The learning rate parameter α limits how quickly learning can occur. In the proposed protocol, it governs how quickly the Q-values can change with a network topology change. If the learning rate is too low, the learning will not adapt quickly to network dynamics. If the rate is too high, then the algorithm cannot reflect the network movements accurately because agents can receive immediate misleading rewards. In QLAODV, the learning rate α is set to 0.8. This is because it is found that 0.8 is the most suitable value for QLAODV, through a lot of experiments and analysis.

In the conventional Q-Learning, discount factor (γ_x) is constant. However, in this case of VANET routing problem, the action is to choose a next hop node. Actually, the action is to choose a link. However, different links have different level of link status. So, the protocol has to discount reinforcements indicated by these links according to their link status. This is why a variable discount factor is used (Eq. 3.5). From Eq. 3.5 and Eq. 3.6, it is easy to know that mobility factor and bandwidth factor are considered in the Q-Value update. Since the reward is discounted when it pass through the nodes, as far as

γ_x is smaller than 1, hop count is implicitly considered in the Q-Value update. It means that, if a constant discount factor is used, only hop count will be considered in route selection. Since γ_x is calculated (Eq. 3.5) before updating the Q-Table, the mobility factor, bandwidth factor and hop count can be considered in route selection. In Eq. 3.3 and Eq. 3.5, square root ($\sqrt{\cdot}$) is used to smooth the value because the experimental results have shown this leads to better outcomes.

Algorithm 1 Q-Learning Algorithm in QLAODV

```
At each node $s$
Initialize Q-table, $Q_s(d,x) = 0, d \in D, x \in N_s$, where $N_s$ is a set of neighbors of $s$ and D is a set of destinations.
for each event do
    if hello timer expires then
        For each d, get maximum $Q_s(d,x), x \in N_s$, namely MaxQValue, and attach them to hello message.
        Send hello message.
    end if
    if receive hello from the neighbor $x$ then
        Get MaxQvalue form the hello message.
        Compute $\gamma_x$ as Eq. 3.5.
        If the hello sender is the destination node, set $R$ to 1, otherwise set $R$ to 0.
        For destination $d$, update Q-Table as Eq. 3.6.
    end if
end for
```

Q-Learning algorithm used in QLAODV is given in Algorithm 1. The nodes exchange link state information and update their Q-Table using hello messages. Each node attaches its MaxQValues, MF_x and BF_x to the hello message before sending it. The node that receives the hello message extracts the corresponding values from the hello packet and executes the Q-Learning algorithm to update its Q-Table. The MaxQValues that a node obtains from a received packet is the Q-Metrics of the neighbor who sent it and it indicates the neighbor's knowledge about the network.

3.3.5. Exploitation, Exploration and Convergence

When forwarding data, QLAODV selects for the next hop the node that has maximum Q-value. This is called exploitation. Nevertheless, to make the exploitation lead to the global optimum, an exploration is required to check whether one neighbor is better than another. In QLAODV, each node updates its Q-values upon reception of hello messages from its neighbors. Since hello messages are exchanged periodically, every node is aware of which neighbor is becoming the preferred choice.

Convergence is an important issue in evaluating an algorithm's validity. There is no guarantee that reinforcement learning always leads to convergence. However, Watkins and Dayan [41] prove that Q-Learning converges to the optimum action-values with probability of 1 so long as all actions are repeatedly sampled in all states and action-values are represented discretely. Fortunately, the proposed Q-Learning algorithm satisfies all the these conditions for convergence. In the proposed algorithm, a node is equivalent to a state and every node uses hello messages to sample all its neighbors. Obviously, the action-values (Q-values) are represented discretely in QLAODV. Therefore, if the hello interval is small enough and the loss of hello messages can be ignored, the proposed algorithm converges to optimum action-values.

3.3.6. Routing Metrics in QLAODV

Many Distance vector routing protocols such as AODV try to find the shortest route. However, the shortest route is not always the best route. QLAODV uses the Q-Learning algorithm to evaluate a path according to its hop count, stability and available bandwidth. QLAODV gives a shorter path a higher value because the discount factor γ_x is smaller than 1. Since QLAODV considers the mobility factor, MF_x, in the calculation of discount factor, it can choose the most stable route. Stability is also reflected in the Q-Metric through the value iteration. As shown in Eq. 3.6, for the first calculation, $Q_s(d,x)$ is zero and this value is discounted by $1-\alpha$ for every iteration. This means that $Q_s(d,x)$ is expected to become larger with each iteration if other elements do not change. In general, if a link's duration time is long, it is more likely to still be durable in the future which is the case of a vehicle traveling in the same direction. QLAODV can also balance the traffic between nodes because it discounts the reward according to the available bandwidth. In short, QLAODV can achieve short, stable and high-bandwidth routes.

3.3.7. Difference Between the Proposed Q-Learning Algorithm and Conventional Q-Learning Algorithm

Vehicular ad hoc networks are distributed networks in which every node works independent to each other. VANET routing problem is different to conventional reinforcement learning problem. Therefore, the conventional Q-Learning is modified to make it suitable for vehicular ad hoc networks. The proposed Q-Learning algorithm is different to conventional Q-Learning algorithm in the following ways.

First, the proposed Q-Learning algorithm uses two types of actions. They are exploration action and exploitation action. Exploration actions are performed periodically to support exploitation action. If the exploration interval is small enough, the Q-values will be converged to their optimal values. Therefore, when choosing a next hop, the agent takes the action with the largest Q-value. However, the selection of exploration intervals is important. The smaller the exploration interval is, the higher control overhead will be. So, a proper exploration interval should be chosen according to the application requirements. In QLAODV, the exploration interval (hello interval) is 1s. This value is used because experimental results have shown that the value leads to better outcomes. (This is also the default setting of original AODV)

Second, the proposed Q-Learning algorithm uses a variable discount factor. Upon receiving a hello message, an agent calculates the discount factor according to the sender's bandwidth status, mobility status. This discount factor is then used to update corresponding Q-values. As a result, multiple metrics are considered in the Q-values, including link stability, bandwidth efficiency and hop counts.

3.3.8. Dynamic Route Change Mechanism to Avoid Link Breakage

Table 3.1 – Composition of RCNG-REQ packet and RCNG-REP packet.

RCNG-REQ	RCNG-REP
Destination IP Address	Destination IP Address
Destination Seq Number	Destination Seq Number
Originator IP Address	Originator IP Address
Originator Seq Number	Life Time
Next Hop	Next Hop

It is possible that the route learned from local communication is already out-of-date because of link breakage in a fast moving network. In order to check whether the route is still available or not, QLAODV uses unicast route change request and route change reply messages. When a route is being used for delivering packets, if a sender node (source node or other forwarder node) finds an alternative path that has a larger Q-Metric than the current route, the sender node will send a unicast packet RCNG-REQ (route change request) to the destination through the neighbor which indicates a better route to the destination. The intermediate nodes will forward the packet according to their Q-Table. Upon receiving the RCNG-REQ packet, the destination node replies with RCNG-REP (route change reply). This means the new path is available if the RCNG-REP reaches the sender node successfully. The sender node then updates its routing table to use the new route. Every forwarder node also updates the corresponding route upon receiving a RCNG-REP. The compositions of the RCNG-REQ packet and the RCNG-REP packet are shown in Table 3.1. Fig. 3.2 depicts the dynamic route change approach of QLAODV.

As shown in Fig. 3.2, node s uses next hop 1 to deliver data packets bound for destination node d. It should be noted that node s could be the source node or another forwarder node. Each node offers connectivity information by broadcasting hello mes-

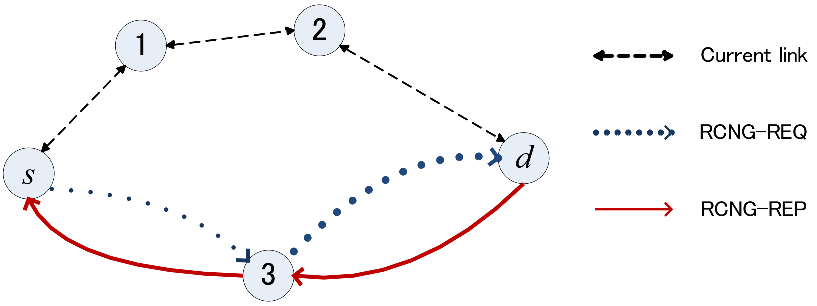


Figure 3.2 – Dynamic route change mechanism in QLAODV.

sages, and utilizes the Q-Learning algorithm to update its own Q-Table when it receives hello messages from its neighbors.

Dynamic route change can be achieved by the following steps.

1. Node s updates its Q-Table upon receiving a hello packet from node 3.
2. Node s then checks its Q-Table and finds that the new path by way of node 3 to destination node d is better than the current route.
3. In order to check the availability of the new path, node s will set a route change timer and initiate a unicast packet RCNG-REQ to destination node d and send it by using neighbor node 3 as the next hop.
4. Upon receiving the RCNG-REQ packet, node 3 knows the packet is for node d.
5. Node 3 sets a route change timer.
6. Node 3 selects the best next hop node d according to its Q-Table and forward the RCNG-REQ to d.
7. Node d receives the RCNG-REQ packet.
8. Node d initiates a RCNG-REP to node s and sends it by way of node 3.
9. Node 3 forwards the RCNG-REP packet to node s.
10. Node s updates its route table upon receiving the RCNG-REP packet.

In this way, without the original route request being broadcast, node s can use the new route to deliver data and thus can reduce the routing overhead compared with other approaches and consequently improve the data delivery ratio. Conversely, if a node (including RCNG-REQ sender node and other forwarder nodes) does not receive the route change reply before the route change timer expires, the corresponding Q-value will be reset to 0, resulting in a route change failure.

3.4. Simulation Results

3.4.1. Simulation Environment

Network Simulator 2 (ns-2) [42] was used to conduct simulations using different mobility models. First, the random waypoint model, currently the most widely used mobility model, was used. Random waypoint model is a common way to evaluate the overall performance of protocols although it does not consider vehicles' specific motion patterns.

Next, the Freeway mobility model and the Manhattan mobility model [43] were used to evaluate the protocols' performance. The Freeway mobility model emulates the motion behavior of mobile nodes on a freeway, while the Manhattan mobility model emulates the movement pattern of mobile nodes on smaller side streets. In the freeway model simulation, a freeway which has two lanes in each direction is used. All lanes of the freeway are 2000 m in length. 80 vehicles are randomly distributed on this freeway and the arrival velocity of each vehicle is 5 m/s. For each of the Manhattan model scenarios, a map of 80 vehicles randomly distributed in a street area of 1000 m × 1000 m is used. The map consists of 3 horizontal streets and 3 vertical streets and every street has one lane in each direction. The distance between intersections is 300 m. The arrival velocity is set to 5 m/s.

Last, a TIGER line map file [44] and real street map based model [45] are used to generate realistic vehicle movement scenarios. A 2500 m × 2500 m square area in

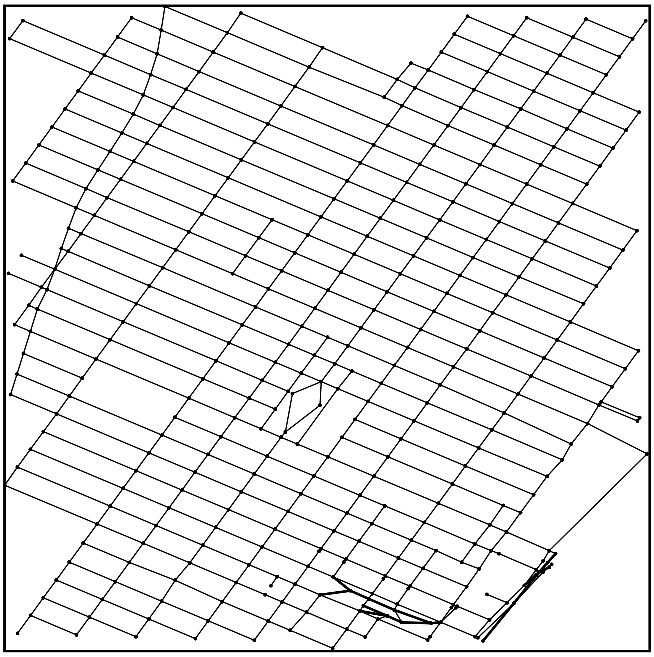


Figure 3.3 – Street scenario corresponding to a 2500 m × 2500 m square area in midtown Manhattan.

Midtown Manhattan in New York City (as shown in Figure 3.3) is used. This area is chosen because it is representative of a large number of city areas in the US. In the freeway mobility model and the Manhattan mobility model, the transmission range is 250 m. Nevertheless, in the real street map based mobility model, a 500 m transmission range is used because it is suggested by Saha and Johnson [45].

The QLAODV protocol was compared with AODV and two other extensions of the AODV protocol (AODV-HPDF and NRD). In all simulations, omnidirectional antennas, IEEE 802.11b standard transmission at 11Mbps and standard 802.11 MAC are assumed. Link layer notification as provided by 802.11 is used to determine connectivity. The standard CMUPri model [42] for a queue of buffer size 50 was used. Simulations used CBR traffic with a packet size of 512 bytes. These simulation parameters were carefully chosen based on the characteristic of VANET applications. Each simulation lasted 500 s and each case was repeated 50 times to give high confidence in the results. All data presented in this paper are the average value of the 50 simulations.

3.4.2. Effect of Mobility

In the Random Waypoint Model, protocol's performances are evaluated over moving velocity. There are 80 nodes randomly deployed in 1000 m × 1000 m area and node movement follows the random waypoint model with zero pause time. The velocity is uniformly distributed in the range between $\upsilon - 1$ and $\upsilon + 1$ m/s, where υ denotes average velocity.

In the Freeway Model and the Manhattan Model, each vehicle accelerates at a rate of ten percent of the maximum allowable velocity per second, if there are no other vehicles ahead of it, until the maximum allowable velocity is reached. In the Freeway Model and the Manhattan Model, simulations are conducted with various values of maximum allowable velocity.

In the real street map based model, the speed limit for each road was based on the type of road as indicated in the TIGER/Line files [44]. In addition, simulation results with various node densities are presented.

For all models, 30 pairs of random connections with a 32 kbps transmission rate were generated. Figures 3.4, 3.5, 3.6 and 3.7 show comparisons of the achieved packet delivery ratio for AODV, AODV-HPDF, NRD and QLAODV for the different mobility models. The packet delivery ratio was calculated as the number of data packets received by the application layer of the destination nodes divided by the number of data packets generated by the source nodes.

Packet delivery ratio

It can be clearly seen that QLAODV outperforms the other three protocols, irrespective of the mobility model. In the Random Waypoint Model and the Freeway Model, as the node velocity increases, the advantage of QLAODV becomes more apparent. This can be explained by the fact that in dynamically changing networks, QLAODV can change to better routes adaptively as the network topology change, whereas other protocols wait until existing routes break before constructing new routes.

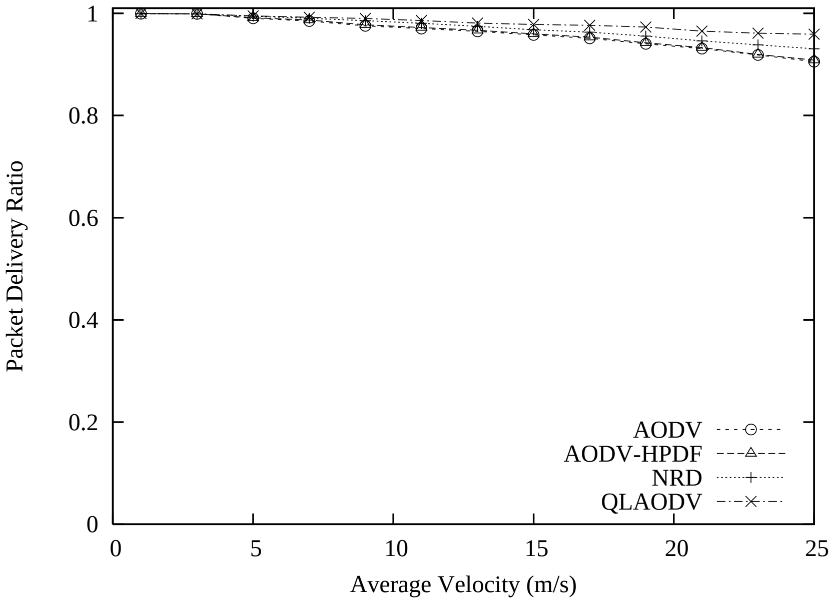


Figure 3.4 – Achieved packet delivery ratio for varying velocities in Random Waypoint model.

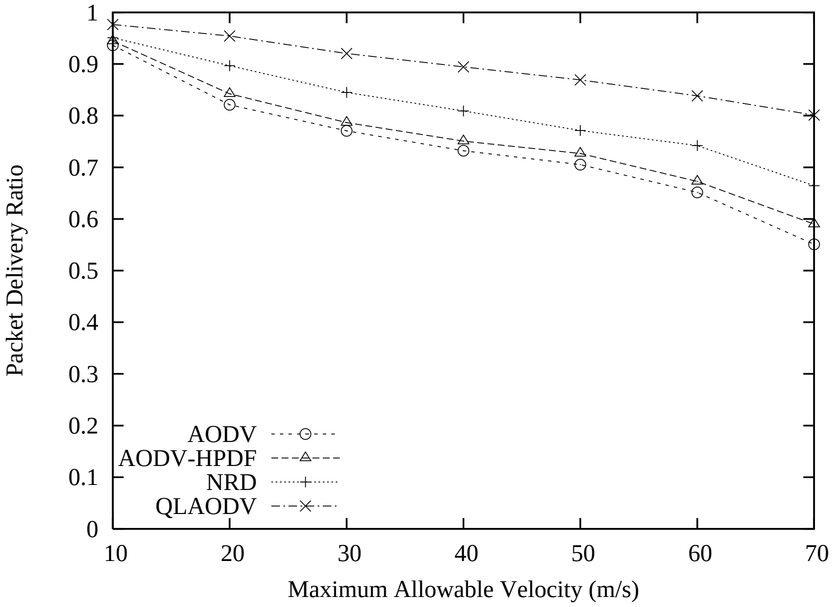


Figure 3.5 – Achieved packet delivery ratio for varying velocities in Freeway model.

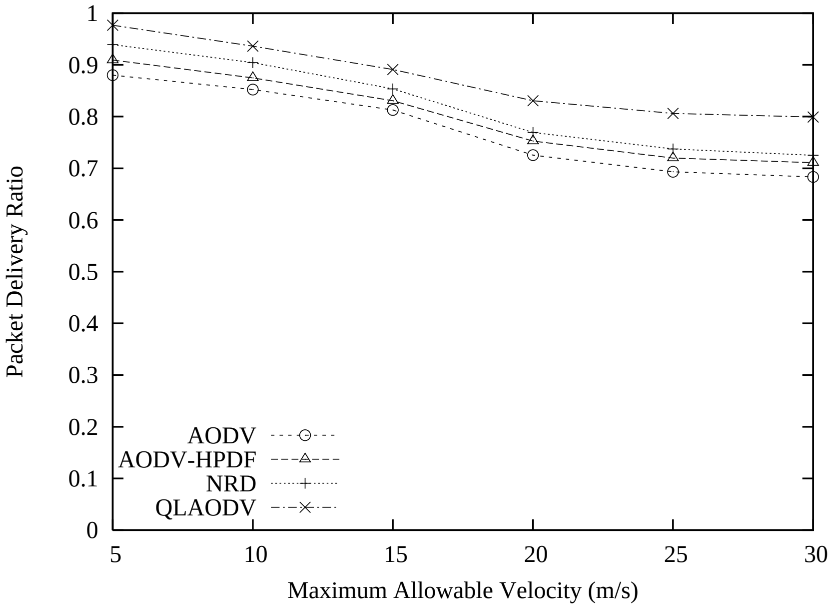


Figure 3.6 – Achieved packet delivery ratio for varying velocities in Manhattan model.

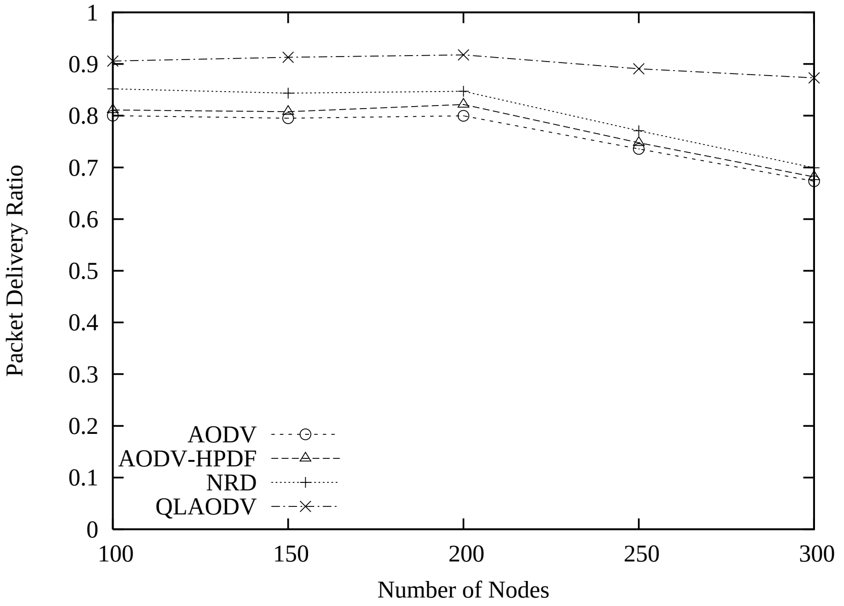


Figure 3.7 – Achieved packet delivery ratio for varying velocities in real street map based mobility model.

In the Freeway Model, since QLAODV takes the stability of routes into account, it results in a high probability of using vehicles moving in the same direction as the source node to forward packets. However, in AODV, a source node initiates a route request to broadcast the packet and the destination node simply replies with the route which has the minimum hop count. Therefore, a node which is moving in the opposite direction to the source node may be chosen as an intermediate node, and in this case the corresponding route is very vulnerable. Consequently, many data packets may be dropped when link failure occurs. NRD may use oncoming vehicles or vehicles moving in other direction to salvage data packets. This results in a significantly higher frequency of route failures. AODV-HPDF also suffers from the same problem, because AODV-HPDF's local repair always leads to non-optimal paths. In QLAODV, vehicles moving in the same direction as the source node always retain a higher Q-Metric than those moving in other directions. Thus, vehicles can use other vehicles moving in the same direction to forward data. Therefore, QLAODV is more efficient than the other three protocols.

In the Manhattan model, QLAODV clearly outperforms the other three protocols in terms of packet delivery ratio even when the vehicles' moving velocity is very low. This can be explained by the following facts. Even when vehicles' velocity is not very high, the relative speed between vehicles may still be high and this results in frequent topology changes. While the other three protocols cannot adapt quickly to network topology changes, QLAODV benefits from its preemptive route change mechanism. Also, since the Q-Learning algorithm takes the hop count into consideration, QLAODV always constructs a shorter route than AODV (as discussed later and shown in Figures 3.24, 3.25, 3.26 and 3.27). This is another factor contributing to QLAODV's advantage. AODV-HPDF and NRD show a decrease in advantage over AODV as a result of increasing mobility. This is because AODV-HPDF and NRD result in longer routes which are easily broken in Manhattan scenarios.

The results for the real street map based mobility model are similar to those for the other mobility models. In AODV-HPDF, upon the occurrence of a link failure, both the upstream node and the source node initiate route discovery. This will become very costly in terms of overheads in high-density networks. This is why AODV-HPDF's advantage decreases with increasing node density. As the number of nodes increases, the flows become more distributed and hence the effectiveness of NRD diminishes. With AODV, when the node density is high, many link failures occur and route request broadcasts consume more bandwidth, leading to a drop in performance. It can be observed that the advantage of QLAODV increases as the number of nodes increases. The reason is that the QLAODV protocol is favored by the increasing number of available paths and it becomes easier to change to a new route before the current one is disconnected.

In AODV, when a link fails, the upstream intermediate node tries to perform a local repair. However, the condition for success of a local repair is that the destination should be no farther than a preset number of hops away from the broken link. If the local repair fails, the buffered packets are dropped. AODV-HPDF utilizes local repair without the hop-distance condition to improve the packet delivery fraction of AODV. However, while offering faster repairs than the route error based end-to-end mechanisms, local repair introduces route non-optimality, and the new route fails shortly after the repair. In NRD, when a link on a route fails due to mobility of nodes, the intermediate node on the failed link can forward packets to one of its neighbors which has already had the route information for the corresponding destination diffused to it. However, NRD only works if nodes around the point of failure have routing information to the same destination. In the case where flows are distributed, NRD cannot provide good performance. Moreover, NRD always results in non-optimal paths which diminishes the advantage of NRD.

Normalized control overhead

A comparison of the normalized control overhead is shown in Figures 3.8, 3.9, 3.10 and 3.11. The normalized control overhead is defined to be the number of control packets generated divided by the number of data packets that arrive at receivers. In Figure 3.8, Figures 3.9 and 3.10, as the node velocity increases, the control overhead of AODV increases because of route errors and route request broadcasts. It can be observed that the normalized control overhead of AODV-HPDF is higher than that of AODV especially

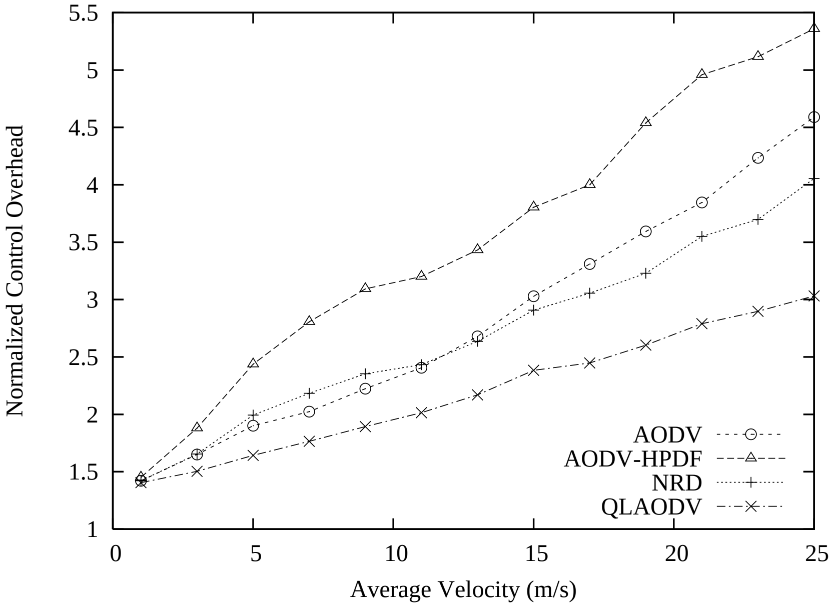


Figure 3.8 – Normalized control overhead for varying velocities in Random Waypoint model.

at high node velocities. In AODV-HPDF, when a link fails, both the source node and the upstream node initiate route discovery. Clearly, this introduces a high overhead. Although the mechanism of NRD for salvaging packets during mobility-initiated link breaks can avoid redundant route requests, the mechanism leads to non-optimal routes and therefore cannot provide a significant improvement. Fortunately, the efficient route change mechanism in QLAODV reduces the number of route errors and therefore results in a low control overhead. As shown in Figure 3.11, the normalized routing overheads of AODV, AODV-HPDF and NRD increase drastically with increasing node density. This is because the protocols use broadcast route discovery when a link failure occurs, which introduces a high overhead in a high-density network. Since QLAODV uses a unicast route change request/reply cycle to discover new routes, the result shows a lower overhead.

Number of route errors

Figures 3.12, 3.13, 3.14 and 3.15 show the number of route errors resulting from the four protocols. It is obvious that a dynamic route change mechanism results in a re-

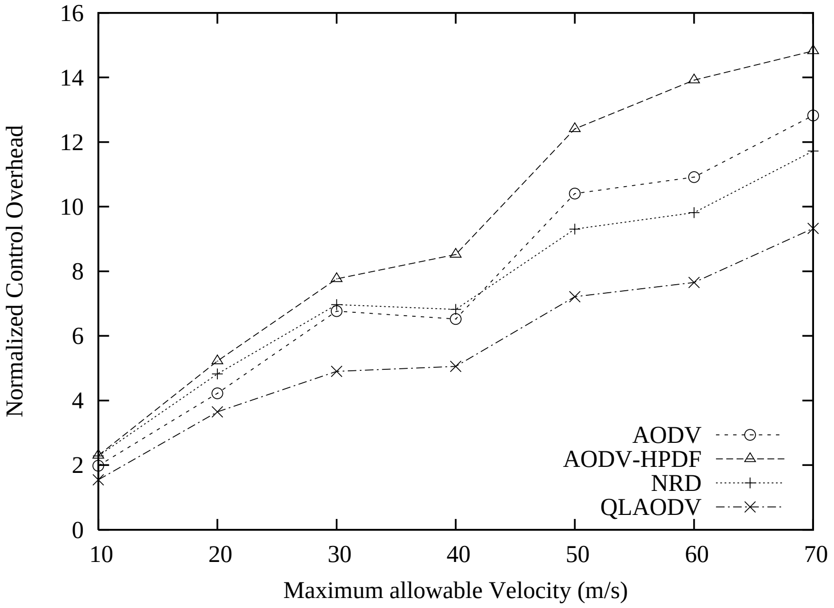


Figure 3.9 – Normalized control overhead for varying velocities in Freeway model.

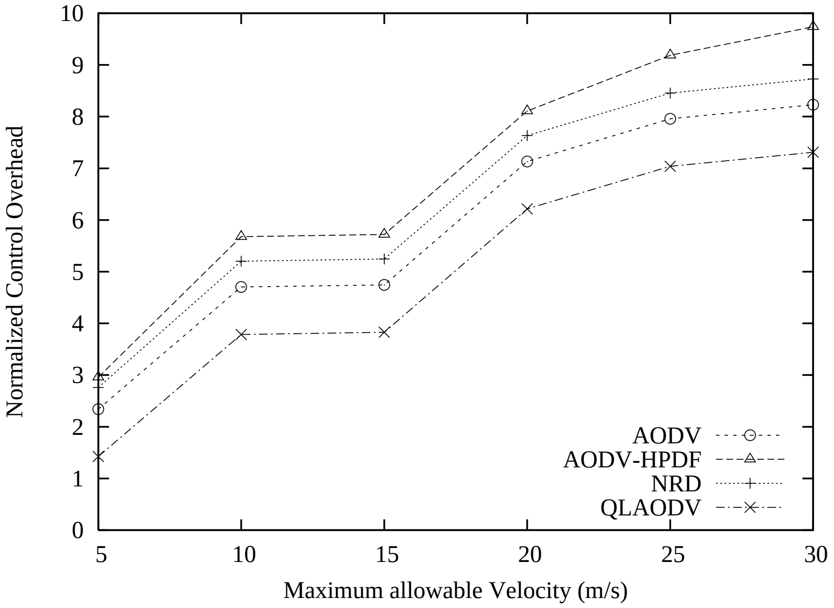


Figure 3.10 – Normalized control overhead for varying velocities in Manhattan model.

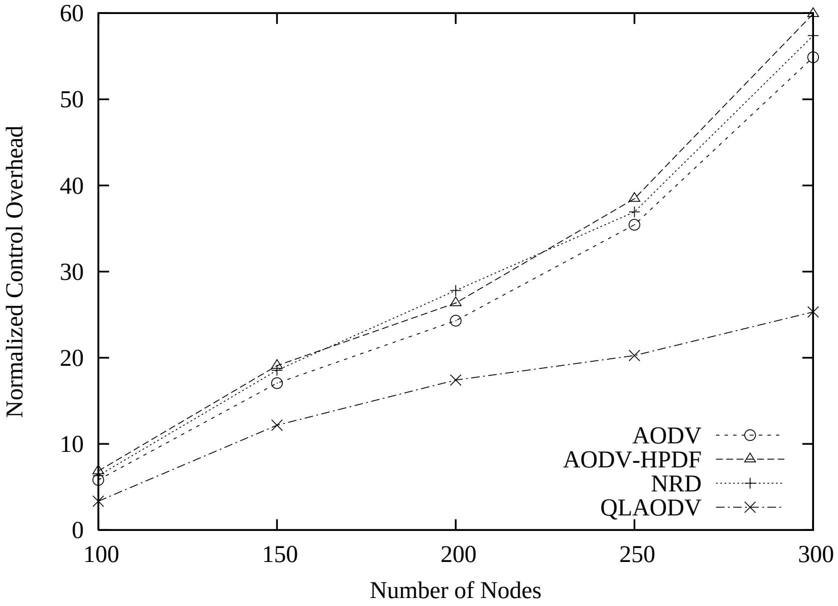


Figure 3.11 – Normalized control overhead for varying velocities in real street map based mobility model.

duction in the number of route errors . In the freeway model, the improvement over AODV is most obvious. This is because QLAODV can change route to use vehicles in the same direction to relay data. These routes are stable and therefore efficiently reduce the number of route errors.

In order to allow Q-Learning to work efficiently in a highly dynamic network environment, the QLAODV protocol uses additional packets, namely the route change request (RCNG-REQ) packet and route change replay (RCNG-REP) packet, to check the availability of candidate routes. Nevertheless, the RCNG-REQ packet and the RCNG-REP packet are sent unicast, and therefore this does not incur too great a network overhead.

Number of RCNG-REQ packets sent by and RCNG-REP packets received by source node

In order to illustrate the efficiency of the route change mechanism with respect to varying velocity, the number of RCNG-REQ packets sent by source nodes and RCNG-REP packets received by source nodes are shown in Figures 3.16, 3.17, 3.18 and 3.19. Error

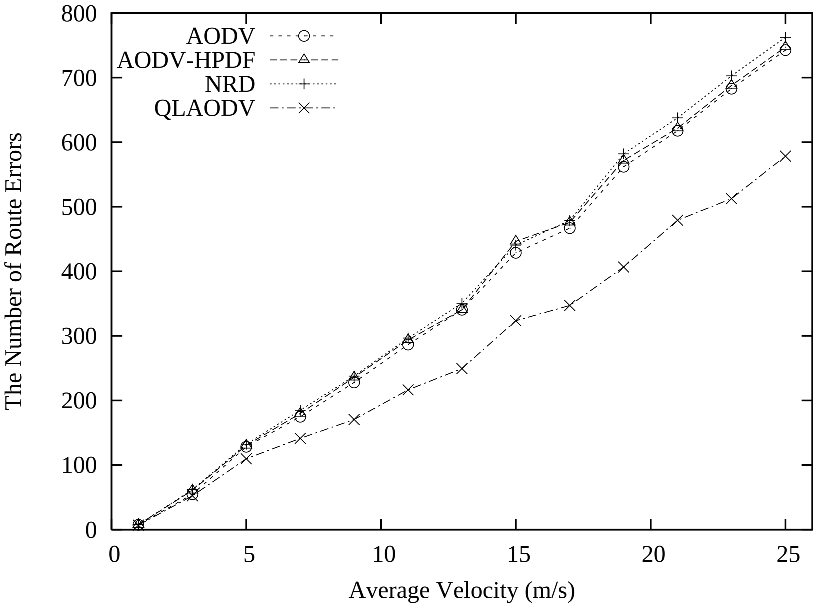


Figure 3.12 – Number of route errors for varying velocities in Random Waypoint model.

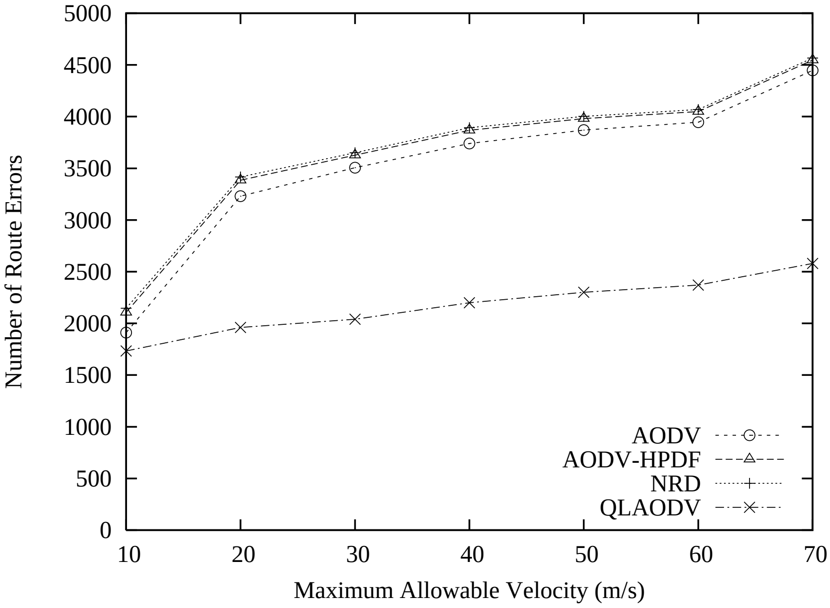


Figure 3.13 – Number of route errors for varying velocities in Freeway model.

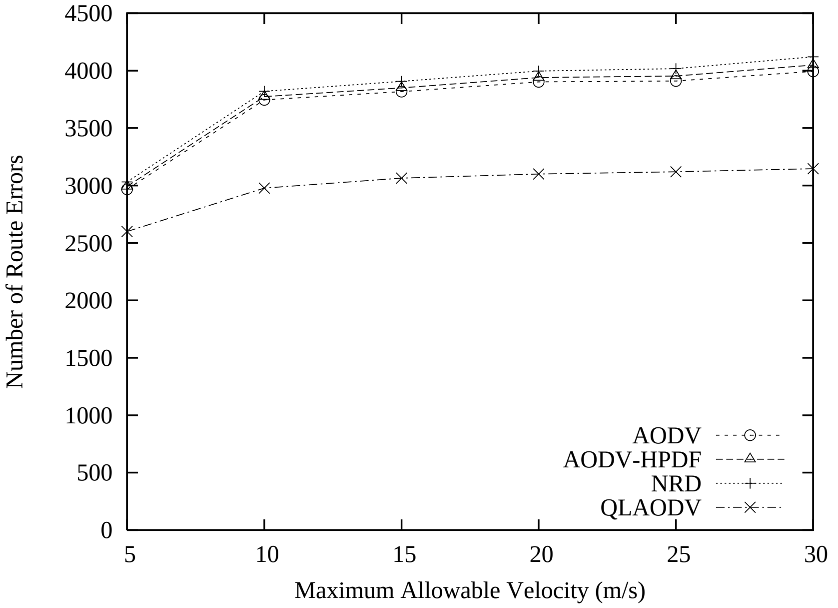


Figure 3.14 – Number of route errors for varying velocities in Manhattan model.

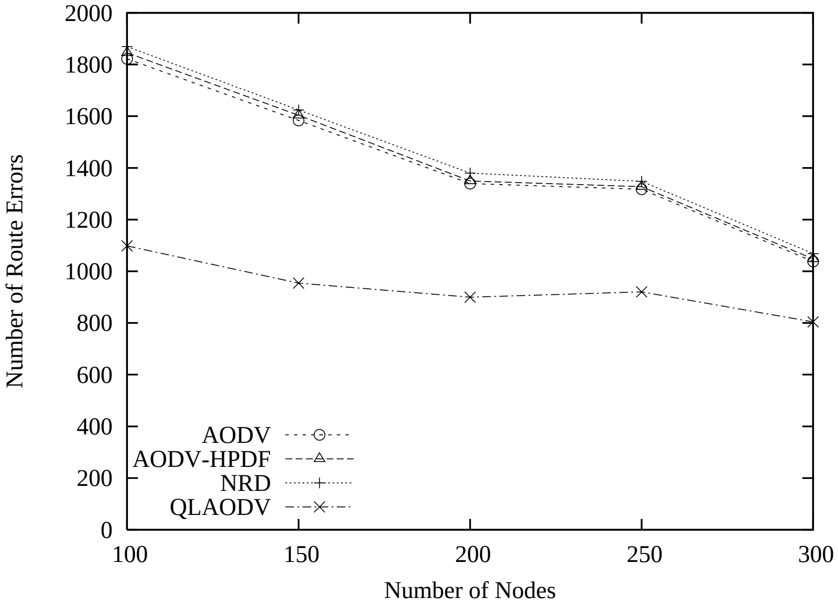


Figure 3.15 – Number of route errors for varying velocities in real street map based mobility model.

bars indicate the 95% confidence intervals. A route change attempt fails if the source node of the RCNG-REQ packet does not receive the corresponding RCNG-REP.

As shown in Figure 3.16, in the Random Waypoint model, many route change attempts fail when the velocity is high. This can be explained by the fact that high node velocity results in frequent topology changes and breakage of the candidate routes, which results in route change failure. This fact also shows that route change request/reply cycle is necessary in highly dynamic networks.

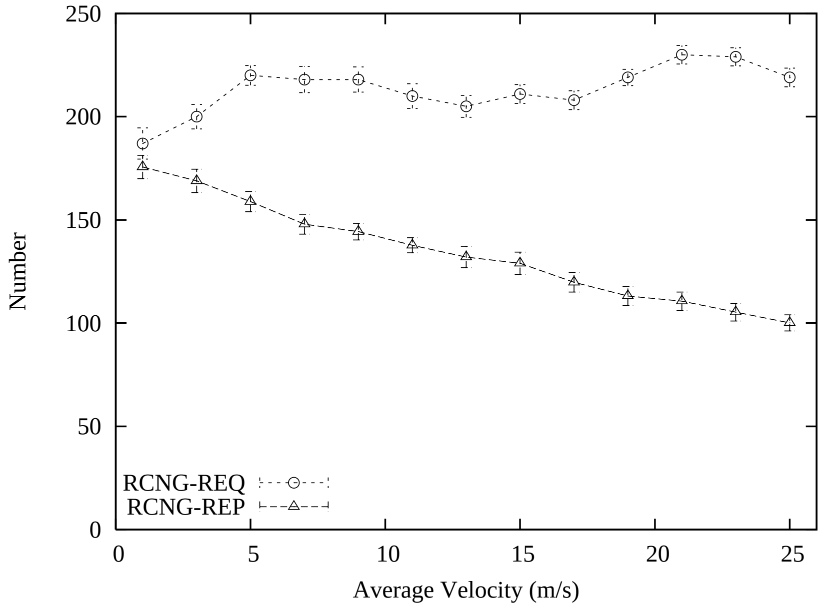


Figure 3.16 – Number of RCNG-REQ packets sent by and RCNG-REP packets received by source node for varying velocities in Random Waypoint model.

Figure 3.17 shows the number of RCNG-REQ packets sent by and RCNG-REP packets received by source node for varying velocities in the Freeway model. When the maximum allowable velocity is 10m/s, some routes using oncoming vehicles can be acceptable. However, when the maximum allowable velocity is increased to 20m/s, these routes become unstable due to a high degree of relative movement. This explains why the number of route change requests increases when the maximum allowable velocity changes from 10m/s to 20m/s. When the maximum allowable velocity increases more, the number of route change requests decreases. This is because vehicles intend to only use vehicles in the same lane to forward data packets even when it results in a longer route. Therefore route changes happen rarely. When the maximum allowable velocity increases to above 50m/s, the number of route change requests increases again because of frequent link changes.

In the freeway model, it can be observed from Figure 3.16 that the occurrence of route change failures is not influenced much by the speed of movement because the relative speed between vehicles moving in the same direction would not be very high.

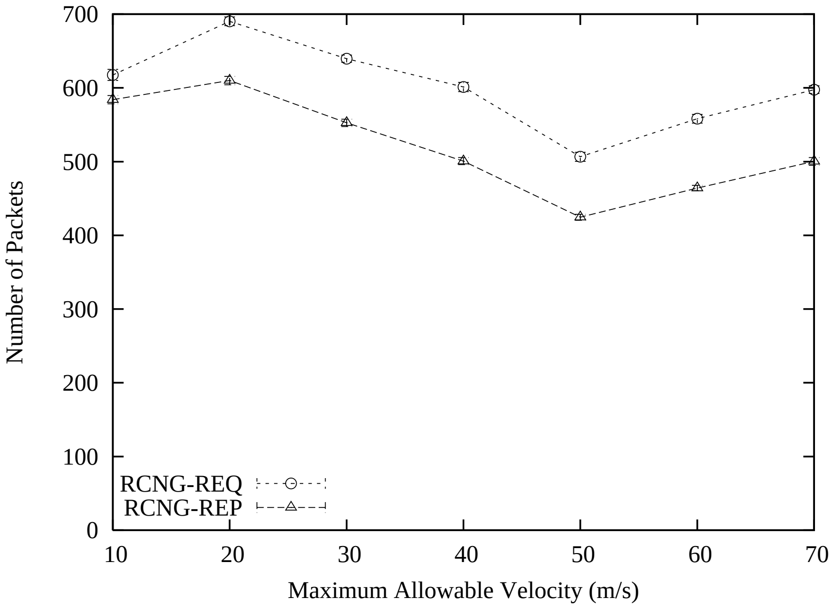


Figure 3.17 – Number of RCNG-REQ packets sent by and RCNG-REP packets received by source node for varying velocities in Freeway model.

In the Manhattan model, with the increasing of node velocity, the number of route change requests increases. The Manhattan model is different to the Freeway model in a way that vehicles can change direction at road intersections. This feature results the node movement in the Manhattan model is relatively unpredictable and is higher than the freeway model. Therefore, the number of route change request increases until the maximum allowable velocity reaches 20m/s. The number decreases slightly when the node velocity increases more. This is because some paths change too frequent to use them to deliver data.

It can be observed that many route change attempts fail when the velocity is high in the Manhattan model. This can be explained by the fact that high node velocity results in frequent topology changes and breakage of the candidate routes, which results in route change failure.

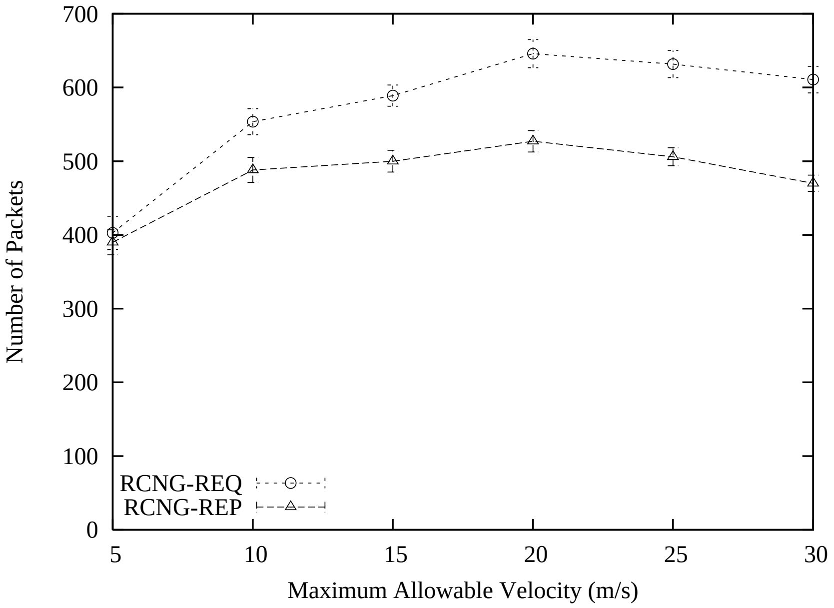


Figure 3.18 – Number of RCNG-REQ packets sent by and RCNG-REP packets received by source node for varying velocities in Manhattan model.

In the real street map based model, when the node density increases, the number of route change requests increases slightly. This is because the number of available paths increases. However, it is observed that the number decreases when the number of nodes increases further. This is because when the number of nodes increases, the average moving speed of vehicles will become slower.

End-to-End delay and route length

End-to-end delay for varying velocities in the four mobility model are shown in the following Figures 3.20, 3.21, 3.22 and 3.23.

As Figures 3.20, 3.21, 3.22 and 3.23 show, the end-to-end delay of AODV-HPDF and NRD is larger than that of AODV and QLAODV. This is not a surprise since AODV-HPDF and NRD have longer route lengths and hence higher delays when compared to AODV. It is observed that QLAODV can construct shorter routes than AODV and thus can provide a lower delay than AODV. To give a numerical proof to this behavior, the

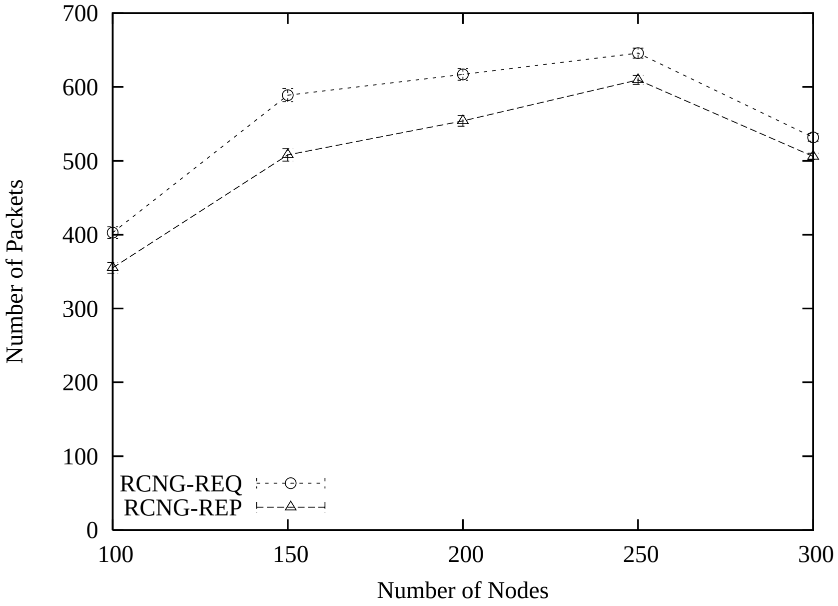


Figure 3.19 – Number of RCNG-REQ packets sent by and RCNG-REP packets received by source node for varying velocities in real street map based mobility model.

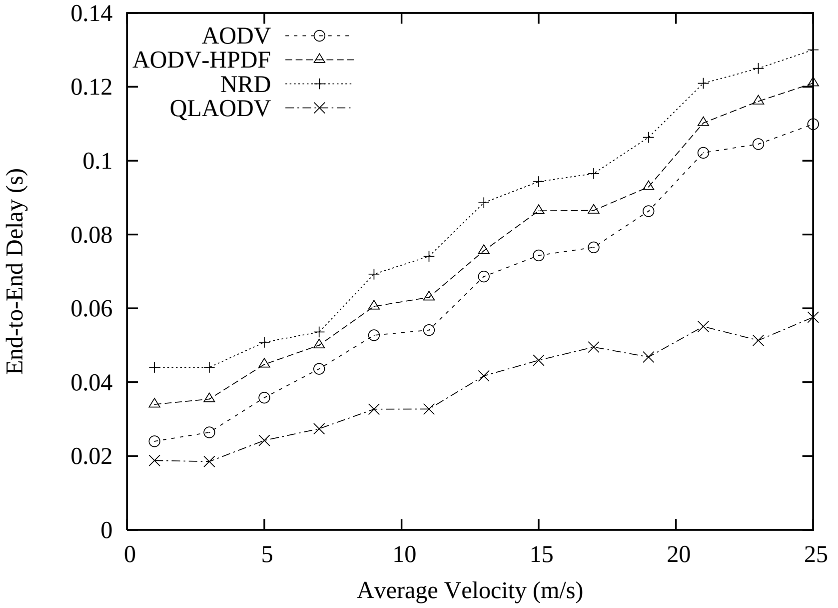


Figure 3.20 – End-to-end delay for varying velocities in Random Waypoint model.

route length comparison of the four protocols are shown in Figures 3.24, 3.25, 3.26 and 3.27. Another reason why QLAODV achieves a good delay performance is that QLAODV reduces the number of route errors and route request broadcasts and so shortens the

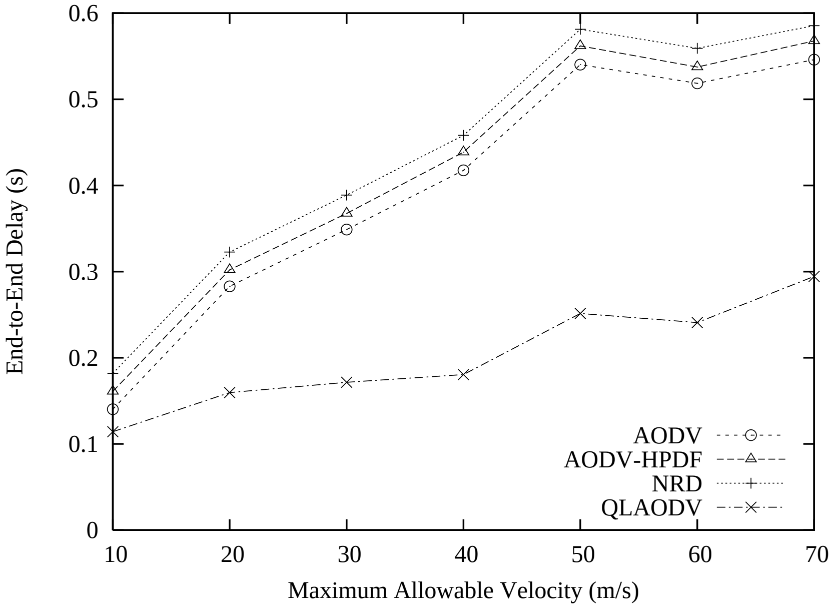


Figure 3.21 – End-to-end delay for varying velocities in Freeway model.

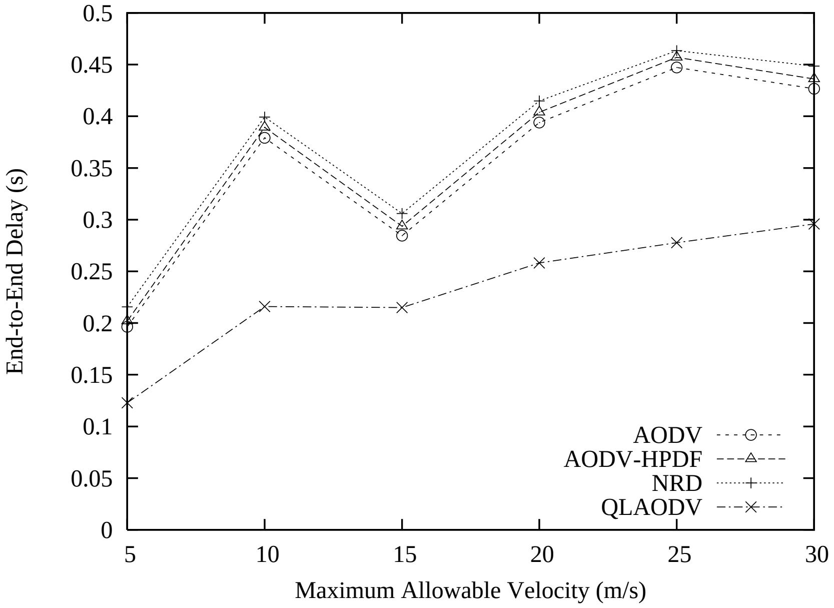


Figure 3.22 – End-to-end delay for varying velocities in Manhattan model.

time packets are waiting in buffers. As QLAODV results in a lower delay than the other three protocols, it can be considered for use in multimedia applications.

It is clear from Figures 3.24, 3.25, 3.26 and 3.27 that QLAODV can construct shorter routes than AODV. AODV-HPDF results in a longer route length due to the local re-

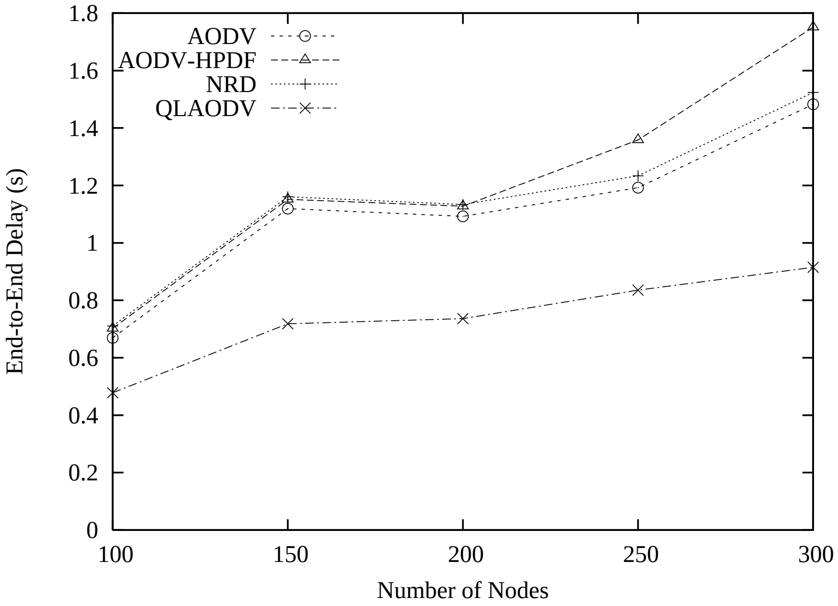


Figure 3.23 – End-to-end delay for varying velocities in real street map based mobility model.

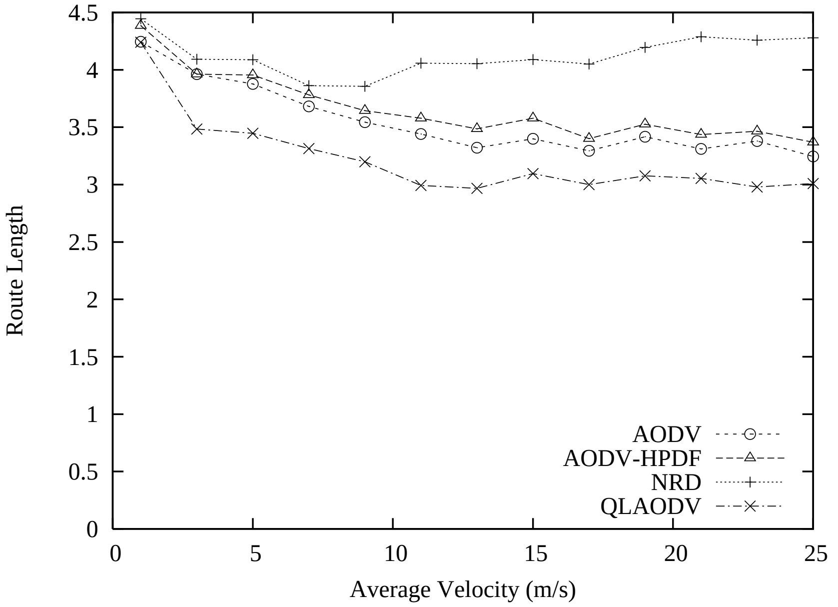


Figure 3.24 – Route length for varying velocities in Random Waypoint model.

pair without the hop-distance condition. NRD can salvage many packets in high-speed scenarios, but it results in a longer route.

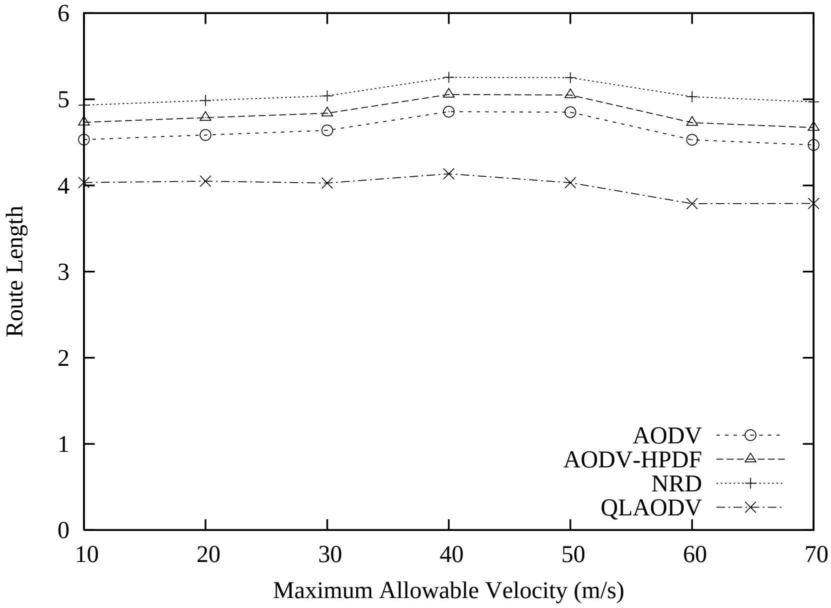


Figure 3.25 – Route length for varying velocities in Freeway model.

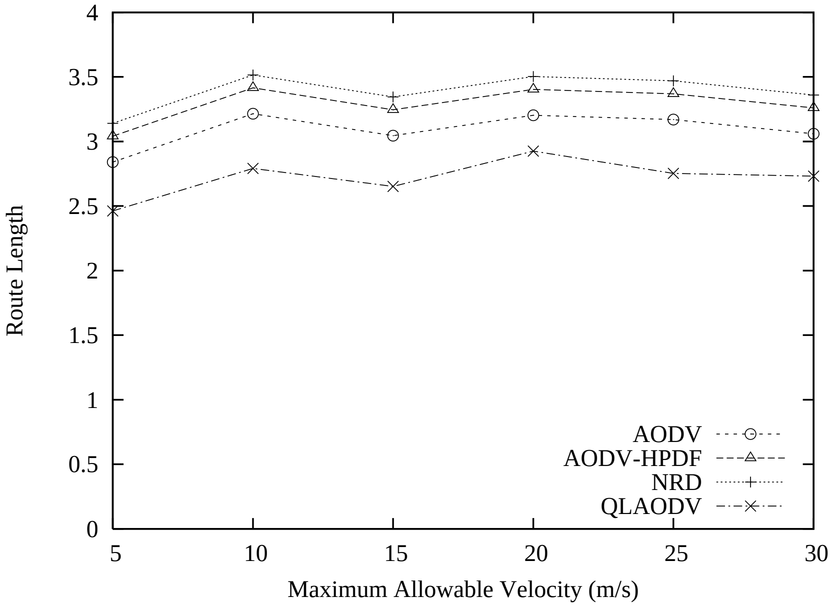


Figure 3.26 – Route length for varying velocities in Manhattan model.

3.4.3. Effects of Parameter values

The proposed protocol has two design parameters: α and γ. The learning rate parameter α limits how quickly learning can occur. In the proposed protocol, α governs how

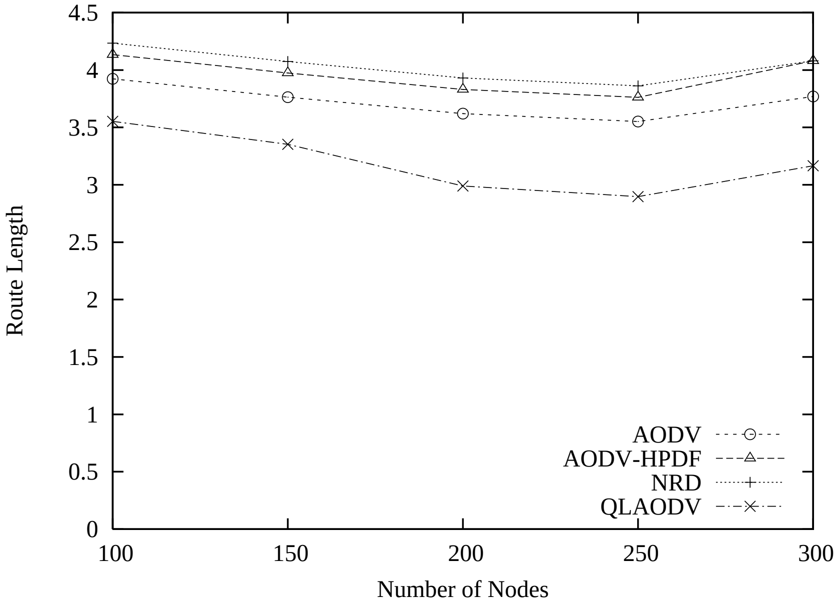


Figure 3.27 – Route length for varying number of nodes in real street map based mobility model.

quickly the Q-values can change with a network topology change. If the learning rate is too low, the learning will not adapt quickly to network dynamics. If the rate is too high, then the algorithm can not reflect the network movements accurately because agents can receive immediate misleading rewards. In the conventional reinforcement learning algorithm, learning rate α may be set to a low value. However, in QLAODV, since a link status information is disseminated in a multi hop manner from the destination node to the source node, α should be relatively high in order to ensure the link status information can quickly reach the source. Therefore, α is set to 0.8 in the QLAODV implementation.

The discount factor γ controls the value placed on future rewards. If γ is low, immediate rewards are optimized, while higher values of the discount factor cause the learning algorithm to more strongly count future rewards.

Figures 3.28, 3.29, 3.30 and 3.31 show effects of learning rate values and discount factor values on QLAODV's performance. Simulation environments are the same to that used in Figures 3.4, 3.5, 3.6 and 3.7 respectively. As shown in Figure 3.28, a relatively higher α can obtain good result in highly dynamic networks. However, if the learning rate is too high (in case of $\alpha = 0.9$), QLAODV suffers from immediate misleading updates. In QLAODV, node movement status and bandwidth status are reflected in the discount factor. Thus, γ's value has slightly effect on QLAODV's performance. Although QLAODV's performance does not so sensitive to parameter values, the pair [$\alpha = 0.8$, $\gamma = 0.9$] show best results.

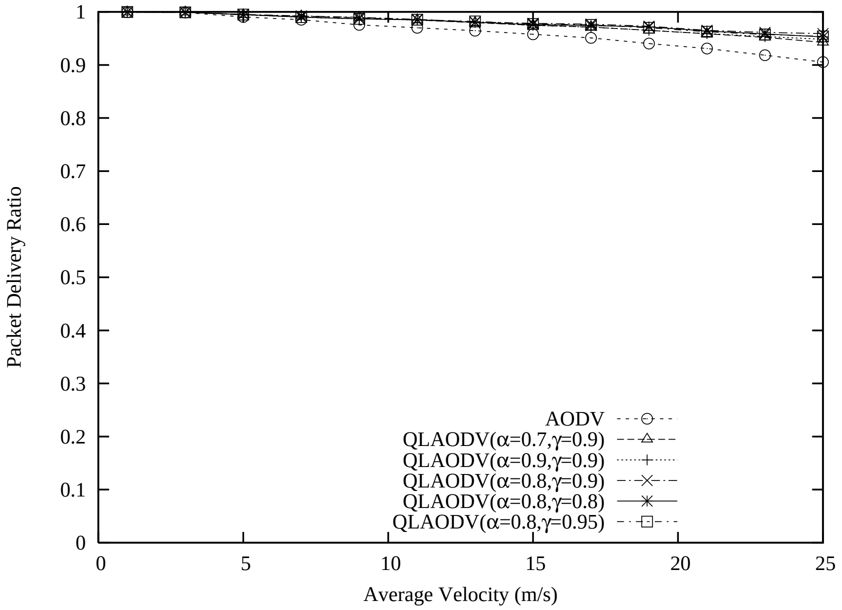


Figure 3.28 – Achieved packet delivery ratio with different parameter values for varying velocities in Random Waypoint model.

As shown in Figure 3.29, in the Freeway model, $\alpha = 0.9$ results a relatively poor packet delivery ratio. This is because a high α may suffer from immediate misleading updates. In the Freeway model, relative movement between two vehicles moving in the different direction can be very high. If the value of α is high, a vehicle may select a oncoming vehicle as a relay node. This does not happen often because QLAODV also considers relative movement using γ_x. However, it could happen if the route using the oncoming vehicle indicates a shorter route. It is also observed that in the Freeway model, γ can use a relatively high value. This is because relative speed between vehicles moving in the same direction is not very high and thus a route which uses these nodes can be relatively long.

Figure 3.30 shows the achieved packet delivery ratio with different parameter values

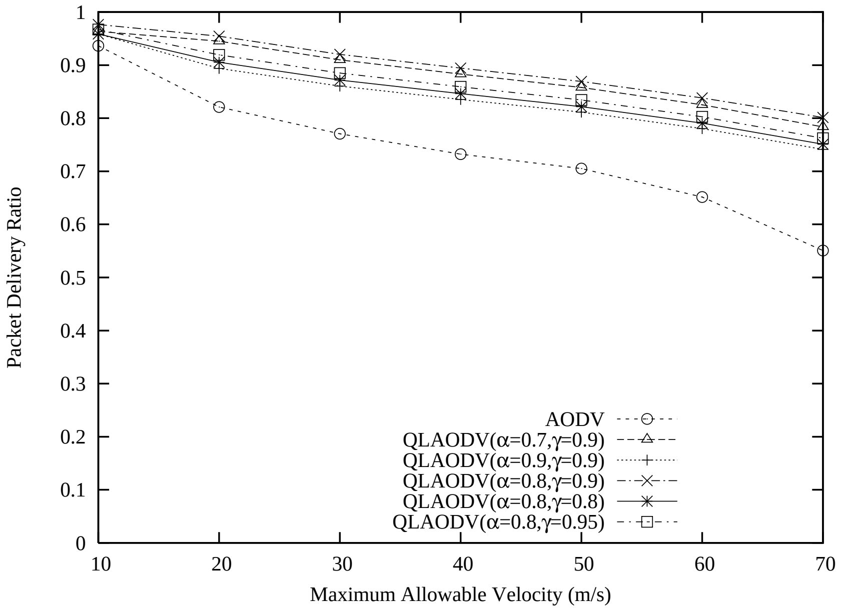


Figure 3.29 – Achieved packet delivery ratio with different parameter values for varying velocities in Freeway model.

for varying velocities in Manhattan model. It can be observed that in the Manhattan model, a higher α is not very bad because it tends to use a new route more quickly. This is acceptable in the Manhattan model in which the life time of a route is relatively short due to frequent link changes.

Figure 3.31 shows the achieved packet delivery ratio with different parameter values for varying velocities in real street map based mobility model. In the real street map based mobility model, it is also observed that the pair $[\alpha = 0.8, \gamma = 0.9]$ show the best result. Therefore, α is set to 0.8 and γ is set to 0.9.

3.4.4. Effect of the Transmission Rate

Figures 3.32, 3.33, 3.34 and 3.35 show the achieved packet delivery ratio, comparing the four protocols for varying transmission rate. For simulation in the Random Waypoint Model, 80 nodes move randomly in an area of 1000 m $\times$ 1000 m. Each node's pause time is 0 and moving velocity is a random value between 10 m/s and 30 m/s. In the freeway model, the maximum allowable vehicle velocity was 40m/s. In the Manhattan

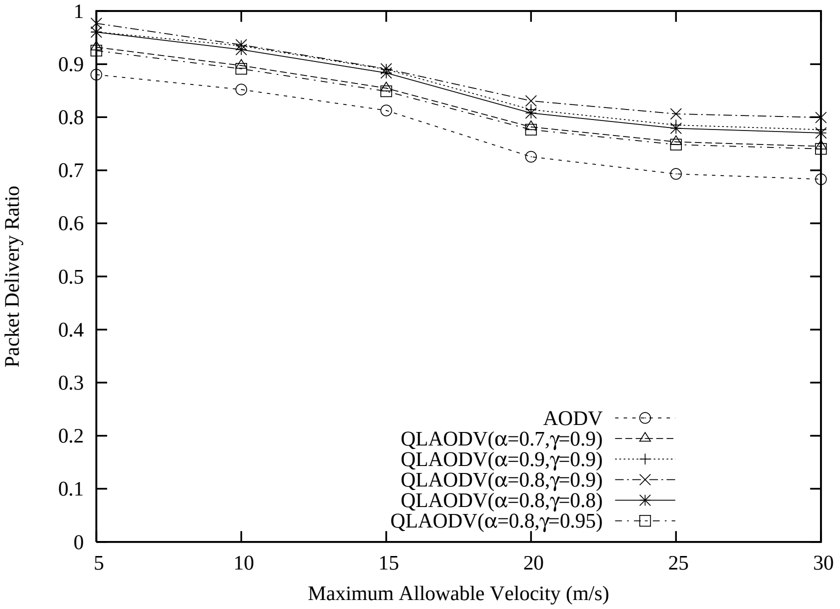


Figure 3.30 – Achieved packet delivery ratio with different parameter values for varying velocities in Manhattan model.

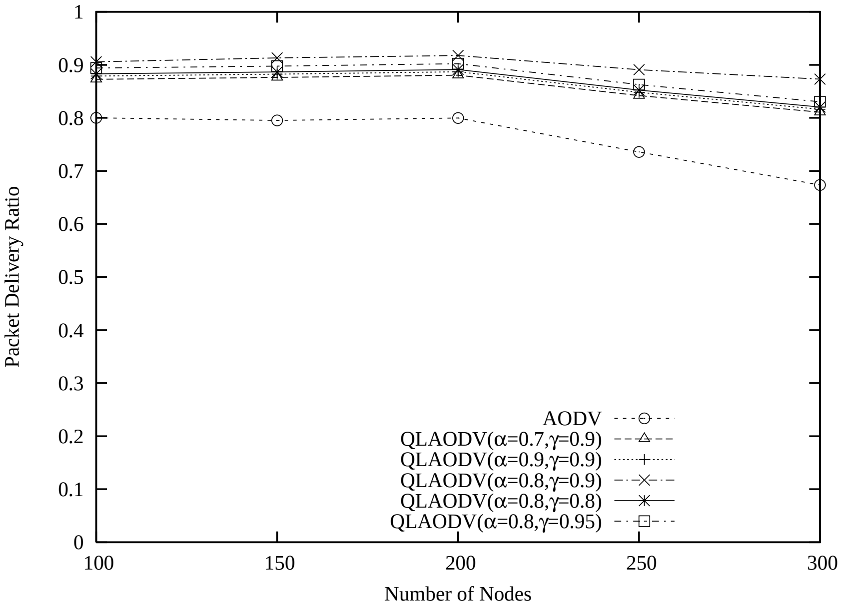


Figure 3.31 – Achieved packet delivery ratio with different parameter values for varying velocities in real street map based mobility model.

model, the maximum allowable velocity was set to 25m/s. 200 nodes were used in the real street map based mobility model. In all simulations models, 30 random CBR connections were generated and simulations were conducted varying the transmission rate of each individual connection from 16 kbps to 1024 kbps.

It is clear from Figures 3.32, 3.33, 3.34 and 3.35 that an increase of transmission rate results in an obvious negative impact on the packet delivery ratio of AODV. This is because the high transmission rate increases channel competition and network collisions. When the date rate is high, the number of data packets dropped upon link failure also increases. AODV-HPDF encounters same problem because of its high overhead. NRD only salvages packets that are dropped due to mobility, and not those dropped due to congestion. As the drops due to congestion become dominant, the NRD mechanism cannot make a significant positive impact on the overall performance. Since QLAODV considers bandwidth efficiency in the selection of the next hop and reduces the control overhead using a dynamic route change mechanism, QLAODV is superior to the other three protocols irrespective of the transmission rate.

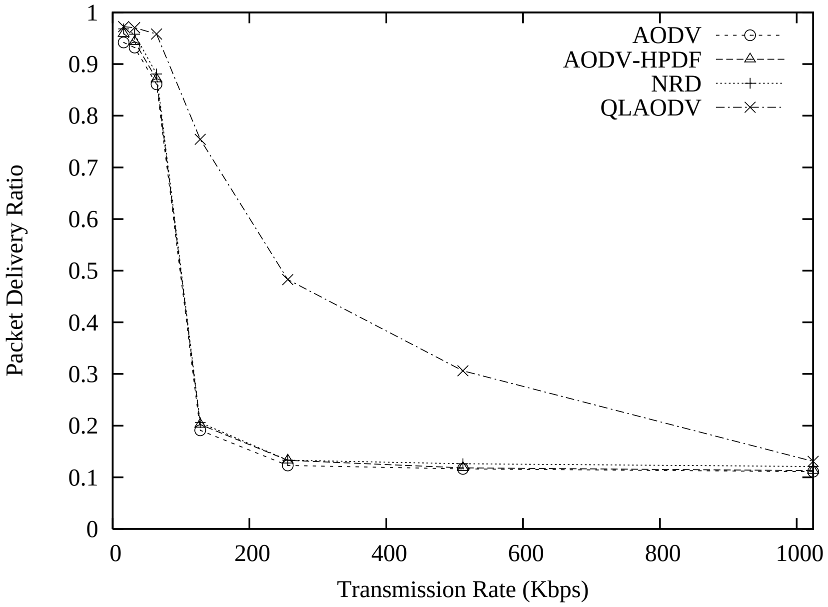


Figure 3.32 – Achieved packet delivery ratio for varying transmission rates in Random Waypoint model.

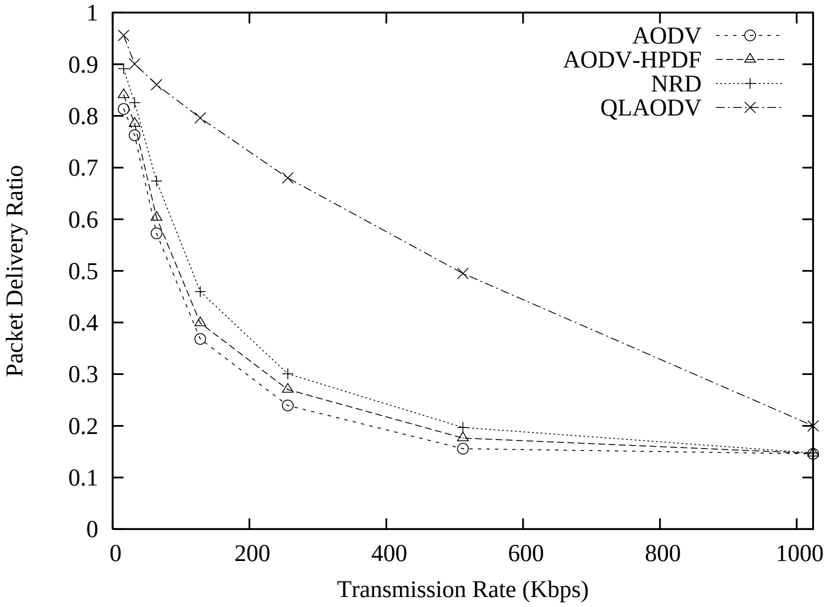


Figure 3.33 – Achieved packet delivery ratio for varying transmission rates in Freeway model.

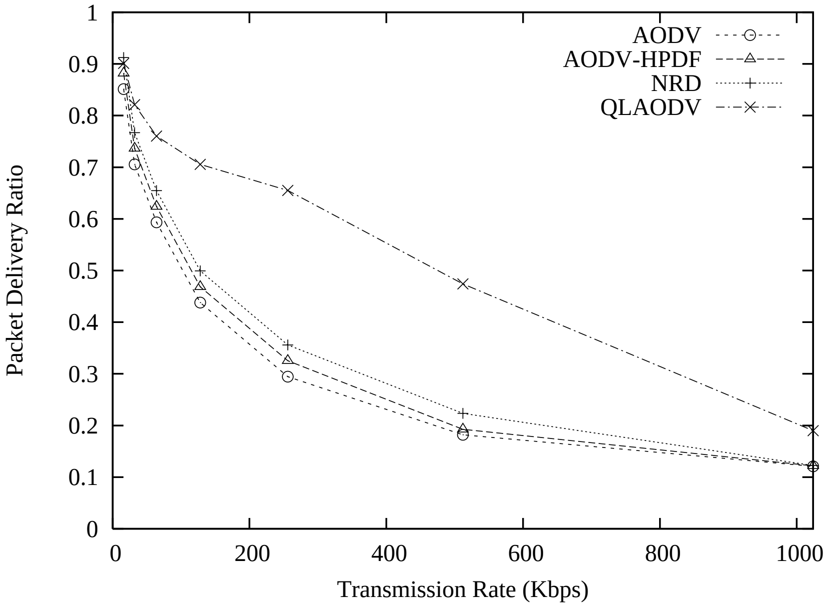


Figure 3.34 – Achieved packet delivery ratio for varying transmission rates in Manhattan model.

3.4.5. Overhead of Link Information Exchange

In QLAODV, every node attaches its MaxQValues to the hello messages to share its link state information with neighbors. Although QLAODV uses a dynamic table to store the

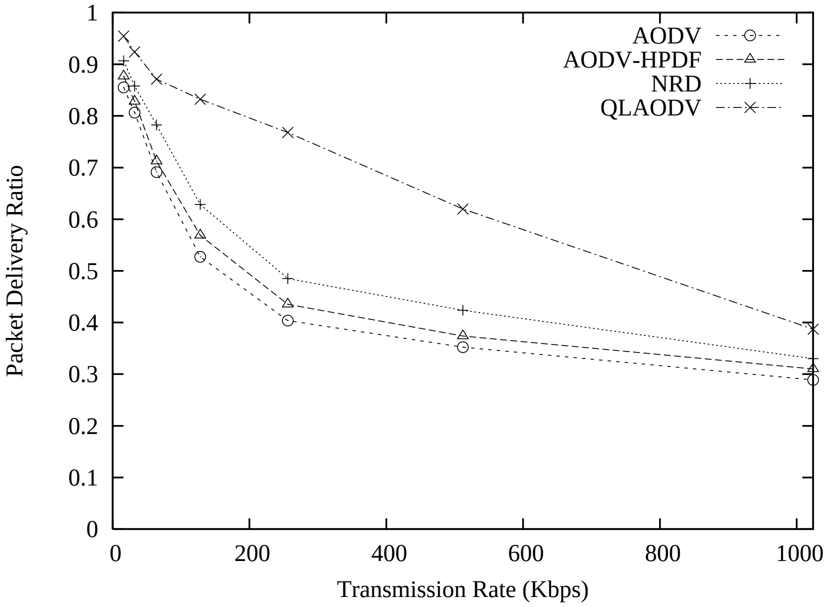


Figure 3.35 – Achieved packet delivery ratio for varying transmission rates in real street map based mobility model.

MaxQValues, with the diffusion of the link status information, the maximum number of elements in the MaxQValues can be equal to the number of nodes in the network. Figure 3.36 shows the average hello message overhead for each hello sender node. This simulation used the freeway model described in Subsection 3.4.1. This freeway has two lanes in each direction. All lanes of the freeway are 2000 m in length. 80 vehicles are randomly distributed on this freeway. The arrival velocity of each vehicle is set to 5 m/s and the maximum allowable velocity is set to 40m/s.

As the number of nodes increases, the information to be attached to the hello messages also increases, resulting in a higher message overhead. As shown in Figure 3.37, hello message overhead of QLAODV increases linearly with increasing node density. In a high density network, the bandwidth consumed by the hello messages in QLAODV can be relatively high. However, QLAODV is still more efficient than AODV because of the dynamic route change mechanism. In AODV, the number of route request packets increases with increasing node density. As a result, total back off time (in MAC layer) would increase dramatically because nodes have to back off for transmissions.

The overhead incurred by hello messages does not significantly impair the advantage

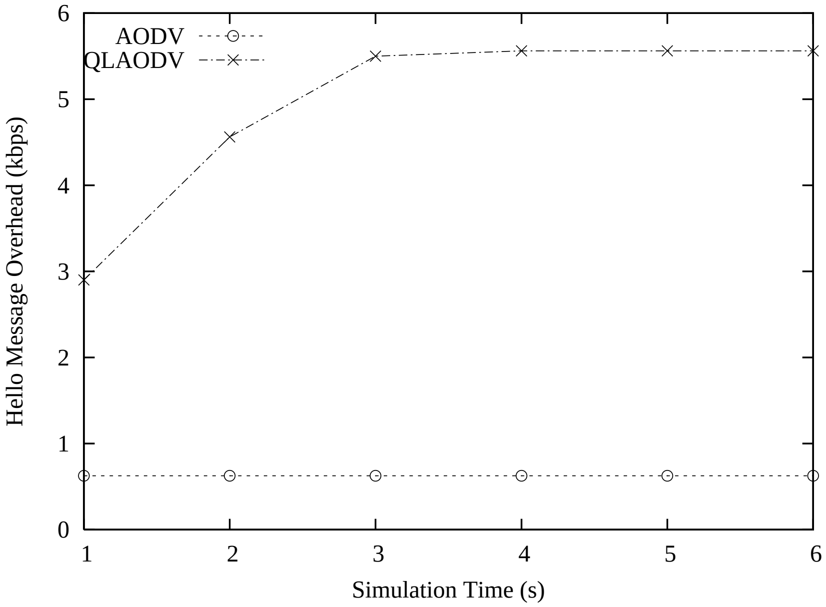


Figure 3.36 – Simulation time versus hello message overhead on each sender node in Freeway model.

of QLAODV. In vehicular ad hoc networks, communication partners of each node are expected to be in a short distance. When a node wants to communicate with another node which is in a long distance, QLAODV can be used with a little modification as below.

A threshold value $HELLO_{Thresh}$ is used to reduce the hello message overhead of QLAODV. Each node attaches a Q-Value to the hello messages only if its value is larger than the threshold because a smaller value would mean an inefficient path. This threshold ($HELLO_{Thresh}$) is a design parameter that should be carefully chosen. The appropriate value of $HELLO_{Thresh}$ depends on the network diameter. If the network is relatively small, a lower value is used. Otherwise, a relatively high $HELLO_{Thresh}$ can be used to reduce the control overhead. How to select a appropriate $HELLO_{Thresh}$ to optimize the QLAODV is considered as a future work.

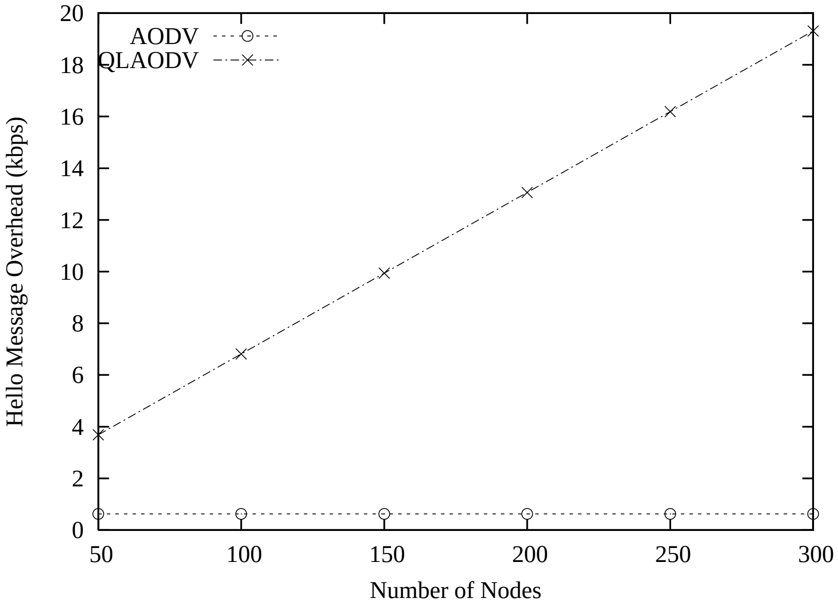


Figure 3.37 – Average hello message overhead on each sender node for varying number of nodes in Freeway model.

3.5. Discussion

3.5.1. Mobility Model

This chapter has provided performance evaluations with different mobility models. In the freeway model, the moving speeds of vehicles can be very high. However, the relative speed between vehicles moving in the same direction would not be very high. Therefore QLAODV benefits from using vehicles moving in the same direction as the source node to forward data. In the Manhattan model and the real street map based mobility model, vehicles have freedom of changing moving direction at an intersection. As a result, frequent link changes occur even when the vehicles' moving velocity is low. By preemptively changing routes before they break, QLAODV can achieve good performance.

3.5.2. Local Connectivity

In AODV, a node can use link layer notification or hello messages to keep track of its continued connectivity to its active next hop nodes. This paper provides experimental results based on the assumption that link layer notification is available. In cases where the link layer notification is unavailable, AODV uses hello messages and AODV's performance drops drastically with increasing node velocity. In that case, the advantage of QLAODV is more apparent. This is because many packets would be dropped because AODV can not detect link failure quickly enough. Similarly, AODV-HPDF and NRD also face this problem. Since the dynamic route change mechanism, QLAODV can handle this because it can switch to a new route before a link break occurs.

3.5.3. Overhead

QLAODV also uses hello messages, to exchange link information. Nevertheless, the messages do not significantly impair the advantage of QLAODV because the messages are sent only periodically. In QLAODV, the hello interval is 1 s, so it will not incur too great an overhead compared to the route request broadcast of AODV in highly dynamic networks. It is also quite reasonable to use hello messages because it is necessary for every vehicle to be aware of its neighbors in a VANET. The simulation results confirm that QLAODV offers a significant performance improvement.

In QLAODV, every node has to maintain a Q-Table, which will consume more memory than the original AODV. However, this is not a problem in vehicular ad hoc networks because vehicles can have enough memory.

3.5.4. Advantages

As described above, AODV-HPDF utilizes a local repair method in which both the upstream node and the source node initiate a route discovery when a link fails. While providing slightly better performance, this mechanism results in high control overheads during situations of high mobility. In NRD, when a link on a route fails due to mobility, the intermediate node on the failed link forwards packets to one of its neighbors to which the route information for the corresponding destination has already been diffused. NRD salvages packets efficiently in the case of multiple streams terminating at a single destination node. However, as the streams become more distributed, NRD's effectiveness diminishes. Moreover, the NRD mechanism can not make a significant improvement when the packet drops due to congestion, as opposed to link failure, become dominant. Fortunately, QLAODV can offer a notable performance improvement in various situation. First, the novel dynamic route change mechanism is more effective than taking action after link failure. Another merit of QLAODV is that it considers hop count, stability and bandwidth efficiency in route selection, making QLAODV very robust to network dynamics.

3.6. Conclusions

This chapter has proposed QLAODV, a routing protocol that uses a reinforcement learning algorithm to handle network state information and a unicast route change request/reply cycle to check the correctness of the information obtained. QLAODV uses a dynamic route change mechanism to reduce the number of route errors and route discoveries. QLAODV can react quickly to network topology changes and can pick the best route for data delivery using newly learned information. QLAODV considers hop count, stability and bandwidth usage in route selection. It is a fully topology-based routing protocol and is therefore easy to implement. Through evaluation of the proposed routing protocol on different mobility models, it is confirmed that QLAODV offers a significant performance advantage over existing alternatives.

Chapter 4

A Novel Multi-hop Broadcast Protocol for Vehicular Safety Applications

4.1. Introduction

Many VANET applications are broadcast based and thus multi-hop broadcast is required to disseminate information to desired receivers. The simplest way to disseminate information is flooding. However, flooding has serious problems. First of all, simple flooding cannot provide enough reliability. Its non-support of retransmission degrades the data delivery ratio and its redundant rebroadcasts causes many collisions. In the heavy traffic condition where the VANET communication is likely to be exploit, many vehicles exist in a dense manner within a radio transmission range. In such a high-density network environment, flooding introduces redundant rebroadcast: many vehicles within a radio transmission range try to rebroadcast a received message, and this redundancy causes high overhead in the data dissemination.

Although there are many proposals on VANET broadcast protocols focusing on the reliability and the efficiency in a vehicular environment, they have some limitations. Proposals focusing on the reliability do not consider high-density environments. On the other hand, those focusing on the efficiency in high-density environments are only designed for dense networks and provide poor performance in the sparse network environment.

In high-density networks, it is possible to reduce broadcast redundancy by selecting a small subset of nodes to relay a broadcast data packet. To efficiently broadcast messages in vehicular ad hoc networks, relay node selection should be handled efficiently. Many methods to select relay nodes are proposed [14, 46–48]. However, none of them considers nodes' mobility in relay node selection. Therefore, they are not suitable for highly dynamic vehicular ad hoc networks.

This chapter first proposes a relay node selection algorithm (enhanced MPR selection algorithm) considering network mobility. Based on the proposed algorithm, a reliable and efficient broadcast protocol that can work well in various traffic conditions is proposed. The proposed protocol employs a hop-by-hop retransmission scheme to provide strict reliability in various traffic conditions. This protocol also provides low overhead in a high-density network environment by means of introducing boundary nodes which are in charge of rebroadcasting. The protocol also works well in a sparse network. The effectiveness of the proposed protocol is confirmed through simulations using the network simulator ns-2.

The rest of this chapter is organized as follows; Section 4.2 proposes an enhanced MPR selection algorithm. Next in section 4.3, the detailed description of the proposed protocol is presented. Section 4.4 evaluates the protocol's performance. Finally, the conclusions are presented in section 4.5.

4.2. Enhanced MPR Selection Algorithm

In order to reduce redundant broadcast in high-density networks, the messages should be only rebroadcast by a subset of neighbors. Without loss of generality, two hop neighbor information is used to select relay nodes. It is assumed that every node broadcasts hello messages periodically. Every vehicle places its one-hop neighbor information to hello messages. Therefore, vehicles are aware of their two-hop neighbors. The proposal does not assume a GPS-like positioning device is available for every vehicle. (The terms node and vehicle are used interchangeable in this thesis.)

4.2.1. Problems in the Original MPR Selection Algorithm

Although the original MPR selection algorithm [14] based broadcast scheme could efficiently reduce redundant rebroadcasts in static networks [20], the original MPR selection algorithm fails in dynamic networks. Figure 4.1 is used as an example. In the figures which are used in this chapter, $TR(x)$ shows the transmission range of node x. As shown in Figure 4.1(a1), S receives hello from its neighbors and updates its two-hop neighbor information. Then the network topology changes to a new state, which is shown in Figure 4.1(a2). S intends to send a broadcast data at this time and select B2 as a relay node due to its out-of-date two-hop neighbor information. Obviously, B2 is no longer the node that provides maximal additional coverage. Additional coverage of node x, $AC(x)$, is used to mean the set of nodes which are one-hop neighbors of the node x but not one-hop neighbors of the sender node s. Specifically, $AC(x)$ is defined as

$$AC(x) = \overline{N(s)} \cap N(x), \tag{4.1}$$

where $N(x)$ and $N(s)$ denote a one-hop neighbor set of node x and one-hop neighbor set of sender node s respectively. It should be noted that node x belongs to $N(s)$.

The original MPR selection algorithm also fails in case of another situation, which is shown in Figure 4.1(b1,b2). S updates neighbor information when B2 is the best relay node as Figure 4.1(b1) shows. The network topology changes to a new state (Figure 4.1(b2)) and S selects node B2 as a relay node based on previous knowledge. As a consequence, B2 could not receive the broadcast packet (B2 is out of the transmission range of S) and the broadcast packet could not be transmitted to two hop neighbors.

Thus, out-of-date neighbor information influences effectiveness of the original MPR selection algorithm. Obviously, it is important to consider node mobility in MPR selection in vehicular ad hoc networks. Therefore, an enhanced MPR selection algorithm considering network mobility is proposed here.

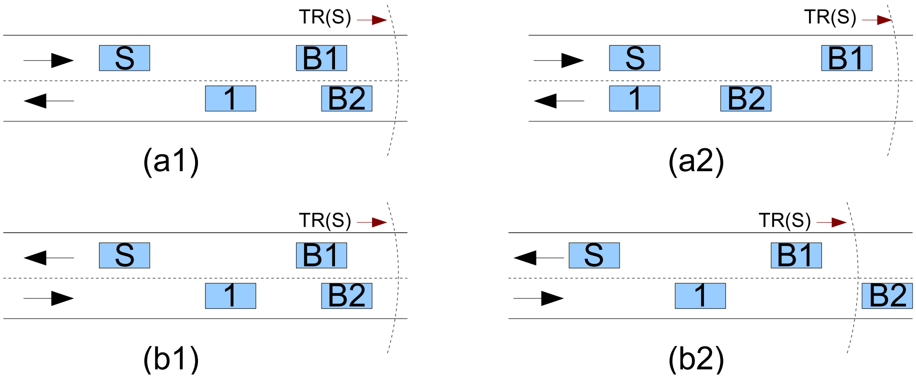


Figure 4.1 – Cases of MPR selection failure.

4.2.2. Enhancement of MPR Selection Considering Network Mobility

Notation

The notation which is used in this chapter is shown in Figure 4.1.

MPR Selection Criteria

The original MPR selection algorithm [14] considers additional coverage (as shown in the Eq. 4.1) only. The additional coverage is an important factor, but not all. In this chapter, predicted MPR fitness (PMF) is defined to evaluate a node whether it is suitable for relaying broadcast packet or not. To calculate $PMF(x)$ for node x, multipoint relay fitness ($MF(x)$) is introduced as

$$MF_i(x) = \frac{|AC_i(x)|}{|N_i(s) \cup N_i(x)|} \tag{4.2}$$

Table 4.1 – Notation.

MF	multipoint relay fitness
PMF	predicted MPR fitness
$PMF_i(x)$	current PMF of node x
$PMF_{i-1}(x)$	previous PMF of node x
$AC(x)$	additional coverage of node x
$\|A\|$	number of elements in set A
$ACN(x)$	$\|AC(x)\|$, number of elements in $AC(x)$
ACN_{min}	minimal ACN between one-hop neighbors
ACN_{max}	maximal ACN between one-hop neighbors
ACN_{Thresh}	a threshold value which is used to determine MPR candidate nodes
$N(x)$	one-hop neighbor set of node x
$N(s)$	one-hop neighbor set of sender node s
$N^2(s)$	two-hop neighbor set of sender node s
μ	a rate which denotes how much current value contribute to the new value
θ	discount factor
$MPR(s)$	multipoint relay set of sender node s

where i indicates the current value.

When a node s receives a hello message from node x, it calculates corresponding $MF_i(x)$. In Eq. 4.2, $N_i(x)$ denotes neighbor set of node x, $|N_i(x)|$ denotes number of x's one hop neighbors. Eq. 4.2 could give higher value to nodes that have larger additional coverage. However, it is not sufficient to only consider the additional coverage in dynamic networks. So, nodes' movement is considered in the calculation of PMF. In order to provide different weights to different levels of movement, discount rate θ is included.

$$\theta = \begin{cases} \sqrt{\frac{|AC_i(x) \cap AC_{i-1}(x)|}{|AC_i(x) \cup AC_{i-1}(x)|}}, & \text{if } AC_i(x) \cup AC_{i-1}(x) \neq \phi \\ 0, & \text{otherwise.} \end{cases} \tag{4.3}$$

where $i-1$ indicates the previous value. Eq. 4.3 could give a bigger value to the same directed vehicles and smaller value to vehicles moving in the opposite direction. In Eq. 4.3, square root ($\sqrt{.}$) is used to smooth the value because experimental results have shown that this leads to better results. If a node x is moving in the opposite direction to the sender, the corresponding θ will be smaller than other vehicles which have similar direction because its additional coverage is frequently changing.

The proposed algorithm also considers nodes' history in PMF calculation. This is because, in general, if a link's duration time is long, it is more likely to be durable in the future. For example, a sender should use vehicles in the same lane or same direction to forward broadcast packets. μ is used to consider node's history in PMF calculation. μ is a rate that denotes how much a current value contributes to the new value.

Considering node's current state, history and movement, a neighbor's PMF is updated as follows.

$$PMF_i(x) \leftarrow (1-\mu)PMF_{i-1}(x) + \mu \times \theta \times MF_i(x). \tag{4.4}$$

$PMF_i(x)$ is updated upon reception of a hello from its neighbor. Every node maintains a PMF ($PMF_{i-1}(x)$) and a AC ($AC_{i-1}(x)$) for every one-hop neighbor. In Eq. 4.4, if it is the first PMF calculation, the $PMF_{i-1}(x)$ is set to 0. Similarly, as far as Eq. 4.3 is concerned, if it is the first AC calculation, the $AC_{i-1}(x)$ will be set to ϕ. The sender node uses these values and the current MF ($MF_i(x)$) and AC ($AC_i(x)$) to calculate the latest PMF ($PMF_i(x)$) as shown in Eq. 4.3 and Eq. 4.4. The node then updates the $PMF_{i-1}(x)$ and $AC_{i-1}(x)$ it maintains. In the proposed algorithm, $PMF(x)$ is reset to zero if the sender did not hear from x in three times the hello interval.

Note that μ is a design parameter that should be carefully chosen. If the value is too small, PMF will not adapt quickly to network dynamics. A higher μ discounts older observations faster. However, if the value is too large, then the PMF cannot reflect network movement tendency because the larger value will be vulnerable to temporary misleading values. Through simulations, it is observed that 0.6 to 0.8 are better values for μ. However, there are not significant differences between them. Therefore, μ is set to 0.7.

θ is used to give different discounts to different levels of movement. Its role is different from μ. μ is used to control how quickly the $PMF_i(x)$ can change with a new $MF_i(x)$. $MF_i(x)$ cannot reflect the neighbor x's relative movement to the sender node. However, the proposed algorithm should select a relatively stable node. Therefore, the proposed algorithm includes θ to consider node movement when selecting a relay node. The following examples are used to explain θ's effect. As shown in Figure 4.2, a network

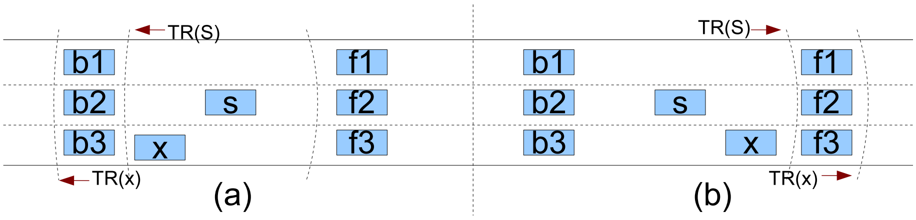


Figure 4.2 – An example to explain the role of θ.

topology changes from (a) to (b). In this case, $MF_i(x)$ does not change because additional coverage of x does not change. However, the new $MF_i(x)$ should be discounted because the relative movement of x' is high. Therefore, θ is included to consider this movement.

MPR *Selection Procedure*

Senders (broadcast source nodes or relay nodes) in vehicular ad hoc networks could be divided to the following two different types, according to their broadcast intentions.

1. There is one type of senders that only need to disseminate messages in one direction. In general, relay nodes (except nodes near an intersection) belong to this type. These senders use algorithm $E1$, which will be described later. Here an intersection is used to mean a road junction where two or more roads either meet or cross at the same level.

2. There also exists another type of sender that require disseminating messages in more than one direction. Broadcast source nodes always have to select at least two relay nodes to guarantee dissemination of messages in both forward and backward directions. Senders which are near to an intersection also need to disseminate messages in more than one direction. This type of senders use algorithm $E2$, which will be described later. In here, it is assumed that vehicles know they are near an intersection or not. This can be achieved by beaconing of access point at the intersection.

Algorithm $E1$**:** Figure 4.3 represents the process for Algorithm $E1$. The sender first calculates a threshold value ACN_{Thresh} as

$$ACN_{Thresh} = ACN_{min} + (1 - \beta) \times (ACN_{max} - ACN_{min}), \tag{4.5}$$

where ACN denotes the number of elements in AC. ACN_{min} is the minimal ACN between neighbors in the forward direction and ACN_{max} is the maximal ACN between neighbors in the forward direction. Neighbors in the forward direction are used to mean the neighbor nodes that are not neighbors of the upstream node. The sender node can get its neighbors in the forward direction by simply excluding the upstream node's neighbors from its one-hop neighbors. From the neighbors in the forward direction, the sender node first selects the nodes that have larger ACN than ACN_{Thresh} as MPR candidates. The sender then specifies the node that has maximal PMF between these MPR candidates as the relay node.

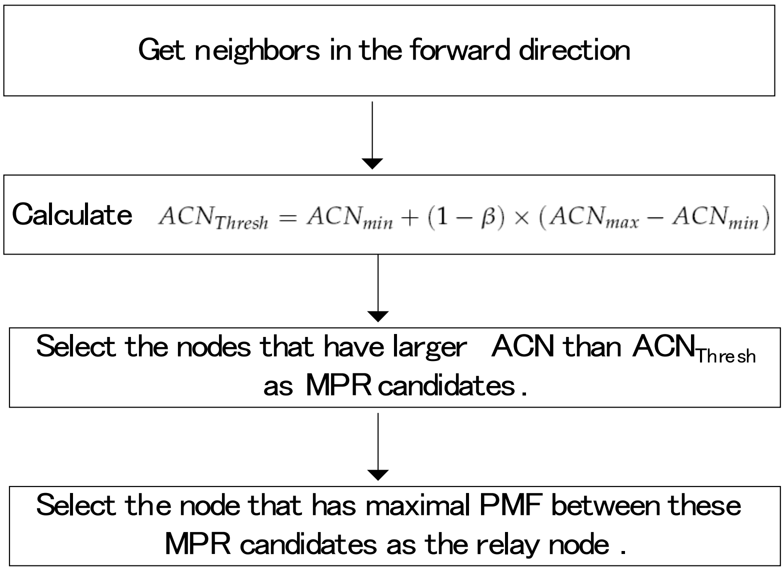


Figure 4.3 – Flow chart for Algorithm $E1$.

Figure 4.4 is used to explain why the proposed protocol selects MPR candidates from neighbors in the forward direction. Node S is a sender node and node S specifies node

C as a boundary node and broadcasts a data packet. Upon reception of the data packet, node C specifies the next relay node. Obviously, node C should only select node F as a MPR candidate because node C has to disseminate information to node f1. However, if node C sets MPR candidates as all one-hop neighbors, node B will be selected as a relay node because node B's ACN is much larger than node F's. Fortunately, in algorithm $E1$, because MPR candidates are selected from the neighbors in the forward direction, node C will only select node F as a MPR candidate.

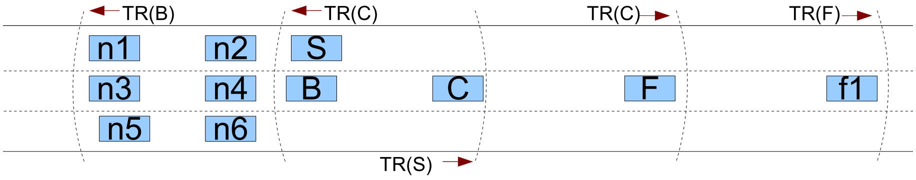


Figure 4.4 – An example for algorithm $E1$.

In algorithm $E1$, the value of β determines the set of MPR candidates. If the value is 0, ACN_{Thresh} will be ACN_{max}. Thus only the nodes that have maximal additional coverage is selected as MPR candidates. If the value is 1, the ACN_{Thresh} is ACN_{min}. In this case, the set of MPR candidates is all its neighbors in the forward direction. This means that the sender node selects the relay nodes totally based on PMFs of its neighbors in the forward direction. As a result, the sender node selects a node that is very near, and this results in inefficient relay. So, β is used to control the value of threshold. Simulation results show that generally, selecting the first quarter of nodes according to the values of ACN results in good performance outcome in various node densities. So β is set to be 1/4.

Algorithm $E2$**:** As shown in Figure 4.5, the process of Algorithm $E2$ is as follows.

1. Start with an empty multipoint relay set $MPR(s)$ where s indicates the sender node.

2. First select those one-hop neighbor nodes in $N(s)$ as multipoint relays which are the only neighbor of some node in two-hop neighbor set ($N^2(s)$), and add these

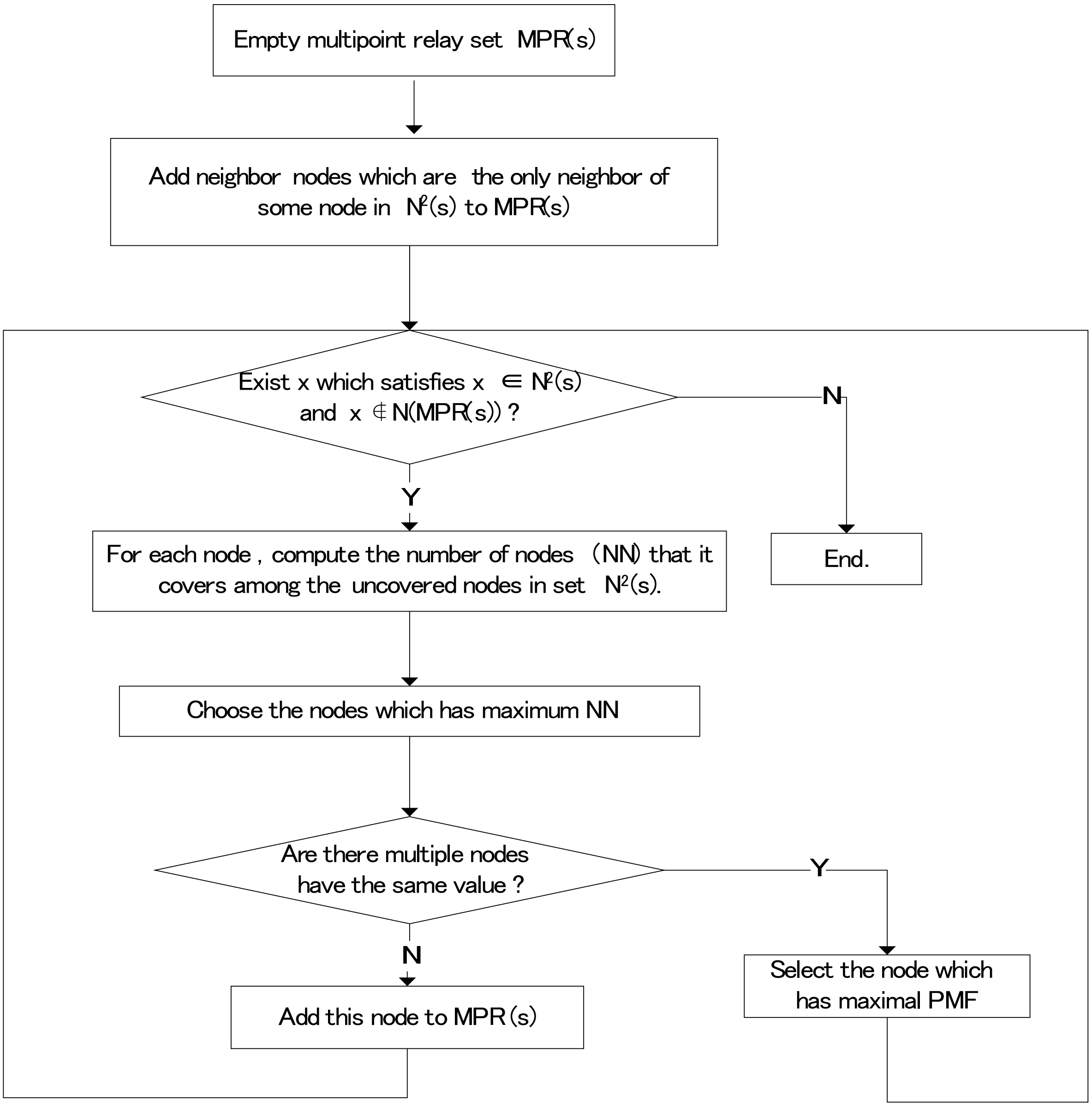


Figure 4.5 – Flow chart for Algorithm $E2$.

one-hop neighbor nodes to the multipoint relay set $MPR(s)$.

3. While there still exist some node in $N^2(s)$, which is not covered by the multipoint relay set $MPR(s)$:

   (a) For each node in $N(s)$, which is not in $MPR(s)$, compute the number of nodes that it covers among the uncovered nodes in set $N^2(s)$.

(b) Add that node of $N(s)$ in $MPR(s)$ for which this number is maximum. If more than one node has the same number, choose the node which has maximal PMF.

As described above, a sender uses algorithm $E1$ or algorithm $E2$ depending on its current state. If the node only needs to disseminate information in one direction, it uses algorithm $E1$ and otherwise uses algorithm $E2$. The enhanced MPR selection algorithm evaluates nodes' MPR fitness based on two-hop neighbor information. The algorithm considers nodes' history and moving tendency in the MPR selection procedure therefore can use better nodes to relay messages. By selecting the relatively stable nodes, the algorithm also increases the probability of disseminating more than one packet using the same relay node. This feature helps broadcast protocol to reduce acknowledgement messages while ensuring reliability. The proposed broadcast protocol uses the enhanced MPR selection algorithm will be explained in the next section.

4.2.3. Effectiveness of the Proposed Algorithm

Due to dynamic features of VANETs, the original MPR selection algorithm [14] did not work well. To solve this problem, the enhanced MPR selection algorithm picks relay node considering mobility. Comparing to other mobile ad hoc networks, relay node selection in VANETs is relatively straightforward. Since lane width is much smaller than transmission range, a sender always needs to select only one forwarder in one direction. Taking advantage of this feature, algorithm $E1$ selects the best relay node.

In the case that senders need to disseminate messages in different direction, algorithm $E2$ can enhance the original MPR algorithm using mobility awareness. In this case, multiple nodes may have similar additional coverage, so choosing the best one is particularly important. However, the original MPR algorithm may select any of them. If the selected nodes havev the opposite direction to the sender, it would result in low dissemination speed or dissemination failure as described above. Algorithm $E2$ enhances the original MPR algorithm when there are multiple candidate nodes that have the same additional coverage range. Since considering the node mobility, the proposed algorithm ensures selecting relatively stable nodes to forward data. In general, the proposed algorithm could eliminate errors of the original *MPR* algorithm in imprecise topology information. In the worst case (in a static network), the proposed protocol performs same as the original *MPR* algorithm.

4.3. Protocol Design

4.3.1. Design Principles

The proposed multi-hop broadcast protocol uses enhanced *MPR* selection algorithm proposed in Section 4.2. The proposed protocol aims to ensure the strict reliability as well as the transmission overhead minimization. As for the strict reliability, the following scheme is used. A hop-by-hop manner is used to provide reliability. Every sender is responsible for assuring reliable broadcast to its one-hop downstream nodes. A sender includes a TO-ACK-LIST in a broadcast packet, and the nodes included in the TO-ACK-LIST reply ACK to the sender when the nodes receive the packet. Three types of acknowledgement methods (explicit ACK, implicit ACK and negative ACK) are used. While broadcasting a data packet, a sender starts a retransmission timer. A sender node maintains a TO-ACK-LIST locally to store nodes from which it has not heard ACK. The sender removes the corresponding node from the list upon reception of an ACK. If the local TO-ACK-LIST is not NULL and the retransmission timer expires, the sender retransmits the packet.

In order to reduce rebroadcast redundancy in high-density networks, the proposed protocol uses only a subset of nodes in the network to relay received broadcast packets. It is assumed that vehicles exchange their neighbor information through hello messages. Every vehicle places its one-hop neighbor information to a hello message and therefore vehicles know existence of their two-hop neighbors. Before broadcasting a packet, a sender uses the enhanced *MPR* selection algorithm to decide relay nodes based on two-hop neighbor information. This chapter describes these relay nodes as boundary nodes. A sender includes the list of its boundary nodes (BOUNDARY-LIST) in a broadcast packet. Upon receiving a broadcast packet, the nodes rebroadcasts the packet if they are included in the BOUNDARY-LIST.

In order to cope with the network topology change, the information, REVERSE-BOUNDARY-LIST, is attached to a broadcast packet. Before sending a broadcast packet, a sender appends their address to the REVERSE-BOUNDARY-LIST. Therefore, REVERSE-BOUNDARY-LIST of a packet is composed of the addresses of the nodes that have forwarded the packet. Every node also needs to maintain a unicast route table, which is used to send ACK. Upon receiving a broadcast packet, every node maintains route entries to nodes contained in the REVERSE-BOUNDARY-LIST. These routes use the sender node (the last node relayed the packet) as the next hop. They are used to deliver ACK to two-hop upstream sender in case the topology changes. The protocol uses these routes for mobility handling in a manner which will be explained later.

The proposed protocol intends to use relatively far nodes to relay packets because they can provide larger progress on distance. It should be noted that bit error rates in 802.11a/b/g/p are relatively high between far nodes than near nodes. However, in this chapter, it is assumed that bit error rates are unaffected by distance between sender node and relay node. The effect of the distance on the bit error rate will be considered in the future work.

4.3.2. Protocol Information and Acknowledgment Scheme

Every sender node maintains a broadcast cache, which consists of entries that include the following fields.

- **Source node address and broadcast ID**
- **TO-ACK-LIST:** This list consists of nodes that should acknowledge upon reception of the corresponding packet.
- **Expire time:** The time of the corresponding packet should be retransmitted in case the packet is not successfully received by all desired receivers.

- **Corresponding broadcast packet:** A copy of the data packet that can be used to retransmit.

A data packet includes the following fields in addition to data itself.

- **Source node address and broadcast ID**
- **BOUNDARY-LIST:** A list consisting of boundary nodes.
- **TO-ACK-LIST**
- **REVERSE-BOUNDARY-LIST:** A list consisting of nodes which have rebroadcasted this packet.
- **Consecutive broadcasting flag:** A flag shows whether this packet belongs to a consecutive broadcast or not.
- **Retransmit flag and retransmit source node address:** Retransmit flag shows whether this broadcast packet is a retransmitted packet or not. Retransmit source node address is the address of the node which initiates retransmission.

As mentioned above, the following three types of acknowledgement methods are used.

- **Explicit ACK:** An explicit ACK should include source node address, broadcast ID, and receiver's address (address of ACK sender).
- **Implicit ACK:** A rebroadcast packet is an implicit ACK to the sender's upstream node. Upon hearing the packet, the upstream node knows the packet has been successfully received by the downstream node.
- **Negative ACK (NACK):** A negative ACK should include all fields of explicit ACK. Upon reception of a NACK, the sender rebroadcasts the packet immediately.

4.3.3. Boundary Specification and TO-ACK-LIST Selection

The proposed protocol always selects nodes that provide larger progress on distance as the boundary nodes. The proposed protocol selects the boundary nodes using Enhanced Multipoint Relay selection algorithm proposed in section 4.2. Every node specifies the boundary nodes before broadcasting a message. In this way, redundant broadcasting can be efficiently reduced. If the sender is not the broadcast source, the BOUNDARY-LIST should not include the upstream node's boundary nodes and nodes which are included in REVERSE-BOUNDARY-LIST.

The broadcast source node's TO-ACK-LIST is simply defined as its one-hop neighbors. If the sender is not the broadcast source, its TO-ACK-LIST excludes nodes that included in the upstream node' TO-ACK-LIST. The TO-ACK-LIST also excludes nodes that included in the packet' REVERSE-BOUNDARY-LIST.

4.3.4. Packet Rebroadcasting

As shown in Figure 4.6, before broadcasting a packet, the source node does the following actions:

1. Update the broadcast cache. Set the expire time field according to delay constraint. Calculate an TO-ACK-LIST for two purposes: Firstly, to maintain locally for future retransmission checking. Secondly, to let downstream nodes know whether the packet should be acknowledged or not. Place TO-ACK-LIST to the data packet.
2. Select boundary nodes and place them to the data packet.
3. Place own address to the REVERSE-BOUNDARY-LIST.

As shown in Figure 4.7, upon receiving a broadcast packet, an intermediate node does following actions:

1. Create an reverse route to nodes included in the REVERSE-BOUNDARY-LIST for delivering ACK.

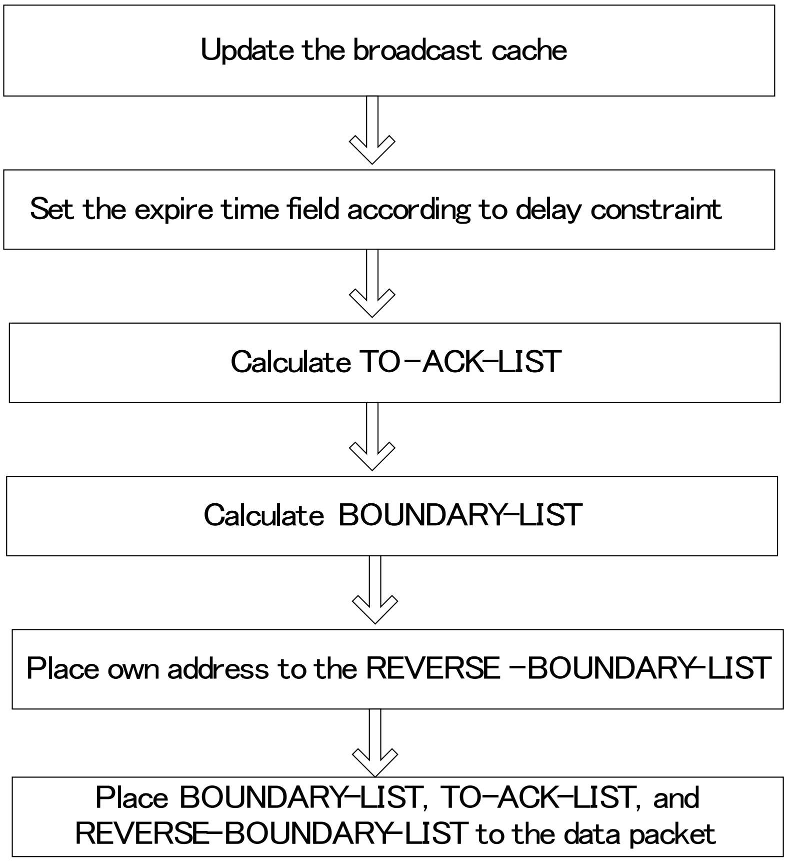


Figure 4.6 – Actions at the source node.

```
2. If (the BOUNDARY-LIST contains the node) then {
     Update the packet's BOUNDARY-LIST according to own neighbor information.
     Append own address to REVERSE-BOUNDARY-LIST.
     Update the broadcast cache. Set the expire time field according to delay constrain. Update the TO-ACK-LIST.
     Rebroadcast. (Rebroadcast is implicit ACK to the upstream node.)
   }else{
     If (TO-ACK-LIST contains the node) then {
       Send an ACK to the upstream node.
```

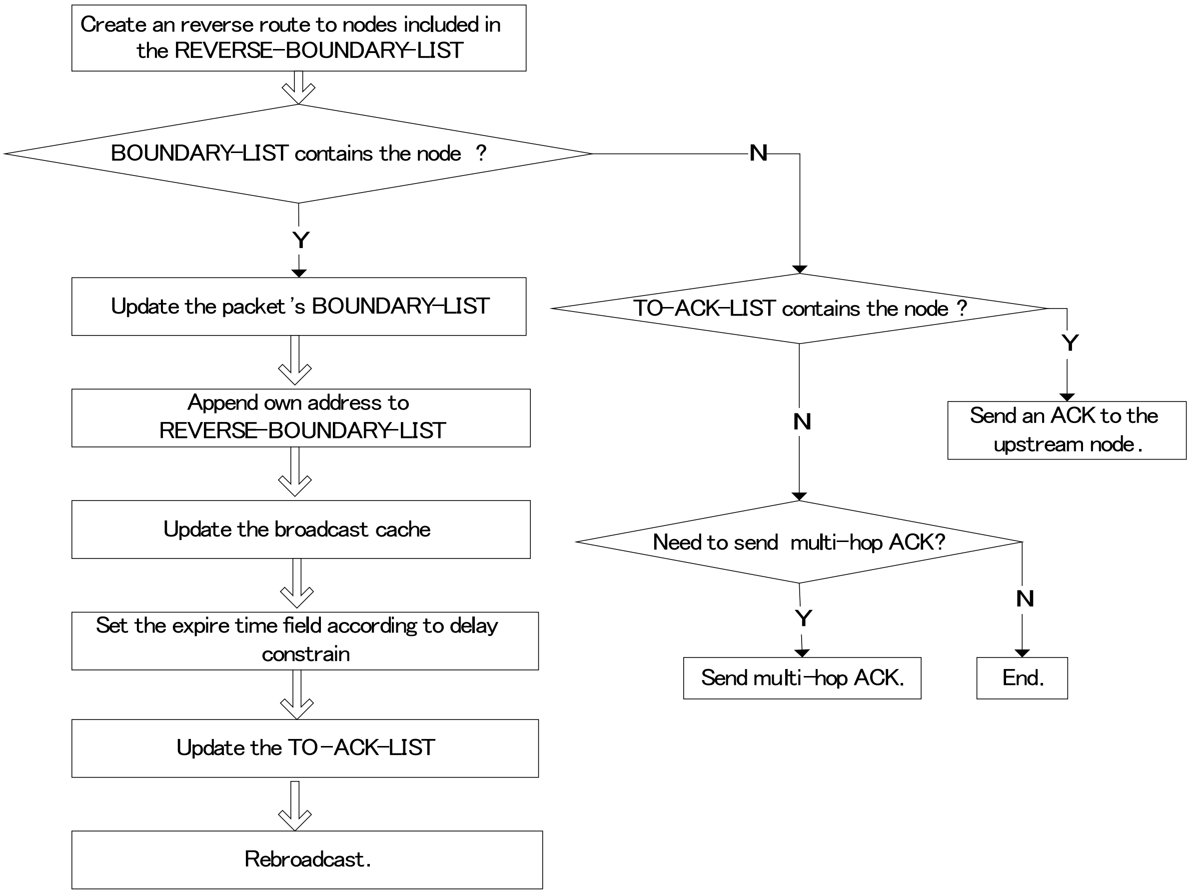


Figure 4.7 – Actions at intermediate nodes.

}**else** {

May send multi-hop ACK to nodes included in REVERSE-BOUNDARY-LIST. (This will be explained later in 4.3.6.)

}

}

As shown in Figure 4.8, upon receiving an ACK (or an implicit ACK), a node does the following actions:

1. According to the ACK's information, get the corresponding entry from broadcast cache.

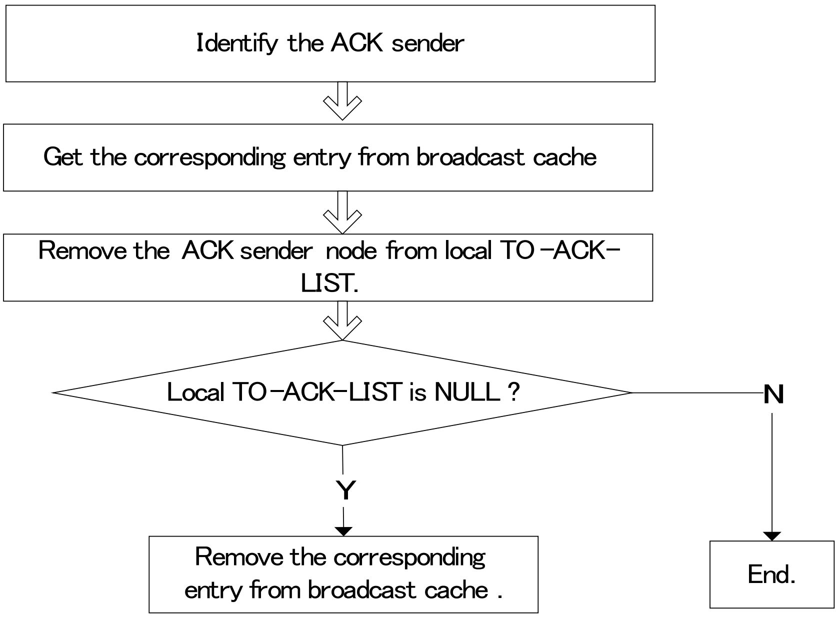
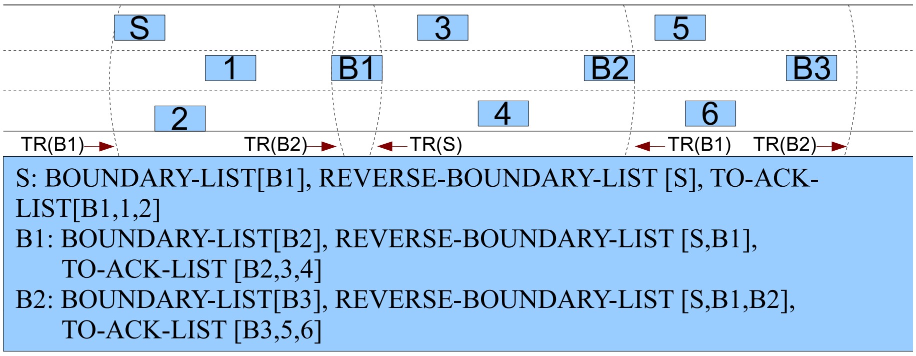


Figure 4.8 – Actions upon receiving an ACK.

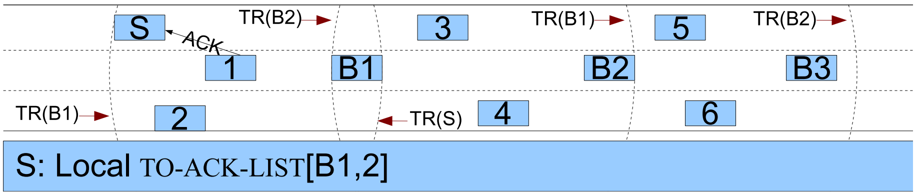
2. Remove the corresponding node (the sender of the ACK) from local TO-ACK-LIST. **If** the local TO-ACK-LIST is NULL **then** remove the corresponding entry from broadcast cache.

As shown in Figure 4.9, node S selects node B1 as boundary node and sets TO-ACK-LIST to [B1, 1, 2]. Node S also appends own address to REVERSE-BOUNDARY-LIST. S broadcasts the message and B1 knows itself is a boundary and then updates the packet's boundary nodes to [B2, 3, 4]. Similarly, B1 appends own address to REVERSE-BOUNDARY-LIST before relaying. When node 1 receives the message from S, it sends ACK to S (in Figure 4.10). But node 1 does not sends ACK to B1 when it receives the message from B1 because it is not specified to do so. Upon reception of the ACK from node 1, S will delete node 1 from local TO-ACK-LIST. As shown in Figure 4.11, if node S does not receive ACK from node 2 before retransmission timer expires, node S will retransmit the message.



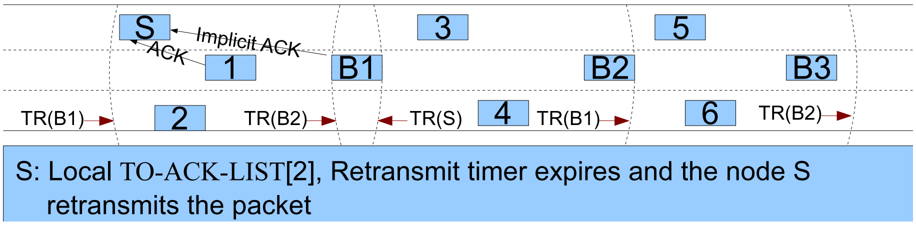
S: BOUNDARY-LIST[B1], REVERSE-BOUNDARY-LIST [S], TO-ACK-LIST[B1,1,2]
B1: BOUNDARY-LIST[B2], REVERSE-BOUNDARY-LIST [S,B1], TO-ACK-LIST [B2,3,4]
B2: BOUNDARY-LIST[B3], REVERSE-BOUNDARY-LIST [S,B1,B2], TO-ACK-LIST [B3,5,6]

Figure 4.9 – Boundary specification.



S: Local TO-ACK-LIST[B1,2]

Figure 4.10 – ACK management.



S: Local TO-ACK-LIST[2], Retransmit timer expires and the node S retransmits the packet

Figure 4.11 – Retransmission.

4.3.5. Retransmission Handling

Every node maintains a retransmission timer and performs retransmission check periodically. As shown in Figure 4.12, when the retransmission timer expires following action is executed.

1. **If** (exist expired broadcast cache entry) **then**{

   Get the corresponding packet and update the packet's retransmit source node address with own address.

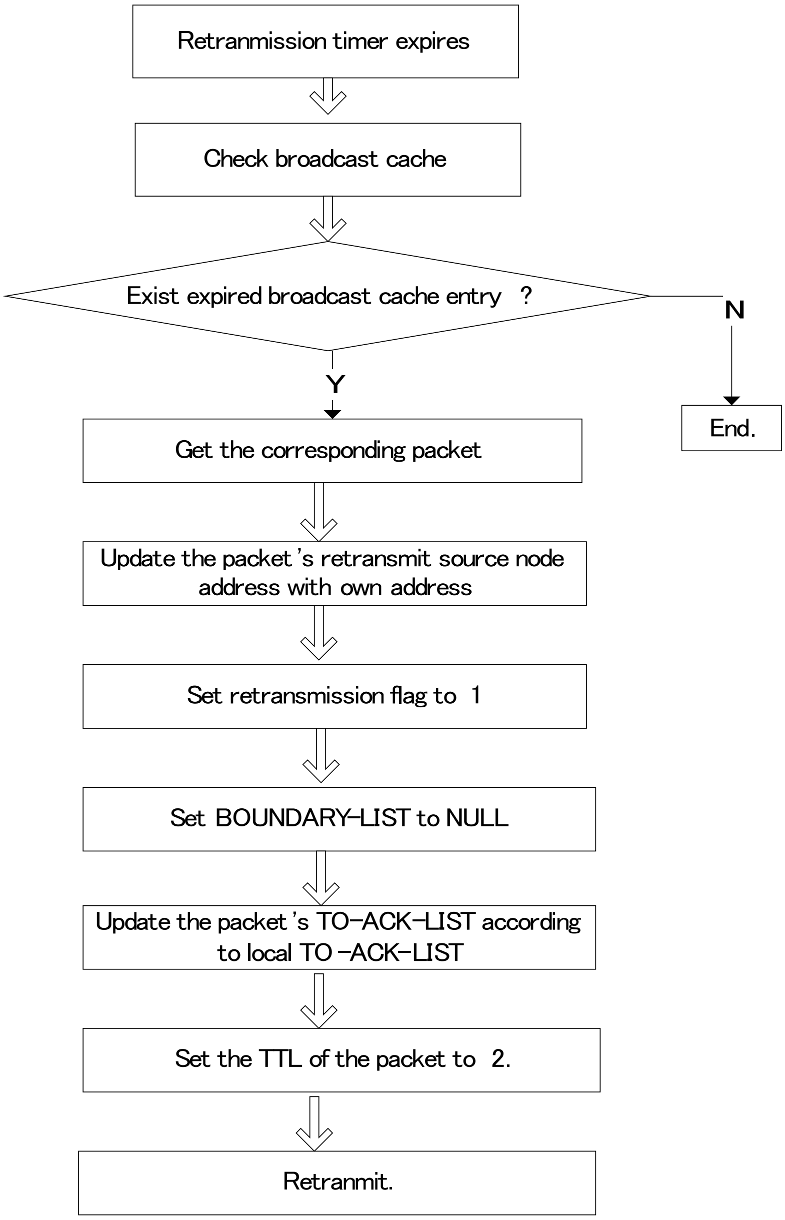


Figure 4.12 – Actions for retransmit handling.

Set retransmission flag to 1 and BOUNDARY-LIST to NULL.

Update the packet's TO-ACK-LIST according to local TO-ACK-LIST.

Set the TTL of the packet to 2. (In general, two-hop flooding is large enough. In case of still have missing receivers, increase TTL by 2 and retransmit.)

```
        Retransmit.
    }
```

Upon receiving a retransmitted data, a node checks if its own address is included in the TO-ACK-LIST of the packet. If so the node sends ACK to the retransmission source node. Otherwise, the node just rebroadcasts the packet.

4.3.6. Mobility Robustness

The proposed protocol also uses ACK messages to handle topology changes. ACK could be one hop or multi-hop. As shown in Figure 4.13(a), suppose that L1 did not receive data from S1 and has moved to new position, which is out of the transmission range of S1. Upon receiving the data from B1, checking the reverse boundary node list, L1 knows the packet has been broadcasted by S1. Since S1 is a neighbor of L1 (in the L1's knowledge), L1 should have received the packet before, but L1 did not receive the packet. This implicates some link changes have happened. Therefore, L1 sends ACK to node S1 although not specified by B1 to acknowledge. The ACK message could arrive at node S1 by the way of B1 and then S1 would know L1 is no longer a neighbor.

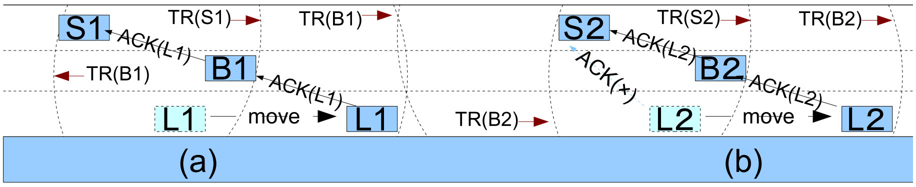


Figure 4.13 – Mobility robustness.

In case of another situation, which is shown in Figure 4.13(b), L2 received data from S2 and sent ACK back to S2. The ACK is lost and S2 retransmits the packet and B2 relays. Even if L2 moved out of the S2's transmission range it also can send ACK to S2 by the way of B2. Accordingly, S2 updates its neighbor information. As described above, vehicles update topology information while broadcasting data packet and therefore the proposed protocol can efficiently handle mobility.

4.3.7. Consecutive Broadcasting: ACK and NACK

In many situations, a sender needs to broadcast many packets to the same relay node. This is called consecutive broadcasting. Actually, the proposed *MPR* selection algorithm tends to use the same relay node if other metrics of candidate nodes are equal. This makes consecutive broadcasting possible without difficulty. In case of consecutive broadcasting, the proposed protocol uses negative ACK (NACK) to reduce the number of control messages. The sender sets the consecutive flag to 1 and sets the next packet arrival time. Every sender should recalculate the next packet arrival time according to packet generation interval, contention delay in MAC layer and propagation delay. The sender also needs to attach a consecutive sequence number (seq no) to the broadcast packet. The consecutive sequence number is used to let downstream nodes know whether this is the first packet of consecutive broadcasting or not. If the packet is not the first packet, TO-ACK-LIST is not required. For example, in case of 3 consecutive broadcasting, the fields of consecutive broadcasting for first, second and last packet are shown in Table 4.2.

Table 4.2 – An example of packets in case of 3 consecutive broadcasting.

packet	consecutive flag	seq no	next packet arrival time
First	1	1	0.5s
Second	1	2	0.6s
Last	0	3	0s

When a node receives a packet with consecutive flag equals one, the node records the next packet arrival time and starts a timer. If the packet is the first packet of consecutive broadcasting, the node sends an ACK to the upstream node and otherwise not. If the next packet did not arrive before expected arrival time, the node sends an NACK to source node and source node would retransmit the packet upon reception of the NACK. If the packet is the last packet of consecutive broadcasting, the sender sets the consecutive flag to 0 in order to notify receivers not to wait for the next packet.

Generally, if the sender node (at IP layer) can predict the next packet generation time, the consecutive broadcasting can be used. For example, applications that generate a constant rate stream can use the consecutive broadcasting. Since the generation rate is constant, the sender node can predict the generation time of the next packet. Another case that can use the consecutive broadcasting is when the application data size is larger than the maximum transmission unit. In this case application data is divided to multiple IP datagrams and thus the sender node is aware of the next packet scheduling time.

4.3.8. Boundary Selection Error Handling

A node may fail to select the boundary nodes. If nothing is selected, it might be because of following two reasons. One is because a rebroadcast cannot provide additional coverage. Another is because this node has insufficient two-hop neighbor information. In the first situation, the node rebroadcasts the packet with TTL equals one. It means every neighbor node can receive this packet, but will not rebroadcast. In the second situation, the node rebroadcast with NULL BOUNDARY-LIST. If a node receives a packet with NULL BOUNDARY-LIST, it rebroadcasts the packet.

4.4. Performance Evaluation

The proposed protocol reduces broadcast redundancy by means of a method in which only boundary nodes relay broadcast packets. Clearly, the protocol is effective in high-density networks. In sparse networks, the proposed protocol is resistant to channel loss because it incorporates a retransmission mechanism. In mobile scenarios, the proposed protocol can update topology information using ACKs without introducing too much overhead.

The proposed protocol uses a subset of neighbor nodes to forward a data packet. In order to check the reception status of all receivers, the proposed protocol uses explicit ACKs when they are required. The sizes of all fields in an ACK message can be seen in Table 4.3. In Table 4.3, Destination node address field is the address of the node this ACK should be sent to. ACK sender node address field is the address of the node that initiates the ACK. Upon reception of the ACK, a node can use ACK sender node address, Broadcast source node address and Broadcast ID to determine which node has received which packet.

Table 4.3 – Sizes of fields in ACK message.

Field	Size
Destination node address	4 bytes
Broadcast source node address	4 bytes
Broadcast ID	4 bytes
ACK sender node address	4 bytes

In the proposed protocol, if a node will not forward the broadcast packet, it sends an explicit ACK to the sender node and otherwise not. That is, the number of explicit ACKs used in the proposed protocol is determined by how many nodes do not forward the packet. It is easy to know that the number of packets used in the proposed protocol is the same to that of flooding. Also according to IEEE 802.11 standard [49], the broadcast frames shall not be fragmented even if their length exceeds the defined fragmentation threshold. Therefore, the number of MAC frames is also the same to that used in flooding.

As far as the MAC frame size is concerned, the ACK frame size used in the proposed protocol is smaller than the data frame size of flooding. However, since the proposed protocol attaches additional information to the broadcast data packet, the MAC data frame size in the proposed protocol can be larger than in flooding. Table 4.4 shows the sizes of additional information within the data packet. This raises a question of how the additional overhead affects the performance of the proposed protocol. The following two facts are used to explain that this overhead is well compensated by the advantages of the proposed protocol. First, in flooding, many packets are dropped because collisions incurred from all neighbors try to rebroadcast a packet at the same time. The proposed protocol efficiently reduces the number of rebroadcasts, so collisions can be avoided. Second, since the ACK frame size is much smaller than the data frame size, the overall overhead of the proposed protocol will always be lower than flooding.

The effect of the additional overhead is explained by an example, which is shown in Figure 4.14. In the figure, the source node S broadcasts a data packet and its one-hop neighbors relay the packet. For simplicity, only the overheads incurred at node S and its one-hop neighbors are considered here. However, the calculation given below can be easily extended to node S' two-hop neighbors and further.

In flooding, a data packet will be broadcasted by $k+1$ nodes (S, $n_1, \cdots, n_k$). In the proposed protocol, two nodes, S and n_k broadcast the data packet and other nodes ($n_1, \cdots, n_{k-1}$) send ACKs to node S. In both protocols, the number of total MAC frames will be $k+1$. However, the total frame sizes are different. In flooding, when the application data size is 512bytes, the MAC data frame size S_d will be $S_d = 512 + 20(\text{IPheader}) + 24(\text{MACheader}) + 4(\text{FCS}) = 560(\text{bytes})$. So, the total frame size will be $S_d \times (k+1)(\text{bytes})$ in flooding.

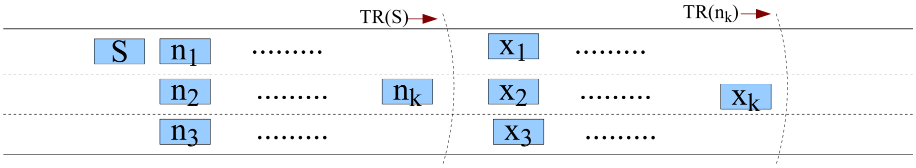


Figure 4.14 – An example for overhead analysis.

In the proposed protocol, MAC data frame size will be affected by the sizes of BOUNDARY-LIST, TO-ACK-LIST and REVERSE-BOUNDARY-LIST. Table 4.5 shows the sizes of those lists.

As Table 4.5 shows, MAC frame size of the data packet sent by node S (S_{d1}) will be $S_{d1} = S_d + 13 + 4 + 4 \times k + 4 = 581 + 4 \times k(\text{bytes})$ where 13 is the total size of fixed length fields which includes Source node address, Broadcast ID, Consecutive

Table 4.4 – Sizes of additional information within the data packet.

Field	Size
Source node address	4 bytes
Broadcast ID	4 bytes
BOUNDARY-LIST	4 bytes × list size
TO-ACK-LIST	4 bytes × list size
REVERSE-BOUNDARY-LIST	4 bytes × list size
Consecutive broadcasting flag	1 bit
Retransmit flag	1 bit
Retransmit source node address	4 bytes

broadcasting flag, Retransmit flag and Retransmit Source node address. In above calculation, 4, $4 \times k$ and 4 are the sizes of BOUNDARY-LIST, TO-ACK-LIST and REVERSE-BOUNDARY-LIST respectively. MAC frame size of the data packet sent by node n_k (S_{d2}) will be $S_{d2} = S_d + 13 + 4 + 4 \times k + 8 = 585 + 4 \times k(\text{bytes})$. Similarly, ACK frame size (S_A) will be $S_A = 16 + 20(\text{IPheader}) + 24(\text{MACheader}) + 4(\text{FCS}) = 64(\text{bytes})$. So the total MAC frame size in the proposed protocol will be $S_{d1} + S_{d2} + S_A \times k = 581 + 4 \times k + 585 + 4 \times k + 64 \times (k - 1) = 1102 + 72 \times k(\text{bytes})$. When k is 2, total MAC frame size of the flooding will be 1,680 bytes and total MAC frame size of the proposed protocol is 1,246 bytes. When k is 32, total MAC frame size of the flooding is 17,920 bytes and total MAC frame size of the proposed protocol will be 3,406 bytes. Therefore, total overhead of the proposed protocol is lower than flooding, especially in the high-density networks.

The above-given descriptions show that the proposed protocol always has lower overhead than flooding even in the extreme situation of one source only sending one packet. In the case where broadcast sources have more than one packet to broadcast consecutively (consecutive broadcasting), the proposed protocol benefits from a negative ACK mechanism. Since explicit ACK is only required in the reception of the first packet from the sender, the protocol's overhead decreases notably.

In order to validate above analysis and further evaluate the proposed protocol's performance, simulations were conducted with ns-2. It is assumed that every node has a transmission range of 250m. Omnidirectional antennas and TwoRayGround propagation model are used. IEEE 802.11 MAC [49] and 512 bytes sized data packets have been used. Other simulation parameters use default setting of ns2.28.

Table 4.5 – Sizes of lists in data packets.

Sender	Field	Items	Size
S	BOUNDARY-LIST	$[n_k]$	4 bytes
	TO-ACK-LIST	$[n_1, \cdots, n_k]$	$4 \times k$ bytes
	REVERSE-BOUNDARY-LIST	[S]	4 bytes
n_k	BOUNDARY-LIST	$[x_k]$	4 bytes
	TO-ACK-LIST	$[x_1, \cdots, x_k]$	$4 \times k$ bytes
	REVERSE-BOUNDARY-LIST	[S, n_k]	8 bytes

In order to capture the realistic character of vehicles' movements to the simulation, Mobility Generator described in [43] is used. A freeway that has four lanes in two different directions is used. All lanes of the freeway are 2000m in length. Maximum velocity is 50m/s and every vehicle accelerates at the rate of ten percent of the maximum allowable velocity if there are no other vehicles ahead of it. In the proposed protocol, the broadcast source node uses algorithm $E2$ and other forwarder nodes use algorithm $E1$ to relay data packets. The sizes of ACK and additional information used in the proposed protocol can be seen in Table 4.3 and Table 4.4. All data presented in this paper are the average value of simulations repeated 10 times with different node movements.

4.4.1. Performance of the Enhanced MPR Selection Algorithm

The effectiveness of proposed MPR selection algorithm is evaluated. It is possible that selected $MPRs$ fail to receive the broadcast data because of vehicles' movements. Figure 4.15 shows the success ratio of original MPR selection algorithm [14] and the proposed enhanced MPR selection algorithm for various maximum velocities. Since two-hop neighbor information is updated on reception of hello messages, two different hello intervals of 0.5s and 1s are considered. 200 nodes are used to acquire enough mobility. In order to evaluate the effect of the enhanced MPR selection more correctly, the proposed retransmission handling mechanism is not used in this simulation.

From simulation results, it is observed that the original MPR selection algorithm's success ratio decreases drastically with increasing node velocity especially in 1s hello interval. This is because the original MPR selection algorithm selects the nodes moving toward different direction as MPR nodes. Those nodes always fail to relay packets successfully because of the vehicles' movements. However, as a result of including mobility prediction in the MPR selection procedure, the enhanced MPR selects relatively stable nodes. Therefore the proposed protocol can achieve high success ratio regardless of node velocity and hello interval. Simulation results confirm that it is important to

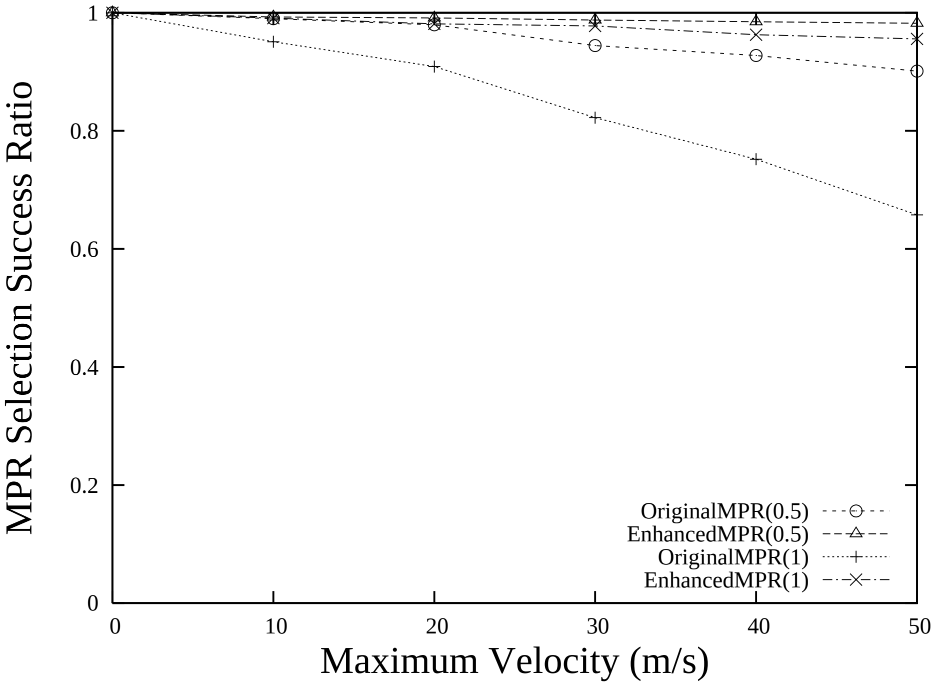


Figure 4.15 – Success ratio for various maximum velocities.

consider vehicles' mobility in *MPR* selection.

4.4.2. Effect of Node Density

In order to evaluate the effect of node density with the proposed protocol, various numbers of nodes ranging from 100 to 500 are used. The proposed protocol is compared with the flooding and other three VANET broadcast protocols (weighted p-persistence, slotted 1-persistence and slotted p-persistence with four slots) proposed in Ref. [3]. Weighted p-persistence, slotted 1-persistence and slotted p-persistence scheme are used because they are efficient and recent broadcast suppression techniques in VANET.

As for flooding, Figure 4.16 shows that delivery ratio decreases drastically with increasing node density. This is due to the broadcast storm problem of flooding. There is a high probability that many nodes that very close to the sender node try to rebroadcast. Therefore, many collisions occur because of the lack of RTS/CTS.

The weighted p-persistence, slotted 1-persistence and slotted p-persistence achieve better performance than flooding in a high-density network due to the reduction of

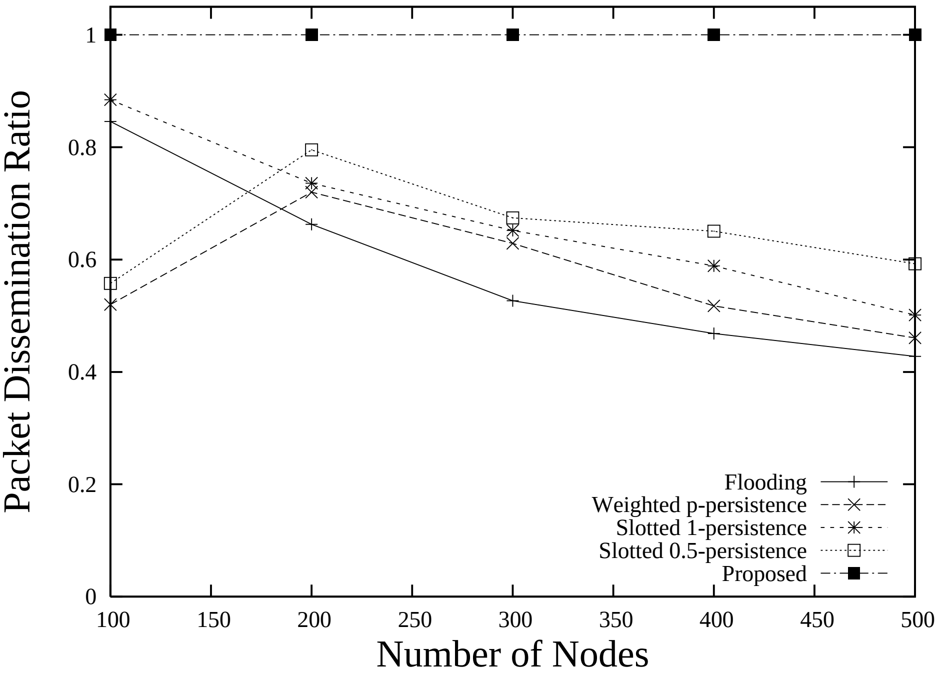


Figure 4.16 – Packet dissemination ratio for various node density.

rebroadcasts. Figure 4.17 shows the number of broadcasts of the protocols. It can be observed that the proposed protocol is more efficient than other protocols. In the proposed protocol, the sender node specifies the boundary nodes and only boundary nodes rebroadcast. The proposed protocol benefits from unicast ACKs and retransmission mechanism and therefore can acquire one hundred percent delivery ratio.

For protocol overhead, at the worst case of no consecutive broadcasting, the proposed protocol's total packet number is near to that of flooding. However, ACKs are smaller than data packets. Hence, total overhead of the proposed protocol is lower than flooding. MAC overhead comparison of the protocols is shown in Figure 4.18. In Figure 4.18, *Proposed* means the proposed protocol with no consecutive broadcasting. *Proposed*(3) denotes senders utilizing consecutive broadcasting for every 3 packets and *Proposed*(10) denotes the senders utilizing consecutive broadcasting for every 10 packets. In this chapter, MAC overhead is simply calculated as the number of sent or received MAC layer frames.

It can be observed from Figure 4.18 that the proposed protocol performs lower MAC

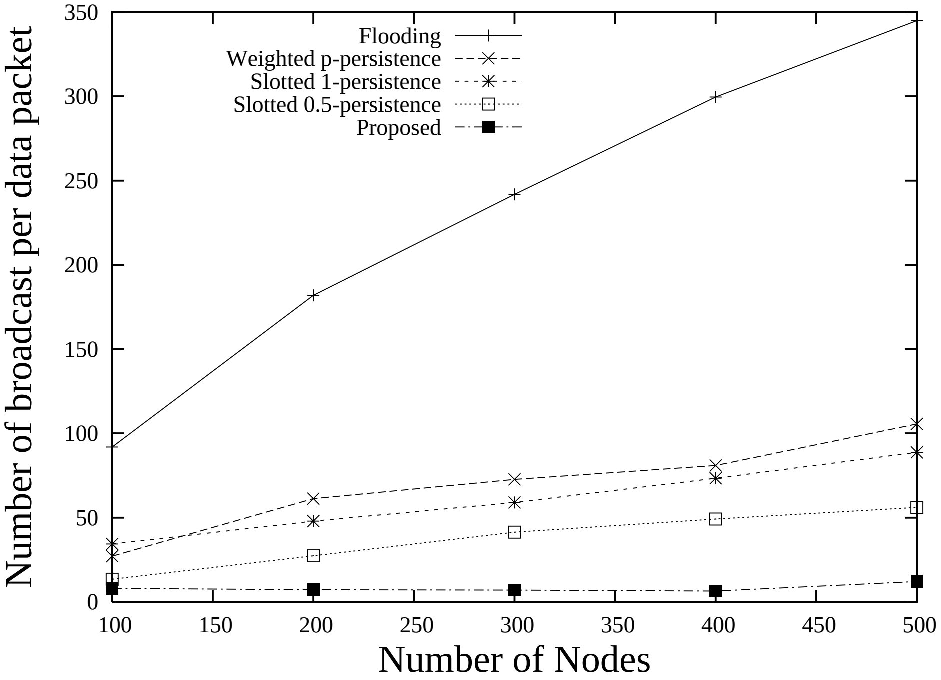


Figure 4.17 – Number of broadcast for various node density.

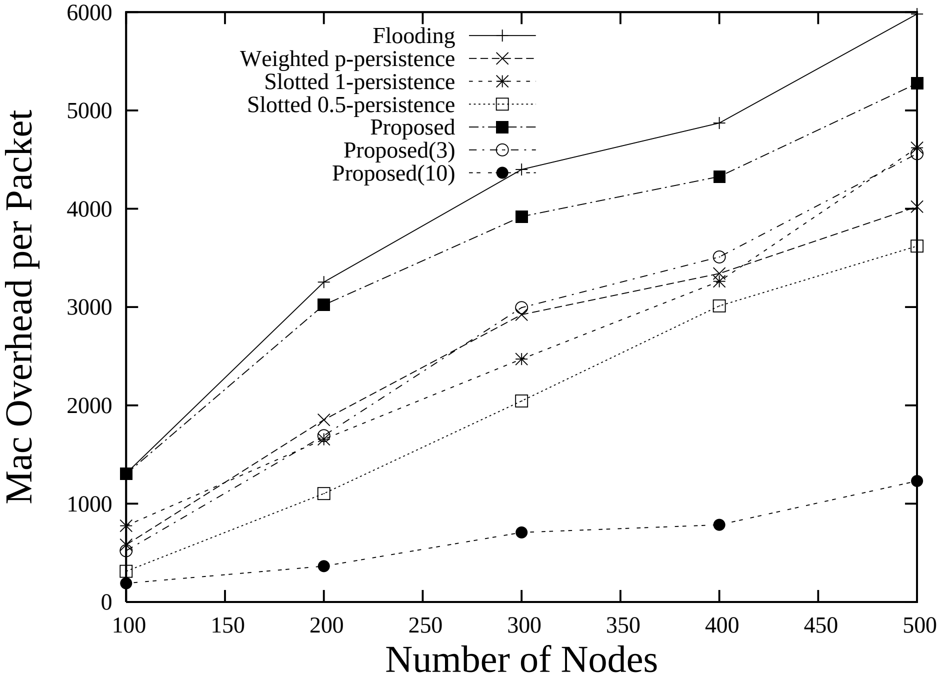


Figure 4.18 – MAC overhead per packet for various node density.

overhead than flooding. The saved rebroadcasts (Figure 4.17) in the proposed protocol can explain this effect. In the case of no consecutive broadcasting, the proposed protocol has higher MAC overhead than the weighted p-persistence, slotted 1-persistence and slotted p-persistence scheme due to ACK messages and retransmission mechanism. In case of the consecutive broadcasting, the receivers only need to explicitly acknowledge the first packet. Therefore, the proposed protocol shows notably lower overhead. Although the proposed method includes ACK messages to improve transmission reliability, this does not significantly increase overhead because those messages are sent unicast. In short, the proposed protocol can significantly improve reliability while keeping MAC overhead at an acceptable level.

4.4.3. Performance over Sparse Networks

A novel VANET broadcast protocol also should work well in sparse networks. Flooding may be considered as an acceptable broadcast scheme in sparse networks. Interestingly, it is observed that if the packet transmission rate is high enough, the simple flooding will be confronted with large number of collisions even in the sparse networks. A sparsely connected single lane network (Figure 4.19) is generated to simulate this effect. There are 10 nodes distributed in a chain manner. Besides the first node and the last node, every node has two neighbors. The first node and the last node have only one neighbor. The distance between two neighbor nodes is 200m. Because the transmission range is 250m, this network is fully connected. Simulation results are plotted in Figure 4.20.

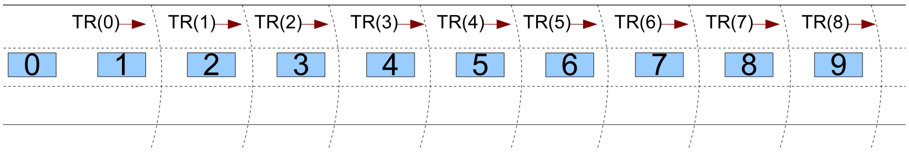


Figure 4.19 – Topology of the sparse Network.

It can be observed that flooding performs poorly when the packet generation rate is high. In the slotted 1-persistence scheme, a node rebroadcasts with probability 1 at the assigned time slot. Hence, the slotted 1-persistence scheme works similar to the flooding in sparse networks. In this simulation, every two neighbor vehicles' distance

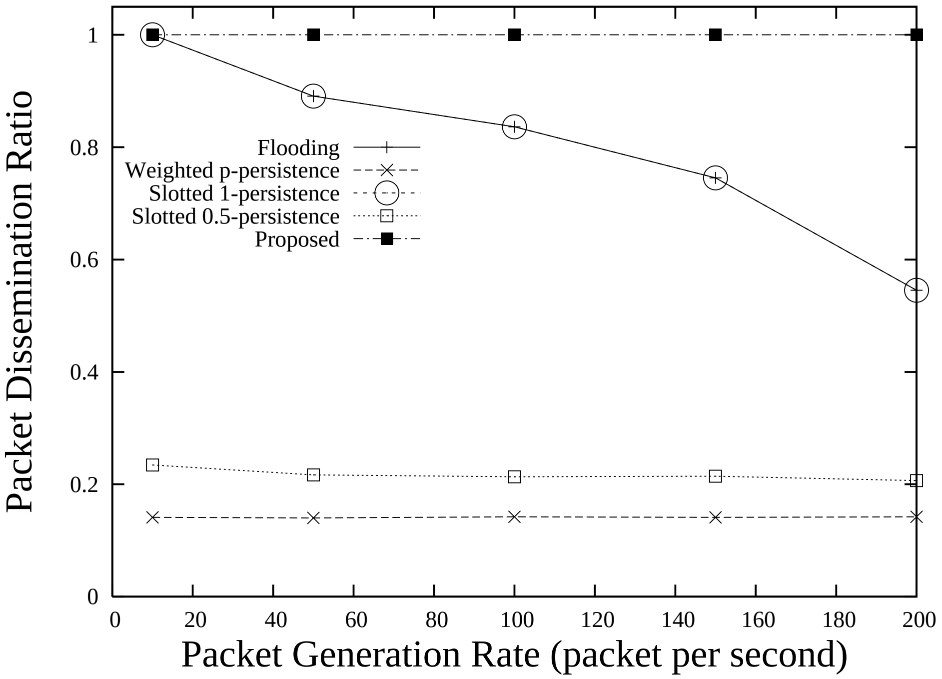


Figure 4.20 – Packet dissemination ratio for various packet generation rate.

is near the transmission range. A node rebroadcasts the packet immediately after the reception of the data packet. Hence, the slotted 1-persistence scheme works exactly same to the flooding. As for the weighted p-persistence and slotted 1-persistence, they behave poor performance in a sparse network because of probabilistic broadcasting. However, thanks to retransmission mechanism, the proposed protocol can achieve one hundred percent delivery ratio.

The protocols' performance over different channel loss rate is simulated. A low data rate of ten packets per second is used. If there is no loss in the wireless channel, the flooding can achieve perfect delivery ratio. However, in the loss channel, as shown in Figure 4.21, the flooding cannot achieve enough penetration. It is obvious that retransmission is required in sparse networks and the proposed protocol benefits from doing so.

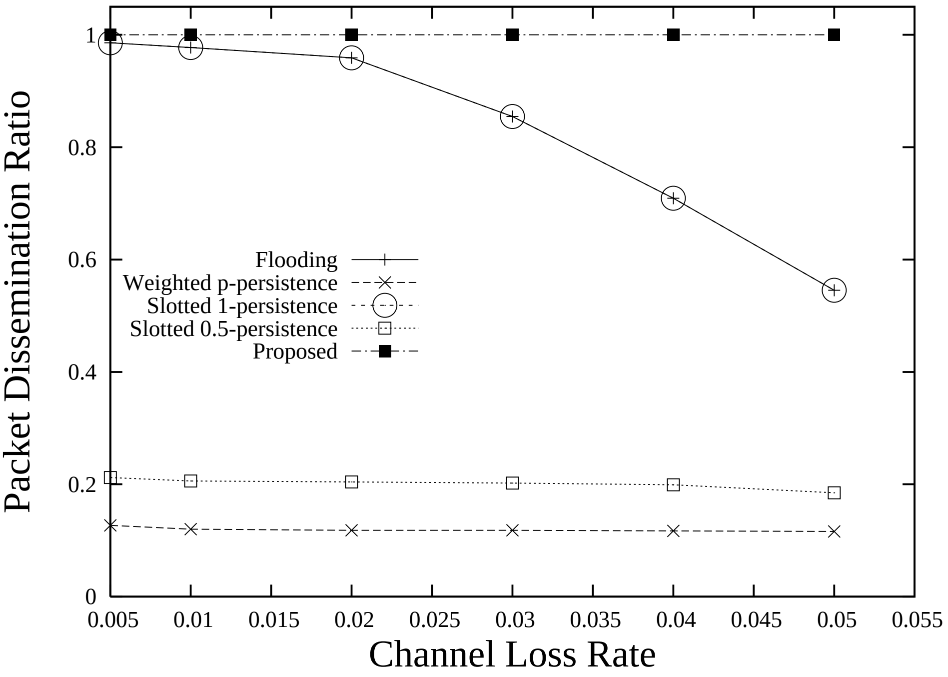


Figure 4.21 – Packet dissemination ratio for various channel loss rate.

4.4.4. Delay

Dissemination delay is an important metric to evaluate a broadcast protocol's performance. The messages should be delivered to intended receivers within the given time. However, flooding cannot disseminate messages quickly enough because of too many redundant rebroadcast. In this simulation, 400 nodes are used. The delay comparison of the protocols is shown in Figure 4.22.

It can be observed that the proposed protocol achieves lowest delay because of the following reasons. The proposed protocol uses boundary nodes to rebroadcast the packets. Consequently, the proposed protocol reduces the number of hops to the desired receivers. The proposed protocol also reduces the number of rebroadcasts and therefore results decreasing contention time.

In flooding, however, the nodes that provide larger progress on distance possibly lose the data packets due to packet collisions. As a result, the packets are delayed because they are delivered through suboptimal paths (longer paths). For the weighted

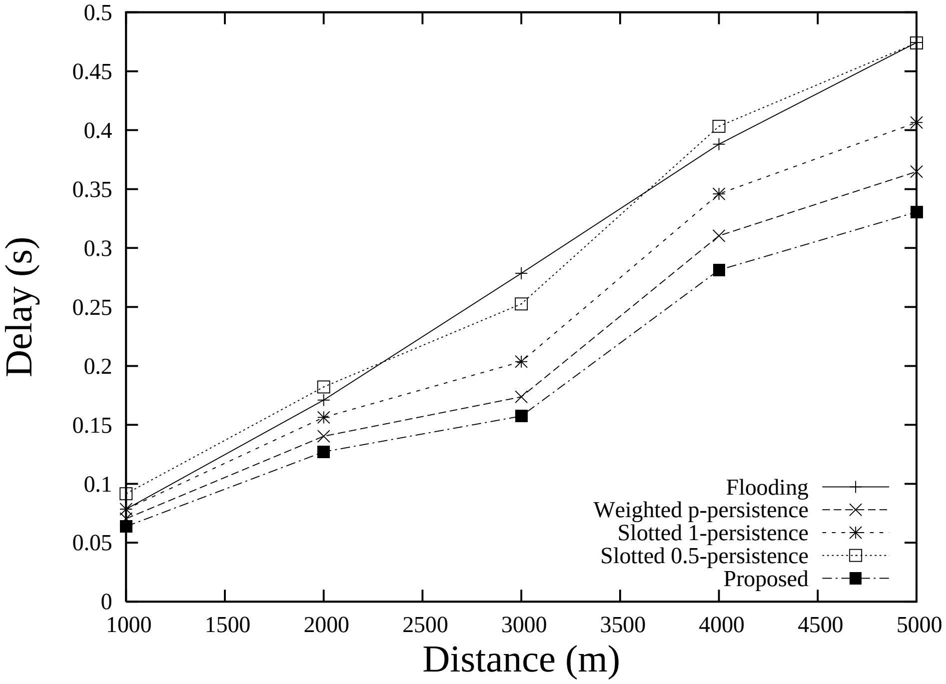


Figure 4.22 – Delay for various distances.

p-persistence, slotted 1-persistence and slotted p-persistence schemes, it can be observed that they perform with acceptable delays. But, they significantly suffer from long delays in a sparse network, due to the scheduling and waiting time required before rebroadcasting [3].

4.5. Conclusions

Reliability is the most important issue in vehicular safety message dissemination. This chapter has proposed a multi-hop broadcast protocol, which can ensure strict reliability. The proposed protocol uses an efficient acknowledgement method to detect whether all desired receivers have received the packet. To mitigate broadcast storms, the proposed protocol uses boundary nodes to relay data packets. In the boundary node selection, the proposed protocol uses an enhanced *MPR* algorithm, which is also proposed in this chapter.

Simulations were used to further evaluate the protocol's performance. Simulation results confirmed that the proposed protocol has notable performance improvement in various traffic conditions compared to other broadcast methods. In summary, the proposed protocol provides an efficient reliable broadcast solution to disseminate safety messages in vehicular ad hoc networks.

Chapter 5

Conclusions and Future Works

5.1. Conclusions

This thesis has discussed requirements of routing protocols for efficient and reliable data transfers in vehicular ad hoc networks. Communications in vehicular ad hoc networks can be point-to-point communications or broadcast communications. As a result, different types of communications have different demands on the network layer.

In point-to-point communications, general-purpose ad hoc routing protocols such as AODV cannot work efficiently due to frequent changes of network topology caused by vehicle's movement. Thus, the routing problem of finding reliable paths from a traffic source to a traffic destination through a series of intermediate forwarding nodes is particularly challenging. Another requirement of this kind of applications is that the routing protocols should efficiently utilize the limited bandwidth resources. To fulfill these requirements, the thesis has proposed a VANET routing protocol QLAODV (Q-Learning AODV) that fits for unicast applications in high mobility scenarios. QLAODV is a distributed reinforcement learning routing protocol, which uses a Q-Learning algorithm to infer network state information and uses unicast control packets to check the path availability in a real time manner in order to allow Q-Learning to work efficiently in highly dynamic network environment. QLAODV is favored by its dynamic route change mechanism and therefore is capable of reacting quickly to network topology changes.

In broadcast applications, the reliability is the most important issue. To provide the reliability, a good broadcast protocol also should efficiently reduce the broadcast redundancy. This is because an inefficient broadcast scheme could saturate the wireless resource and consequently cannot provide efficiency. To disseminate safety messages, the thesis proposes a reliable and efficient multi-hop broadcast routing protocol for vehicular ad hoc networks. The proposed protocol provides the strict reliability in various traffic conditions. This protocol also performs low overhead by means of reducing rebroadcast redundancy in a high-density network environment. Since the proposed protocol uses an enhanced multipoint relay (MPR) selection algorithm that considers vehicles' mobility to select relay nodes, it is robust to network mobility. Due to its low delay, high reliability and low overhead, the proposed protocol can be a reliable and efficient data transfer solution to the broadcast applications in vehicular ad hoc networks.

5.2. Future Works

Although plenty of works have been done for efficient and reliable data transfers in mobile ad hoc networks and vehicular ad hoc networks from different aspects, there is still a lot of unsolved issues. To make data transfers in vehicular ad hoc networks more efficient and reliable, future works will consider the following issues.

1. Estimating the link quality is very important to develop an efficient and reliable point-to-point data transfer solution. When selecting a route, QLAODV considers bandwidth efficiency, hop count and vehicle movement. The author is currently working on a more accurate link quality estimation based point-to-point data deliver protocol. In this protocol, a more realistic physical layer model which includes fading will be considered.

2. Future work on an accurate admission control in vehicular ad hoc networks is of great interest. A network should reserve enough bandwidth for vehicular safety application to ensure safety related broadcast messages could be disseminated quickly to desired receivers in case of an accident. A new flow also should not violate previously made guarantees. Therefore, an admission control is a major task to develop a good point-to-point data transfer protocol. However, it is a very challenging work to make an accurate admission control in vehicular ad hoc networks due to node mobility and dynamic network link quality incurred from fading and collisions. In the future work, an accurate admission control scheme for vehicular ad hoc networks considering mobility and dynamic link quality will be considered.

3. Group communications, i.e. video conferencing and on-line gaming, can be possible applications in vehicular ad hoc networks. The difference of these applications from conventional broadcast applications is that the messages only need to be disseminated in the groups. Therefore, a multicast protocol can be used to deliver data more efficiently. Using multicast data delivery, secrete data transfer can be possible, which normal broadcast protocols do not consider. It is thus efficient and reliable multicast also can be an interesting research topic in vehicular ad hoc networks.

4. This thesis has proposed a reliable and efficient multi hop broadcast protocol based on general IEEE 802.11 MAC layer. It might be interesting if the reliability and efficiency issues are considered in a cross layer cooperative manner. Future work on a cross layer broadcast protocol, which efficiently incorporates the physical layer, MAC layer and routing layer, would be of practical significance.

Bibliography

[1] S. Chaabouni, M. Frikha, and M. Meincke, “Traffic models for inter-vehicle communications,” Proc. Second International Conf. on Inf. and Commun. Technologies, Damascus, Syria, pp.773-778, Apr. 2006.

[2] Y. Toor, P. Muhlethaler, A. Laouiti, “Vehicle Ad Hoc networks: applications and related technical issues,” IEEE Communications Surveys & Tutorials, vol.10, no.3, pp.74-88, 2008.

[3] N. Wisitpongphan, and K.O. Tonguz, “Broadcast Storm Mitigation Techniques in Vehicular Ad Hoc Networks,” IEEE Wireless Communications, vol.14, no.6, pp.84-94, 2007.

[4] V. Namboodiri, and L. Gao, “Prediction-Based Routing for Vehicular Ad Hoc Networks,” IEEE Trans. on Vehicular Technology, vol.56, no.4, pp.2332-2345, 2007.

[5] T. Taleb, E. Sakhaee, A. Jamalipour, K. Hashimoto, N. Kato, and Y. Nemoto, “A Stable Routing Protocol to Support ITS Services in VANET Networks,” IEEE Trans. on Vehicular Technology, vol.56, no.6, pp.3337-3347, November 2007.

[6] Q. Yang, A. Lim, and P. Agrawal, “Connectivity Aware Routing in Vehicular Networks,” Proc. IEEE Wireless Commun. and Networking Conf., Las Vegas, USA, pp.2218-2223, Apr. 2008.

[7] B. Ducourthial, Y. Khaled, and M. Shawky, “Conditional Transmissions: Performance Study of a New Communication Strategy in VANET,” IEEE Trans. on Vehicular Technology, vol.56, no.6, pp.3348-3357, November 2007.

[8] G. Lu, G. Manson, and D. Belis, “Enhancing routing performance for inter-vehicle communication in city environment,” Proc. ACM international workshop on Performance monitoring, measurement, and evaluation of heterogeneous wireless and wired networks, Torremolinos, Spain, pp.82-89, 2006.

[9] H. Menouar, M. Lenardi, and F. Filali, “An Intelligent Movement-based Routing for VANETs,” ITS World Congress 2006, London, United Kingdom, Oct. 2006.

[10] W. Wang, F. Xie, and M. Chatterjee, “TOPO: Routing in Large Scale Vehicular Networks,” Proc. 66th IEEE Vehicular Technology Conf., Baltimore, USA, pp.2106-2110, Oct. 2007.

[11] O. Abedi, M. Fathy, and J. Taghiloo, “Enhancing AODV routing protocol using mobility parameters in VANET,” Proc. IEEE/ACS International Conf. on Computer Systems and Applications, Doha, Qatar, pp.229-235, Apr. 2008.

[12] L.P. Kaelbling, M.L. Littman, and A.W.Moore, “Reinforcement learning: A survey,” Journal of Articial Intelligence Research, vol.4, pp.237-285, 1996.

[13] C. Watkins, “Learning from Delayed Rewards,” PhD thesis, King’s College, Cambridge, 1989.

[14] A. Qayyum, L. Viennot, and A. Laouiti, “Multipoint Relaying for Flooding Broadcast Messages in Mobile Wireless Networks,” Proc. 35th Annual Hawaii Intl. Conf. on System Sciences, Big Island, Hawaii, pp.3866-3875, 2002.

[15] F. Li, and Y. Wang, “Routing in vehicular ad hoc networks: A survey,” IEEE Trans. on Vehicular Technology, vol.2, no.3, pp.12-22, June 2007.

[16] M. Mauve, A. Widmer, and H. Hartenstein, “A survey on position-based routing in mobile ad hoc networks,” IEEE Network Magazine, vol.15, no.6, pp.30-39, December 2001.

[17] R. Jain, A. Puri, and R. Sengupta, “Geographical routing using partial information for wireless ad hoc networks,” IEEE Personal Commun., vol.8, no.1, pp.48-57, February 2001.

[18] B. Karp, and H.T. Kung. “GPSR: Greedy perimeter stateless routing for wireless networks,” Proc. 6th IEEE International Conf. on Universal Personal Communications, pp.562-566, 1997.

[19] S.R. Das, R. Castaneda, J. Yan, “Simulation-based performance evaluation of routing protocols for mobile ad hoc networks,” Mobile Networks and Applications, vol.5, no.3, pp.179-189, 2000.

[20] T. Clausen, and P. Jacquet, “Optimized Link State Routing Protocol (OLSR),” RFC 3626, October, 2003.

[21] C. Perkins, and E. Belding-Royer, “Ad hoc On-Demand Distance Vector (AODV) Routing,” RFC 3561, Jul. 2003.

[22] Z. J. Haas, “A new routing protocol for the reconfigurable wireless networks,” Proc. 6th Annual International Conf. on Mobile and Ubiquitous Systems, San Jose, USA, pp.1-8, Jul. 2006.

[23] M. Zhaomin, Z. Hao, M. Kia, and N.Pissinou, “MURU: A multi-hop routing protocol for urban vehicular ad hoc networks,” Proc. 3rd Annual International Conf. on Mobile and Ubiquitous Systems, San Jose, USA, pp.1-8, Jul. 2006.

[24] O. Tonguz, N. Wisitpongphan, F. Bai, P. Mudalige, and V. Sadekar, “Broadcasting in VANET,” Proc. Mobile Networking for Vehicular Environments, pp.7-12, 2007.

[25] Y. Liu, F.Y. Li, and H.-P. Schwefel, “Reliable Broadcast in Error-Prone Multi-hop Wireless Networks: Algorithms and Evaluation,” Proc. IEEE Global Telecommunications Conf., Washington, USA, pp.5329-5334, 2007.

[26] H. Jiang, H. Guo, and L. Chen, “Reliable and Efficient Alarm Message Routing in VANET,” Proc. 28th Intl. Conf. on Distributed Computing Systems Workshops, Beijing, China, pp.186-91, 2008.

[27] S. Khakbaz, and M. Fathy, “A Reliable Method for Disseminating Safety Information in Vehicular Ad Hoc Networks Considering Fragmentation Problem,” Proc. Fourth Intl. Conf. on Wireless and Mobile Communications, Athens, Greece, pp.25-30, 2008.

[28] B. Blaszczyszyn, A. Laouiti, P. Muhlethaler, and Y. Toor, “Opportunistic Broadcast in VANETs (OB-VAN) Using Active Signaling for Relays Selection,” Proc. 8th Intl. Conf. on ITS Telecommunications, Phuket, Thailand, pp.384-389, 2008.

[29] W. Usaha and J.A. Barria, “Reinforcement learning for resource allocation in LEO satellite networks,” IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics, vol.37, no.3, pp.515-527, 2007.

[30] W. Usaha and J. Barria, “A Reinforcement Learning Ticket-based Probing Path Discovery Scheme for MANETs,” Ad Hoc Networks, vol.2, no.3, pp.319-334, 2004.

[31] W. Usaha and J.A. Barria, “QoS Routing in MANET with Imprecise Information Using Actor-Critic Reinforcement Learning,” Proc. IEEE Wireless Communications and Networking Conference, Kowloon, China, pp.3382-3387, 2007.

[32] C. Gaskett, D. Wettergreen, A. Zelinsky and E. Zelinsky, “Q-Learning in Continuous State and Action Spaces,” Proc. Australian Joint Conference on Artificial Intelligence, Sydney, Australia, pp.417-428, 1999.

[33] C. Liang, and H. Wang, “An ad hoc on-demand routing protocol with high packet delivery fraction,” Proc. IEEE International Conf. on Mobile Ad-hoc and Sensor Systems, Fort Lauderdale, USA, pp.594-596, Oct. 2004.

[34] M. Quwaider, J. Rao, and S. Biswas, “Neighborhood Route Diffusion for Packet Salvaging in Networks with High Mobility,” Proc. IEEE International Conf. on Performance, Computing and Communications, Austin, USA, pp.168-175, Dec. 2008.

[35] J.A. Boyan, and M.L. Littman, “Packet routing in dynamically changing networks: A reinforcement learning Approach,” Advances in Neural Information Processing Systems, vol.6, pp.671-678, 1993.

[36] Y.H. Chang, T. Ho, and L.P. Kaelbling, “Mobilized ad-hoc networks: A reinforcement learning approach,” Proc. First International Conf. on Autonomic Computing, New York, USA, pp.240-247, May 2004.

[37] J. Dowling, E. Curran, R. Cunningham, and V. Cahill, “Using feedback in collaborative reinforcement learning to adaptively optimize MANET routing,” IEEE Trans. on Systems, Man and Cybernetics, vol.35, no.3, pp.360-372, May 2005.

[38] Y. Wang, M. Martonosi and L.-S. Peh, “Predicting link quality using supervised learning in wireless sensor networks,” ACM SIGMOBILE Mobile Computing and Communications Review, vol.11, no.3, pp.71-83, 2007.

[39] M. Caleffi and L. Paura, “Bio-inspired Link Quality Estimation for Wireless Mesh Networks,” in Proc. the 10th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, Kos, Greece, June. 2009.

[40] R.D. Renesse, M. Ghassemian, V. Friderikos, and A. H. Aghvami, “Adaptive admission control for ad hoc and sensor networks providing quality of service,” Technical Report, King’s College London, 2005.

[41] C. J.C.H. Watkins, and P. Dayan, “*Q*-Learning,” Machine Learning, vol.8, no.3-4, pp.279-292, 1992.

[42] The Network Simulator - ns-2, http://www.isi.edu/nsnam/ns/, Accessed on February 23, 2010.

[43] F. Bai, N. Sadagopan, and A. Helmy, “Important: a framework to systematically analyze the impact of mobility on performance of routing protocols for adhoc networks,” Proc. 22nd Annual Joint Conf. of the IEEE Computer and Commun. Societies, San Francisco, USA, pp.825-835, Apr. 2003.

[44] U.S. Census Bureau - TIGER/Line,
http://www.census.gov/geo/www/tiger/, Accessed on February 23, 2010.

[45] A.K. Saha, and D.B. Johnson. “Modeling Mobility for Vehicular Ad Hoc Networks,” Proc. First ACM Workshop on Vehicular Ad Hoc Networks, Philadelphia, USA, pp.91-92, Oct. 2004.

[46] J. Wu, and F. Dai, “A Generic Distributed Broadcast Scheme in Ad Hoc Wireless Networks,” IEEE Trans. on Computers, vol.53, no.10, pp.1343-1354, 2004.

[47] M. Khabbazian, and V.K. Bhargava, “Efficient Broadcasting in Mobile Ad Hoc Networks,” IEEE Trans. on Mobile Computing, vol.8, no.2, pp.231-245, 2009.

[48] J. Wu, W. Lou, and F. Dai, “Extended Multipoint Relays to Determine Connected Dominating Sets in MANETs,” IEEE Trans. on Computers, vol.55, no.3, pp.334-347, 2006.

[49] IEEE Std 802.11, “Wireless LAN medium access control (MAC) and physical layer (PHY) specifications,” ANSI/IEEE 802.11 Std, Aug. 1999.

List of Abbreviations

MANET	Mobile Ad hoc Network
VANET	Vehicular Ad hoc Network
AODV	Ad Hoc On Demand Distance Vector Routing
QLAODV	Q-Learning Ad Hoc On Demand Distance Vector Routing
AODV-HPDF	An ad hoc on-demand routing protocol with high packet delivery fraction
NRD	Neighborhood Route Diffusion for Packet Salvaging in Networks with High Mobility
ZRP	Zone Routing Protocol
OLSR	Optimized Link State Routing Protocol
RREQ	Route Request
RREP	Route Reply
RERR	Route Error
RCNG-REQ	Route Change Request
RCNG-REP	Route Change Reply
MPR	Multipoint Relay
CBR	Constant Bit Rate
UDP	User Datagram Protocol
GPS	Global Positioning System
GPSR	Greedy Perimeter Stateless Routing
IEEE	Institute of Electrical and Electronics Engineers
MAC	Medium Access Control
RTS	Request To Send
CTS	Clear To Send
ACK	Acknowledgement
NACK	Negative Acknowledgement
IP	Internet Protocol
FCS	Frame Check Sequence
TIGER	Topologically Integrated Geographic Encoding and Referencing system
ns-2	The Network Simulator - ns-2

Acknowledgments

This PhD thesis would not have been possible without the support of many people and organizations. I am honored to have a chance to express my deep appreciation here.

First and foremost, I would like to express my great appreciation to my supervisor Professor Toshihiko Kato, had it not been for his constant guidance, instructions and encouragement, I would certainly never have successfully completed this thesis. He has been a great and wonderful supervisor.

I would also like to express my great appreciation to Professor Hiroyoshi Morita, Professor Yoshikatsu Tada, Professor Hiroshi Nagaoka, Professor Akihiko Ohsuga, Professor Tsutomu Yoshinaga and Associate Professor Hisashi Koga. Their suggestions and comments on this thesis have been extremely valuable.

Many thanks are due to Associate Professor Satoshi Ohzahata for his invaluable comments on all aspects of my research. His advice is a great source of inspiration for my research.

I wish to thank Assistant Professor Kumekawa Kazuya, for his years of suggestions, guidance, teachings and his willingness to listen when I needed to talk about everything technical.

Special thanks are due to Japan Student Services Organization (JASSO) and Tonen International Scholarship Foundation for their financial support during the course of this research. Many thanks go to my laboratory colleagues, especially Alireza Nemaney Pour and Yaping Gong, for their advice and kindness.

Last, but definitely not least, I would like to thank my family members for their encouragements and love. It would not have been possible to complete this thesis without their support.

Author Biography

Celimuge Wu was born in Inner Mongolia, China, on May 8, 1979. He received his M.E. degree from Beijing Institute of Technology, Beijing, China, in 2006. Since 2007, he has been with the Department of Information Network Science, Graduate School of Information Systems, the University of Electro-Communications, Tokyo, Japan, working toward the Ph. D. degree. His research interests include mobile ad hoc networks, networking architectures and protocols.

Celimuge Wu is a member of the Institute of Electronics, Information and Communication Engineers (IEICE) of Japan and Information Processing Society of Japan (IPSJ).

List of Publications Related to the Thesis

Jouranl Paper

[1] Celimuge Wu, Kazuya Kumekawa and Toshihiko Kato, "A novel multi-hop broadcast protocol for vehicular safety applications," Journal of Information Processing, vol.51, no.3, pp.930-944, Mar. 2010.

(Related to the content of Chapter 4)

[2] Celimuge Wu, Kazuya Kumekawa and Toshihiko Kato, "Distributed reinforcement learning approach for vehicular ad hoc networks," IEICE Transactions on Communications, vol.E93-B, no.06, June. 2010. (To appear).

(Related to the content of Chapter 3)

International Conference

[1] Celimuge Wu, Kazuya Kumekawa and Toshihiko Kato, "A MANET protocol considering link stability and bandwidth efficiency," Proc. International Conference on Ultra Modern Telecommunications & Workshops, pp.1-8, Oct. 2009.

(Related to the content of Chapter 3)

[2] Celimuge Wu and Toshihiko Kato, "Reliable and efficient data dissemination in VANETs," Proc. IASTED International Conference on Internet and Multimedia Systems and Applications, pp.105-114, Jul. 2009.

(Related to the content of Chapter 4)

[3] Celimuge Wu and Toshihiko Kato, "An enhanced reinforcement routing protocol for inter-vehicular unicast application," Proc. IASTED International Conference on Internet and Multimedia Systems and Applications, pp.170-175, Mar. 2008.

(Related to the content of Chapter 3)