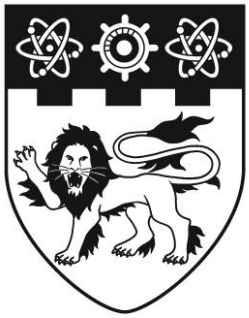
On the design of hybrid and polymorphic routing protocols for mobile ad hoc networks

Chen, Lei

2009



NANYANG TECHNOLOGICAL UNIVERSITY

ON THE DESIGN OF HYBRID AND POLYMORPHIC ROUTING PROTOCOLS FOR MOBILE AD HOC NETWORKS

CHEN LEI

SCHOOL OF COMPUTER ENGINEERING

2009

On the Design of Hybrid and Polymorphic Routing Protocols for Mobile Ad Hoc Networks

Chen Lei

School of Computer Engineering

A thesis submitted to the Nanyang Technological University
in fulfillment of the requirement for the degree of
Doctor of Philosophy

2009

Acknowledgment

I am deeply grateful to my supervisor Dr. Foh Chuan Heng and my former supervisor Dr. Adel Ben Mnaouer for their advice, support and patience during my candidature period. Without their guidance, I would not finish my work so fruitful.

I would like to express my gratitude to Dr. Andreas Jurgen Kassler for his supervision, friendship and moral support during my project.

I would like to thank all my friends and colleagues in the Centre for Multimedia and Network Technology (CeMNet), with whom I have worked together, for their helpful discussions and friendship. I am especially grateful to Ms. Chua Poo Hua, who provides me well support of research environment.

I am indebted to the examiners who have spent their precious time to give suggestions to improve the thesis.

Last but not the least important, I would like to express my profound gratitude to my beloved parents, for their love, understanding and support during my candidature period and more. Thank you very much, I love you.

Contents

List of Figures

List of Tables

Abstract

With the rapid growth of the development of wireless communication technology, we witness the increase in the popularity in the research of wireless Mobile Ad hoc Networks (MANETs). Since MANETs are networks with no fixed infrastructure, nodes within a MANET communicate with each other through a multi-hop route. Thus, effective routing protocols for MANETs are in urgent need to realize MANETs. This thesis introduces new concepts and designs for routing protocols in MANETs.

We first design routing protocols for MANETs using the hybrid approach. We propose two hybrid routing protocols, namely, ZODMRP and ZMAODV. The hybrid approach attempts to combine the benefits of proactive and reactive routing protocols into a single routing protocol. This combination allows a node implementing a particular hybrid routing protocol to operate in two different routing behaviors at the same time. Usually, hybrid routing protocols use some simple rules which combine the operations of proactive and reactive routing protocols to optimize the benefits of the two selected protocols. We notice that the concurrent operation of two behaviors in a hybrid routing protocol makes the protocol consumes more resources. The rules for deciding the routing behaviors of a hybrid routing protocol that often used also have limited ability to optimize a hybrid routing protocol performance under a wide range of network conditions.

To address the shortcomings of the hybrid approach for routing protocol design, we introduce a new concept of the routing protocol design. In this design, a node implements a number of routing protocols and adaptively selects an appropriate routing behavior to operate based on the environmental conditions and

the resource availabilities. We call this *polymorphic routing* protocol design. The adaptiveness of the polymorphic routing protocol gives it the ability to achieve low resource usage and high performance under various network conditions.

We demonstrate the concept of polymorphic routing protocol design by proposing three polymorphic routing protocols, namely, P_ZODMRP, Optimized Polymorphic Hybrid Multicast Routing (OPHMR) protocol and Polymorphic Unicast Routing Protocol (PURP). Each of these three routing protocols is designed for a targeted network environment. P_ZODMRP is a multicast routing protocol that uses zone routing technique from Zone Routing Protocol (ZRP) for its proactive behavior and the routing selection technique from On-Demand Multicast Routing Protocol (ODMRP) protocol for its reactive behavior. A specific polymorphic routing algorithm is added into the design to achieve polymorphic behavior. OPHMR is an enhanced version of P_ZODMRP where multipoint relay (MPR) technique from Optimized Link State Routing (OLSR) protocol is introduced into the design to optimize the control packets propagation in its proactive behavior. Finally, we design PURP for the Vehicular Ad hoc Networks (VANETs) using the concept of polymorphic routing protocol design. PURP uses a different polymorphic routing algorithm from P_ZODMRP and OPHMR based on the characteristic of VANETs. The proactive behavior of PURP is derived from ZRP and the reactive behavior comes from Ad hoc On-Demand Distance Vector (AODV) routing protocol. Also, the MPR mechanism from OLSR is still used in PURP. For all the protocol designs, we conduct simulation experiments to study their performance and highlight their performance advantages.

Chapter 1

Introduction

In recent years, we witness the increase in use of wireless and mobile communications in daily activities. The current wireless and mobile communications rely heavily on infrastructure networks such as 2G/3G cellular networks, the IEEE 802.11 wireless local area networks [1], the emerging IEEE 802.16 wireless metropolitan area networks [2], and others. The deployment of infrastructure networks is not only costly, but also time consuming. The complementary solution to infrastructure networks is the wireless mobile ad hoc networks (MANETs) [3]. MANETs are wireless mobile networks with no infrastructure. This type of network provides solutions to situations where infrastructures are difficult or not economical to deploy. Deployment of a MANET in places such as a disaster area or a battlefield results in shorter deployment time and lower setup cost comparing to that of an infrastructure network.

With no fixed infrastructure, MANETs rely on nodes to relay packets across the networks to the destination. In MANETs, each node communicates directly with its neighbors to forward packets. Each node essentially acts as a router to

find destination and forward packets to achieve communications.

One of the main challenges in developing an efficient and practical MANET is the design of a routing protocol. MANETs possess many characteristics different from those infrastructure networks. In particular, MANETs use wireless channels as the media which have limited bandwidth. The key element in MANETs, the mobile device, has limited computing power. In addition, the mobile device is often powered by battery which has finite power resources and limited operational lifetime. The device mobility feature in MANETs implies the constant change in network topology. This frequently changing network topology makes the maintenance of routing information difficult. This is because any established routing information can be made invalid quickly due to the changing topology.

Since the traditional routing protocols used in existing wired networks are not capable for handling those unique characteristics in MANETs, a new approach is necessary in order to design a routing protocol for MANETs.

1.1 Routing Issues in Mobile ad hoc Networks

One of the key concerns of a routing protocol for MANETs is the performance [4]. The general measures of performance in routing protocols for MANETs are throughput and latency. In particular, throughput measures the amount of data packets successfully transmitted from the source to the destination in a MANET given a certain load. Latency measures the time delay from when a data packet is generated to when the packet has reached the destination.

In order to achieve high performance in a routing protocol operation, the protocol must achieve high throughput and low latency. However, a common argument

in MANET performance is that a routing protocol that achieves high throughput (resp. low latency) produces high latency (resp. low throughput).

Many routing protocol design techniques are proposed to achieve the trade off between throughput and latency so as to meet specific throughput and latency requirements in a particular environment. In general, these techniques can be described by two broad classes, namely, proactive and reactive.

In proactive operation, a node periodically sends out packets to update its neighbors about the routing information in order to maintain consistent routing information in the network. The constant background maintenance of routing information among all nodes produces ready routing information for a node to route packets to the destination. The well prepared routing information in each node at all time allows routing of data packets in the network immediately after the packets are generated. This leads to an overall low latency in packet routing. However, the constant background routing information updates consume bandwidth. In some setups where channel bandwidth is limited, such bandwidth consumption may starve data packet transmissions which defeats the purpose of data packet routing.

On the other hand, a node in reactive operation does not require constant maintenance of routing information. When a packet is generated in a node, the node starts the route discovery procedure. The route discovery procedure may take some time to establish a potential route between the source node and the destination. This procedure increases the data packet routing latency. Since route establishment is performed when a data packet is generated, this eliminates the need for background control packet exchanges and allows the channel to be mainly utilized for data packet routing.

Comparing between the proactive and reactive operations, routing protocols using proactive operation often give lower latency and relatively lower throughput. Whereas, routing protocols using reactive operation often have higher throughput but higher latency. A well designed routing protocol for MANETs should balance between throughput and latency to meet a specific target of throughput and latency requirements.

Apart from the performance of a routing protocol, there are several other considerations in the design of a routing protocol due to the characteristics of MANETs. Some important examples include battery lifetime, computational power, QoS and security.

1.2 Polymorphic Concept for Routing Protocol Design in Mobile ad hoc Networks

In this thesis, we propose a new routing protocol design concept, the *polymorphic* protocol design. In this design, a mobile node detects designated environmental conditions and resource availabilities as factors and selects an appropriate routing behavior from pre-implemented behaviors for operation.

Polymorphic routing protocol design allows a polymorphic routing protocol to meet some specific objectives in the operation. Objectives such as high packet delivery ratio, low end-to-end packet routing, long operational lifetime for battery powered devices, and others are commonly considered objectives in routing protocol design. In the polymorphic routing protocol design, specific objectives of the routing protocol are first defined.

Given the objectives, it is then necessary to identify factors [5] that may influence the achievement of objectives. For example, battery level is a factor to be considered when the objective is set to achieve prolonged operational lifetime of battery powered mobile nodes. The factors are considered as inputs for the decision making in polymorphic routing protocol. Based on these factors, the behavior of the polymorphic routing protocol is adapted accordingly.

A set of routing behavior modes is then proposed. The polymorphic routing protocol will select a behavioral mode to operate based on the factors. Relationships between behavior modes and factors must be established, where inputs are factors and the output is the selected behavioral mode. For example, a low battery power and low mobility indications may enforce a reactive behavior which consumes less power to operate.

A polymorphic routing protocol consists of a polymorphic algorithm that describes the relationships between factors and routing behavior modes. The designed polymorphic algorithm allows the node using polymorphic routing protocol to adapt its behavioral mode based on the environment and its condition. A routing protocol that matches the behavioral mode is selected to be implemented for the routing protocols. The resultant polymorphic routing protocol switches among the selected routing protocols to operate based on the environment and its condition.

It is recognized that each existing routing protocol in the literature has its own advantages and shortcomings when operating in a particular environment and/or conditions in a MANET. There is no single routing protocol that can outperform others and maintain its benefits in all conditions. Our polymorphic routing protocol design approach allows specific objectives to be achieved for certain targeted

network environments. Rather than being a new routing protocol, a polymorphic routing protocol combines a collection of routing protocols with each routing protocol outperforming others in a particular condition. Based on the conditions, the polymorphic routing protocol selects the most appropriate routing protocol to operate. It is this adaptation that allows the polymorphic routing protocol to produce the best outcome based on the defined objectives.

1.3 Overview

In this thesis, I summarize the work that I have done during my candidature period for the degree of Doctor of Philosophy.

In Chapter 2, we describe the characteristics of mobile ad hoc networks (MANETs), and discuss the impacts of these characteristics on the routing protocol design. Then we review the existing routing protocols related to our research.

In Chapter 3, we experiment on hybrid multicast routing protocol design. We propose two new hybrid multicast routing protocols, ZMAODV and ZODMRP [6], by combining zone routing concept from Multicast Zone Routing Protocol (MZR) [7] with Multicast Ad hoc On-Demand Distance Vector (MAODV) [8] routing protocol and On-Demand Multicast Routing Protocol (ODMRP) [9] respectively. The main difference between ZMAODV and ZODMRP is their multicast group topologies, since the multicast group topology of MAODV is a core-based tree, the topology of ODMRP is mesh and the topology of MZR is a source-based tree. This difference allows us to examine the influence of multicast group topologies on the performance in hybrid multicast routing protocols. We first describe the detailed protocol format of these two new protocols, then introduce the route finding

and maintenance processes. Finally, we analyze their performance by comparing them to some well-known peer protocols through simulation.

In Chapter 4, we introduce our new concept of routing protocol design, precisely, the polymorphic routing protocol design. We illustrate the polymorphic routing protocol design by proposing two new power-aware multicast routing protocols, P_ZODMRP [10] and Optimized Hybrid Multicast Routing (OPHMR) [11] protocol. These two protocols allows a node to change behavior according to its power level and other factors such as mobility speed and vicinity density. We design a polymorphic routing algorithm which is responsible for behavior selection under different situations. We define the proactive behavior of P_ZODMRP using MZR and the reactive behavior using ODMRP. OPHMR is an enhanced version of P_ZODMRP. In OPHMR, we introduce the multipoint relay technique from Multicast Optimized Link State Routing (MOLSR) [12] protocol into P_ZODMRP to optimize the control packet propagation of the proactive behavior. Through simulation experiment, we show that the enhancement achieves even better performance result.

In Chapter 5, we apply the polymorphic concept to the design of a routing protocol for Vehicular Ad hoc Networks (VANETs). This type of MANETs has received increasing attention in the recent years, but its special characteristics have made existing routing protocols for MANETs not suitable for VANETs. Applying polymorphic routing design approach to routing protocol design for VANETs appears as a promising approach. We first summarize the characteristics of VANETs and review the existing routing protocols for VANETs. Then we propose our new polymorphic protocol, Polymorphic Unicast Routing Protocol (PURP) [13], which is a unicast routing protocol for VANETs. We illustrate the behavior se-

lection algorithm of PURP and show the effectiveness of the algorithm using simulation. The proactive behavior of PURP is based on Zone Routing Protocol (ZRP) [14] and Optimized Link State Routing (OLSR) protocol, and its reactive behavior is based on Ad hoc On-demand Distance Vector (AODV) [15] routing protocol. This chapter finally studies the performance of PURP with simulation results showing the performance advantages of PURP.

In Chapter 6, we summarize the main contributions of our work, draw important conclusions, and suggest future work directions in this area of research.

1.4 Contribution

The main contribution of this thesis is threefold. Firstly, we have proposed two hybrid multicast routing protocols [6], ZMAODV and ZODMRP. ZMAODV uses the Zone Routing concept from MZR as its proactive behavior and MAODV as its reactive behavior. ZODMRP combines MZR and ODMRP as its proactive behavior and reactive behavior respectively. We evaluate the performance of these two protocols by comparing them to their peer protocols. The results reveal that combined behavior usually outperforms pure behaviors and in general, mesh-based protocols have better performance than tree-based protocols in multicast routing design.

Secondly, we have introduced polymorphic concept in multicast routing protocol design. Using this new design concept, we propose our first polymorphic routing protocol, called P_ZODMRP [10]. This protocol decides a node's behavior based on the remain power, mobility and vicinity density of the node. The proactive behavior of this protocol comes from ZRP and the reactive behavior

adopts from ODMRP. After comparing P_ZODMRP with its corresponding hybrid protocols (ZMAODV and ZODMRP), we reach the conclusion that the adaptive behavior determination of polymorphic routing protocols leads to a better performance than simple combining behaviors. We further extend the protocol design of P_ZODMRP to achieve performance optimization of control packet propagation. We call this extended version of P_ZODMRP the OPHMR [11], [16]. This protocol refines its proactive behavior by introducing the Multipoint relay (MPR) mechanism from OLSR so as to optimize the control packet propagation. The evaluation of this protocol's performance shows that a proper selection and definition of routing behaviors for a polymorphic routing protocol also results in an enhanced performance.

Thirdly, we have demonstrated the polymorphic routing protocol design for vehicular ad hoc network (VANET), which we call PURP [13]. PURP mixes AODV and ZRP where it uses AODV as its reactive behavior and ZRP as its proactive behavior. Its routing behavior changes based on traffic load, mobility and vicinity, which are the three network environment factors in VANET. We measure throughput and delay performance of PURP. Comparing with its peer protocols, PURP achieves better performance in typical network conditions. This again shows the benefits of polymorphic routing protocol design.

Publications due to the work presented in this thesis are

1. Adel Ben Mnaouer, Lei Chen, Chuan Heng Foh and Juki Wirawan Tantra, "The OPHMR: An Optimized Polymorphic Hybrid Multicast Routing Protocol for MANET", IEEE Transactions on Mobile Computing, Vol. 6, No. 5, P551-563, May, 2007.

2. Lei Chen, Chuan Heng Foh and Adel Ben Manouer, "An Optimized Poly-

morphic Hybrid Multicast Routing Protocol (OPHMR) for Ad Hoc Networks", the IEEE International Conference on Communications, Wireless Ad Hoc and Sensor Network, Istanbul, Turkey, 2006.

3. Adel Ben Mnaouer, Lei Chen, Chuan Heng Foh and Juki Wirawan Tantra, "A New Polymorphic Multicast Routing Protocol for MANET", the IEEE International Conference on Communications, Seoul, Korea, 2005.

4. Lei Chen and Adel Ben Mnaouer, "Performance Evaluation of New Hybrid Multicast Routing Protocols for Ad-hoc Networks", in Proceedings of 9th IEEE International Conference on Communications Systems (ICCS04), Singapore, 2004.

and the submitted publications are

5. Lei Chen, Adel Ben Mnaouer and Chuan Heng Foh, "A Generic Polymorphic Unicast Routing Protocol for VANETs", submitted to the IEEE Transactions on Vehicular Technology.

Chapter 2

Literature Review

In this chapter, we address the characteristics and the designing issues of routing protocols for Mobile Ad hoc Networks (MANETs). We discuss different protocol design approaches by analyzing their characteristics and their performance under various circumstances. In addition, we review some routing protocols that exhibit a certain limitation in operation. In particular, we discuss routing protocols with power constraint. Finally, we summarize the reviewed routing protocols with discussion on their advantages and shortcomings. The discussion also forms the arguments and foundations for the polymorphic routing protocol design approach.

2.1 Characteristics of MANETs and their impact on routing protocol design

The unique characteristic of MANETs is infrastructureless. MANETs make use of mobile nodes to perform routing in the networks. Since mobile nodes are not stationary, their movement causes changes in the network topology. The con-

stantly changing network topology makes it inapplicable to use traditional routing protocols for wired networks in MANETs. As a result, a new class of routing protocols for MANET routing is required. This new class of routing protocols requires some additional considerations in the design. Some commonly considerations are described as follows.

- Infrastructureless Concern: In an infrastructureless network environment, all mobile nodes act as routers to actively contribute to the operation of MANETs. Mobile nodes relay packets on wireless channels from the source node to the destination. As the locations of mobile nodes are not deterministic, the routing protocol design for MANET must take care of the discovery of the path between the source node and the destination. The design for an effective path discovery is an important issue for all the routing protocols in MANETs.

- Topology Changing Concern: One of the important features in MANETs is mobile communications. All mobile nodes may move away from their original locations to new locations from time to time. As mobile nodes actively participate in routing of packets, their movements may cause a communication link breakage in multi-hop packet relaying, which in turns invalid an earlier established path between two communicating mobile nodes. As a result of the changed network topology, a new routing path must be established to recover the communication from the link breakage. In some environments where mobile nodes move constantly, the change in network topology occurs continuously. The routing protocol design for MANET must provide an effective method to maintain continuation of routing ser-

vice.

- Bandwidth Concern: All the mobile nodes communicate using wireless channels in MANETs. As the bandwidth for wireless channels is limited, this also limits the network bandwidth of MANETs. A routing protocol in MANETs should have the ability to use the bandwidth effectively. Designs such as limiting the traffic volume of control packets or reduces redundant data packet relaying are some common methods to improve the efficiency of bandwidth usage in routing protocols.

- Battery Life Concern: Mobile devices are usually battery powered. As the power supply of batteries is finite, the efficient use of power for routing may also be an important routing protocol design issue. A more efficient routing protocol consumes less power for the routing operation which in turns increases the lifetime of a mobile node. An increased lifetime operation of mobile nodes also increases the lifetime of the MANETs.

2.2 Types of Routing protocols in MANETs

Given the vast varies of MANET routing protocols, we recognized that there are many other ways to describe MANET routing protocols. In the following, we discuss several ways to view a MANET routing protocol related to the proposed protocols given in this thesis.

We can first describe MANET routing protocols based on their behaviors which consists of proactive, reactive and hybrid. We can also describe MANET routing protocols according to their source-destination relationship which consists

of unicast, multicast and broadcast. There is also a special type of MANET routing protocols that is constraint aware. Some commonly considered constraints for routing design include QoS, route and power.

2.2.1 Behavior

There are many ways to describe MANET routing protocols. One common way is to describe MANET routing protocols based on the mobile nodes' behaviors in the network. According to the behaviors of a node, the routing protocols can be grouped into three kinds, they are, proactive routing protocols , reactive routing protocols and hybrid routing protocols.

Proactive routing protocols follow those design approach used in traditional wired networks. In proactive routing protocols, mobile nodes in a MANET periodically exchange routing information among themselves to establish their routing tables. These distributed routing tables collectively describe the entire network topology. Data packets generated from a source node are forwarded according to the established routing table in each forwarding mobile nodes to the destination.

The application of proactive routing protocols to MANETs introduces some performance concerns. Due to the bandwidth limitation in MANETs, the periodic exchange of routing information may reduce the available bandwidth for data transmission. Notice that the constant change in network topology due to mobility in MANETs may invalidate the established routing tables causing packet routing failure. To overcome such routing failure and promote the robustness of the proactive routing protocols, the frequency of routing information exchange needs to increase. However, the cost of this increase is the excessive control packet

transmissions which result in increased consumption of bandwidth and battery power.

Reactive routing protocols, on the other hand, are introduced to eliminate the need for periodic routing information exchange among mobile nodes in a MANET. Reactive routing protocols perform route establish on demand. In reactive routing protocols, mobile nodes do not require to establish ready routing tables at all time. Whenever a request for packet routing occurs, the mobile node attempts to discover a path connecting itself and the destination. Once the route is confirmed between the source node and the destination, data packets are forwarded using the established route.

Due to its reactive behavior, the reactive routing protocols do not require routing information to be exchanged. Channel bandwidth and battery power consumption due to the control packet transmission reduces greatly compared to that of the proactive routing protocols. However, reactive routing protocols require an additional procedure, that is route discovery, to establish a route before any data transmission can take place. The time required to discover a route is counted as the delay in the data transmission. Hence, the data transmission delay is usually longer for the reactive routing protocols compared to that of the proactive routing protocols. Besides, a change in network topology may cause link breakage which then invalidates the established route. This immediately causes failure in packet routing. The reactive routing protocols must provide further mechanisms to detect and correct link breakage during data transmissions to increase their robustness.

From the above discussion, we can see that either proactive routing protocols or reactive routing protocols can provide solutions to a wide network configurations and conditions. Since different types of protocols may offer better perfor-

mances than others under a certain condition, combining various types of protocols into a single routing protocol appears as a useful approach to capitalize on each protocol's strength. Routing protocols based on such an approach are named as hybrid routing protocols.

A simple hybrid routing protocol is usually built on the principle that a mobile node is able to behave either proactively or reactively under different conditions. Since a hybrid routing protocol inherits the benefits of both the reactive and the proactive behaviors, it maintains its high performance in the situation when either the proactive or the reactive routing protocols can offer that high performance. The challenge in protocol behavior hybridization is the ability to define proactive and reactive behaviors to suit most common network conditions.

A common hybridization design is to enforce mobile nodes to operate simultaneously in both the proactive and the reactive behaviors, with some restricted operation imposed in the proactive behavior to reduce bandwidth and power consumption. For example, a hybrid routing protocol may enforce a mobile node to operate in both the proactive and the reactive behaviors, but the proactive behavior, the exchange of route information is restricted to a particular network range measured in the hop counts.

2.2.2 Source-Destination Relationship

We can also describe routing protocols for MANETs based on relationship of source-destination. There are three kinds of relationships, namely, one-to-one, one-to-many, and one-to-all. Based on this differentiation, MANET routing protocols are separated into three groups. They are unicast, multicast, and broadcast,

which specifies a single mobile node as the destination, a collection of mobile nodes, and all mobile nodes as the destinations, respectively.

For unicast routing protocols, the data is designated to one specific destination. This kind of routing protocols establishes a loop-free path from the source node to the destination and then maintains the route between the source-destination pair during the data transmission. The main goal of such protocols is to identify the most efficient path towards the destination when the source node has data to send, and to provide a mechanism to deal with the link breakage, transmission error and any other failures during the data transmission period.

On the contrary, the multicast routing protocols specifies a collection of mobile nodes as the destinations. A destination group is specified using a multicast IP address. Every mobile node within the networks may join and leave a group independently. This kind of routing protocols is responsible for managing the destination group in the form of *graphs* and performing one-to-many routing of packets. In the destination group management, the multicast routing protocols will provide a mechanism to deal with the mobile node joining, leaving, and link breakage. In the packet routing, the multicast routing protocols will establish a one-to-many path for the packet forwarding.

As for the broadcast routing protocols, the destination set consists of all mobile nodes in the network. The common method for broadcast routing is flooding routing. This kind of routing protocols is rarely used as a routing protocol for data packets in a network. However, flooding routing is used to support other routing protocols in order to establish some necessary information for their operations. Flooding routing remains as an important mechanism in the design of routing protocols of other kinds.

2.2.3 Constraint Awareness

There is a small class of MANET routing protocols that add constraint awareness into the consideration of routing protocol design. Some commonly considered constraints or factors are power, quality of service (QoS), route. Route-aware routing protocols (e.g. [17], [18], [19], [20]) attempt to select the minimum hop counts for the packet routing. Power-aware routing protocols (e.g. [21], [22], [23], [24]) attempt to find the route with minimum power consumption or to avoid utilizing mobile nodes with low battery power for the packet routing. QoS-aware routing protocols (e.g. [25], [26], [27], [28]) attempt to provide service differentiation to traffic of different classes in routing. These are protocols that set additional rules to achieve specific targets during packet routing.

2.3 Proactive Routing Protocols

In this section, we review several important proactive routing protocols for MANETs. One of the earliest proactive routing protocols for MANETs was proposed by Perkins and Bhagwat called Highly Dynamic Destination-Sequenced Distance Vector routing protocols (DSDV) [29] in 1994. It is a typical distance vector proactive routing protocol. It inspires many other proactive routing protocols, for example, Babel [30]. Seeing link state approach as a potential for MANET routing protocol design, Optimized Link State Routing (OLSR) [31] was proposed in 2001 by Clausen *et al.*. The authors continue improving it and the most current version of OLSR is specified in [32] by Clausen and Jacquet in 2003. OLSR is based on link state algorithm. To minimize the impact of flooding in the MANET environment, OLSR uses an optimized mechanism for flooding. The authors fur-

ther propose Multicast OLSR (MOLSR) [12] which is the enhanced version of OLSR to support multicast transmission. Furthermore, recently, Clausen *et al.* propose the Optimized Link State Routing version 2 (OLSRv2) [33] in 2007, which adds new features to OLSR.

Apart from the above discussed proactive routing protocols for MANETs, there are also many other proactive routing protocols each with its special feature for some network environments and conditions. Some important examples include Source Tree Adaptive Routing Protocol (STAR) [34] proposed by Garcis-Luna-Aceves and Spohn in 1999, Topology Broadcast based on Reverse-Path Forwarding routing protocol (TBRPF) [35] by Ogier, Templin and Lewis in 2004, and Direction Forward Routing (DFR) [36] proposed by Lee *et al.* in 2006.

2.3.1 DSDV

Highly Dynamic Destination-Sequenced Distance Vector routing protocol (DSDV) [29] is a distance vector routing protocol. It is based on the Distributed Bellman-Ford routing protocol. In DSDV, data packets are forwarded among the mobile nodes using routing table which is stored at each node in the network. Each node maintains a routing table which lists all available destinations and the hop count together with the next hop node address to each destination. Each routing table entry stores routing information for one possible destination, and each entry is tagged with a sequence number which is originated by the destination so as to maintain the freshness of the routing information.

To maintain the consistency of the routing table, each node within the network periodically transmits update packets to all of its neighbors. When a significant

change in topology occurs, an update packet is transmitted immediately.

The update packet includes routing information based on the routing table of the node which originates the transmission. The needed information contained in the update packet includes the destination address, the hop count to the destination and the sequence number originated by the destination. The update packet also includes the address of the node which originates the packet, together with a new sequence number generated by the node.

The receiver of the update packet updates its routing table using the information stored within the packet. The node compares the sequence number between each entry of its routing table and the received update packet with the same destination. If the sequence number in the routing table is older, or the sequence number is equal but the hop count in the routing table is larger, the corresponding entry needs an update. The hop count in the corresponding entry is set to the hop count in the update packet plus one. The next hop address is set to the address of the node which originates the update packet. The new sequence number is also updated accordingly.

The receiver of the update packet also advertises this information. The receiver first adds an increment to the hop count and then broadcasts the new information.

Since the nodes in the network are mobile, when they move from place to place, link breakage may occur. When the link breakage is detected (such as by data link protocols), any route through the node as the next hop is immediately assigned as an infinite metric and assigned an updated sequence number, and such modified routes are immediately disclosed in a broadcast routing information packet. When a node receives an infinite metric, and it has a later sequence number with a finite metric, it triggers a route update broadcast to propagate the

important information to the network.

Some works [37] [38] aim to evaluate the performance of DSDV. These works reveal that with the nature of proactive routing protocols, DSDV always has low latency in the data transmission. The control overhead of DSDV maintains in a relatively constant level since DSDV uses periodical updating to maintain topology information. The throughput of DSDV decreases greatly in high mobility scenarios, that is because DSDV needs network wide topology information to calculate the route, and high mobility makes topology unstable, thus, many routes are incorrect due to the late updates, especially for those long distance routes.

2.3.2 OLSR and MOLSR

Optimized Link State Routing (OLSR) [32] protocol is a proactive unicast routing protocol for MANETs. This protocol is based on the classic link state mechanism. The key point of this protocol is its multipoint relay (MPR) mechanism. Each node selects a set of its neighbors as MPRs. Only selected MPRs can forward control packets which intend to disseminate into the entire network. Also, the MPRs have to declare link state information in the network. OLSR uses the link state information MPRs declared for their MPR selectors (by whom the node is selected as MPR) to provide shortest path (measured in hop count) routes to all nodes in the network.

Neighbor detection and MPR calculation

Each node i records four routing tables (node sets) to maintain the neighborhood and MPR information. They are Neighbor Set (NS), 2-hop Neighbor Set (2NS), MPR set and MPR selector set. Node i also maintains a parameter *will-*

ingness which indicates the desire of node i to carry and forward traffic for other nodes. The NS stores the neighborhood information of node i. All immediate neighbors of node i is listed in the NS together with their willingness, and each link between node i and its neighbor has a lifetime. The 2NS stores all nodes that can be reached by node i in 2 hops. Each entry of the 2NS contains a 2-hop link towards one 2-hop neighbor. The entry also contains a neighbor through which node i can reach the 2-hop neighbor. The lifetime of the 2-hop link is also included in the entry. The 2NS should list all possible 2-hop links to every 2-hop neighbor, and the set is used to calculate MPR of node i. The MPR set stores all MPRs that node i selects within its neighbors. The MPR selector set stores all neighbors which select node i as their MPR.

To detect neighbors and perform MPR calculation, node i periodically broadcasts HELLO message to its neighbors. The HELLO message includes the NS and MPR set of node i. On receiving the HELLO message, every node uses the received information to update its node sets. After processing the received HELLO message, the node discards it without forwarding.

For example, a node j receives a HELLO message from node i. It first updates its NS to include node i as its neighbor. If node i already exists in node j's NS, the lifetime and the willingness of the corresponding entry is updated accordingly. After that, node j updates its 2NS using the NS of node i stored in the HELLO message. Each neighbor of node i, except node j itself, is a 2-hop neighbor of node j through node i. So, node j updates its 2NS by creating nonexisting entries and updating the lifetime of existing entries. If node j is within the MPR set of node i shown in the HELLO message, node j updates its MPR selector set to add node i into it.

With the information stored in the NS and 2NS, a node can calculate its MPR. In the design, MPR set is selected such that a node can communicate with any of its 2-hop neighbor via at least one node from its MPR set. MPR also ensures that a broadcast transmission can reach all other nodes in the network.

The calculation of MPR follows a heuristic algorithm. We will further discuss of this algorithm in Chapter 4.

Topology Discovery and Routing Table Calculation

OLSR calculates topology information by periodically broadcasting topology stored by each node. Each node maintains a topology set (TS). Each entry of the set refers to one $(destination,\ last-hop)$ pair. The *destination* is reachable by the TS maintainer in one hop from the node indicated as *last-hop*. The entry also contains a sequence number and its lifetime. Different to classical link state routing protocol, only the node selected as MPR performs the periodic topology broadcasting.

Each MPR node broadcasts a Topology Control (TC) packet to build up the TS. The TC packet contains a list of the neighbors of the node which generates the packet. The list of neighbors in each TC packet can be partial, but within a certain period, all TC packets from one node should cover the complete list of its neighbors. The TC packet also includes a sequence number of the neighbor list stored in the packet. Every time detecting a topology change of the NS, the sequence number should increase by one so as to indicate this topology change.

Upon receiving a TC packet, the node should update its TS according to the received packet. If there are entries in the TS that the last-hops are equal to the node which generates the packet, i.e. the originator, and the sequence numbers in those entries are greater than that in the packet, the packet is discarded. Otherwise,

all the entries are removed if their last-hops are the same as the originator, and their sequence numbers are smaller than that in the packet. After that, if there still exists entries that have their last-hops the same as the originator and the destinations of these entries exist in the neighbor list of the packet, lifetime of these entries are updated. Then, new entries are inserted into the TS with the remaining of the neighbor list as the destinations and the originator as the last-hop. The sequence number of the packet is also copied into the new entries. Then the received TC packet is forwarded if the receiving node is the MPR of the node from which the packet is received.

Each node uses its NS, 2NS and TS to calculate the routing information. The routing information is stored in a routing table. Each routing table entry contains the destination address, next hop towards the destination, and hop count to the destination.

The routing table is calculated as follows. All the neighbors are inserted into the routing table. The destination and next hop fields of these entries are set to the neighbors address respectively. The hop count is set to 1. Then the strict 2-hop neighbors are inserted into the routing table according to the 2NS. The hop count for these entries is set to 2. Also, a counter h is set to 2. Then the node processes the TS. If there is an entry in the TS that the destination of the entry does not exist in the routing table, but the last-hop exists, and the hop count to the last-hop in the corresponding TS entry is equal to h, a new entry is inserted into the routing table. The destination of the new routing table entry is that of the TS entry, the next hop of the new routing table entry copies the routing table entry whose destination is the last-hop of the TS entry, and the hop count of the new routing table entry is $h + 1$. The procedure repeats with the value h increments by one until all the

entries in TS are handled, or no new entries are inserted in the routing table with the increment of h.

Thus, the routing table is built up and the node could use the routing table to route data packets.

An improvement of OLSR was proposed to provide multicast support for OLSR. This new protocol is called Multicast Optimized Link State Routing (MOLSR) [12] protocol. MOLSR creates source-based tree to maintain multicast group topology. Thus, a tree is built with its root at the source node and the tree includes all nodes belongs to one designated multicast group.

Each node willing to participate the multicast transmission (called multicast router) maintains three routing tables: MC_router_table, MC_tree_table and Multicast_routing_table. MC_router_table maintains a list of multicast routers within the network. The source tree is built only along the multicast routers. MC_tree_table maintains the source tree information. Each entry of MC_tree_table contains the source node address, the multicast group address, the upstream node (defined as the next node towards the source node) address and a list of the downstream nodes (the possible nodes that have one further hop away from the source node along the source tree) information. Multicast_routing_table maintains the shortest path to all multicast routers in the network. All the nodes along the shortest path are multicast routers. This table contains the destination address and the next hop towards the destination. The creation and maintenance of the Multicast_routing_table is with the same manner as OLSR creates and maintains it routing table.

Each multicast router periodically broadcasts MC_CLAIM packet throughout the network. This packet contains no special information and the packet is only used for notify the network that the originator of this packet is willing to partic-

ipate the multicast transmission. The node which receives this packet updates it MC_router_table to store the existence of the multicast router.

When a source node has data to send to one multicast group, it periodically broadcasts a SOURCE_CLAIM packet to build up and maintain the source tree. The SOURCE_CLAIM packet contains the addresses of all multicast groups the source node has data to send to.

When a node receives a SOURCE_CLAIM packet and it is the member of the multicast group indicated by the received packet, it responses to the packet. It first checks its MC_tree_table to see whether there is an existing entry refers to the source tree indicated in the SOURCE_CLAIM packet. If not, the node inserts a new entry to record this source tree information. Then it sends a CONFIRM PARENT packet to the upstream node. The upstream node is chosen from all possible next hop nodes towards the source node in the Multicast_routing_table. If the node already has the entry for the source tree, the node updates the corresponding entry to extend its lifetime.

When the upstream node receives the CONFIRM PARENT packet, it first checks that whether it has the corresponding source tree information. If not, a new entry is created and inserted to the MC_tree_table, and a CONFIRM PARENT packet is generated for itself. The new generated CONFIRM PARENT packet is sent to its upstream node of the tree, which comes from the Multicast_routing_table. If the node has the source tree information but it does not know that the originator of the receive CONFIRM_PARENT packet is one of its downstream nodes of the tree, it inserts the originator's information into the downstream nodes list of the corresponding entry.

Each node participating in a source tree periodically sends out CONFIRM

PARENT packet to notify the upstream node of its existence. When the node detects a topology change and the upstream node address changes, the node generates a CONFIRM PARENT packet to inform the new upstream node. Then the node may generate a LEAVE packet to disable the old upstream node if the old upstream node is still reachable.

Once a leaf node (the node has no downstream nodes in the tree) of the source tree wants to leave the tree, it sends a LEAVE packet to the upstream node. If the upstream node then becomes a leaf node of the tree and it is not a member of the multicast group, it also generates a LEAVE packet and sends the packet to its upstream node. This process continues until an upstream node has more than one downstream nodes in the tree or an upstream node is a multicast group member.

A source tree is built up and maintained throughout the transmission. The source node uses the source tree to propagate data packets to all members of the destination multicast group.

Note that all the packets that are going to disseminate throughout the network (for example, MC_CLAIM and SOURCE_CLAIM) are forwarded using the same manner in OLSR. The neighbor detection and route discovery of OLSR are still part of the operation in MOLSR to build up the MPR set and calculate the routing table. Thus, the operation for MOLSR described above is additional to OLSR to provide multicast support.

There are many works proposed to conduct an analysis of performance for OLSR and MOLSR, for example, [39], [40], [41] and [42]. These researches show that with the MPR mechanism, OLSR significantly decreases the amount of broadcasting control packets by eliminating a large amount of redundancy transmissions. This feature eases the main drawback of proactive routing protocols

for MANETs, which is large amount of control packets generated for topology updates leading to congestion. Also, OLSR maintains the performance characteristics of proactive routing protocols, such as relatively low latency for data transmission and low throughput in high mobility scenarios due to frequent topology changes.

2.3.3 TBRPF

Topology Broadcast based on Reverse-Path Forwarding routing protocol (TBRPF) [35] is a proactive, link state routing protocol. Each node in TBRPF maintains a topology information of the network and updates its stored information using a localized updating mechanism. TBRPF can handle the unidirectional scenarios. For the convenient purpose, we use (i, j) to denote a one hop directed link from node i to node j in the following introduction, that is node i can directly send packet to node j. The reverse link (j, i) may not exist due to the unidirectional condition.

TBRPF includes two main modules, they are the TBRPF neighbor discovery (TND) module and the routing module.

The TND module allows each node to quickly detect the neighbors and quickly detect link breakages. The TND uses HELLO messages which report only changes in the status of links.

Each TBRPF node maintains a neighbor table. The neighbor table stores state information for each neighbor detected. The status of the link between itself and each neighbor may be set to 1-WAY, 2-WAY, or LOST. The HELLO messages are sent periodically. The neighbor table determines the contents of the HELLO messages, and the table is updated based on the received HELLO mes-

sages. Each HELLO message contains three lists of neighbors: NEIGHBOR REQUEST, NEIGHBOR REPLY, and NEIGHBOR LOST. Each HELLO message also contains a sequence number, which is incremented with each transmitted HELLO message. This sequence number is used to maintain the freshness of the HELLO message so as to detect link breakage.

The routing module performs topology discovery and route computation. Each node running TBRPF maintains a source tree T, which provides shortest paths to all reachable nodes. The root node of T is the node itself. Each node calculates and updates its source tree based on partial topology information stored in its topology table.

A node periodically updates part of its source tree to the neighbors, this partial reported tree is called reported subtree RT. The periodical updating includes two parts. The first one is called *periodic update*. Whole RT is reported to the neighbor in periodic update. Periodic update has lower update frequency. The other one is called *differential update*. Only changes of the RT are reported in the differential update and the update frequency is higher. To decrease the amount of control overhead, all the topology updates are included in the same packet as a HELLO message.

To build up the RT, each node first calculates its reported node set RN. node i will include a neighbor j in its RN if node i can determine that one of its neighbors may select node i to be the next hop on its shortest path to node j. Thus, node i needs to calculate shortest paths from each neighbor to each other neighbor. The computation is limited up to two hops, and using only the neighbors or node i itself as an intermediate node. After the computation, node i could determine which neighbors are within its RN. Then node u is included in RN if the next

hop on the shortest path from node i to node u is within RN.

Each node has a relay priority which indicates the willingness of the node to propagate packets. A node with a large relay priority reports a large part of its source tree.

A node received an update packet will not forward the received packet but use it to calculate the shortest path to nodes so as to build its source tree. The shortest path here concerns not only the hop count, other metrics pre-defined by users can also be used to calculate the shortest path. The calculation is based on Dijkstra's algorithm.

2.4 Reactive Routing Protocols

In this section, we introduce several important reactive routing protocols for MANETs. In 1994, Johnson proposed Dynamic Source Routing (DSR) protocols [43], which is one of the first reactive routing protocols for MANETs. The most current version of DSR is [44]. DSR uses source routing technology to indicate the route. The original design of DSR was not optimized in all network conditions, some performance enhancements were proposed later, including adding flow state for DSR [45] proposed by the same authors of DSR in 2001.

Ad hoc On-Demand Distance Vector (AODV) routing protocol is another popular reactive routing protocol for MANETs. AODV was introduced by Perkins and Royer in 1999 [46]. The most current version of AODV is [15]. The authors also proposed an enhanced version of AODV to support multicast transmission in 2000 [8], named Multicast Ad hoc On-Demand Distance Vector (MAODV) routing protocol. Many researchers extend the concept of AODV to design new

reactive routing protocols, notable examples include Ad hoc On-Demand Multipath Distance Vector (AOMDV) protocol [47] proposed by Marina and Das in 2001, and Reliable Ad hoc On-Demand Distance Vector (RAODV) routing protocol [48] proposed by Khurana *et al.* in 2006.

For multicast reactive routing protocol, apart from the above examples, one of the popular examples is On-Demand Multicast Routing Protocol (ODMRP) [9]. It was proposed by Lee *et al.* in 1999 [49]. It is a typical mesh-based multicast routing protocol. It builds up mesh to represent multicast group topology and uses a unique group forwarding mechanism to provide redundant data transmission. Some research was done to improve ODMRP, for example, Yoneki and Bacon proposed Content-Based Routing with On-Demand Multicast [50] in 2004, and Klos and Richard III proposed Reliable ODMRP (RODMRP) [51] in 2001.

2.4.1 DSR

The Dynamic Source Routing Protocol (DSR) [44] is a reactive unicast routing protocol for MANETs. A source node executes route discovery procedure and uses source routing technique to establish and maintain the route. In source routing technique, the source node includes the entire route path to the destination in the header of the data packet, and all the intermediate nodes along the route propagate the data packet following the route stored in the header. The DSR protocol consists of two main mechanisms: Route Discovery and Route Maintenance. Since all nodes do not actively maintain routing tables, when a source node wants to send packets to the destination, it begins with the route discovery procedure. Route discovery procedure specifies the methods for a source node to establish a

route between the source node and the destination. After the route establishment, all nodes involved in the forwarding maintain the route. When link breakage is detected, the node executes the route maintenance procedure to find an alternative route to the destination. In the following, we give details of the DSR operation.

Route Discovery

Each node maintains a Route Cache which contains the possible route to a destination. The route may be discovered previously. When a node, called the source node, has data packets to send to the destination, and it has no route information to the destination in its Route Cache, the route discovery procedure is initiated. The source node originates a Route Request packet (RREQ) and transmits it as a local broadcast packet. With this broadcast, the RREQ will reach all nodes in the network. The RREQ includes the address of the source node and the destination as well as a unique request identification generated by the source node. The RREQ also includes a list, called forwarder list, to include nodes through which the packet has been forwarded.

When a node receives a RREQ, it first checks whether this received packet is duplicated. That is, when the received RREQ carrying the same source node address, destination address and request identification with one received previously, or the node finds itself in the forwarder list of the received packet indicating a loop, the received packet is a duplicated one and should be discarded.

The node then checks whether it is the destination. If it is not the destination, the node adds its own address to the forwarder list of the Route Request packet, then it transmits the packet as a local broadcast packet so as to propagate the RREQ to the destination.

The RREQ will finally reach the destination. The destination then replies to

the RREQ packet. Apart from the destination, those nodes containing the route information to the destination in their Route Cache may also reply to the RREQ packet. Upon receiving the RREQ packet, these nodes, called repliers, generate a Route Reply packet (RREP) and sends the it back to the source node. The RREP contains the list of the forwarders which is included in the RREQ.

Once the source node receives the RREP from the replier, the source node updates its Route Cache using the forwarder list given in the RREP as the route to the destination. Then the source node sends out the data packet following the received route using the source routing technique.

Route Maintenance

After the route establishment, each node forwarding data packets is responsible for the link connectivity from itself to the next hop. In many data link protocols, an explicit acknowledgement is returned to the transmitter to confirm the packet transmission. Failure in packet transmission at the data link layer indicates link breakage that may be due to the mobility of a node, power failure of a node, or other reasons.

When an intermediate node receives a data packet and detects that the link to the next hop indicated by the packet is broken, the node first searches in its Route Cache for other possible routes to the destination. If there are such routes, the node selects the shortest one and sends the data packets following the newly selected route. Then the node generates a Route Error message to the source node to inform this link breakage and the new route. If there are no such routes, the node simply inform the source node with the Route Error message. The source will redo the Route Discovery in this case.

In terms of performance, many aspects of the performance of DSR has been

evaluated in the literature [37], [38], [52], [53], [54] and [55]. All these works show that DSR has an outstanding performance in eliminating total number of control overhead. Since DSR uses the excessive route caching mechanism, many intermediate nodes that overhear routing information through other data transmission cache the route information which in turns help accelerate route discovery procedure. However, in terms of control overhead (in bytes), the performance advantage of DSR is not as significant as others because of the source routing mechanism of DSR. This is because each data packet contains a header which indicates the route, and the header is counted as control overhead.

Another drawback of DSR is that since DSR does not provide any efficient mechanism to verify the validity of the overheard routes, such aggressive route caching mechanism may lead to incorrect selection of routes, especially in high speed scenarios. In such scenarios where nodes are often moving fast, the probability of link breakage is high. The overheard routes may quickly become unavailable. Since the node cannot identify whether the overheard routes are still available, it may select the unavailable routes and pollute the Route Cache of other nodes.

2.4.2 AODV

Ad hoc On-Demand Distance Vector (AODV) [15] routing protocol is a famous reactive routing protocol for MANETs. In AODV, the source node begins the route discovery procedure only when there is a demand for data transmission. Each node in AODV maintains a routing table which stores routing information towards the destinations and determines the route towards the destination based

on the routing table.

In the routing table, each route is stored as an entry. Each entry of the routing table includes the destination address, the hop count to the destination, the next hop towards the destination and the lifetime of this route. AODV uses a sequence number mechanism to maintain the freshness of the routing information in the routing table. Each node maintains its own sequence number. On generating a control packet, the node includes its sequence number in it. Every other node records the sequence number of that node in the corresponding routing table entry according to the received control packets from that node. Only when the received sequence number is greater than that stored in the entry, or the sequence numbers are equal but the hop count of the received packet is less than that in the entry, the entry can be updated according to the receive control packets. We call this *updating criteria*. Also, when a node generates a control packet, it inserts the sequence number of the node it requests for if it knows. Another node on receiving the control packet will only respond to it when the node has entry for the requested node and the sequence number of the entry is greater than or equal to the received one. We call this *responding criteria*.

Route Discovery Procedure

When a node needs a route to a destination and it does not have any available routes, the node generates a Route Request packet (RREQ) and broadcasts it. The RREQ includes the source node address and the destination address. The RREQ also includes a source sequence number and a destination sequence number. The source sequence number is generated by the source node and the destination sequence number comes from the routing table of the source node. If the source node does not know any available sequence number of the destination, the un-

known sequence number flag of the RREQ will be set. Each RREQ has a unique RREQ_ID to detect packet duplication, and a hop count is also set to the RREQ to indicate the length that the RREQ has traveled. When initializing the RREQ, the hop count is set to zero. To avoid unnecessary network-wide dissemination of RREQs, a TTL value is set in RREQs to limit their propagation.

When a node receives a RREQ, it first checks whether the received packet is a duplicated one. This is done by comparing the source node address and RREQ_ID of the received RREQ with previously received RREQs. If the RREQ is a duplication, the node simply discards it. If not, the node updates its routing table to store the reverse path to the source node according to the received RREQ packet.

If the node's routing table has no entry describing the source node of the RREQ, it creates the entry and puts information from the RREQ packet to that entry. The destination address and destination sequence number of the entry are copied from the received packet accordingly. The hop count of the entry is set to the hop count field of the RREQ plus one. The next hop of the entry is the node address from which the RREQ is received.

If the node's routing table has such entry referring to the source node, the node further checks whether it meets the updating criteria. If so, the entry needs update. The update method is the same as the creation of a new entry.

On receiving the RREQ, if the node is not the destination, or it has no active routes (routing table entries meet the responding criteria) to the destination, or the TTL value of the RREQ is greater than 1, the node updates the RREQ and rebroadcasts it. The TTL value of the RREQ decremented by one, the hop count of the RREQ increases by one, and the destination sequence number of the RREQ is set to that stored in the node's routing table or the received one in the RREQ packet,

whichever is larger. After the RREQ is generated, the new RREQ is broadcasted. Note that the node should not update its routing table for the destination sequence number even if the received one in the RREQ is larger.

If the node is the destination, or it possesses an active route to the destination, the node generates a Route Reply packet (RREP) responding to the received RREQ. The RREP construction is described as follows. The RREP copies the source node address and the destination address from the RREQ. If the node is the destination itself, it first updates its sequence number so as to guarantee its own sequence number is no less than the received one in RREQ, then it puts the sequence number into the RREP. The hop count of the RREP is set to zero. Otherwise, the node is an intermediate node, the node puts the destination sequence number stored in its routing table into the RREP and copies the hop count stored in its routing table to the RREP. Once the RREP packet is generated, the node unicasts the RREP to the source node according to its routing table.

When a node receives the RREP, the node first updates its routing table by inserting a new entry of the destination or updating an existing entry if any. The existing entry can be updated when the received sequence number is larger than that in the entry, or the sequence number is the same but the received hop count plus one is smaller then the one in the entry. If an entry is created or updated, the next hop is set to the node from which the RREP is received, and the hop count value of the entry is set to the received hop count plus one. After then the node propagates the RREP according its own routing table.

Once the RREP reaches the source node, the source node begins the data transmission by sending the data packet to the next hop indicated in its routing table. Each intermediate node along the path uses its own routing table to determine the

path to the destination.

Link breakage and Route Error

In AODV, link breakage is intended to be repaired locally. For example, a link (i, j) is along an active path to a destination. The (i, j) here refers to a one hop link through which the data packet is transmitted from node i to node j. We call node i as the upstream node of (i, j) and node j is the downstream node. When (i, j) is detected broken, node i initiates a RREQ for that destination with an increased destination sequence number. The purpose to increase the destination sequence number is to avoid loop. Since upper stream nodes (the nodes between the source node and node i along the path) along the path have no knowledge of the link breakage, if they receive the RREQ, the increased destination sequence number stored in the RREQ could prevent the upper stream nodes to respond. Thus, the loop is avoided. The TTL value of the RREQ is limited so as to perform the repair locally. If the destination is not far from the upstream node of the broken link, it will receive the RREQ and repair the link breakage. During the local repair procedure, data packets are buffered and the data transmission will resume once the link breakage is repaired. Usually, local repair is transparent to the source node such that the source has no knowledge about the link breakage nor the local repair.

If the local repair fails, or a node receives a data packet but the node has no active route towards the destination of the data packet, the node generates a Route Error packet (RERR) and sends the packet to the source node. Once the source node receives the RERR, the source node regenerates the route discovery procedure if needed.

In the literature, [37], [38], [40], [41], [53] and [54] are some of the works

studying the performance of AODV. All these works indicate that AODV has better performance than other protocols in high speed scenarios. Due to its local repair mechanism and a relatively efficient mechanism to stale information [53], AODV can quickly respond to link breakage and provides link recovery as soon as possible. Although AODV usually has more control packets generated than other protocols, the amount of control packets in AODV is acceptable; besides, these control packets guarantee the validity of the selected routing path and increase the throughput performance of AODV.

2.4.3 MAODV

Based on AODV, Multicast Ad hoc On-Demand Distance Vector (MAODV) [8] Routing protocol is designed to provide multicast support. MAODV uses similar route discovery procedure with AODV, but to support multicast transmission, MAODV has some unique characteristics. MAODV uses a core-based tree to maintain the topology of multicast group, each multicast group selects one node within it as the group leader and the group leader would be the root of the tree. The core-based tree is constituted by the multicast group members and other nodes that are needed to build the tree. In the following description, we define the upstream node of a node i as the node which is the immediate one hop neighbor of node i along the path from node i to the root within the tree. Similarly, upper stream nodes of node i are those nodes which constitute the path from node i to the root. All nodes which have node i as their upstream node are node i's downstream node nodes. A leaf node means that the node has no downstream nodes.

Similar to AODV, MAODV is a table driven routing protocol that nodes main-

tain their routing tables to record routing information. MAODV inherits the routing table of AODV to record unicast routing information. In the following description, we use the term, unicast routing table, to refer to such routing table. MAODV also introduces a multicast routing table to maintain multicast routing information. Each entry of the multicast routing table represents a multicast tree the node belongs to. Each entry contains the multicast group address, the leader address of the multicast group, the upstream node address and all possible downstream nodes address. MAODV uses a similar sequence number mechanism from AODV to maintain the freshness of the information in routing tables. The difference between AODV and MAODV is that in MAODV, each multicast group also has a sequence number, this sequence number is maintained by the leader of the multicast tree.

Route Discovery Procedure

The route discovery procedure is required when a node wants to join a multicast group. The node begins the route discovery procedure with RREQs transmission. The RREQ is constructed with the J flag set. The RREQ also contains a sequence number generated by the source node and the latest sequence number that the source node may have for the multicast group. The destination address in the RREQ packet is always set to the multicast group address. If the source node knows the address of the multicast group leader and the route to the leader, it includes the multicast group leader address in the RREQ and unicasts it to the multicast group leader. Otherwise, the source node broadcasts the RREQ.

After transmitting the RREQ, the node waits for the reception of RREPs. The node may resend RREQ if it does not receive any RREPs in a predefined time period. If the node still cannot receive any RREPs after several RREQ transmission,

it concludes that the multicast group is no longer exist. The source node then start the group by declaring itself as the leader for others to join.

If a node wants to transmit packets to a particular group but the node does not have the routing information to the group, the route discovery procedure is used for the node to establish a route to the multicast group. The procedure follows that for a source node to join a group, except that the J flag in the RREQ is reset for this case. Besides, if the node fails to receive an RREP after several RREQ transmission attempts, the node concludes that the multicast group is unreachable and that all packets for the group will be discarded.

When a node receives a RREQ, the node checks the J flag. If the J flag is set indicating joining request, a reply to RREQ may be issued by the node if it meets the responding criteria. If the J flag is not set indicating data transmission request, a node that has a route to the multicast group can reply to the RREQ if the responding criteria is met. Otherwise, if the nodes are not responsible to reply RREQ, they are required to forward the RREQ to their neighbors. Except updating RREQs before forwarding, they also create a reverse route and store it in their routing table. This information may be used to route the RREP back to the source.

A node replies to the received RREQ by using a Route Reply Packet (RREP). The node first updates its multicast route table then generates a RREP. The RREP contains the current multicast group sequence number and the address of the multicast group leader. The node then unicasts the RREP back to the source based on the routing information setup in the intermediate nodes during their RREQ forwarding. The RREP will be forwarded on the path which the corresponding RREQ traveled but in the reverse order. During the RREP transmission, each

intermediate node also updates its multicast route table to record the routing information. Duplicate RREP can be detected by updating criteria. It discards all the other RREPs.

It is likely that the source node which originated the RREQ receives more than one RREP message. The source node usually waits for a period of time to collect all RREP messages. After that, the source node uses updating criteria to select the route. Finally, the source node sends an Activation Packet (MACT) through the route which is selected. All the other nodes which propagated the RREP but are not along the selected route expire the recorded multicast routing information after a period of time with no MACT received. Thus, route is built up from a specific node to the multicast group.

Group Hello Procedure

The leader of the multicast group periodically broadcasts a Group Hello message (GRPH) to ensure all member nodes maintain consistency and up-to-date information of the multicast group leader. The GRPH includes group leader address, multicast group address and the multicast group sequence number initiated by the leader.

A node receiving the GRPH updates its route table to record the multicast group information and multicast group leader information accordingly. It then rebroadcasts the message to its neighbors. The node may detect duplicate GRPH packets, and they are simply ignored.

Multicast Tree Pruning

When a node wants to leave the multicast group, it only needs to update its multicast routing table to mark itself out from the multicast group. If it knows that it is the leaf node of the multicast tree, the node just sends a MACT to the

upstream node of the tree with prune flag set and the upstream node removes the multicast route information of that node.

Repair Link Breakage

When a broken link is detected, the nodes detected the broken link first remove the link state information from their multicast route tables, and the downstream node of the broken link is responsible for the repair of the link breakage. Such node broadcasts a RREQ which has the J flag set, and also includes the address of the leader of the multicast group as well as the hop count to the leader. The RREP will come from a node which is a member of the multicast tree and meets the responding criteria. The processing and propagation of the control packets in this procedure is the same as the route discovery procedure.

Since the repair procedure is invoked by the downstream node of the broken link, it is possible that the repaired link is not through the upstream node of the broken link. In this case, the upstream node may become leaf node of the multicast tree if it has no other downstream nodes of the multicast tree. If it is not a member of the multicast group, and it will become a leaf node of the multicast tree after the link repair, it no longer needs to stay in the multicast tree. So it first waits a period of time to ensure that the link has already been repaired. After that, if it detects that it does not need to stay in the multicast tree (as described above), it begins the prune procedure.

Tree Partitions

When network topology changes in a MANET, link breakage may occur, which may also lead to the partitioning of a tree. In tree partition, the node that is a member of the multicast group which initiates the RREQ then becomes the new leader of the multicast group. To indicate so, the node increases the multicast

sequence number and broadcasts a GRPH with U flag set. Other nodes that receive the GRPH updates the route table in order to accept the new multicast group leader.

If the node initiated the rebuilding is not a group member, it chooses one of the downstream nodes and sends a MACT with G flag set to that downstream node. Then the node changes the direction to the node it chose by marking that node as an upstream node. The node receives such MACT knows that it should become the new leader ,and if it is a multicast group member, it broadcasts a GRPH with U flag set. If it is not a multicast group member, it repeats the procedure until a group member is reached.

One special case may occur during the rebuilding of multicast group. That is, the node initiating the RREQ is not a member of the multicast group, it has only one downstream node, and it did not receive any MACT packets with G flag set from its upstream node node. In this case, the node prunes itself from the tree. This is done by unicasting to its downstream nodes a MACT packet with Prune Flag set. This procedure may also repeat until the downstream node node does not meet the criteria (is a multicast group member or has more than one downstream nodes).

Studies on performance evaluation of MAODV are available in the literature, some examples are [56] and [57]. It is reported that MAODV can provide high performance on throughput in multicasting. MAODV uses core-based tree to maintain multicast group topology. When link breakage occurs, MAODV must activate link repair to recover the core-based tree since there are no redundant path in tree topology. This tree recovery procedure usually causes large amount of control packets generated, also extends the end-to-end delay in MAODV. Also, since it

is the downstream node's responsibility to repair the broken link, the retransmission must start at the multicast group leader, which further extends the end-to-end delay. Meanwhile, since MAODV intends to create the core-based tree locally, the hop count for one possible path is usually longer. Long hop count can also result in higher latency and increase the probability to handle link breakage in MANETs.

2.4.4 ODMRP

On-Demand Multicast Routing Protocol (ODMRP) [9] is a reactive multicast routing protocol using mesh to maintain multicast group topology and forwarding data packets. Forwarding groups are built to represent multicast groups in the form of mesh. Each multicast group has one forwarding group. The forwarding group includes all source nodes that have data to send to the multicast group, the nodes that belong to the multicast group, and all intermediate nodes. The data packet is flooding within the forwarding group so as to provide redundant data packet transmission.

In ODMRP, the source node is responsible for the mesh creation and group membership updating in on demand manner. When a node, called the source node, has packets to send to a multicast group, and it has no knowledge of route and group membership to that multicast group, it begins the mesh creation procedure by flooding a member advertising packet stating the multicast group. This packet is called Join Query.

The Join Query includes the address of the source node and a unique identifier generated by the source node. The unique identifier is used to detect transmission

duplication. The Join Query also contains the hop count recording the number of hops that this packet has traveled so far, and the address of the previous forwarded node (also referred to as the upstream node). The hop count and upstream node address are for the receiving nodes to update its routing table. A Time-To-Live (TTL) value is set in the Join Query to limit the packet propagation. The TTL value is usually determined by the network size so as to ensure that the Join Query could propagate to the entire MANET.

Every intermediate node received a Join Query first stores the source node address and the unique identifier of the packet to detect transmission duplication. Duplicate transmissions are discarded; otherwise, the node updates its routing table using the upstream node address of the packet as the next node towards the source node. If the Join Query has a TTL value higher than one, the node forwards the Join Query to its neighbor. The forwarded Join Query has TTL value decremented by one, the hop count increased by one, the upstream node address set to its address.

When the Join Query reaches the multicast group member, the member replies with a Join Reply. The Join Reply includes a series of $(source\ node, next\ hop\ node)$ pairs. Each pair indicates a source node which has data to send to the multicast group, and the corresponding next hop node towards the source.

The Join Reply is forwarded back to the source node. Along the way, the intermediate nodes forwarded the packet set its FG_FLAG (Forwarding group flag) to register itself as a member of the forwarding group, and other nodes detected the packet simply ignores it.

When the forwarding group is built, the source node periodically broadcasts the Join Query. Thus, the forwarding group built up procedure repeats periodically

to maintain the freshness of the membership information and update the routes if topology change occurs.

There is no need to explicitly notify the leaving of the group due to periodically updating Join Query and Join Reply. Once a source node has no data to send to the multicast group, the source node just stops sending the Join Query. For a multicast group member wanting to leave the group, it simply stops replying the Join Reply. A member of the forwarding group will leave the group only when it does not receive any Join Reply for that group within a pre-configured time interval.

The source node uses the forwarding group to propagate the data packets. When an intermediate node receives the data packet sent to one multicast group, it propagates the data packet only when its FG_FLAG to the multicast group is set.

As reported in [56], [57] and [58], we see the performance of throughput is outstanding in ODMRP comparing to other multicast routing protocols, while the amount of control packets in ODMRP is limited to a relatively low level. This is because ODMRP uses group forwarding mechanism in data transmission and mesh to maintain multicast group topology. Mesh topology gains benefits from having alternative paths to the destination. It does not need to repair a broken link when a link fails. With group forwarding mechanism, data packets flood throughout the mesh to provide redundant data transmission. Combining these two features reduces the impact of link failure on performance. However, redundant data transmission is still counted as a waste of channel resource, it can be calculated as overhead. So, if the forwarding group size is large in terms of the number of nodes, such overhead could also do harm to the performance. Another issue of ODMRP is that the Join Request is broadcasted periodically; this creates

background traffic to maintain the tree, especially when the tree size is large.

2.5 Hybrid Routing Protocols

The hybrid routing protocol design for MANETs was first introduced by Haas in 1997. They proposed Zone Routing Protocol (ZRP) [59]. They further improved ZRP and the most current version of ZRP is [14]. The design of ZRP tries to combine proactive behavior and reactive behavior together by limiting the proactive behavior into a fix propagation range. They even improved the concept of ZRP and designed Independent Zone Routing (IZR) protocol [60] in 2004. In IZR, the limited propagation range for proactive behavior dynamically changes. Also, an enhanced version of ZRP to provide multicast support, Multicast Zone Routing Protocol (MZR) [7], was proposed by Devarapalli *et al.* in 2001. MZR adopts the limited proactive range concept from ZRP and expands both the proactive and reactive behavior to enable multicast transmission.

Sharp Hybrid Adaptive Routing Protocol (SHARP) [61] is another hybrid routing protocol for MANETs. It was designed by Ramasubramanian *et al.* in 2003. In SHARP, some selected nodes perform proactive behavior and all other nodes perform reactive behavior to achieve balance on the trade-off between proactive and reactive behaviors. In addition, some other works were done to design hybrid routing protocols from other aspects. For example, An and Papavassiliou proposed Mobility-based Hybrid Multicast Routing (MHMR) protocol [62] in 2001. MHMR is a hierarchical hybrid routing protocol which builds up hierarchical clusters and performs similar to traditional cellular networks.

2.5.1 ZRP

Zone Routing Protocol (ZRP) [14] is a popular hybrid unicast routing protocol framework in MANETs. In ZRP, nodes perform proactive and reactive behaviors simultaneously so as to benefit from both behaviors. Since the main drawback of proactive behavior is its large amount of control packets generated for routing information exchange, ZRP introduces a zone routing concept which aims to limit the proactive information exchange locally, so as to minimize the amount of control overhead.

The zone routing concept defines a zone area for every node in ZRP. A uniform parameter R, zone radius, is set to every node in the network. The zone radius is calculated in terms of hop count. With this zone radius, a node can determine its zone area. Zone members of a node are all nodes that can be reached by that node within R hops. The zone radius is determined by the network conditions, for example, the density of the network, the mobility of the nodes within the network, and others.

Each node periodically sends update packet to maintain the topology information of its zone. The propagation range of the update packet is limited within the zone area. We assume that the link status is bidirectional, so when a node becomes a zone member of some nodes, these nodes automatically become the zone member of that node. Thus, with this zone updating mechanism, each node can have the topology information of its zone area.

The zone topology information each node maintains is used for the reactive behavior in route discovery and route maintenance. When a node has data to send to a destination, if the destination is a zone member of the source node, the

source node can directly send the data to the destination with known path within the zone. Otherwise, the source node invokes a reactive route discovery procedure by broadcasting a request packet. Unlike the pure reactive protocols that only the destination can reply the request packet, all zone member of the destination can also reply to that request. Thus, once the route between the source node and a zone member of the destination is built up, the route between the source node and the destination is also built up.

ZRP is separated into two parts, the Intrazone Routing Protocol (IARP) [63] and the Interzone Routing Protocol (IERP) [64]. IARP defines the criteria of the proactive behavior of ZRP, and IERP defines the criteria of the reactive behavior of ZRP. Any practical proactive routing protocols can be converted to IARP, and any practical reactive routing protocols can be converted to IERP. The conversion needs to follow some guidelines.

When a proactive routing protocol is converted to IARP, the periodically update packet must be set to a Time-To-Live value which equals to R. When a zone member receives the update packet, it updates its zone routing table to record the routing information in the update packet. The node can forward the update packet only when the TTL value of the packet is larger than 1. The source node of the update packet may have routing information towards the node which is outside of its zone, such information cannot be included within the update packet. The routing information included in the update packet must be compatible to the request of IERP so that IERP can use such information for route discovery and maintenance. Every node must periodically refresh its zone routing table to remove possible routing information of the nodes outside the zone.

When a reactive routing protocol is converted to IERP, it must provide a mech-

anism to import the zone routing table into its reactive routing table. In route discovery and route maintenance procedure, the IERP must be able to make use of such zone routing information. The IERP must incorporate a Broadcast Resolution Protocol (BRP) [65] as its underlying broadcast service. That is, when a node wants to broadcast a control packet in IERP, the BRP takes the responsibility for the broadcasting. The BRP aims to eliminate the redundant control packet transmission in broadcasting with the zone routing information. For example, when a node wants to broadcast a request packet, it first builds up a broadcast tree with itself as the root within its zone. The broadcast tree covers all peripheral nodes (the nodes whose hop counts to that root node are equal to R). Then the node sends the request packet to the neighbors in the broadcast tree. When a node receives a new request packet, it first checks whether it is within the broadcast tree of the forwarder of the packet. If so, the node discards the received packet. Then the node checks whether it can reply to the request packet. If not, the node builds up its broadcast tree for forwarding the packet. The broadcast tree covers all peripheral nodes which are not within the zone area of the forwarder. Then the node forwards the request packet to the neighbors in its broadcast tree. The procedure of broadcasting reply packets or other packets are the same as above. The IERP must incorporate the BRP as its underlying broadcast service.

Some works [55], [66] have been conducted to evaluate the performance of ZRP. These works reveal that the zone radius and update interval of IARP are two key points in ZRP. Different conditions need different parameter settings in ZRP. Inappropriate parameters can be disastrous in ZRP. For example, large zone radius and high update frequency in a dense network can cause network channel exhaustion, and downgrade the throughput in ZRP. Also, proper selection of IARP

and IERP implementation is another important issue in ZRP. Based on different protocols to implement the IARP or IERP, the benefits of the base protocol can be introduced into ZRP as well as its drawbacks.

2.5.2 MZR

Multicast routing protocol based on Zone Routing (MZR) [7] is designed to provide multicast routing support for ZRP. MZR incorporates the zone routing concept from ZRP to maintain multicast routing information within the zone. MZR uses an on-demand method to build a source-based tree to provide routes and propagate data packets.

The zone routing in MZR is the same to ZRP. The difference of MZR and ZRP is their reactive behavior.

Multicast Tree Creation

When a node wants to send data to the multicast group, it begins to create the source-based multicast tree. First the node unicasts a TREE-CREATE packet to each node within its zone. As the TREE-CREATE packet is propagated to a zone node, reverse route entries are created at each intermediate node. When a node which is interested in the multicast session receives the TREE-CREATE packet, it creates a multicast route entry and replies to the source a TREE-CREATE-ACK packet. As the TREE-CREATE-ACK packet reaches the source node, the corresponding multicast route entry at each intermediate node is completed and activated. When the multicast route entry is activated at a node, it means a tree branch on which data can be forwarded is created.

Once the multicast tree is created within the zone, the source node propa-

gates the tree to the entire network. The source node sends a TREE-PROPAGATE packet to each peripheral node. On receiving the TREE-PROPAGATE packet, the peripheral node creates a multicast route entry for the session and the peripheral node begins to build the multicast tree in its zone.

Such process repeats until the multicast tree within the entire network has been built.

Tree Refresh

Each multicast route entry at a node has a timer to indicate its lifetime. The entry will be removed once the timer expires. This timer is used to eliminate stale multicast routing information. In order to maintain the source-based tree, the source node periodically sends TREEREFRESH packets down the tree. When a tree member receives the TREEREFRESH packet, it updates the timer of the corresponding multicast route entry. Once the source node decides to stop data transmission, it simply stops this refresh procedure. All tree member nodes then remove the corresponding multicast route entry when the timer expires.

Reaction to Link Breaks

When a broken link is detected, the downstream node of the broken link is responsible for reconfiguring the tree. The node first broadcasts a JOIN-PACKET in its zone to find a multicast group member in the zone. When a node within the zone receives the JOIN-PACKET and it is a multicast group member, it replies a JOIN-ACK packet to the node initiates the repair process. Thus the broken link of the tree is repaired.

If the node receives no replies within its zone, it propagates the join request throughout the entire network. It sends a JOIN-PROPAGATE packet to the peripheral nodes. These peripheral nodes search in their own zones to find a mul-

ticast member. If so, the node replies a JOIN-ACK to the source node of the JOIN-PROPAGATE. If multicast group member is not found, the node continues to propagate the join request procedure until some nodes find a node within their zones that belongs to the multicast group, the JOIN-ACK is sent back to the node which performs the repair procedure and the tree is repaired.

Tree Prunes

When a tree member wants to leave the tree and it is a leaf node of the tree, it sends a TREE-PRUNE packet to its upstream node in the tree and removes the corresponding multicast route entry. The upstream node on receiving the TREE-PRUNE packet removes the link of the downstream node in the corresponding multicast route entry. If the upstream node turns into leaf node and it is not a multicast group member, it repeats the prune procedure.

2.5.3 MHMR

Mobility-based Hybrid Multicast Routing (MHMR) [62] protocol is a hierarchical hybrid multicast routing protocol for MANETs. MHMR inherits the concept from traditional cellular networks for routing maintenance and data transmission.

Considering the traditional cellular networks, the whole network area is geographically separated into cells. Each cell has one fixed base station built to cover the whole area of the cell. Every mobile node within a cell communicates directly with the base station of the cell. If a mobile node wants to communicate with another node, the node sends the data to its base station and the base station is responsible for forwarding the data packet to the destination. If the source node and the destination are within the same cell, the base station simply send the data to

the destination. Otherwise, the base station propagates the data to the base station of the destination cell, and then that base station sends the data to the destination.

Similar to the traditional cellular networks, MHMR separates the nodes of the whole MANET into groups, which is called clusters. MHMR uses geographical location information to aggregate nodes into clusters. The geographical location information is maintained by special location-aware devices equipped in all nodes, i.e. GPS devices. Each cluster has one cluster head performs the similar jobs of base station in cellular networks. The cluster head acts as a local coordinator in terms of transmissions within the cluster. Unlike the traditional cellular networks, the cluster head is not a special built, fixed location data access point. Instead, the cluster head is dynamically selected among all nodes within the cluster using a specific mobility-based cluster algorithm [67].

MHMR also maintains a mesh to represent the topology of a multicast group. Unlike other mesh-based multicast routing protocols that the mesh includes all member nodes of the multicast group together with all intermediate nodes, only cluster heads can be a mesh member in MHMR.

Every cluster head maintains a multicast routing table to store the multicast information. If a node of one cluster wants to join or have data packets to one certain multicast group, it notifies its cluster head of this intension. If the cluster head has no knowledge about the destination multicast group in its multicast routing table, the cluster head sends out a JOIN REQUEST to all its neighbor cluster heads (the cluster heads of the neighbor cluster). Once a cluster head receives a JOIN REQUEST, if it also has no knowledge of the multicast group indicated in the JOIN REQUEST, it records this REQUEST for reverse path then forwards the REQUEST to the neighbor cluster heads.

If a cluster head is a member of the multicast group mesh according to the received REQUEST, the cluster head updates its multicast routing table then generates a JOIN REPLY to response the received REQUEST. The JOIN REPLY is forwarded along the reverse paths until it reaches the source cluster heads. All the cluster heads along the reverse paths also update their multicast routing table to record this mesh information.

The cluster head periodically sends JOIN REQUEST to maintain the mesh. If a source node wants to leave the multicast group, it sends the cluster head a stop message. Then, the cluster head stops sending JOIN REQUEST. After some time, when the neighbor cluster heads within the same mesh stop detecting JOIN REQUEST, the neighbor cluster heads update their multicast route table accordingly.

Once the mesh is built, the source node can begin the data transmission. The source node sends the data packets to its cluster head (the source cluster head). The cluster head then calculates a sub-tree of the destined multicast group mesh according to its multicast routing table. The sub-tree covers all cluster heads belong to the mesh, and the root of the sub-tree is the source cluster head. Then the data packets are propagated along the selected sub-tree. Upon receiving the data packet, the cluster head disseminates the data packet to multicast group members within its cluster.

2.5.4 SHARP

Sharp Hybrid Adaptive Routing Protocol (SHARP) [61] is a hybrid adaptive unicast routing protocol for MANETs. Similar to ZRP, SHARP also uses zone rout-

ing concept to limit the range of proactive behavior. The zone radius is calculated in terms of hop count. Unlike ZRP, the zone radius is not uniformly pre-configured by the protocol in SHARP. On the other hand, each node dynamically determines its own zone radius. The determination of zone radius is based on the upper layer application requirements of the node. Thus, when a node (core node) decides its zone radius, every node whose hop count to the core node is no more than the zone radius is the zone member of the core node.

The key concept of SHARP is that only selected nodes perform proactive behavior. When a node is likely to receive data from many sources (hot destination), the node determines its zone radius and invokes the proactive behavior. All zone members of the hot destination then perform proactive behavior until the hot destination decides to stop being proactive. During the proactive behavior, a directed acyclic graph (DAG) is constructed within the zone. The DAG is rooted at the hot destination. Every zone member has at least one route to the hot destination based on the DAG. On the other hand, all other nodes which are not members of any hot destination perform reactive behavior for route discovery.

The proactive behavior in SHARP is derived from DSDV and Temporally Ordered Routing Algorithm (TORA) [19]. The authors named the proactive behavior SHARP Proactive Routing (SPR) protocol. SPR consists of two components, *construction protocol* and *update protocol*. The *construction protocol* uses flooding throughout the zone to build up the DAG, and the *update protocol* uses one hop broadcasting to refresh routing information and handle link breakage. The update interval of *construction protocol* is greater than that in *update protocol* since flooding throughout the zone usually cause large amount of control packet generated. In the following description, we use two terms, upstream node and

downstream node, to identify one hop neighbors of a node in the DAG. For node i, its upstream nodes are the nodes which are the next hops of node i towards the core node; and its downstream nodes are the nodes which choose node i as their next hop towards the core node.

Each node within the zone (including the zone members and the core node) maintains a zone routing table to record the DAG information. The zone routing table of a node includes a downstream node list which contains all downstream nodes in the DAG, an upstream node list which contains all upstream nodes in the DAG, and a hop count which indicates the distance from the node to the core node. Each item in the upstream node list and downstream node list indicates the link to one of its neighbors in the DAG, and the item has an active label to indicate the validity of this link. Explicitly, when an item is inserted into the upstream node list or downstream node list, it is labeled active automatically.

The core node performs the *construction protocol* by periodically broadcasting a *construction packet* to all zone members. The *construction packet* includes the core node address, a sequence number which identify the *construction packet*, a Time-To-Live (TTL) value which is set to the zone radius. The TTL value indicates the life time of the packet as well as the hop count of the packet. The hop count of the packet is calculated as $hop\ count = zone\ radius - TTL + 1$. On receiving the *construction packet*, each zone member processes the received packet and updates it upstream node list and down stream node list accordingly. Finally, all zone members receive the *construction packet* and the DAG is built up.

After that, every node within the zone performs *update protocol* to maintain the DAG. Each node periodically broadcasts an *update packet* to all its one hop

neighbors within the zone. The *update packet* includes the upstream node list, the downstream node list and the hop count of the node.

When a node receives an *update packet* from its neighbor, the node updates its upstream node and downstream node list according to the received packets.

Thus, the routing information of the DAG is refreshed and any new link to the DAG is added. When a node finds that there is one link within its upstream node list or downstream node list is labeled de-active for several rounds of updating, it indicates that the link is broken. If the broken link occurs in the downstream node list, the node simply removes this link from the downstream node list. If the broken link occurs in the upstream node list, and there are other upstream nodes within the upstream node list, the node can simply removes the broken link since alternative path to the core node exists. Until links to all upstream nodes are broken, the node increases its hop count by 2 and reverses all its links to the downstream nodes. Then, it generates an *update packet* and sends it to all its downstream nodes to inform this link reverse.

From the description above, we can see that the link breakage recovery in SHARP may result in a non-optimal DAG (Not all paths in the DAG are shortest). But with the *construction protocol*, such non-optimal DAG only exists between two consecutive constructions, thus minimizes the impact of the non-optimal DAG.

The reactive behavior in SHARP is derived from AODV. When a node wants to send data to a destination, if it is in reactive behavior, or it is a zone member but the destination is not the core node of the zone, the node broadcasts an RREQ to find the route. When a node receives an RREQ and it is the destination or it is a zone member and its core node is the destination, it replies the RREQ to build up the route. Besides, any other reactive routing protocol can be introduced into

SHARP to replace AODV as the reactive behavior.

2.6 Power-Aware Routing Protocols

The limitation of the battery lifetime creates a great challenge to routing protocol design in MANETs. Because of the limited battery lifetime for each mobile device, reducing the power consumption which in turns extending the battery operating hours for a mobile device can be an important consideration in designing routing protocols in MANETs. Thus, a number of routing protocols aiming on power-awareness have been proposed in the literature.

Usually, power-aware routing protocols can be distinguished into two types based on their route selection mechanism. The first one aims to select a route with minimal transmission power. The other is going to select the route has the maximal residual battery power. Minimum Total Transmission Power Routing (MTPR) [68], which is proposed by Scott and Bambos in 1996, is a typical power-aware routing protocol of the first type. Singh *et al.* proposed Min-Max Battery Cost Routing (MMBCR) [69] in 1998 which represents the second type. But both of these two types have some potential constraints that they only consider the transmission power or battery lifetime of one route. Since the nodes along the route may have different transmission environments or not only join one data transmission, the selected route may not be the optimal choice while considering transmission scenario. Some power-aware routing protocols were proposed to overcome the constraints. In the following part, we will describe two well designed protocols which take both the power conservation and traffic scenario together into consideration. One is the Maximum Residual Packet Capacity (MRPC) routing

protocol [70], proposed by Misra and Banerjee in 2002; the other is the Minimum Drain Rate (MDR) routing protocol [71], proposed by Kim *et al.* in 2002.

2.6.1 MRPC

Maximum Residual Packet Capacity (MRPC) [70] is a power based routing protocol which aims to select a route that maximizes the residual battery power currently available at the most critical node (the one with the least residual battery power). To achieve this goal, MRPC selects a route which maximizes the total number of packets that can be transmitted over it. In MRPC, it is assumed that all other transmissions through the nodes along the selected route are stopped so as to limit the route selection along the path. Particularly, MRPC does not define any specific routing selection algorithm. Instead, MRPC intends to define a power-aware metric for use during routing selection.

To formalize this concept, we assume that the residual battery power at node i is B_i. Also, we assume that the transmission energy for node i to transmit a packet to node j over link (i, j) is $E_{i,j}$. Note that in MRPC, we assume that a node can dynamically adjust its transmission power to one of its neighbors based on the their distance. Thus, $E_{i,j}$ is $\propto d^K (K \geq 2)$, where d is the distance between node i and node j.

With given B_i and $E_{i,j}$, the maximum number of packets that node i can forward through (i, j) is $\frac{B_i}{E_{i,j}}$. Accordingly, we can define a node-link cost metric, $C_{i,j}$ for the link (i, j) as

$$C_{i,j} = \frac{B_i}{E_{i,j}}. \tag{2.1}$$

The key point in this formulation is that the cost metric includes both a node-specific parameter (the battery power) and a link specific parameter (the packet transmission energy across the link).

Let S and D denote the source and destination nodes for a specific route respectively. P is the the path from S to D which includes link (i, j). P is selected by a routing selection algorithm. Clearly, the node with the smallest value of $C_{i,j}$ along a chosen P determines the maximum number of packets that may be potentially forwarded through P. We use the term "lifetime" to indicate the maximum number of packets forwarded through a path, so the "maximal lifetime" associated with route P is seen to be

$$Life_P = \min_{(i,j)\in P}\{C_{i,j}\}. \tag{2.2}$$

The MRPC algorithm then selects the route $P_{condidate}$ that maximizes the "maximal lifetime" of communication between S and D.

$$P_{condidate} = \arg\max\{Life_p \mid P \in all\ possible\ routes\}. \tag{2.3}$$

Note that B_i and $E_{i,j}$ subject to the time of computation, the random traffic patterns will potentially make the currently selected paths less optimal in the future. Since MRPC is essentially a power-aware metric definition for routing selection, a routing protocol that uses MRPC for multi-hop wireless networks will include mechanisms for periodic and distributed route calculation.

One important issue to be concerned in MRPC is retransmission caused by possible packet transmission error over links. Since smaller d causes smaller $E_{i,j}$ which essentially increases $Life_P$, a selected P from S to D is more likely to be

consisted with a bunch of short links, that is increases the hop counts of P. Due to the nature of MANETs, large hop counts can essentially increase the possibility of retransmission, and retransmission can also consume batter power. Thus, a route P may not be optimal without concerning retransmission.

To avoid this constraints, we calculate $E_{i,j}$ with the following equation [72].

$$E_{i,j} = \frac{T_{i,j}}{(1 - p_{i,j})^L}. \tag{2.4}$$

$T_{i,j}$ is the energy involved in a single packet transmission attempt, and essentially $T_{i,j} \propto d^K (K \geq 2)$. $p_{i,j}$ is the packet transmission error possibility over link (i, j). L is a constant parameter where $L \in \{1, 2, ...\}$. in [72], L is 1 when hop-by-hop retransmissions (a reliable link layer) are present; when in the absence of hop-by-hop retransmission, the transmission cost is well approximated by $L \in \{3, ..., 5\}$.

2.6.2 MDR

The Minimum Drain Rate (MDR) [71] routing protocol incorporates the drain rate metric into the routing selection procedure. The drain rate denotes one node's power consumption speed which is determined by not only traffic through one route, but also all possible traffic through this node. MDR defines the calculation of drain rate and presents the criteria of route selection using drain rate. Any underlying routing protocols can incorporate MDR as their route selection criteria and the operation of them do not need to change.

When the remaining power is the only metric used to determine the best route for one data transmission, this route may not be optimal due to other data trans-

missions through nodes along this selected route. Moreover, if a node is selected along a route for one data transmission because the node has large amount of residual battery power, other data transmissions are more willing to chose this node along their routes if possible. In this case, such node exhausts its battery power quickly due to heavy traffic load on it.

Thus, a route selection metric which considers not only the residual battery power but also traffic load information is needed. The MDR defines the energy drain rate as the route selection metric. The energy drain rate of a node indicates how much average energy is consumed by the node per second. To calculate the energy drain rate, Each node n_i monitors its energy consumption in a given past interval and calculates the energy drain rate value of this interval DR_{sample} by averaging the amount of energy consumption into every second of the interval.

Every T seconds, n_i computes its energy drain rate, denoted by DR_i, by utilizing the well-known exponential weighted moving average method (see 2.5). For example, at time t, n_i wants to calculate the new DR_i of this time. With the most newly calculated DR_{sample} and the previous drain rate DR_{old} calculated at time $t - T$, the new drain rate is calculated as

$$DR_i = \alpha \times DR_{old} + (1 - \alpha) \times DR_{sample}. \tag{2.5}$$

Note that, DR_i is subject to the calculation time, thus, a route selected at one time may not be optimal with time goes by. To mitigate this problem, underlying routing protocols should provide techniques to take into account the continuously changing drain rates of network nodes for obtaining new routes if needed. In proactive routing protocols, all nodes should maintain the route and update the

power information periodically. In reactive routing protocols, all source node should periodically perform route discovery in order to find a new route even when no link breakage occurs.

The ratio $\frac{RBP_i}{DR_i}$, where RBP_i denotes the residual battery power at node n_i, indicates when the remaining battery of node n_i would be exhausted. In other words, this ratio represents how long the remaining energy can keep up the connections with current traffic condition. The corresponding cost function can be defined as

$$C_i = \frac{RBP_i}{DR_i}. \tag{2.6}$$

The maximum lifetime of a given path r_P is determined by the minimum value of C_i over the path, that is,

$$L_P = \min_{\forall n_i \in r_P} C_i. \tag{2.7}$$

The MDR therefore intends to select the route r_M with the highest maximum lifetime value among all possible routes r_* between the source and the destination nodes, that is,

$$r_M \doteq r_* = \max_{\forall r_i \in r_*} L_i. \tag{2.8}$$

Finally, MDR does not guarantee that the total transmission power is minimized over a chosen route. However, based on a γ threshold, MDR can be applied when all routes have nodes with low battery capacity (i.e., below the threshold) in order to prolong the lifetime of both nodes and connections as well as to minimize

the total transmission power consumed per packet.

Chapter 3

Design Analysis of New Hybrid Multicast Routing Protocols

In this chapter, we describe our two newly proposed hybrid multicast routing protocols, which are ZMAODV and ZODMRP [6]. When they are properly designed, hybrid routing protocols have the potential to perform better than a pure reactive or proactive protocol in the sense that they try to strike a balance between improved delay performance and lower overhead activities. Lowering the control packet overheads is known to extend the longevity of the battery power of mobile nodes. To name one, the Multicast Zone Routing (MZR) protocol [7] proposes an elegant hybrid multicast routing protocol based on the concept of zone routing, where the protocol behaves proactively within the zone radius to maintain routing information constantly. When a data communication is needed, the protocol uses a source-based tree to find the path to the destination.

Multicast routing protocols have a unique characteristic that the destination is constituted by a group of designated nodes. Thus, building up and maintaining a

multicast group topology for data transmission are important issues in multicast routing protocol design. Commonly, the multicast group members are structured into one of the three topologies, namely, the source-based tree (e.g. [73], [74]) with the source node as the root of the tree, the core-based tree (e.g. [75], [76]) with the core node as the root, and the mesh(e.g. [77], [78]).

The source-based tree requires a tree setup for every source node to a multicast group with the source node as the root of the tree. This setup provides the shortest path from the source node to all intended destinations since the tree establishes shortest paths from the root to all intended destinations. However, in case of multiple source nodes in a multicast group, multiple trees will be established leading to excessive control packets.

On the other hand, in the core-based tree, only one tree is setup for one multicast group. Each source node must establish a path to the core node to disseminate its data packets to all intended destinations via the core node. Using a single tree for a multicast group helps reduce control packets for the tree management. However, all data packets are channeled to the core node creating a bottleneck in packet dissemination.

Unlike the tree topology, in the mesh topology, all the source nodes, members of the destination group and the intermediate nodes are forming a mesh topology for data transmission. Although duplicated data packets occur in mesh topology, alternative routes to the destination is the advantage of mesh topology as they minimize the impact of link failure. However, massive redundant data transmission may cause starvation of channel bandwidth. The extreme case of the mesh topology is that the mesh consists of all nodes within the network, thus, the data transmission in this mesh becomes network-wide flooding. Thus, the key point

of the mesh topology is to design a mesh creation algorithm which can achieve balance between the alternative paths and the redundant data transmission.

Since MZR uses a simple path finding procedure, and the multicast data delivery tree it uses is the source-based tree, the reactive behavior can be a drawback of the performance for MZR. We believe that combining the zone routing feature from MZR with well-performing reactive behavior which uses other data transmission topology should result in better performance. In our design, we propose two hybrid routing protocols, namely ZMAODV and ZODMRP, where ZMAODV combines zone routing with Multicast Ad hoc On-Demand Distance Vector (MAODV) [8] routing protocol, and ZODMRP combines zone routing with On-Demand Multicast Routing Protocol (ODMRP) [9]. One reason for choosing MAODV and ODMRP is because they are shown to perform well in the literature [56], [57], [58]. The fact that different multicast strategies employed in MAODV and ODMRP, namely, MAODV uses core-based tree and ODMRP uses mesh, allows examination of the impact of different employed strategies on the performance. We also want to examine that by adding proactive behavior in MAODV and ODMRP, what would be the effect to these two well-designed reactive protocols.

After describing the two routing protocols, we study their performances using GloMoSim [79] based simulations. Simulation comparison among ZMAODV, ZODMRP, MAODV, ODMRP ,MZR and MOLSR [12] are given. It is shown that our proposed protocols outperform their predecessors. The performance study of these two protocols also form the basis of the behavior selection for our polymorphic routing protocol design that is introduced in the next chapter.

3.1 Description of Zone Routing

Our proposed hybrid routing protocols use zone routing as their proactive behavior. In this section, we provide description of zone routing. In zone routing, Each node maintains a Zone Routing Table(ZRT) to store the routing information and multicast information, and periodically sends messages to the nodes within its zone, in order to exchange the topology and multicast routing information with the nodes within its zone. The data and packet structures of our protocols are described in the following.

3.1.1 Zone Routing Table entries

Each entry in the ZRT of a node has the following structure.

- Destination:

  This field contains the address of the destination, and the other information about the destination.

- Nexthop:

  This field contains the address of the next hop that leads to the destination. When the node has packets to send to the destination, it sends the packet to the nexthop.

- Hop Count:

  This field contains the number of hops from the node to the destination.

- Lifetime:

This field contains the last update time of the corresponding entry. The purpose of this field is to ensure that the information in this entry remains current. Otherwise, the information will be removed.

- Multicast Information:

  This field holds a list of records that contains the multicast information of the destination.

  Each record of the multicast information has the structure described below:

  - Multicast Group Address:

    This field contains the multicast group address that the destination belongs to.

  - Multicast Group Leader Address:

    This field contains the address of the leader of the multicast group to which the destination belongs. This field is used only in ZMAODV.

  - Multicast Group Sequence Number:

    This field contains the current sequence number of the multicast group. This field is also used only in ZMAODV.

3.1.2 Packet Structure

- The structure of Zone Update packets used in ZMAODV is shown in Fig. 3.1. The fields in the packet are explained as follows.

  - Type

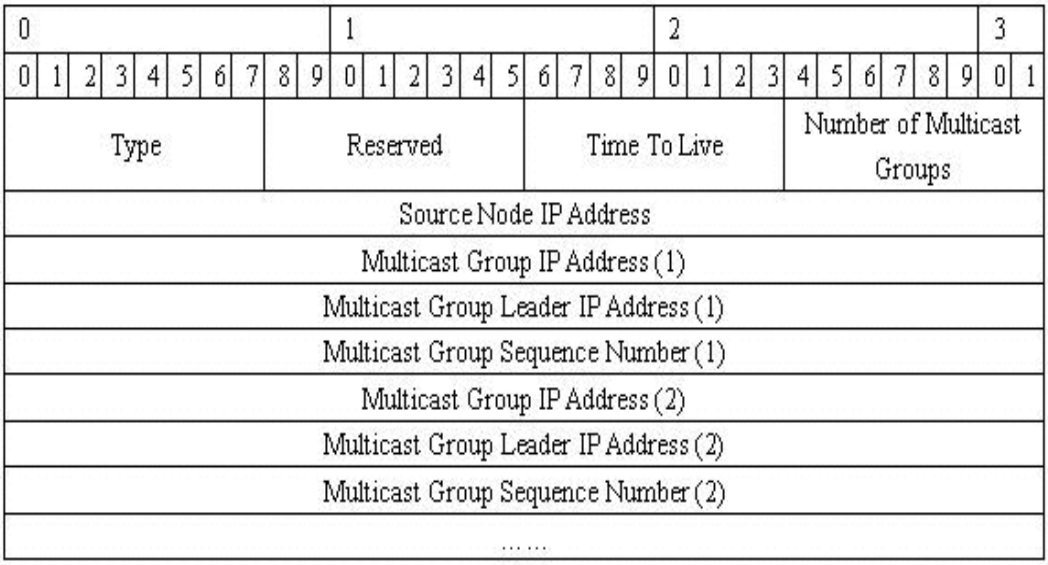


Figure 3.1: Packet structure of ZMAODV.

This section is set to 7 in ZMAODV. It indicates that the packet is for Zone Update.

– Reserved

  This section is reserved. It does not carry any useful information.

– Time To Live (TTL)

  When a node originates (originator) the Zone Update packet, it sets the TTL equal to the Zone_Radius. As the packet propagates each hop, the value is subtracted by 1. The packet is discarded when the TTL reaches 0.

– Number of Multicast Groups

  Since the originator may belong to several multicast groups, this field indicates the number of multicast groups to which the source node belongs.

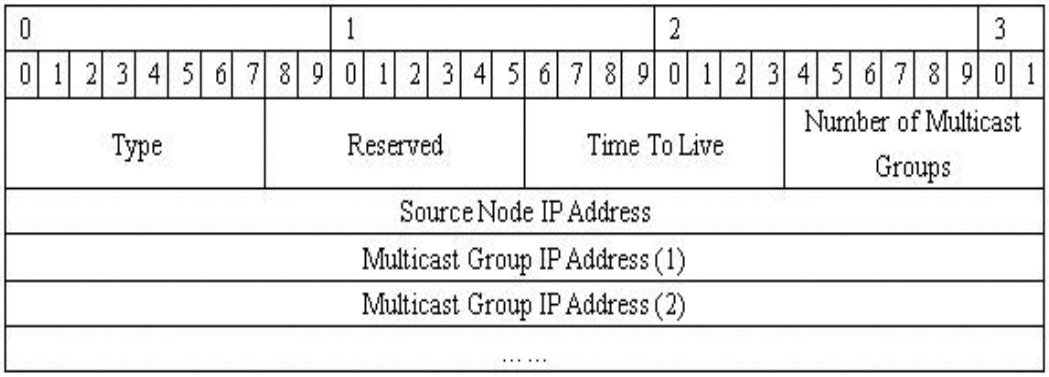


Figure 3.2: Packet structure of ZODMRP.

- Source Node IP Address

  This field stores the IP address of the originator.

- Multicast Group IP Address

- Multicast Group Leader IP Address

- Multicast Group Sequence Number

  These three sections build up a Multicast Group Information set which contains the multicast group information the originator belongs to. Since the originator may belong to several multicast groups, each set contains one multicast group information and the total number of sets is equal to the Number of Multicast Groups field.

- The structure of Zone Update packets used in ZODMRP is shown in Fig. 3.2 with descriptions given in the following.

  - Type

    This field is set to 4 in ZODMRP. It indicates the packet is for Zone Update.

- Reserved

  This field is reserved. It does not carry any useful information.

- Time To Live (TTL)

  The originator of the Zone Update packet sets the TTL equal to the Zone_Radius. As the packet propagates each hop, the value is subtracted by 1. The packet is discarded when the TTL reaches 0.

- Number of Multicast Groups

  Since the originator may belong to several multicast groups, This field indicates the number of multicast groups to which the source node belongs.

- Source Node IP Address

  This field stores the IP address of the originator.

- Multicast Group IP Address

  This field contains the multicast group information to which the originator belongs. Since the source node may belong to several multicast groups, this field repeats until all multicast group IP addresses are specified in the packets. The total number of multicast group IP Addresses is equal to the Number of Multicast Groups field.

3.1.3 The Operation of Zone Updating

- **Initiate Zone Update Packet**

Each node within the network periodically sends out the Zone Update packet. The node first evaluates its multicast routing table to obtain the correspond-

ing information, then it builds up the Zone Update packet by appropriately inserting corresponding information into the packet. The type field is set to the corresponding number based on the used protocol, that is, 7 in ZMAODV or 4 in ZODMRP. The TTL field is set to the Zone_Radius as pre-configured. The Source IP Address field is set to the address of the node itself. The Number of Multicast Groups section is set to the number of multicast groups to which this node belongs, and this value carries the total number of records in its multicast routing table. After the packet is constructed, the node broadcasts the packet to all its neighbors.

- **Update Zone Routing Table and Forward the packet**

  Each node receives the Zone Update packets from its neighbors. After receiving a Zone Update packet, it first evaluates its ZRT to find out whether there is an entry whose Destination field equals to the Source Node IP Address section of the received Zone Update packet. If there is no such entry, the node inserts a new entry in the ZRT to store the information within the Zone Update packet. Otherwise, the node compares the Hop Count field of the found entry with the TTL value of the received packet so as to determine whether the corresponding entry needs update. If the value of the Hop Count field is larger or equal to $Zone\_Radius - TTL + 1$, the node updates its corresponding entry.

  When updating an existing entry or inserting a new entry into ZRT, the updated information is based on the received Zone Update packet. The Destination field is set to the Source Node IP Address section of the received packet. The Nexthop field is set to the IP address of the node from which the

update packet is received. The Hop Count field is set to $Zone\_Radius - TTL + 1$. The Lifetime field is set to the current time. The Multicast Information field copies the Multicast Group Information sections in ZMAODV or Multicast Group IP Address sections in ZODMRP.

After updating the ZRT, the node decrements the TTL value of the packet by 1. If the value is 0, the node does nothing. Otherwise, the node broadcasts the packet to its neighbors.

- **Remove Stale Entries**

  Periodically, each mobile node checks its ZRT to remove any stale information. A per-configured parameter Zone_lifetime is used to determine whether the information is stale. The node compares $t_{current} - t_{lifetime}$ with Zone_lifetime, where $t_{current}$ is the current time and $t_{lifetime}$ is the value recorded in the Lifetime field in the ZRT. If $t_{current} - t_{lifetime}$ is larger than or equal to Zone_lifetime, the entry is considered stale and should be removed from the ZRT.

3.2 Reactive Features

This section presents the reactive features of ZMAODV and ZODMRP. The zone routing has already provided a node the multicast information of nodes within the zone, such information helps the node discover a route. The only modification to the protocol operation is the route finding process in MAODV and ODMRP for ZMAODV and ZODMRP respectively. The details of this modification are presented below.

3.2.1 ZMAODV

The data and packet structures in ZMAODV is the same as that of the MAODV. We only modify the process of sending Route Request (RREQ) in ZMAODV.

When a node needs to send a RREQ, it first looks in its ZRT to see whether there are nodes in the zone that belong to the multicast group which the node intends to join or has packets to send to. When it finds some nodes that belong to the multicast group in its zone, it compares other information, such as the multicast group sequence number and the multicast group leader address, to ensure that the information recorded in the zone routing table remains valid. Thereafter, the source node unicasts a RREQ to the nearest found node and then waits for the reply.

If it finds no nodes in its zone, the source node broadcasts a RREQ setting its TTL value to Zone_Radius to ensure the broadcast is constrained in the zone. Only the border nodes handles the RREQ. The border nodes are the nodes that have hop count to the source node equals to the Zone_Radius. The other nodes in the zone just propagate the RREQ to the border nodes. When a border node receives the RREQ, it looks up its ZRT and continues the path finding procedure until some nodes belonging to the multicast group receive the RREQ. These nodes then generate Route Replies (RREPs) and send the RREPs to the source node following the reverse path.

When a node wants to send RREQ to either join a multicast group, send packets to a group, or fix a detected broken link, it follows the procedure described above. As for the other operations, such as sending RREP, sending MACT, group hello, detecting and repairing broken link, and selecting group leader, are all based

on the original MAODV.

3.2.2 ZODMRP

The process of sending Join Query in ZODMRP is similar to that of sending RREQ in ZMAODV. The difference is that when a node finds other nodes in its zone belonging to the multicast group, it does not only unicast the Join Query to the node with the shortest path but also to all nodes belonging to the multicast group. This helps build up the forwarding group as a mesh.

3.3 Simulation and Analysis

3.3.1 Simulation Scenarios

In the simulation, we compare ZODMRP, ODMRP, ZMAODV, MAODV, MZR and MOLSR. The simulation of these protocols was implemented within the GloMoSim [79] library. The parameters of simulation are listed in Table.3.1.

The following metrics are used in the performance evaluation:

- Packet delivery ratio;
- Number of control packets transferred per data packet delivered;
- Number of control + data packets transferred per data packet delivered;
- Average end-to-end delay.

We evaluate the above metrics against mobility speed, number of senders and the multicast group size.

Total number of nodes	50
Simulation area	1000m×1000m
Propagation range	225m
Channel capacity	2Mbps
Simulation time	500s
MAC protocol	The IEEE 802.11 MAC [1]
Traffic type	constant bit rate (CBR)
Mobility model	random waypoint model [80]
Pause time	0s
Zone update interval	5s
Zone radius	3
Zone lifetime	180s
Packet sending rate	10packets/s
Packet size	512 bytes

Table 3.1: The parameters for the simulation.

3.3.2 Results and Analysis

1) Mobility Speed

Experimental Scenario:

Each node moves with a predefined speed as the maximum speed. The maximum speed is varied from 0m/s to 60m/s. In this scenario, we had 20 multicast members and 5 source nodes.

Fig. 3.3 illustrates the packet delivery ratio of the protocols under different speeds. It can be noticed first that the ODMRP group (i.e., ODMRP and ZODMRP) shows better performance than the MAODV group (MAODV and ZMAODV). The ZODMRP performs better than the pure ODMRP, while the ZMAODV achieves slightly gains against the MAODV. The MZR performs the worst. Its performance degrades sharply with speed increase. We can also see that MOLSR has a good performance in this scenario. Since MOLSR is a proactive routing protocol, it provides robustness when mobility speed is high which usually leads to large amount

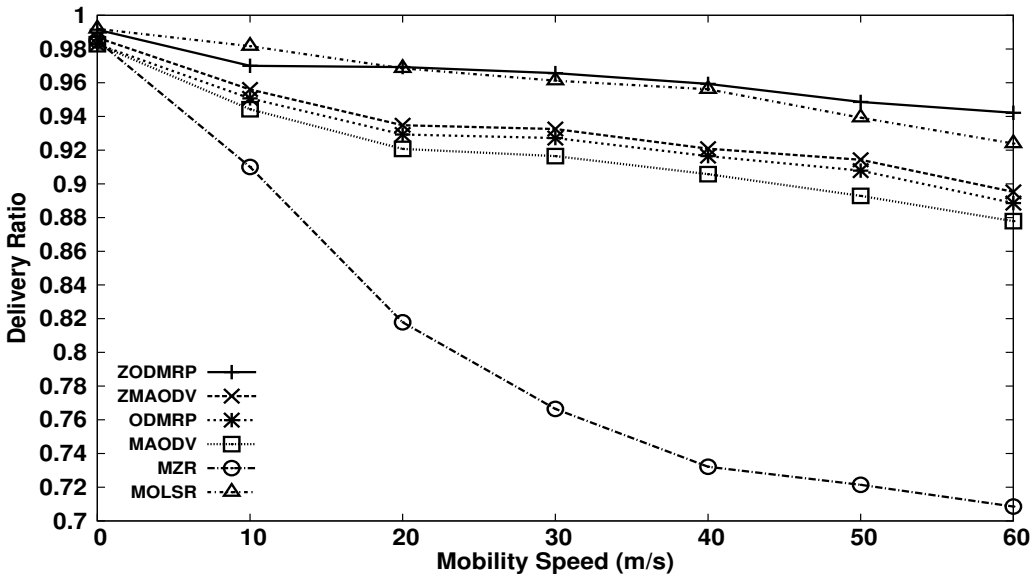


Figure 3.3: Delivery ratio vs. mobility speed.

of link failure.

Fig. 3.4 illustrates the number of control packets transferred per data packet delivered under different speed. We can see that the plot of ZODMRP is almost flat. This is because the zone routing mechanism together with the forwarding group technique in ZODMRP are effective on promoting robustness. That is, in the high speed scenario, the high volume link breakage occurs, ZODMRP can use the stored routing information to rapidly repair the broken link.

The MZR generates the highest number of control packets in all six protocols. Since the MZR uses a source-based tree and with the increase of the mobility speed, the broken link occurs frequently, MZR needs many control packets to repair the link, and thus increases the control overhead.

It can also be noticed that when the speed is low, the performance of ZODMRP is worse than MAODV. This is because the zone routing is a proactive activity

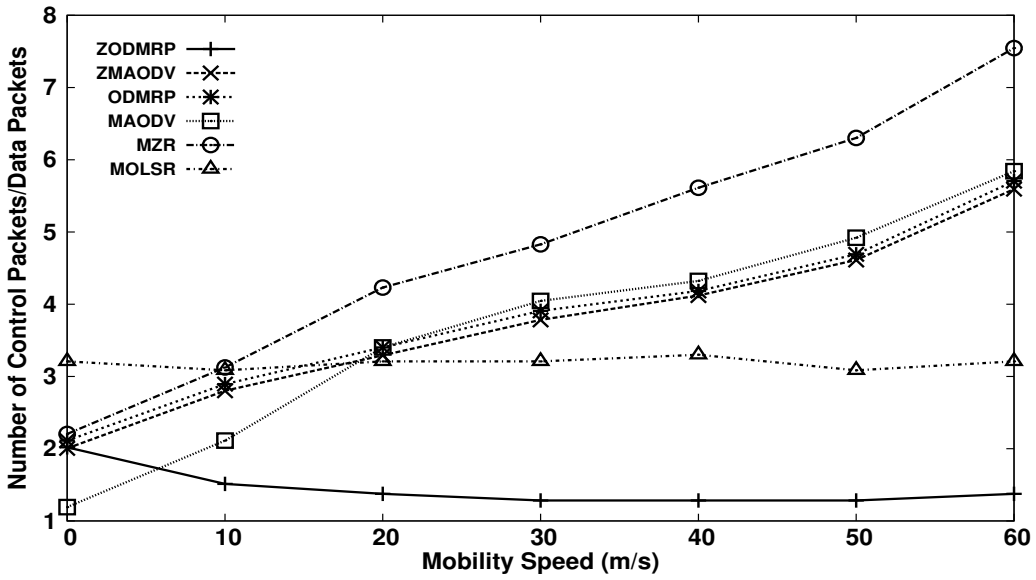


Figure 3.4: Number of control packets per data packets delivered vs. mobility speed.

that always generates nearly a fixed number of control packets serving the Zone Update process. When the speed is low, the total amount of control packets is relatively small, and the Zone Update Packets constitutes a large portion of it. However, when the speed increases, the amount of other control packets also increases, making the Zone Update Packets relatively a small portion of the overall control packets. In addition, the zone routing also saves a number of request control packets. When the speed is high, the protocols with zone routing have achieved a slightly better performance than the pure protocols. MOLSR has a relatively constant performance in this test. It is a typical performance for pure proactive protocols.

Fig. 3.5 illustrates the average delay of the protocols under different speeds. We can see that the ZODMRP almost has stable average delay, thus because the

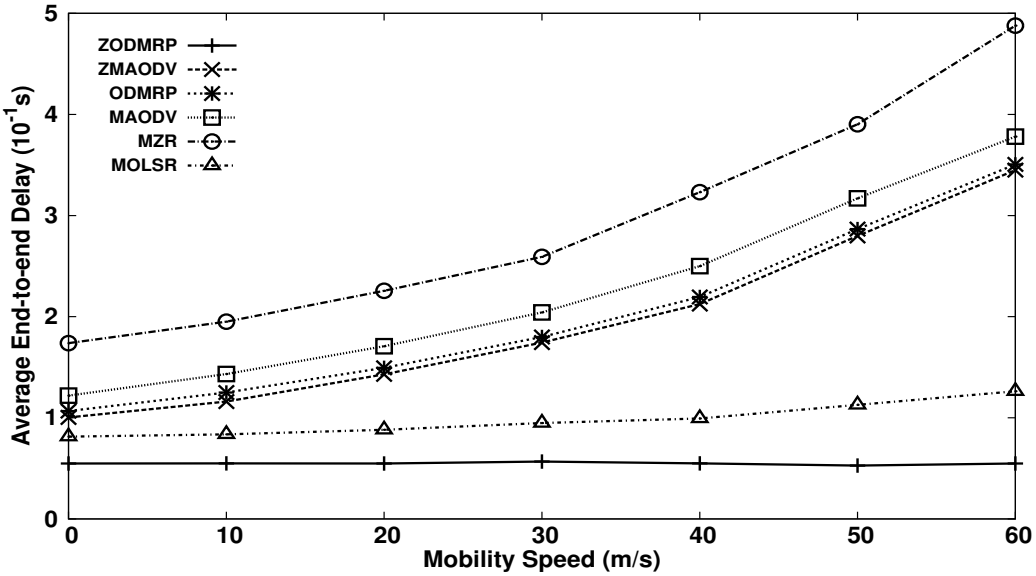


Figure 3.5: Average end-to-end delay vs. mobility speed.

forwarding group mechanism which the ZODMRP used as well as the zone routing mechanism provide more robustness on high speed scenario (which usually leads to high topology change rate). The mesh-based topology and the forwarding group feature remove the need for ZODMRP to handle the link breakage, the zone routing mechanism helps the ZODMRP to rapidly response to topology changes; thus the performance of average delay could be improved.

Also we can see that the protocols using zone routing could have lower average delay than those without zone routing. That is because the zone routing feature allows the nodes to have the multicast information of the neighbors, and such information helps the nodes find the route or repair the broken link leading to reduced the average delay of transmission.

Again, when the mobility speed is high, MOLSR provides a better performance due to its pure proactive behavior which provides robustness in this sce-

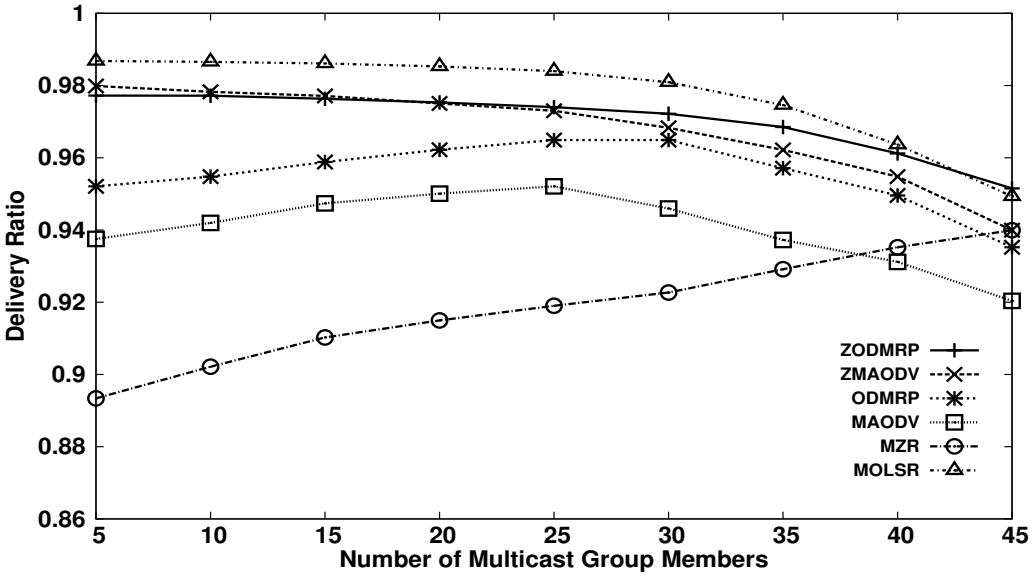


Figure 3.6: Delivery ratio vs. multicast group members.

nario.

2) Multicast Group Size

Experimental Scenario:

We varied the number of multicast members from 5 to 45 members, and we fixed the number of senders at 5, with a maximum speed of 5m/s.

Fig. 3.6 illustrates the packet delivery ratio as a function of multicast group members. We can see that the ZODMRP and ZMAODV outperform the others. With the group member increases, the ZODMRP achieves a little better performance than ZMAODV.

Unlike the other protocols, the MZR has an increasing delivery ratio as the increase of the multicast group size. This is because as the multicast group size increased, almost all the nodes within the network belong to the multicast group, and this increases the probability that a node finds a multicast group in its zone.

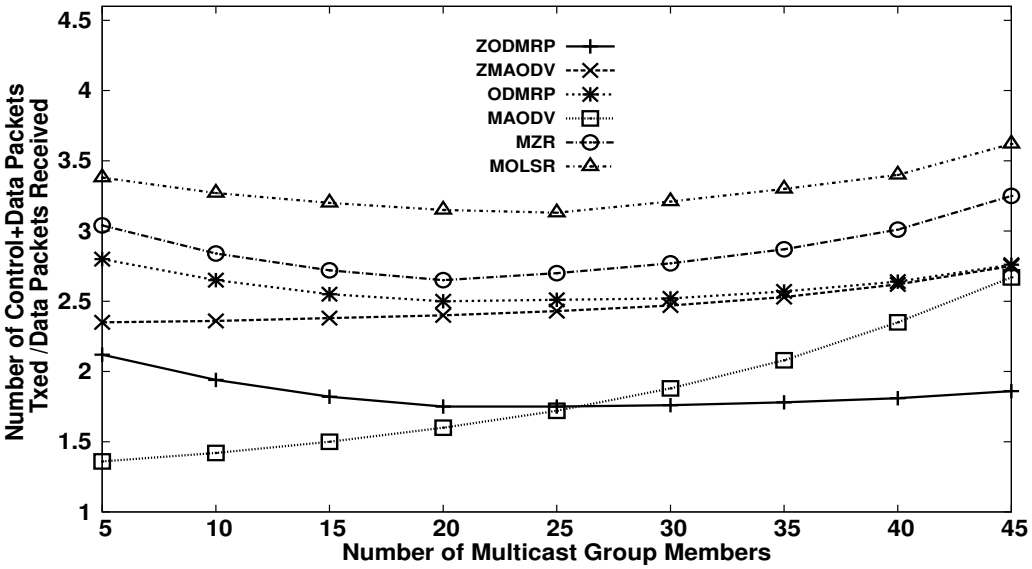


Figure 3.7: Number of control + data packets transferred per data packets received vs. multicast group members.

So when broken link occurs, the node could easily repair the link within the zone. In case of small multicast group size, when the node repairs the broken link in MZR, it may search the whole network to find a multicast group member.

We can notice that when the group size is low, MOLSR has the best performance among all these six protocols. But when the group size is high, the performance falls much more rapidly than that of ZODMRP. This is because in MOLSR, each multicast router (the node joining the source-based tree for data packets propagation) periodically broadcasts update packet to maintain the source-based tree. Thus this increases the amount of control overhead.

Fig. 3.7 illustrates the number of control plus data packets transferred per data packets received as a function of multicast group members. It can be seen that ZODMRP and ZMAODV have relatively flat plots. That is because the zone

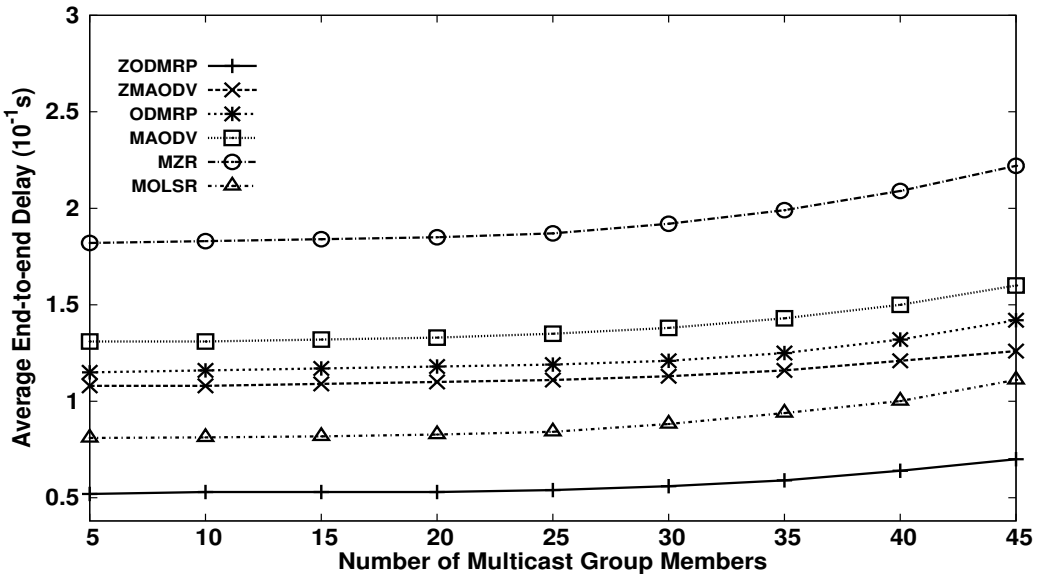


Figure 3.8: Average end-to-end delay vs. multicast group members.

routing issue of these two protocols can effectively help the nodes to create and maintain the multicast group with their stored zone routing information.

MAODV has the best performance when the group size is low. This is due to the pure on-demand feature of MAODV. As the group size is low, fewer nodes participating the maintenance of the multicast group, thus the number of generators of control packets is low. When the group size is high, there are more generators of control packets, the control overhead increases greatly for MAODV.

This figure illustrates our discussion of MOLSR on the performance of delivery ratio in Fig. 3.6. When the group size is high, the control overhead increases more rapidly for MOLSR than ZODMRP.

Fig. 3.8 illustrates the average delay as a function of multicast group members. ZODMRP has a better performance than the other protocols. Besides, the protocols with zone routing have a better performance than those without zone routing.

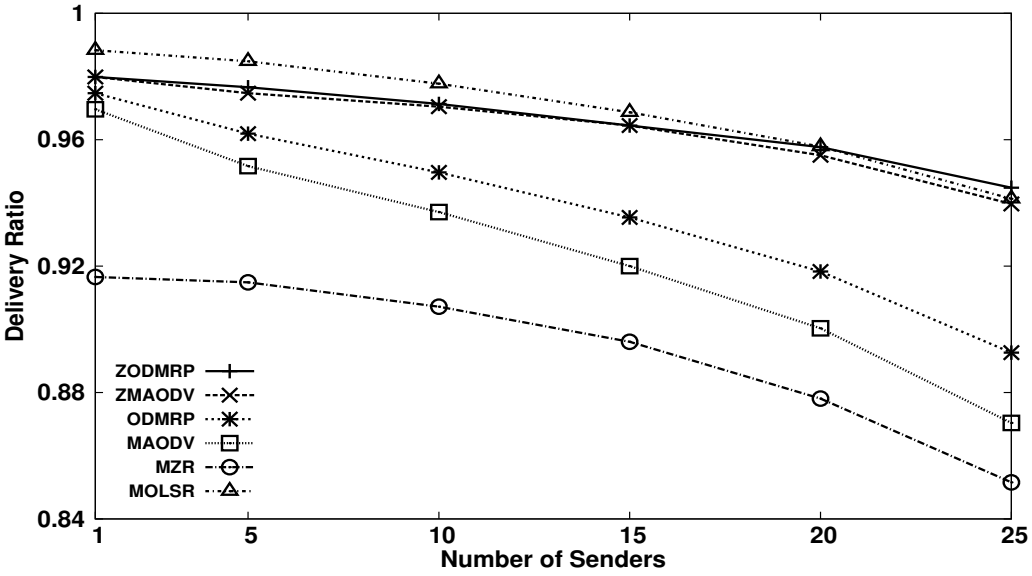


Figure 3.9: Delivery ratio vs. number of senders.

Since the zone routing feature could help the node find the path to the destination and to repair a broken link, it leads to reduced transmission delay.

Again, the end-to-end delay of MOLSR increases more rapidly than ZODMRP due to the large amount of control overhead.

3) Number of Senders

Experimental Scenario:

We fixed the multicast group size at 20, and the node maximum speed at 5m/s. The number of multicast senders ranges in a set 1, 5, 10, 15, 20, 25.

Fig. 3.9 illustrates the packet delivery ratio as a function of the number of senders. We can see the protocols in the new zone routing group (ZODMRP and ZMAODV) outperform their peer protocols as the zone routing information can help the senders locate the destinations more efficiently. MOLSR has the best performance when the number of senders is low.

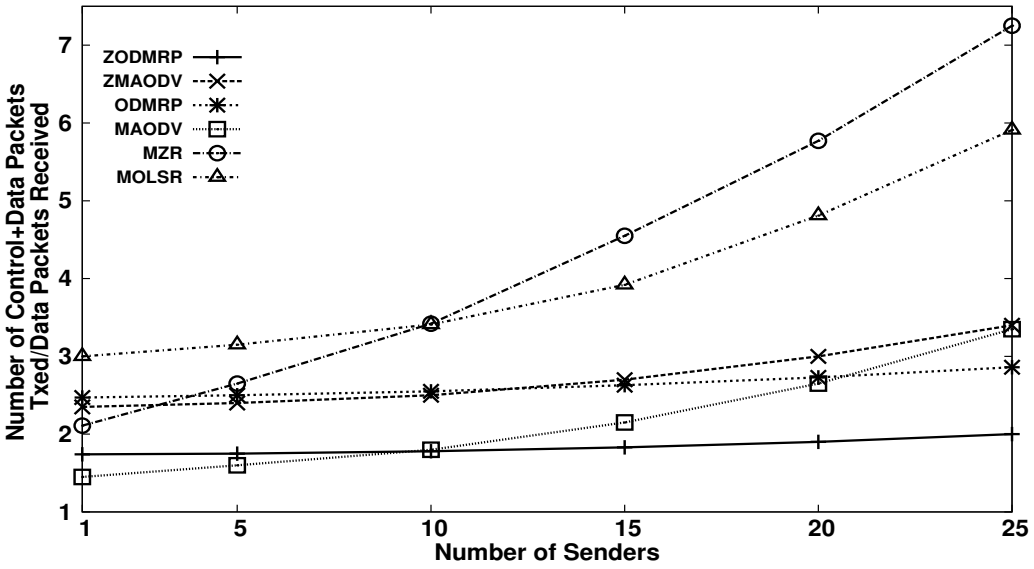


Figure 3.10: Number of control + data packets transferred per data packets received vs. number of senders.

Fig. 3.10 illustrates the number of control plus data packets transferred per data packets received. The ODMRP group protocols (ZODMRP and ODMRP) have a more steady performance since they maintain the forwarding group for data transmission and such mesh topology helps overcome the potential link breakage. When the number of senders is low, MAODV outperforms ZMAODV since there is a fixed number of control packets generated for zone updating in ZMAODV. However, the zone routing mechanism helps improve the performance when the number of senders increases. That is because the increasing of control overhead for joining group or link breakage caused by increasing the number of senders can be decreased using the zone routing information.

Because the MZR uses source-based tree topology, each source node builds up its own tree and with the number of senders increases, the control overhead to build up and maintain the tree increases too. Thus this gives MZR the worst

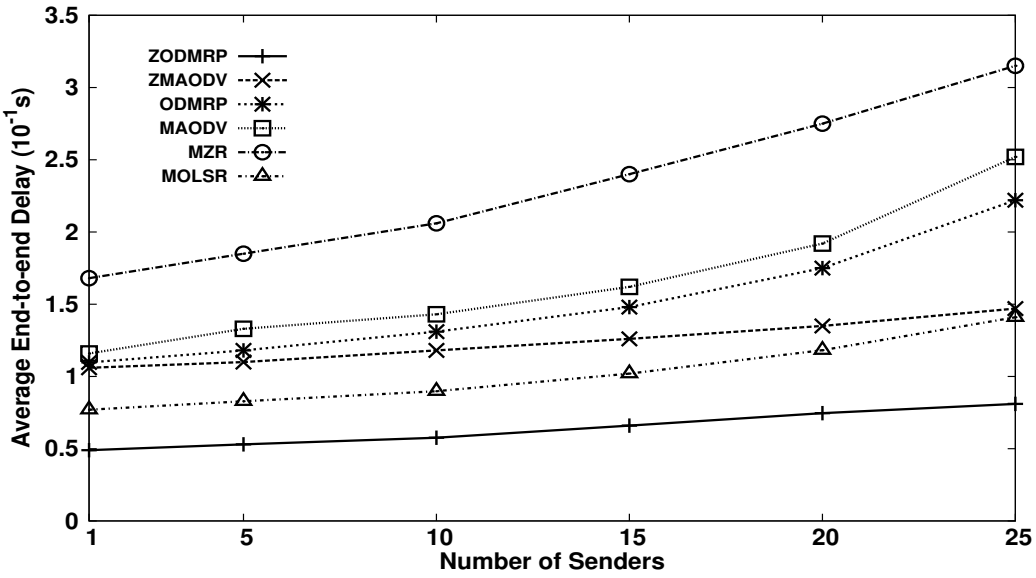


Figure 3.11: Average end-to-end delay vs. number of senders.

performance among all six protocols.

When the number of senders is high, the source-based tree used in MOLSR causes high control overhead since more senders result in more trees in this scenario. Each sender (as the tree root) periodically updates the routing information, and each node may join multiple trees which may turn most of them into multicast routers. Multicast routers also periodically update their routing information. Thus leads to a high control overhead.

Fig. 3.11 illustrates the average delay as a function of the number of senders. The average delay of ZODMRP is quite low and nearly stable. Again, due to the benefits of zone routing, protocols with this feature have a better performance than their peers.

3.3.3 Summary

In this chapter, we have conducted a performance study of two multicast protocols, MAODV and ODMRP that are augmented with an additional proactive behavior based on the Zone Routing Protocol. The newly developed protocols are hybrid in nature and the performance evaluation presented here has shown that these protocols can benefit from the added proactive behavior. In addition, we concluded that the zone routing based proactive behavior affects reactive protocols in different ways depending on the topology of the multicast group.

In terms of delay, the zone routing behavior has been found to have beneficial effects on both the ODMRP and the MAODV. Its effect on the delivery ratio was found to be better with the MAODV rather than with the ODMRP. In general, the performances of both protocols have been found to improve sensibly by incorporating the zone routing behavior. This work is part of a project looking at finding the best combination of proactive and reactive behaviors that will lead to dynamic and multi-behavioral routing protocols.

Chapter 4

Proposed Polymorphic Multicast Routing Protocol in Ad Hoc Networks

In this chapter, we first propose a new power-aware polymorphic multicast routing protocol for Mobile Ad hoc Networks (MANETs), P_ZODMRP [10]. We design a polymorphic algorithm for P_ZODMRP as the behavior selection criteria. The polymorphic algorithm considers the node's power level, mobility level and vicinity density level as the factors. For the routing protocol selection for various behavioral modes, we choose On-Demand Multicast Routing Protocol (ODMRP) [9] as the reactive behavior and the zone routing concept from Multicast Zone Routing (MZR) protocol [7] as the proactive behavior. After demonstrating the polymorphic algorithm, we introduce the detailed routing operation of P_ZODMRP. We evaluate the performance of P_ZODMRP by comparing it with the other protocols through simulation.

Secondly, we propose another power-aware polymorphic multicast routing protocol for MANETs, the Optimized Polymorphic Hybrid Multicast Routing (OPHMR) protocol [11], [16]. OPHMR is an enhanced version of P_ZODMRP. It uses the same polymorphic algorithm as P_ZODMRP with many similar operations. In OPHMR, we introduce the Multipoint Relay (MPR) technique from Multicast Optimized Link State Routing (MOLSR) protocol [12] so as to optimize the control packets propagation in the proactive behavior. We also perform simulation to evaluate the performance of OPHMR by comparing it to P_ZODMRP and other peer protocols.

4.1 Polymorphic Algorithm

4.1.1 Polymorphic Algorithm Description

In this proposed conception of the protocol design, a node selects its own behavior based on its battery power level, mobility speed level and vicinity density level. We define two power level thresholds P_TH1 and P_TH2, where $P\_TH1 > P\_TH2$. We also define mobility level threshold M_TH and vicinity density level threshold V_TH. These four thresholds are the criteria for the polymorphic algorithm to determine a node's behavior. Furthermore, we define four behavioral modes for each node. The four behavioral modes are *Proactive Mode 1 (PM1)*, *Proactive Mode 2(PM2)*, *Reactive Mode (RM)* and *Proactive Ready Mode (PRM)*.

- *Proactive Mode 1 (PM1)*:

  A node in this mode periodically sends out update packets to maintain the topology information. A pre-configured zone radius R value is set as the

Time-To-Live (TTL) value to the update packets. The update interval is set to i, where i is a fixed time interval. When the node receives update information from other nodes, it updates its Neighborhood Routing Table (NRT) using the information received. The structure of NRT is described later in Section 4.2.1. The node then minuses the TTL value of the received update packet by one. If the new TTL value is not zero, the node propagates the update packet. Otherwise, the node discards the update packet. The lifetime of the corresponding NRT entry of a node in *PM1* is $2\times i$.

- *Proactive Mode 2 (PM2)*:

  The node in this mode performs a similar operation as that in *PM1*. The difference between *PM1* and *PM2* is that the update interval of *PM2* is set to $2\times i$, where i is the same value as in *PM1*. The lifetime of the corresponding NRT entry of a node in *PM2* is $3\times i$.

- *Reactive Mode (RM)*:

  The node in this mode does not send out update packets and when receiving update packets from other nodes, the node just discards them. The lifetime of the corresponding NRT entry of a node in *RM* (if any) is $4\times i$.

- *Proactive Ready Mode (PRM)*:

  The node in this mode does not periodically send out update packets but updates its NRT when receiving the update packets from other nodes. The node in this mode propagates the received update packets in the same manner as in *PM1*. The lifetime of the corresponding NRT entry of a node in *PRM* (if any) is $4\times i$.

The polymorphic algorithm driving the targeted polymorphic routing protocols includes two parts: the main algorithm and the mobility speed routine. We assume that the node is capable of detecting its battery power level, its mobility level and its vicinity density level. The measurement of a node's battery power level is remaining power over the total battery power. The measurement of a node's mobility level is its speed. We calculate the number of one hop neighbors of a node as its vicinity density level.

Algorithm 1 Polymorphic Algorithm

if $Power > P\_TH1$ **then**
  if the node is not in *PM1*, it switches to *PM1*.
  then it notifies neighbors about the mode switch.
else
  if $Power < P\_TH2$ **then**
    if the node is not in *RM*, it switches to *RM*.
    then it notifies neighbors about the mode switch.
  else
    Perform the mobility speed routine.
  end if
end if

Algorithm 2 Mobility Speed Routine

if $Mobility > M\_TH$ **then**
  if $Vicinity < V\_TH$ **then**
    if the node is not in *PM2*, it switches to *PM2*.
    then it notifies neighbors about the mode switch.
  else
    if the node is not in *PRM* switches to *PRM*.
    then it notifies neighbors about the mode switch.
  end if
else
  if the node is not in *RM* switches to *RM*.
  then it notifies neighbors about the mode switch.
end if

When a node's power level is high ($> P\_TH1$), the node is set to *PM1* mode, so that it can be able to maintain topology information and react faster to topology changes. Thus, nodes are allowed to operate in *PM1* mode only if the power level is high enough. On the other hand, when the node's power level is low ($< P\_TH2$), the node is forced into the *RM* mode in order to extend its battery life.

Data transmission is one of the operations consuming high power in nodes. When a node is in proactive behavioral mode, the additional periodical update packets propagation consumes the battery power of a node, and this shortens the lifetime of the node. Thus, when a node's power level is low, *RM* that consumes less power helps extend the battery operation time. However, when a node's power level is high, the duly updated topology information achieved by the proactive behavioral mode can offer better performances, thus a node should perform *PM1* in high power level environment.

It can be argued here that the *PRM* and *RM* modes can be combined into the *PRM* mode only, in order to make good use of the valuable routing information of the packets discarded in the proposed *RM* mode. That may be a valid point. However, it may be argued against it that if a node is in real shortage of power, then every single unit of power will be needed to send its own data and to extend its battery life and hence its service time. Therefore, we separate *PRM* and *RM* in case that when a node's power level is extremely low, it can choose *RM* to further extend its battery life.

When a node's power level is within P_TH1 and P_TH2, the mobility speed routine is performed to help determine the node's behavior. The mobility speed routine is described next.

When the mobility speed level of a node gets high, this implies that the topology around the node is expected to change quickly. Thus, the node is required to behave proactively in order to maintain better connectivity and awareness of the topology changes. This is triggered when the node's mobility speed level gets higher than the M_TH threshold.

With regards to high mobility levels of the node, it is to be noted that, in general, almost all routing protocols will fail when the node moves too fast. Our proposed protocol tries to minimize the probability of failure by switching to a proactive mode when high mobility is detected.

The next consideration here relates to the node's vicinity density level. When it is high, it means that there are many nodes within the power range of the node. Thus, if we let the node engage in a proactive mode, the update packets would consume the channel capacity and jam the network with higher probability. Thus, when the vicinity density level is high, the node is forced into *PRM* (semi-proactive) behavior. The *PRM* mode is more conservative with regards to proactiveness.

4.1.2 On Receiving Notification

When a node switches its behavior, it will broadcast a notification to all its neighbors to inform them about the change. When other nodes receive such information from the source node, they may react to the change accordingly. The reactions are given as follows.

- PM1→PM2: When a node receives a notification of the behavior switching from *PM1* to *PM2*, the node extends the lifetime of the entry of the source

node in the NRT. The new lifetime is set to $3\times i$.

- PM1→RM/PRM: When a node receives a notification of the behavior switching from *PM1* to *RM* or *PRM*, the node extends the lifetime of the entry of the source node in the NRT. The new lifetime is set to $4\times i$.

- PM2→PM1: When a notification of behavior switching from *PM2* to *PM1* is received, the lifetime of entry of the source node is reduced from $3\times i$ to $2\times i$.

- PM2→RM/PRM: When a notification of behavior switching from *PM2* to *PM1* is received, the lifetime of entry of the source node is changed from $3\times i$ to $4\times i$.

- RM/PRM→PM1: When a notification of behavior switching from *RM* or *PRM* to *PM1* is received, the node adds an entry of the source node in the neighborhood routing table. The lifetime of the entry is set to $2\times i$.

- RM/PRM→PM2: When a notification of behavior switching from *RM* or *PRM* to *PM2* is received, the node adds an entry of the source node in the neighborhood routing table. The lifetime of the entry is set to $3\times i$.

- RM⇔PRM: When a notification of behavior switching from *RM* to *PRM* or from *PRM* to *RM* is received, the node takes no action.

4.1.3 Special Handling

There is such a condition that a node is in *PM1* or *PM2* but all the neighbors of that node are in *RM*, so that the node could not send out any update packets and

could not hear any update packets from others. To address this case, we have an addition rule that if a node in *PM1* or *PM2* and have not heard any update packets within a fixed time interval $3 \times i$, the node switches its status to *RM*.

4.2 P_ZODMRP

We have implemented a routing protocol using the polymorphic algorithm described above. We choose the zone routing concept from MZR as the proactive behavior, and using the ODMRP as the reactive behavior. Since the reactive behavior is adopted from ODMRP, we maintain the multicast group topology as a mesh.

4.2.1 Routing Tables

Each node maintains two routing tables, one is the Neighbor Table (NTable), the other is Multicast Routing Table (MRTable).

The NTable acts as the NRT we described in the algorithm, and actually only nodes in proactive like modes maintain it. The main structure is similar to that of the Zone Routing Table in ZODMRP. Each entry specifies a neighbor in the zone. Each entry contains the routing information to that neighbor, including hop count and next hop address. In addition, each entry contains the multicast routing information of the node, such as the multicast group the node belongs to. Each entry is assigned a lifetime, and the entry that exceed the life time would be removed from the NTable.

The nodes in *PM1*, *PM2* and *PRM* maintain the NTable using the update packets they received from others. The nodes in those modes also periodically flush

the NTable to remove the stale entries. The nodes in *RM* periodically flush the NTable to remove the stale entries, but do not update or insert any entries in their NTables.

Each node should also maintain a Multicast Routing Table (MTable) to record the multicast routing information of itself and store the multicast routing topology. The structure of the MTable is the same as the routing table in ODMRP.

4.2.2 Packet Structure

The packet structure in P_ZODMRP is the same as the packet structure in ZODMRP, and we only introduce a new type of packets in the protocol, i.e. the notification packet.

The notification packet is constructed with 32 bits. The first 8-bit is for packet type, and it is set to 5 for Notification Packet in P_ZODMRP. The next 16-bit is for the type switching. Each bit is for one particular type, and there are 12 types of possible switching. The first 4 bits are always set to 0.

Table.4.1 describes the type switchings and their corresponding codes. The last 8-bit is reserved.

4.2.3 Path Finding Procedure

When a node has packets to a multicast group or wants to join the multicast group, it begins the path finding procedure.

If the node is in *RM*, it sends out a Join Query as the same way in ODMRP and then waits for the Reply.

If the node is in *PM1*, *PM2* or *PRM*, the node first looks inside its NTable to

0000000000000001	PM1→PM2
0000000000000010	PM1→PRM
0000000000000100	PM1→RM
0000000000001000	PM2→PM1
0000000000010000	PM2→PRM
0000000000100000	PM2→RM
0000000001000000	PRM→PM1
0000000010000000	PRM→PM2
0000000100000000	PRM→RM
0000001000000000	RM→PM1
0000010000000000	RM→PM2
0000100000000000	RM→PRM

Table 4.1: The meaning of the bit in switch type section.

see whether there are nodes that belong to the destination multicast group. If so, the node unicasts Join Queries to all these nodes and waiting for reply. Otherwise, the node broadcasts a Join Query.

When an intermediate node receives a Join Query, and it is a member of the multicast group, it generates the Reply and sends it back to the source node initiated the Join Query, then updates the MTable to record the route. For the node which cannot reply for the Join Query, it first checks its own behavior. If it is in *RM*, it just propagates the Join Query and records the Join Query in the route cache. If the node is in *PM1*, *PM2* or *PRM*, the node looks inside its own NTable to find the destination multicast group member. If there are members inside its zone, it unicasts the Join Query to all the members. Otherwise, the node just propagates the Join Query. When the Join Query reaches a node that can reply, a Reply will be returned to the source node.

When the source node receives the Reply, it updates its MTable to record the route and begins data transmission.

We emphasize here that our protocol allows different modes of operations for the participating nodes in a forwarding route. In a route, each node tries to determine the destination node according to its own strategy (proactive or reactive). Thus, the nodes try to find next forwarding nodes by using their own routing tables which are established in the background for proactive nodes, or using broadcasting for reactive nodes. This feature ensures the avoidance of any hysterical behavior, as in the worst case of large disparity in the modes of operations of the nodes, the *RM* constitutes the last resort for synchronization of behaviors.

4.2.4 Polymorphic Algorithm

Periodically, each node within the network executes the polymorphic algorithm to determine its operational behavior. Each node first examines its power level. If the power level is exceeds P_TH1 (or falls below P_TH2), it switches to *PM1* (or *RM*). Otherwise, if the power level of the node stays between P_TH1 and P_TH2, it then uses its mobility speed and vicinity density levels to further determine its behavior. Details of this behavioral adjustment is given in Section 4.1.1.

If a node switches its behavior, the node generates a Notification Packet and broadcasts the packet to all its neighbors. The TTL of the notification packet is set to one, and only the one hop neighbors receive the notification packet. The neighbors detected the notification packets follow the procedure described in Section 4.1.2 to process the packets.

4.2.5 Energy Consumption Model

The energy consumption model that we are using is implemented from L.M.Feeney's work [81]. In this model, the network performance has four possible energy consumption states: *transmit* and *receive* are for transmitting and receiving data. The *idle* mode is used when the interface can transmit or receive, and the *sleep* mode is reserved for the case when the interface can neither transmit nor receive and where nodes have an extremely high power consumption.

The cost for a node to send or receive a network-layer packet is modeled as a linear function. There is a fixed cost associated with channel acquisition (b) and an incremental cost (m) proportional to the size of the packet. The basic equation for the computation of power consumption cost is

$$Cost = m \times size + b. \tag{4.1}$$

The total cost of a packet is the sum of the costs incurred by the sending node (s), and all receivers. Potential receivers include the destination (d), any node n within wireless range of s ($n \in S$ (the set of all neighbor nodes of s)), and any node r within wireless range of d ($r \in D$ (the set of destination nodes)).

In our implementation, we do not concern about the power consumption in *sleep* mode. Such consumption is extremely low and in the polymorphic algorithm there is no *sleep* mode, so we use *off* mode as the substitution. Nodes in *off* mode could neither transmit or receive and have no power consumption.

In addition, in [81], Equation 4.1 was extended to cover the cases of broadcasting traffic, point-to-point traffic and discarding traffic (for discarding overheard traffic by nodes not concerned with the transmission). For instance, different val-

ues of m and b were defined, for sending, receiving or discarding packets. These equations were adopted, *as-is*, in our model for energy consumption computation. The reader is referred to [81] for more details about the energy model adopted here.

4.3 Performance Evaluation of P_ZODMRP

4.3.1 Simulation Scenarios

We have performed two sets of simulations. In the first set, we have varied and evaluated the value i to see the effect of update interval within different conditions. In the second set, we perform the simulation among P_ZODMRP, ZODMRP, ZMAODV and MOLSR, the value i is set to 5 for P_ZODMRP.

The simulation of these protocols was implemented within the GloMoSim library[79]. Table 4.2 illustrates the parameters in the simulation.

There are two set of simulations. The simulation time for the first set is 500s and the initial power of all the nodes are full. In some scenarios of the second simulation set, the simulation time is set to 1000s and the power level of nodes are set to vary from one to another. This setting is done to validate the effect of the proposed protocol on the delivery ratio when in the course of the simulation some nodes will die off due to lack of battery power. Thus, we can check the effect of the polymorphic protocol on battery longevity. Thus we set 20% of the nodes to have 100% power, 20% of the nodes have 90% power, 20% of the nodes have 80% power, and 40% of the nodes have 75% power. The following metrics were used in the performance evaluation:

Total number of nodes	50
Simulation area	1000m×1000m
Propagation range	225m
Channel capacity	2Mbps
MAC protocol	The IEEE 802.11 MAC [1]
Traffic type	constant bit rate (CBR)
Mobility model	random waypoint model [80]
Pause time	0s
Power model	L. F. Feeney's work [81]
Zone update interval	5s
Zone radius	3
Zone lifetime	180s
Packet sending rate	10packets/s
Packet size	512 bytes
m_{send}	0.000405
m_{recv}	0.000157
b_{send}	0.067594
b_{recv}	0.037701
Total power	10000
P_TH1	85%
P_TH2	50%
V_TH	6
M_TH	20m/s

Table 4.2: The parameters for the simulation.

- Packet delivery ratio;
- Number of control packets transferred per data packet delivered;
- Average percentage of power conservation;
- Average end-to-end delay.

We evaluate the above metrics against mobility speed, network traffic load and the total number of nodes.

4.3.2 Results and analysis

1) Mobility Speed

Experimental Scenario:

There are 50 nodes within the area, each node moves constantly with a predefined speed. The node maximum movement speed was varied from 0m/s to 60m/s. In this scenario, we had 20 multicast members and 5 source nodes.

Fig. 4.1 depicts the delivery ratio against mobility speed for different values of the parameter i. It can be seen that the curves for $i = 5$ and $i = 8$ have nearly the same performance. The protocol performs badly when $i = 2$ or $i = 10$. When $i = 2$, the update interval is short and thus many update packets are generated occupying much of the channel capacity and reducing the delivery ratio.

Fig. 4.2 shows the average end-to-end delay as a function of mobility speed. Again, when $i = 5$ and $i = 8$, the performance is better. With the increase of the mobility speed, the average delay is higher when $i = 2$, that is because when the battery power falls below P_TH1, more nodes switch to *PM2* from *PRM* when

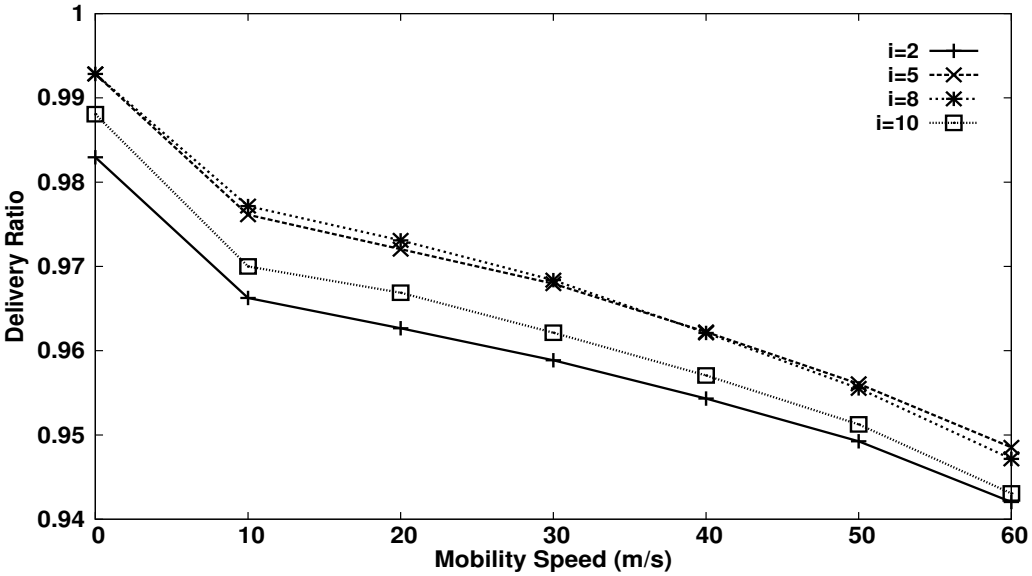


Figure 4.1: Delivery ratio vs. mobility speed.

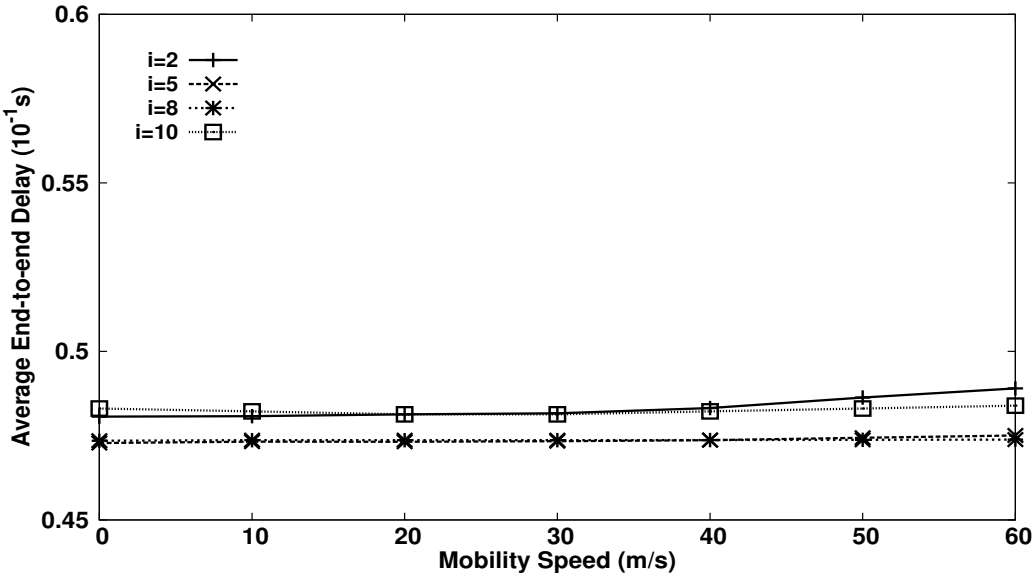


Figure 4.2: Average end-to-end delay vs. mobility speed.

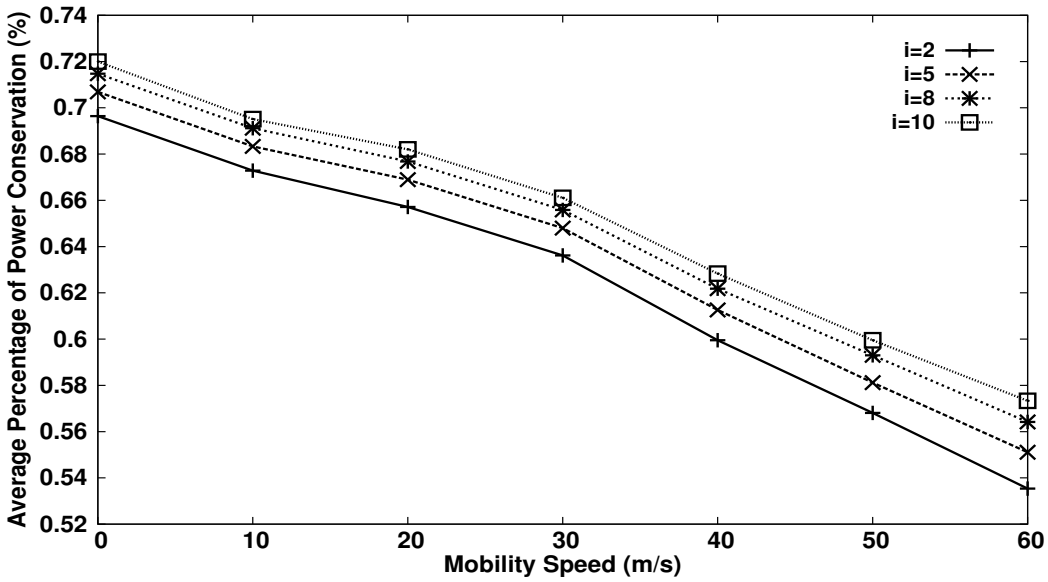


Figure 4.3: Average percentage of power conservation vs. mobility speed.

the speed is high, which makes more control overhead generated and causes more control overhead and channel usage than other situations.

Fig. 4.3 shows the average power conservation as a function of mobility speed. When $i = 10$, the update interval is the longest which leads to the least control packets generated, and thus could save up the power usage.When $i = 2$, the worst performance is recorded.

Fig. 4.4 shows the delivery ratio as a function of mobility speed. We could see that the P_ZODMRP has the best performance, and the performance of ZODMRP and the MOLSR are very near to the P_ZODMRP. ZMAODV has the poorest performance. All nodes have equal initial power here.

Fig. 4.5 shows the delivery ratio as a function of mobility speed, when variable power levels and longer simulation is considered. Since some nodes would have used up their battery power and gone off, the delivery ratio decreases greatly. We

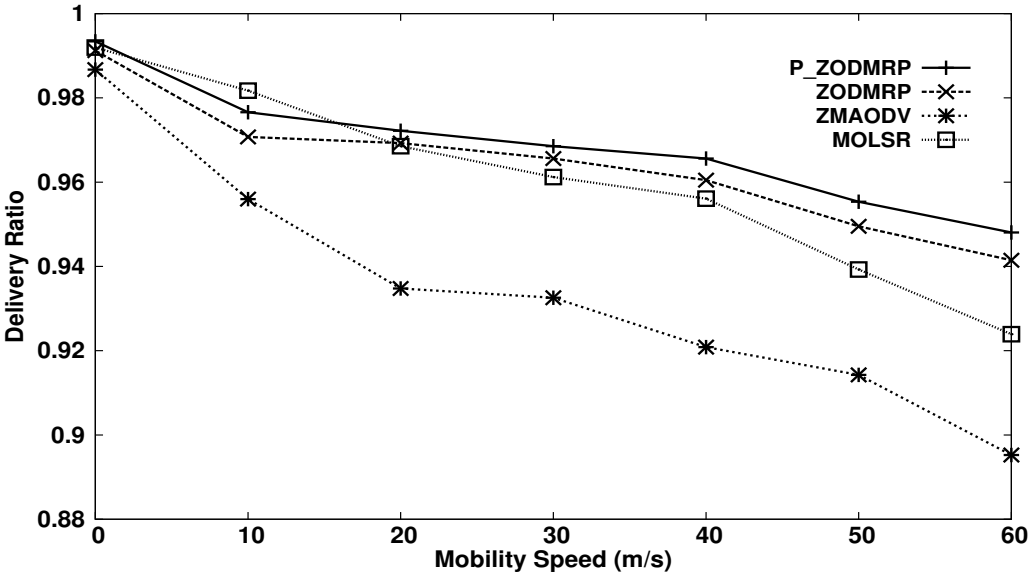


Figure 4.4: Delivery ratio vs. mobility speed.

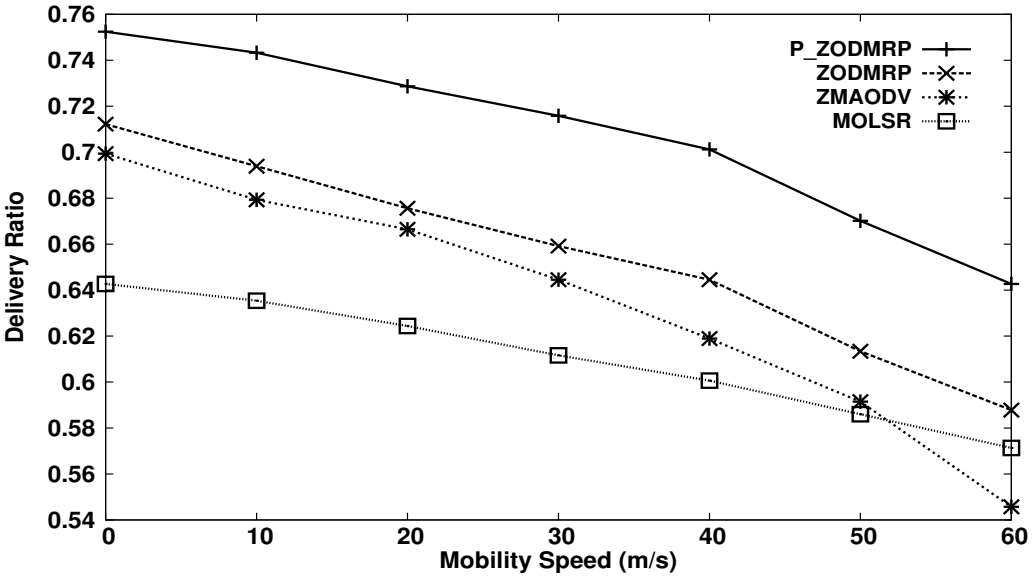


Figure 4.5: Delivery ratio vs. mobility speed.

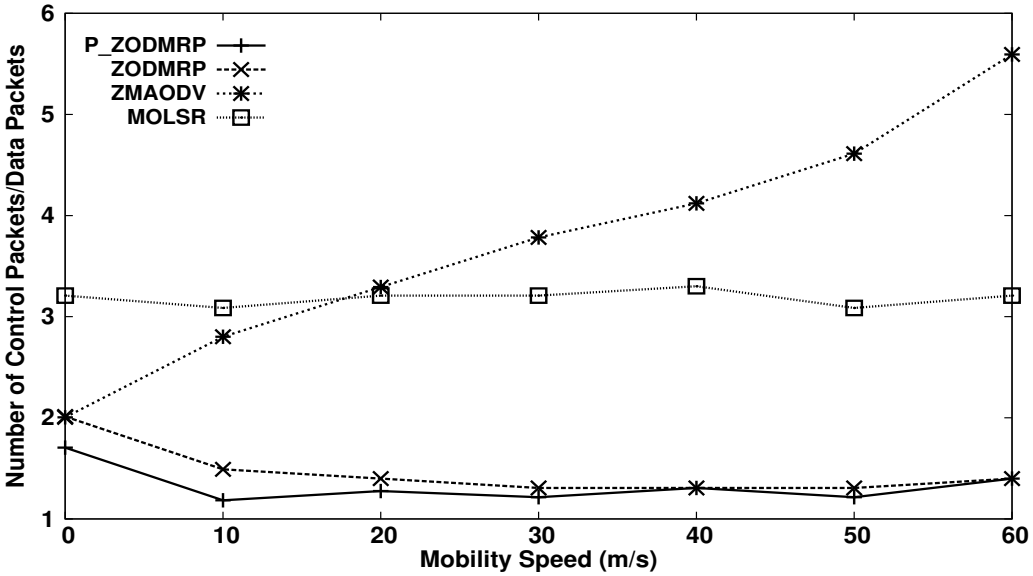


Figure 4.6: Number of control packets per data packets vs. mobility speed.

can see that in this scenario, the P_ZODMRP exhibits the best performance. The battery life of nodes using P_ZODMRP has been extended so that the delivery ratio could increase greatly. In other experiments, we have found that in terms of end-to-end delay, the performance of the P_ZODMRP is almost comparable to that of the ZODMRP. The main addition is mostly in battery life extension.

Fig. 4.6 shows the number of control packets per data packets as a function of mobility speed. We can see that P_ZODMRP outperforms the other protocols while ZMAODV performs the worst when the mobility speed is high. Since P_ZODMRP uses mesh topology for multicast group to provide redundant routes, P_ZODMRP does not have to deal with the high volume link breakage involved in high speed scenario, which leads to a relatively constant amount of control packets of P_ZODMRP. Meanwhile, with the polymorphic features, P_ZODMRP generates less periodically update packets than ZODMRP.

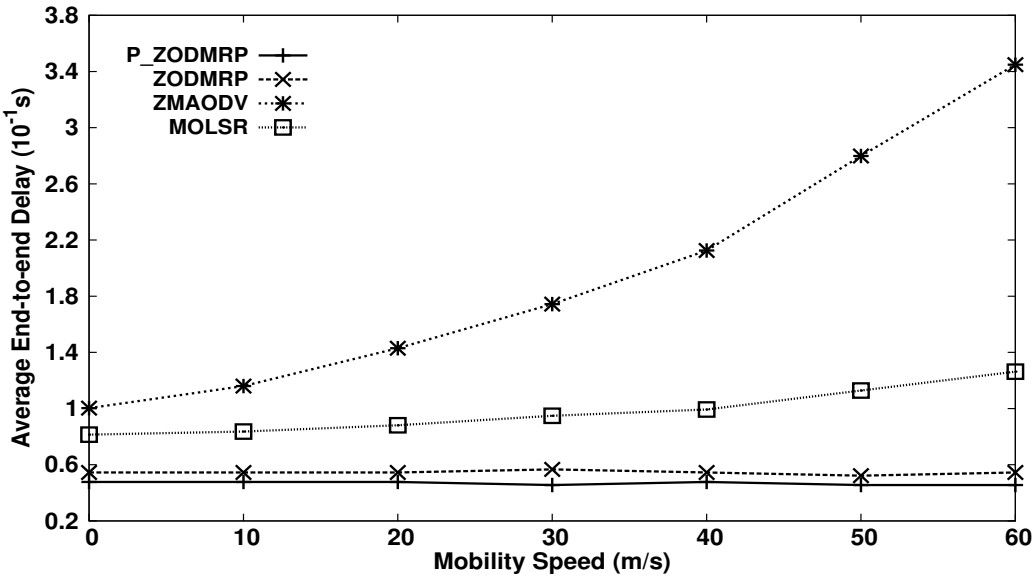


Figure 4.7: Average end-to-end delay vs. mobility speed.

Fig. 4.7 shows the average end-to-end delay as a function of mobility speed. We can see that with the speed increases, the P_ZODMRP has the best performance. The performance of ZODMRP is similar to the P_ZODMRP but since the polymorphic features could reduce the control packets on periodical update packets, thus could reduce the channel usage. The ZMAODV has the worst performance because with the increase of the mobility speed, the link breakage occurs more frequently which produces more control packets for the tree rebuilding.

Fig. 4.8 shows the average power conservation as a function of mobility speed. We can see that P_ZODMRP could extend node's battery life better than the other routing protocols. With the increase of the speed, the power usage of ZMAODV increase greatly because ZMAODV generates more control packets in high speed.

2) Nodes Vicinity Density

Experimental Scenario:

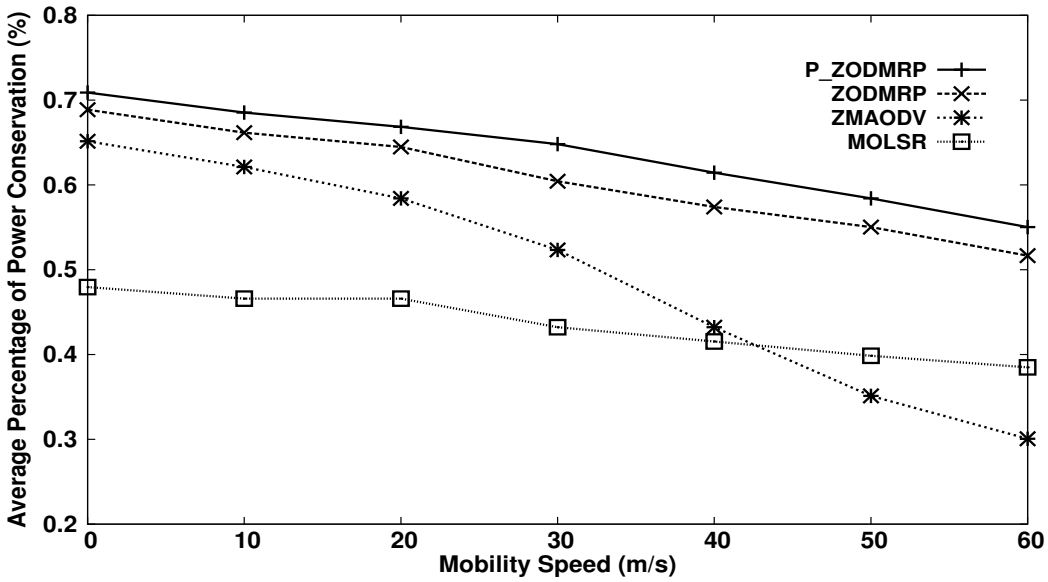


Figure 4.8: Average percentage of power conservation vs. mobility speed.

The total number of nodes within the area varies from 20 to 80, each node moves with a predefined maximum speed 5m/s. In this scenario, we had 15 multicast members and 5 source nodes.

Fig. 4.9 shows the effect of parameter i on delivery ratio with a range of vicinity density. Usually, a node in a high density situation is likely to have more neighbors, which causes more control overhead generated during the propagation of control information leading to channel resource exhaustion. When $i = 2$, the high update frequency makes the situation even worse. Thus, when $i = 2$, the throughput performance is the worst, while on the contrary, when $i = 10$, the performance is the best among all settings. However, in the situation when the vicinity density is low, high update frequency does not have as significant impact to the performance as the number of neighbors is high. Hence, we see that all settings give similar performance. In fact, the setting that $i = 2$ has slightly better

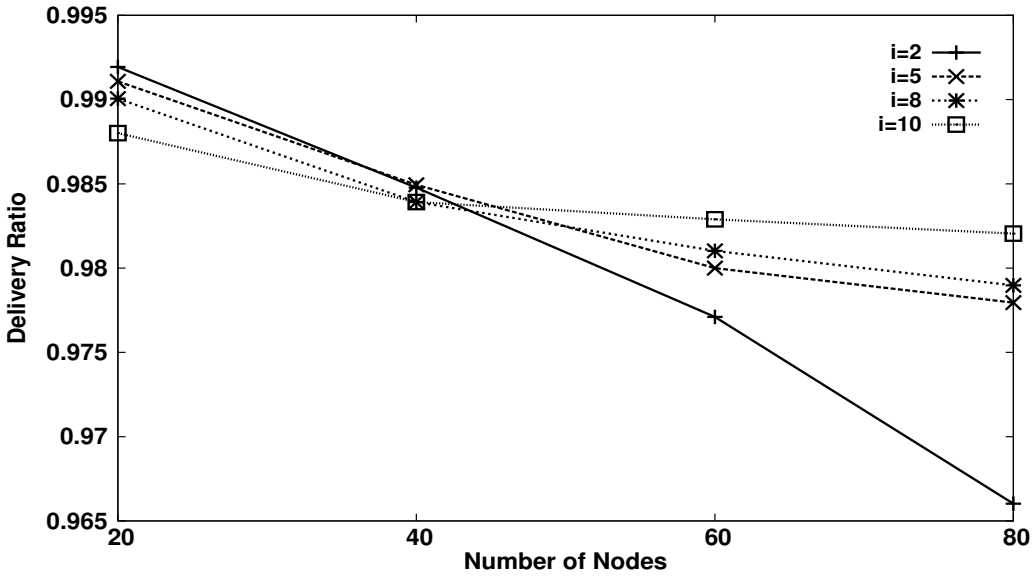


Figure 4.9: Delivery ratio vs. nodes vicinity density.

performance among all in low vicinity density due to its more frequent update of routing information leading to more up-to-date routing information for accurate packet forwarding.

Fig. 4.10 shows the average end-to-end delay as a function of nodes vicinity density for several values of i. The setting of $i = 5$ and $i = 8$ were found to perform better. Also, we can see that in low vicinity density scenario, a higher update frequency ($i = 2$ compares to $i = 10$, and $i = 5$ compares to $i = 8$) has better performance since a higher frequency provides more robustness. On the other hand, in high vicinity density level, a higher update frequency leads to a higher volume of update packets generated, thus causes the performance to be worse than a lower update frequency.

Fig. 4.11 shows the average power conservation as a function of nodes vicinity density. Again, when $i = 10$, the update interval is the lowest, thus, the control

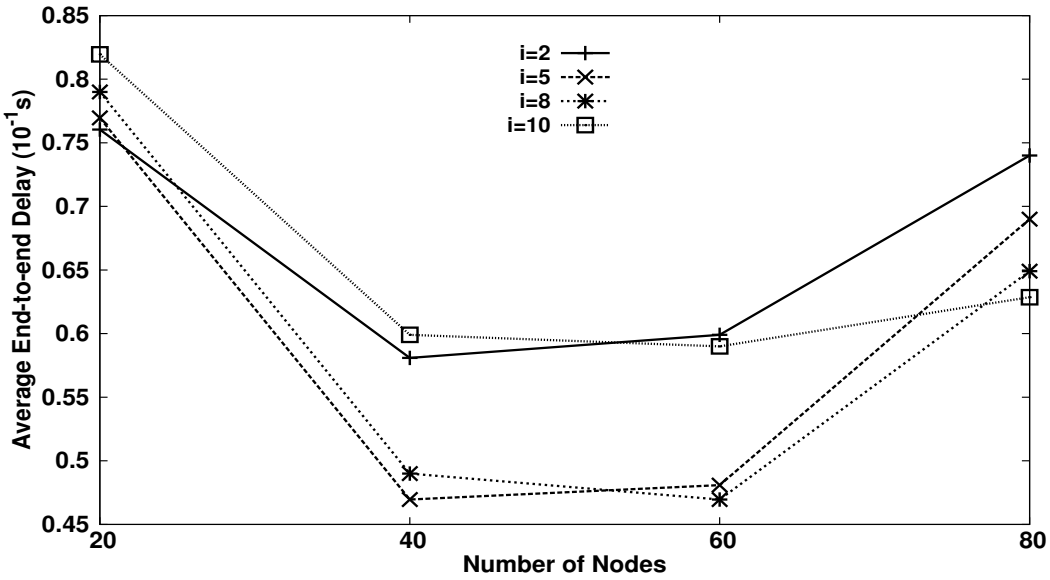


Figure 4.10: Average end-to-end delay vs. nodes vicinity density.

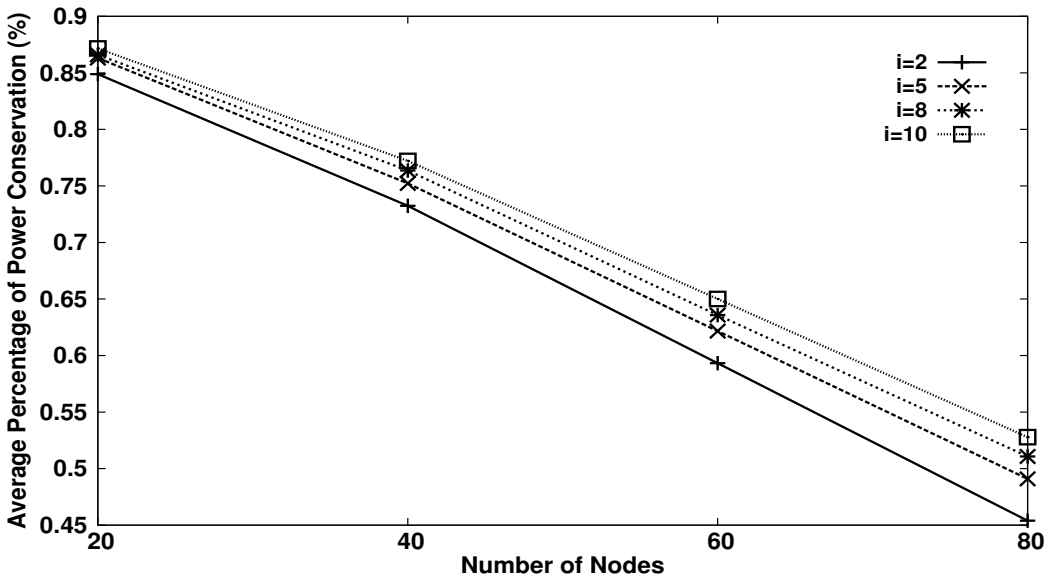


Figure 4.11: Average percentage of power conservation vs. nodes vicinity density.

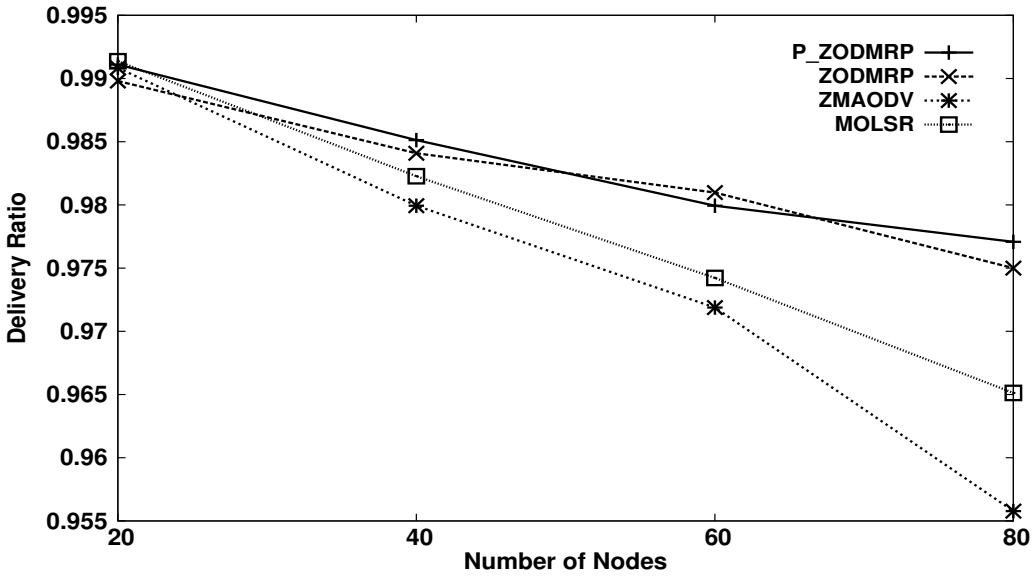


Figure 4.12: Delivery ratio vs. nodes vicinity density.

overhead is the least among four settings which brings the best performance.

Fig. 4.12 shows the delivery ratio as a function of nodes vicinity density when the simulation time is 500s. We can see that the P_ZODMRP and ZODMRP have the best performance because they use the mesh to maintain multicast group topology. The mesh topology could guarantee the data packets to reach the destinations and does not need to handle the link breakage.

Fig. 4.13 shows the delivery ratio as a function of nodes vicinity density when the simulation time is 1000s. We can see that the performance of all four protocols slightly increases with the increasing of the vicinity density. That is because in this scenario, some nodes could exhaust the battery and be turned off. When the vicinity density is low, the number of neighbors of a node is already low, as well as some of the neighbors may be turned off during the simulation, the node may be isolated from the others. Thus makes the delivery ratio even lower. Again we

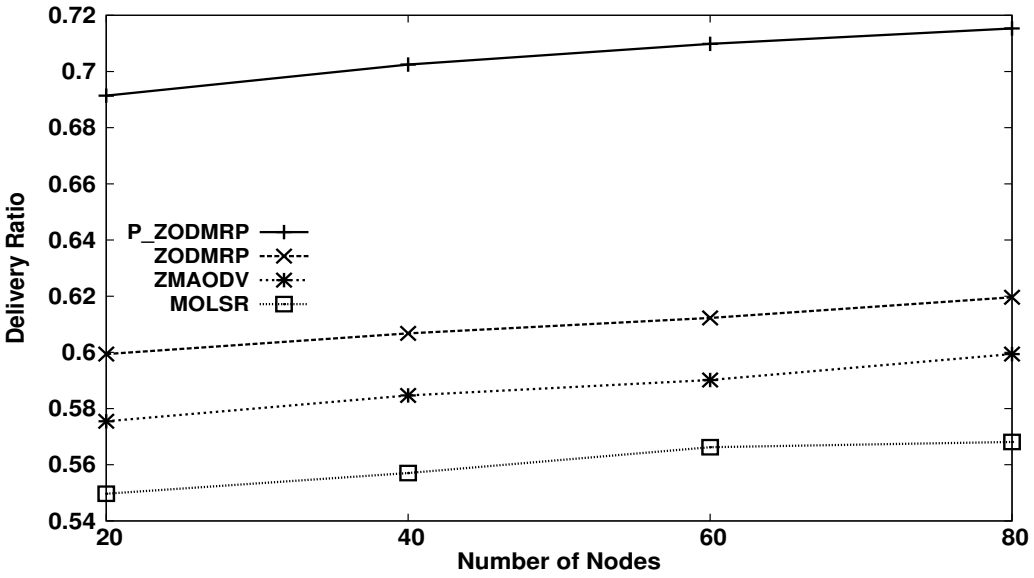


Figure 4.13: Delivery ratio vs. nodes vicinity density.

can see that the P_ZODMRP has the best performance because of its power saving issues.

Fig. 4.14 shows the number of control packets per data packets as a function of nodes vicinity density. Since the MOLSR is a pure proactive protocol and its update packets should reach all nodes through the network, the number of control packets increases greatly. In addition, with the increase of the number of nodes, the MAODV was found to generate higher control packets.

Fig. 4.15 shows the average end-to-end delay as a function of nodes vicinity density. When the density is low, some nodes may get isolated and cause an increase in delay. When the density is high, nodes could have more neighbors thus build better path finding information, but would generate more control overheads which occupy the channel and delay data packets. That is why the average delay of all the four protocols is high when the density is either low or high. Again,

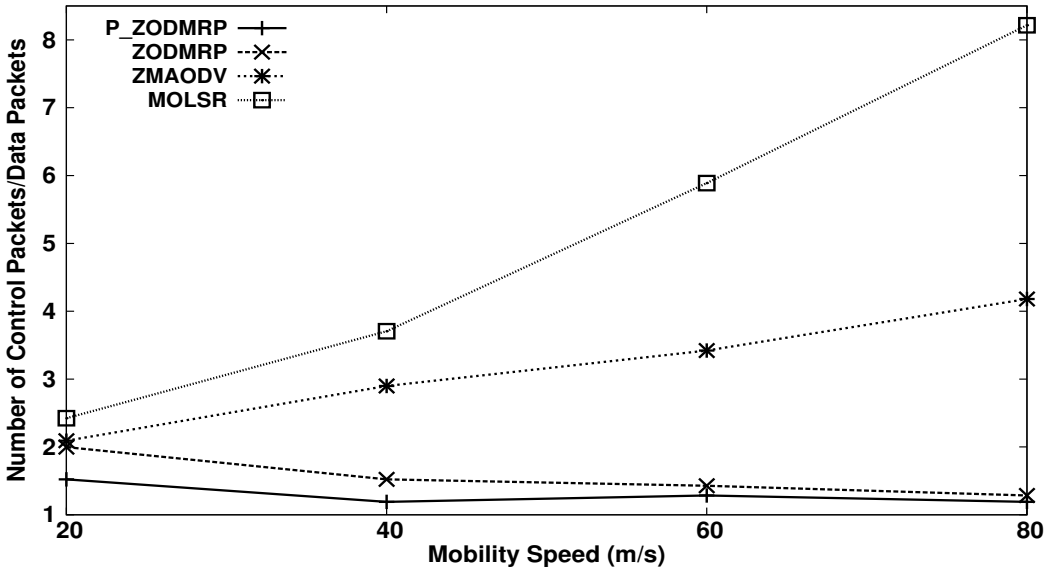


Figure 4.14: Number of control packets per data packets vs. nodes vicinity density.

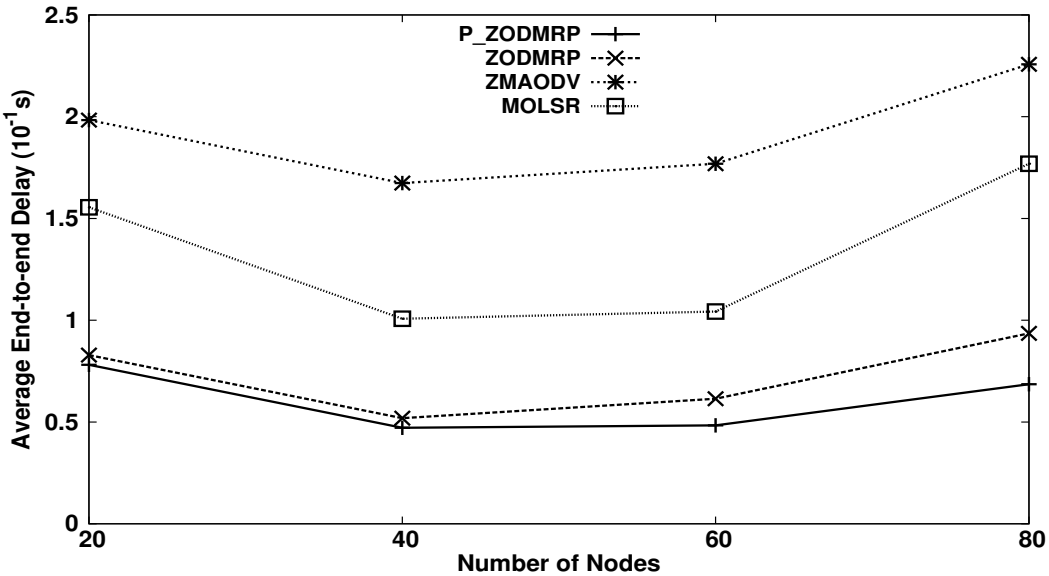


Figure 4.15: Average end-to-end delay vs. nodes vicinity density.

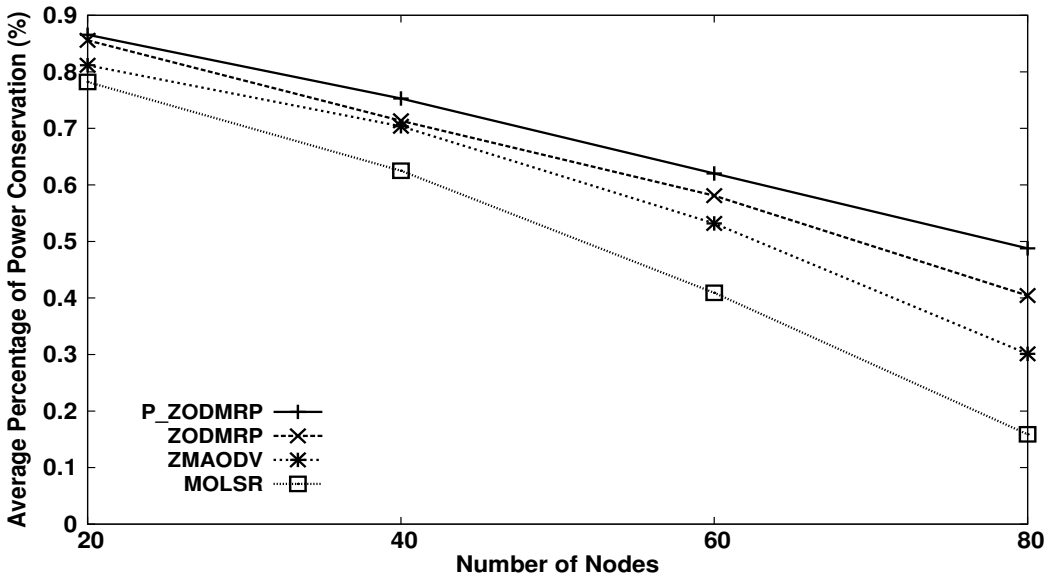


Figure 4.16: Average percentage of power conservation vs. nodes vicinity density.

the polymorphic protocol achieves the best performance by reducing the control packets than the ZODMRP.

Fig. 4.16 shows the average power conservation as a function of nodes vicinity density. Since the power model used is mainly concerned with the packets been transferred, the MOLSR used the most power within these protocols because the large amount of its periodically update packets. The P_ZODMRP saves up in terms of the number of control packets it transferred, and achieves the best performance.

3) Network Traffic Load

Experimental Scenario:

There are 50 nodes within the area and the number of packets the sources send varies from 1 to 50 packets per second, each node moves with a predefined maximum speed 5m/s. In this scenario, we had 20 multicast members and 5 source

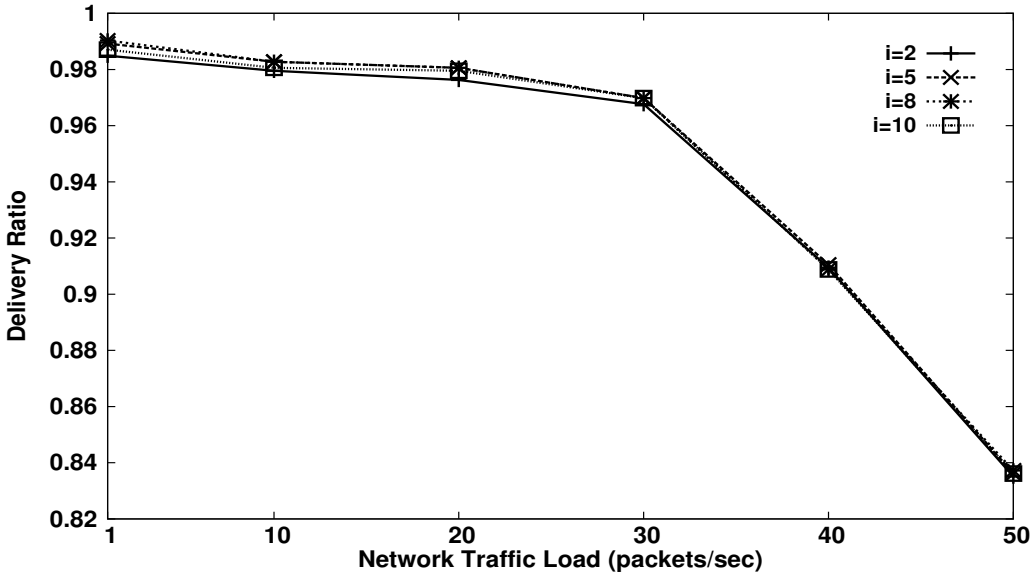


Figure 4.17: Delivery ratio vs. network traffic load.

nodes.

Fig. 4.17 shows the delivery ratio as a function of network traffic load. With the increase of the traffic load, as the data packets used up most of the channel capacity, different update intervals have very limited effect on the delivery ratio. Thus, especially when the load is high, the delivery ratios are nearly the same for all four settings.

Fig. 4.18 shows the average end-to-end delay as a function of network traffic load. Again since the channel is mainly occupied by the data packet transmission, different settings only affect a little in all the conditions.

Fig. 4.19 shows the average power conservation as a function of network traffic load. When $i = 10$, the long update interval generates the least control packets, hence this setting produces the best performance. On the contrary, $i = 2$ has the worst performance.

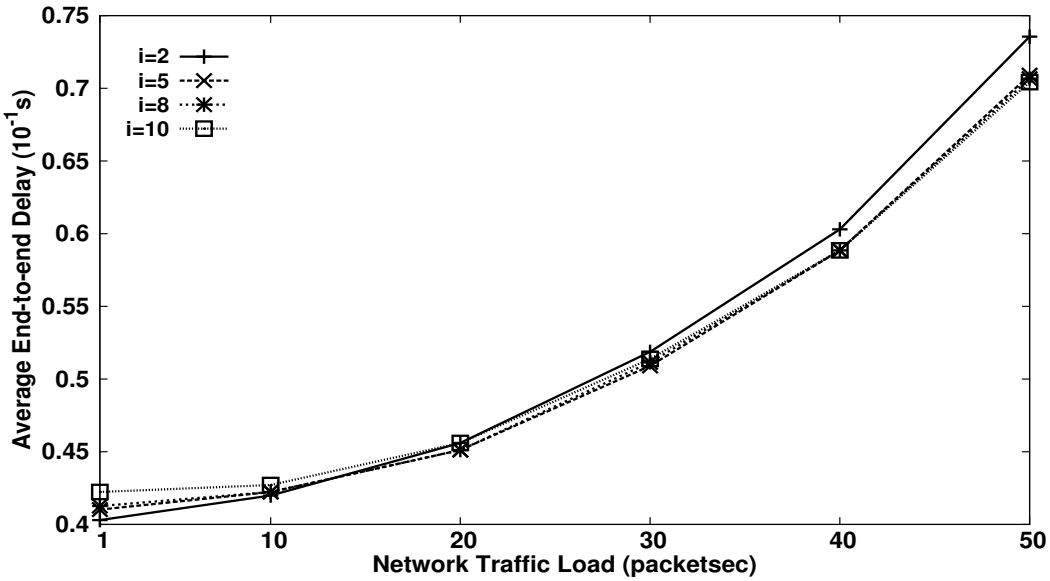


Figure 4.18: Average end-to-end delay vs. network traffic load.

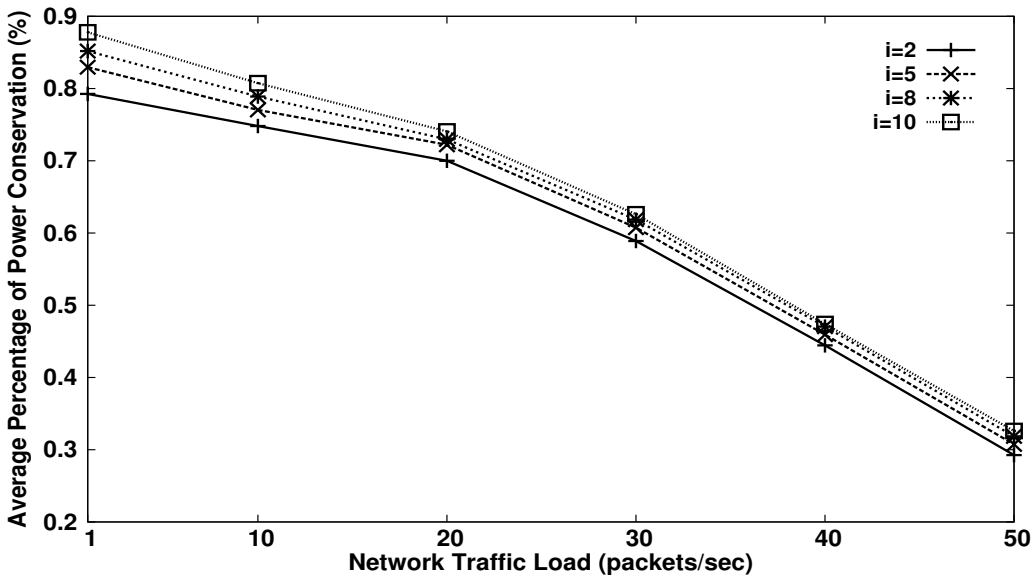


Figure 4.19: Average percentage of power conservation vs. network traffic load.

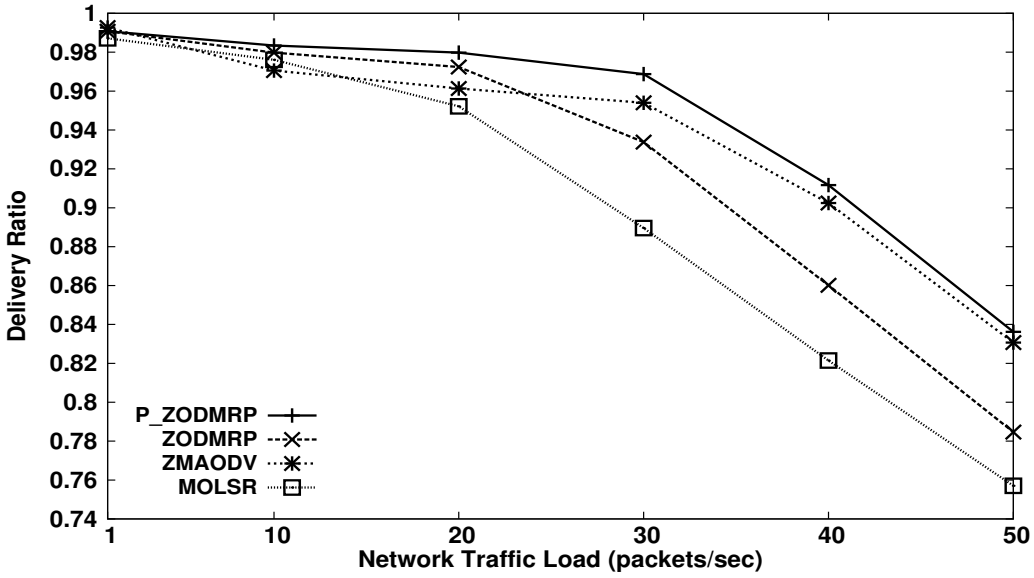


Figure 4.20: Delivery ratio vs. network traffic load.

Fig. 4.20 shows the delivery ratio as a function of network traffic load when the simulation time is 500s. We can see that the performance of all the four protocols decreases greatly when the traffic load is high. As the traffic load increases, the number of data packets increases, and the channel capacity is used up by data packet transmissions thus causes the traffic jam. Again the P_ZODMRP has the best performance because of the control packets saving mechanism of the polymorphic features and the mesh topology.

Fig. 4.21 shows the delivery ratio as a function of network traffic load when the simulation time is 1000s. With the increase of the traffic, clear decline of performance is observed. However, the performance of P_ZODMRP is by far the best among all four protocols. The gain is clearer than in the case of equal initial battery power setting in Fig. 4.20.

Fig. 4.22 shows the average end-to-end delay as a function of network traf-

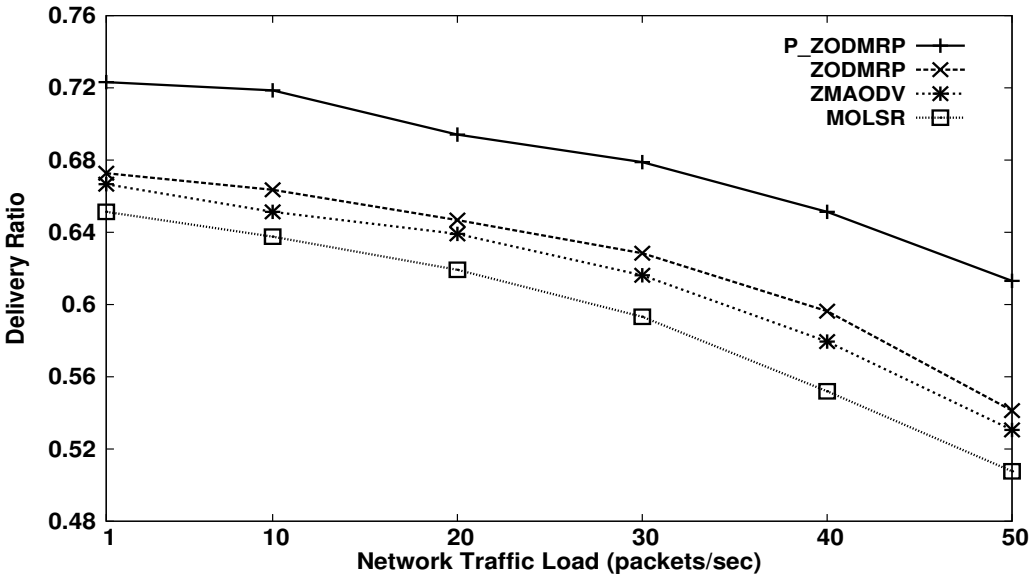


Figure 4.21: Delivery ratio vs. network traffic load.

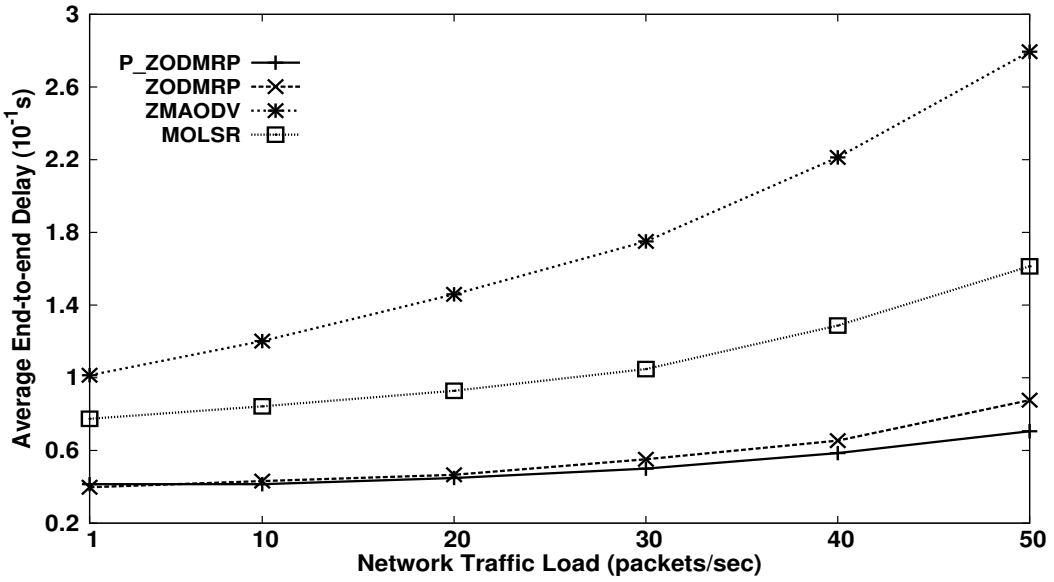


Figure 4.22: Average end-to-end delay vs. network traffic load.

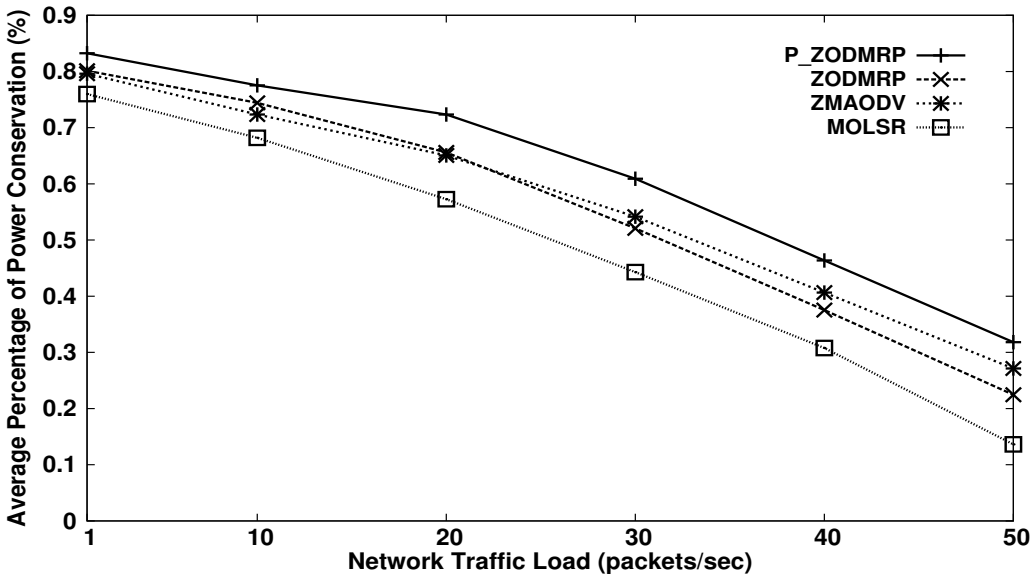


Figure 4.23: Average percentage of power conservation vs. network traffic load.

fic load. We can see that with the load increases, the average delay increases in all four protocols. The ZMAODV has the poorest performance due to the large amount of control overhead for repairing link breakage. When the load is high, P_ZODMRP performs better than ZODMRP because its polymorphic feature helps reducing control overhead on periodical update packets.

Fig. 4.23 shows the average power conservation as a function of network traffic load. As the transmission of data packets constitutes the main part of power consumption, in this scenario, the power of nodes in all of the four protocols decreases greatly. Again, with its power saving mechanisms, the P_ZODMRP performs better than the other three.

4.4 Optimized Polymorphic Hybrid Multicast Routing (OPHMR) Protocol

OPHMR is an enhanced version of P_ZODMRP. They use the same polymorphic algorithm to determine the node's behavior. OPHMR and P_ZODMRP share many similarity in terms of operation, except for the control packet propagation technique. Precisely, we introduce the Multipoint Relay (MPR) mechanism from OLSR [32] into OPHMR for control packet propagation in proactive modes. The MPR mechanism allows a node to select a subset of its neighbors as the relay set. Only the nodes in the relay set take the responsible for propagating the control packets. We further modify the operation of proactive behaviors in P_ZODMRP to make the proactive behavior in OPHMR suitable for the new propagation mechanism. The following subsections of this section illustrate the MPR mechanism and our modified proactive behavioral operations.

4.4.1 The Multipoint Relay Mechanism

The multipoint relay (MPR) based mechanism of the OLSR [32] is used to perform an optimized forwarding Mechanism in OPHMR. Each node maintains a two hop Neighborhood Table (*2NTable*). The 2NTable is used to calculate the MPR information. When a node receives an update packet, it uses the neighborhood information in the packet to calculate the two hop neighborhood and updates the corresponding entries in the 2NTable.

MPR nodes are selected to forward broadcast messages during the flooding process. This technique substantially reduces the message overhead as compared

to the classical flooding mechanism, where every node retransmits each message on receiving it for the first time.

Each node has its MPR set and broadcasts its MPR information in the periodic update packets. When propagating the periodic update packets, only the MPRs forward update packets.

We use the heuristic algorithm proposed for the OLSR to compute the MPR with slight adaptation.

MPR Computation [32]: The following definitions are given first.

N: represents the subset of neighbors of thecurrent node.

N2: represents the set of two hop neighbors of thecurrent node.

D(y): The degree of a one hop neighbor y (where y is a member of N). It is defined as the number of symmetric neighbors of a node y, excluding all the members of N and excluding the node performing the computation. The flowchart given in Fig. 4.24, summarizes the essence of the adapted version of the heuristic algorithm proposed in [32].

4.4.2 Proactive Operations in OPHMR

When a node is in *PM1* or *PM2*, it periodically sends out update packet. This time, the update packet not only includes the information as P_ZODMRP, it also includes all its one hop neighborhood information. These one hop neighborhood information is used to calculate the MPR set of this node.

When a node receives the update packets, if it is in *PM1*, *PM2* or *PRM*, the node uses the one hop neighborhood information in the update packet to update the 2NTable. If the node is in the MPR set of the source node, the node propagates

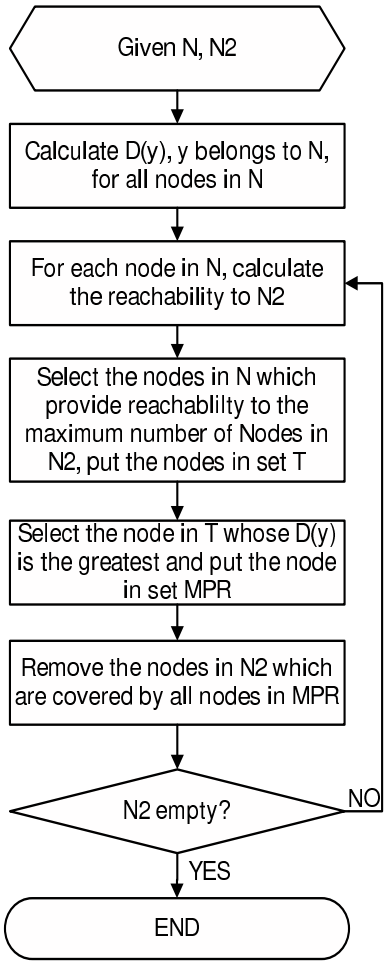


Figure 4.24: Flowchart for the MPR operation.

Simulation time	1000s
Simulation area	2000m×2000m
Propagation range	225m
Channel capacity	2Mbps
MAC protocol	The IEEE 802.11 MAC [1]
Traffic type	constant bit rate (CBR)
Mobility model	random waypoint model [80]
Pause time	0s
Power model	L. F. Feeney's work [81]
Zone update interval	5s
Zone radius	3
Zone lifetime	180s
Packet sending rate	10packets/s
Packet size	512 bytes
m_{send}	0.000405
m_{recv}	0.000157
b_{send}	0.067594
b_{recv}	0.037701
Total power	10000

Table 4.3: The parameters for the simulation.

the update packet, and the other nodes discard the update packet.

4.5 Performance Evaluation of OPHMR Protocol

4.5.1 Simulation Scenarios

We have performed a simulation based comparison of the OPHMR against the P_ZODMRP, the ODMRP and the MOLSR.

The simulation of these protocols was implemented using the GloMoSim library [79]. Some common parameters are listed in Table 4.3.

The two parameters, R and i, were pre-configured for OPHMR, where R denotes the zone radius (in number of hops) and i is the tuning factor used for deter-

mining the update interval and table entries' lifetime (as described in Section 4.1).

The power distribution among nodes was set to be variable from a node to another to emulate a realistic environment. This setting is done to validate the effect of the proposed protocol when in the course of the simulation some nodes die off due to lack of battery power. Thus, we can check the effect of the polymorphic behavior of OPHMR protocol on extending battery longevity. The power level of each node was set as the percentage of power residue out of the total capacity. We have set 20% of the nodes to have 100% power, 20% of the nodes to have 90% power, 20% of the nodes have 80% power, and 40% of the nodes have 75% power.

Three metrics were used in the performance evaluation, the packet delivery ratio, the end-to-end delay, and the average percentage of power conservation. The latter, is defined as the average level of power among all active nodes sampled over time.

The above metrics were evaluated against mobility speed, network traffic load and the total number of nodes. In the first part of the simulation, the threshold values were set as: P_TH1 = 85%, P_TH2 = 50%, V_TH = 6 and M_TH = 20m/s.

These settings aim to extend battery operation time. Since nodes with battery power above P_TH1 consume high power due to constant communications activities, the threshold should be high. We believe 85% is an appropriate threshold as if it too high, the nodes quickly escape from this mode making the mode useless, or if it is too low, battery operation time reduces. As for P_TH2 setting, we set it such that the communications activities are minimized when power level goes below half as battery may deplete fast at this power level.

We choose V_TH1 to be 6 due to the fact that maintaining at least six neigh-

bors leads to high reliability of a network in terms of connectivity for medium sized networks (see [82], [83], [84], [85]).

As for the M_TH setting to 20m/s (or 72km/h), we chose that because for vehicular MANETs in a metro area, vehicles are usually restricted to travel at around 60km/h.

Furthermore, in the last experiment, we have evaluated the effect of the power threshold setting on the performance of the OPHMR protocol.

4.5.2 Sensitivity Analysis

1) Effect of Mobility Speed:

Experimental Scenario:

In this scenario, 150 nodes were spread within the defined area. node maximum mobility speed was varied from 0m/s to 60m/s. The traffic load was set to 20 packets per second, and 40 multicast members and 10 source nodes were considered.

Fig. 4.25-4.27 shows the protocols' performance as a function of the mobility speed. Fig. 4.25 shows the delivery ratio versus mobility speed. We can see that the OPHMR has the best performance among all four protocols especially at high speed. When the speed reaches 60m/s, the OPHMR could score 13% advantage over the ODMRP's performance. In addition, we can see that the setting of a higher zone radius and smaller update interval (i.e., the setting of $R = 3$ and $i = 5$) generates better deliverability.

Fig. 4.27 shows the average end-to-end delay versus mobility speed. The OPHMR scored the best performance especially when $R = 3$ and $i = 5$. It

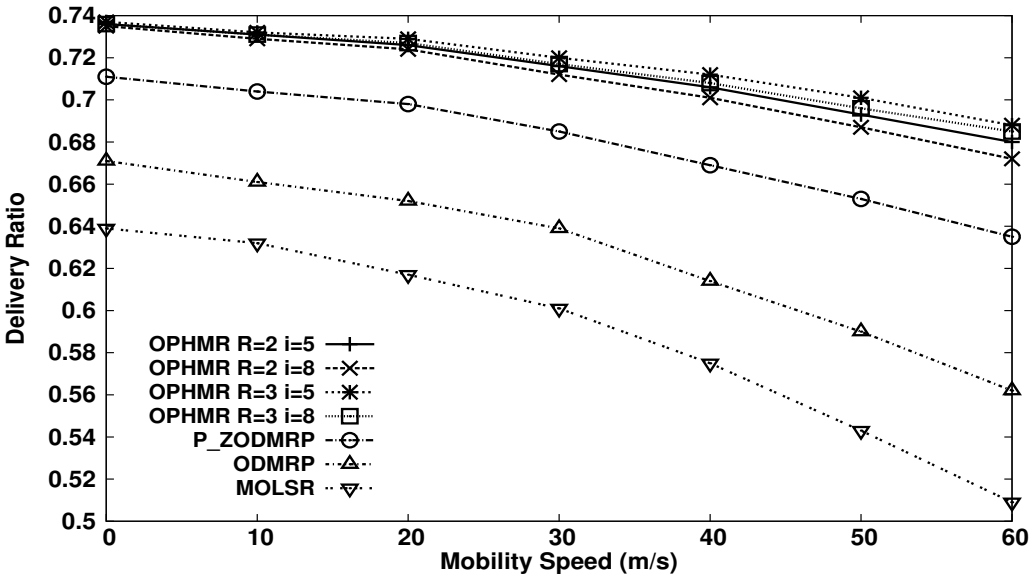


Figure 4.25: Delivery ratio vs. mobility speed.

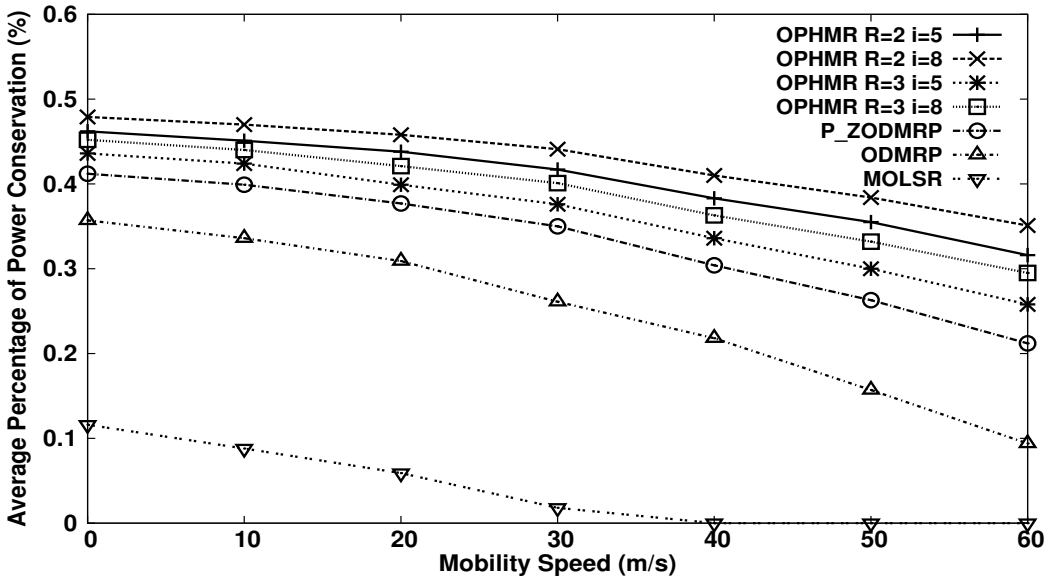


Figure 4.26: Average percentage of power conservation vs. mobility speed.

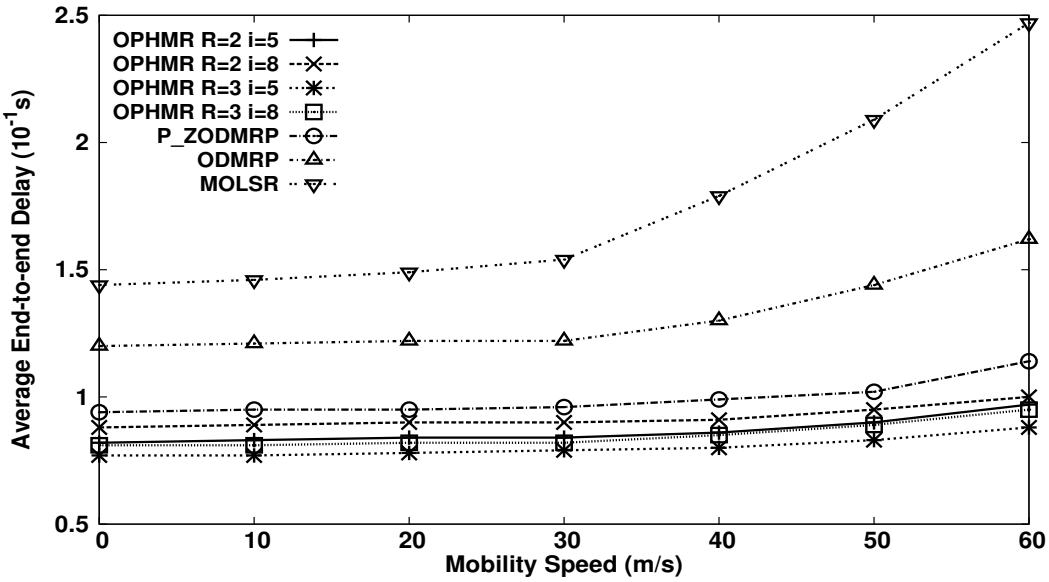


Figure 4.27: Average end-to-end delay vs. mobility speed.

shows that a more proactive behavior (with high zone radius R and low update interval i) has indeed lead to improved performance (as compared to the case of lower values of these parameters). In addition, all the protocols using mesh topology maintained a nearly constant performance. The OPHMR could have an enhancement of 80ms over the ODMRP.

Fig. 4.26 shows the average percentage of power conservation. This metric is defined as the sum of battery levels of all active nodes divided by their number. Because of its embedded MPR mechanism, the OPHMR could save up more power than the P_ZODMRP and the ODMRP. With their polymorphic behavior, the OPHMR and the P_ZODMRP protocols were able to achieve better performance than non-polymorphic ones. When the speed is at 60m/s, the OPHMR was able to save about 25% more of power usage than the ODMRP. In addition, for the OPHMR the least proactive behavior (with $R = 2$ and $i = 8$) was ben-

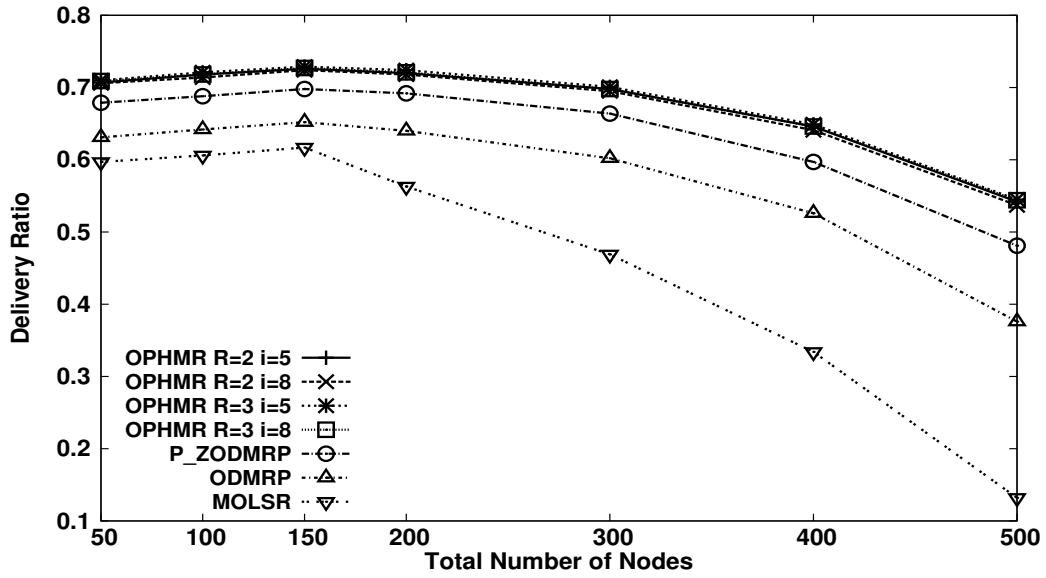


Figure 4.28: Delivery ratio vs. Nodes vicinity density.

eficial for power conservation, and the most proactive (with $R = 3$ and $i = 5$) one resulted in less power savings. This results show that power conservation is inversely proportional to higher proactiveness.

2) Effect of Nodes' Vicinity Density:

Experimental Scenario:

The total number of nodes within the defined area was varied from 50 to 500. Each node moves with a predefined maximum speed of 20m/s. The traffic load is 20 packets per second. Again in this scenario, we had 40 multicast members (forming a single multicast group) and 10 source nodes.

Fig. 4.28-4.30 shows the protocols' performance as a function of the vicinity density level. Fig. 4.28 shows the delivery ratio, and Fig. 4.30 shows the average end-to-end delay versus nodes vicinity density. We can observe that with an increase of nodes' vicinity density, there are more neighbors that cause the perfor-

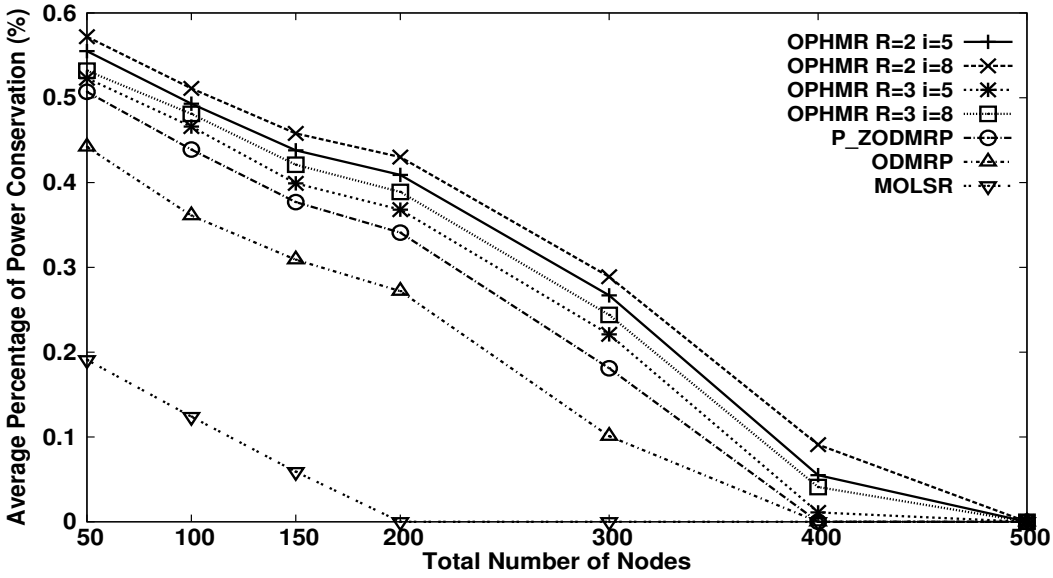


Figure 4.29: Average percentage of power conservation vs. Nodes vicinity density.

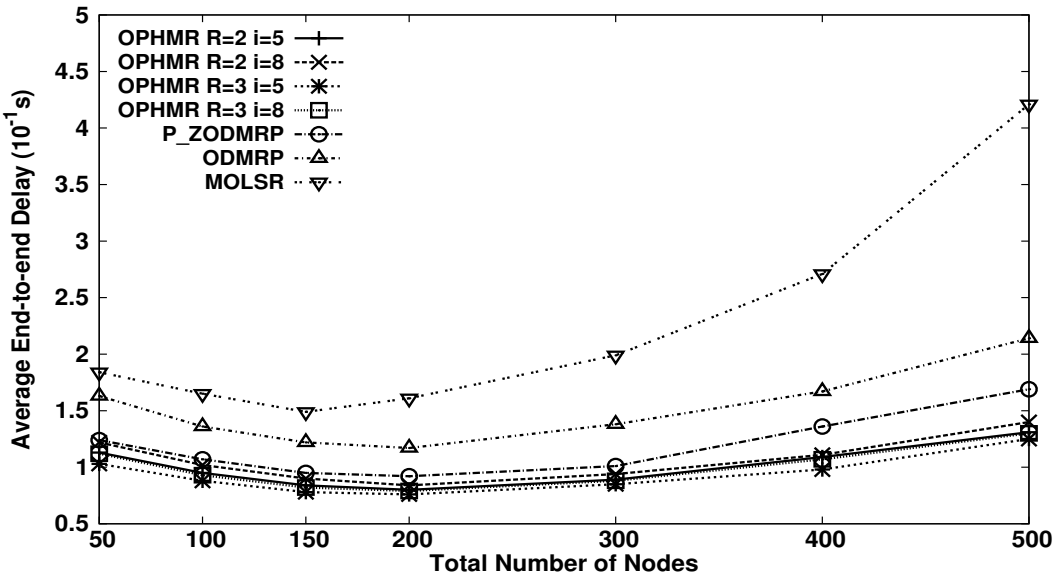


Figure 4.30: Average end-to-end delay vs. Nodes vicinity density.

mance of the pure reactive protocol to decrease sharply.

It can be seen from Fig. 4.28 and Fig. 4.30 that when the total number of nodes is low, the proactive behavior in the polymorphic protocols could increase the performance with its provision of fresher information of nodes' neighborhood. When node vicinity density is high, more neighbors could generate more control overhead for pure proactive protocols, and the reactive behavior of the polymorphic protocols could reduce the amount of the control packets while guaranteeing a good performance. In addition, for higher vicinity density the OPHMR superiority over both the P_ZODMRP and the ODMRP was clear. This is mainly due to the MPR based optimization scheme.

Another general observation related to effect of node density, is that there is (and that's expected) an optimal number of nodes per area that guarantees the best performance (in the figure 200 nodes seems to be the optimal number for the current setting for all the protocols except the MOLSR). This can be used as a guideline for setting the vicinity density threshold value.

Fig. 4.29 shows the average power conservation against nodes vicinity density. We can see that due to the polymorphic behavior of the OPHMR, it could save up more power usage and extend the battery life of the nodes. When the total number of nodes is 200, the average power level of the MOLSR reaches zero. When the total number of nodes is 400, the average power level of the ODMRP and the P_ZODMRP reaches zero, but the OPHMR could still prolong the battery life of some nodes up to an area density of 500 nodes. Again, for the OPHMR, the results confirm the positive effect of the lower proactivity levels on the power conservation (Best results are for the setting of $R = 2$ and $i = 8$).

3) Effect of Network Traffic Load:

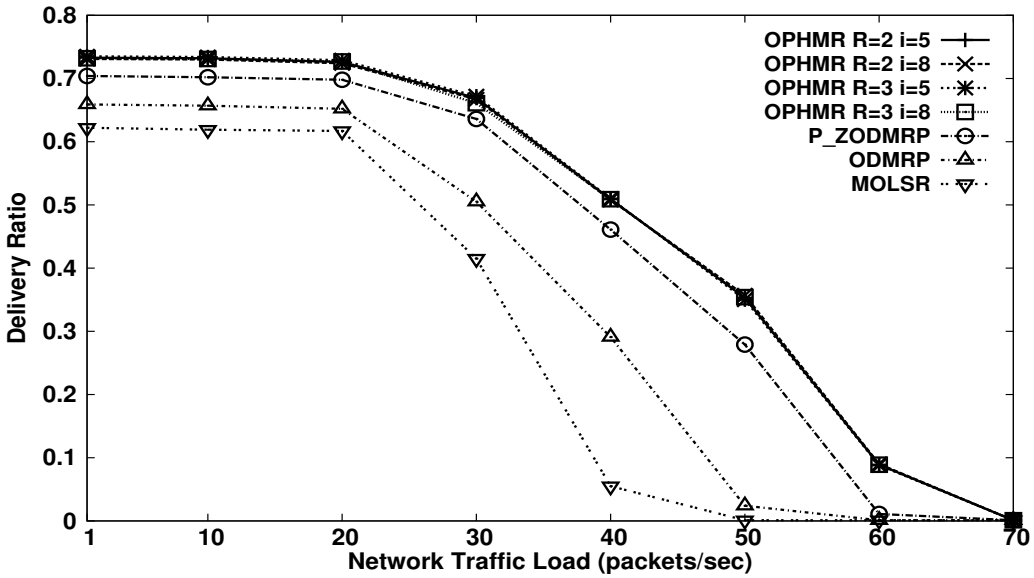


Figure 4.31: Delivery ratio vs. traffic load.

Experimental Scenario:

There are 150 nodes spread within the defined area and the number of packets the sources send was varied from 1 to 70 packets per second. Each node moves with a predefined maximum speed of 20m/s. The same setting of 40 multicast members and 10 source nodes was maintained in this scenario as well.

Fig. 4.31-4.33 depicts the protocols' performances as a function of traffic load. Fig. 4.31 shows the delivery ratio, and Fig. 4.33 shows the average end-to-end delay, both versus traffic load.

With the increase of the traffic load, most of the channel capacity is used by the data packets, and the deliverability is perfect until the saturation starts to appear. Above 40 packets per second the network performance degrades sharply (both in deliverability ratio and in latency). However, the OPHMR has distinguished itself from its peers with higher deliverability and lower latency.

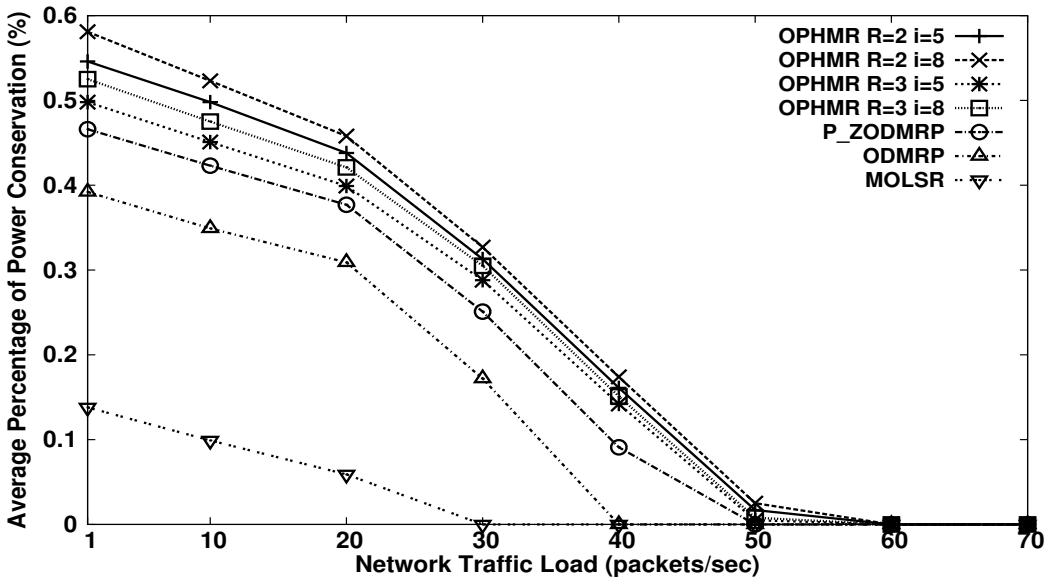


Figure 4.32: Average percentage of power conservation vs. traffic load.

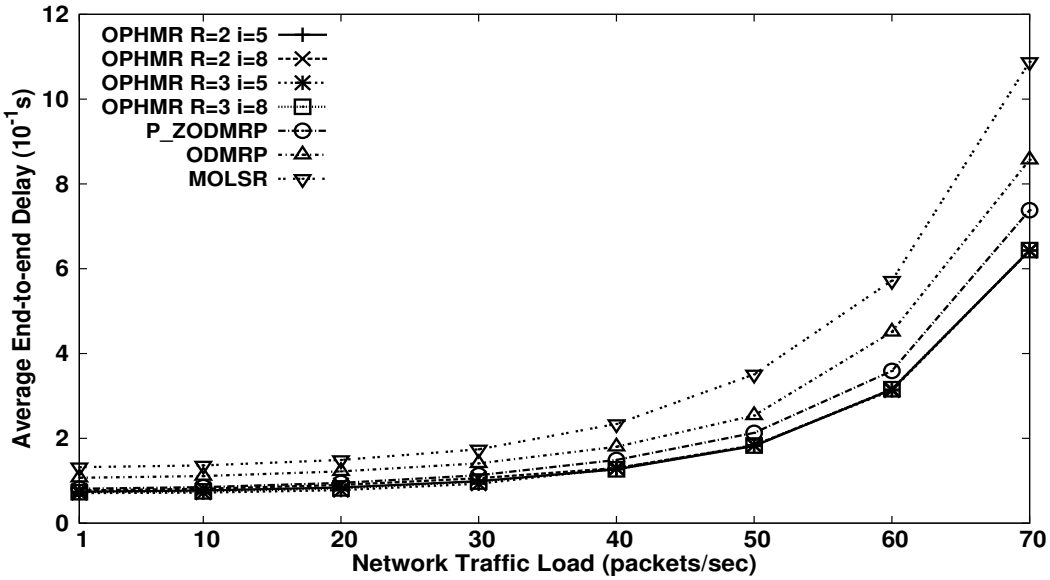


Figure 4.33: Average end-to-end delay vs. traffic load.

We also need to mention that the different settings of the zone radius R and the update interval i didn't have a perceivable effect. However, that effect is perceivable in the case of Fig. 4.29 that plots the average power conservation against traffic load. Again higher update intervals and lower zone radius were found to benefit power conservation.

In addition, in terms of power conservation the OPHMR showed superiority only below a traffic load of 50 packets per second. After that cut-off value, the performances of the OPHMR and the P_ZODMRP are almost identical. This can be attributed to the fact that above the cut-off value many nodes would have died off resulting in a weaker density, and thus the optimizing scheme of the OPHMR would have lost its effectiveness. The MOLSR was a great loser in all the simulations.

4) Performance Variation over Time:

Experimental Scenario:

We have also plotted the performance variation of the considered protocols with time in two different simulation settings. In this scenario, 150 nodes were considered. Traffic was set to 20 packets per second, and the maximum mobility speed is 20m/s. 40 multicast members and 10 source nodes were considered.

Fig. 4.34 and fig. 4.35 shows the performance of the protocols at different time-stamps. Fig. 4.34 shows the delivery ratio over time. We can see in the figure that in the first 500 seconds, the delivery ratios of all the protocols (except MOLSR) are relatively constant and the differences among them are small. With passage of time some nodes start using up their energy power and go off. This resulted in a decrease in delivery ratio, and also in the average power level. However, again, the polymorphic protocols outperformed the others, and the OPHMR

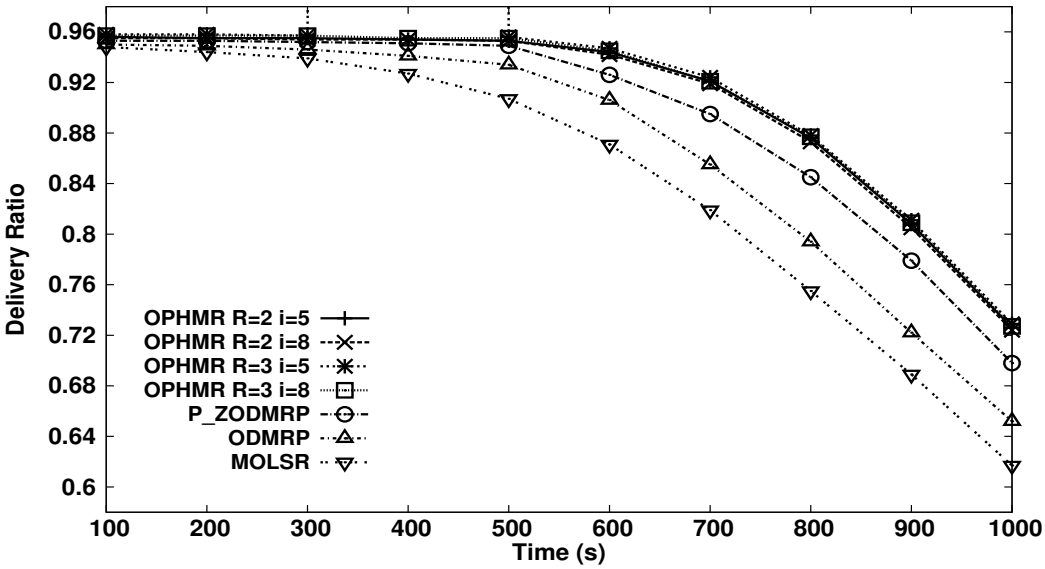


Figure 4.34: Delivery ratio vs. time.

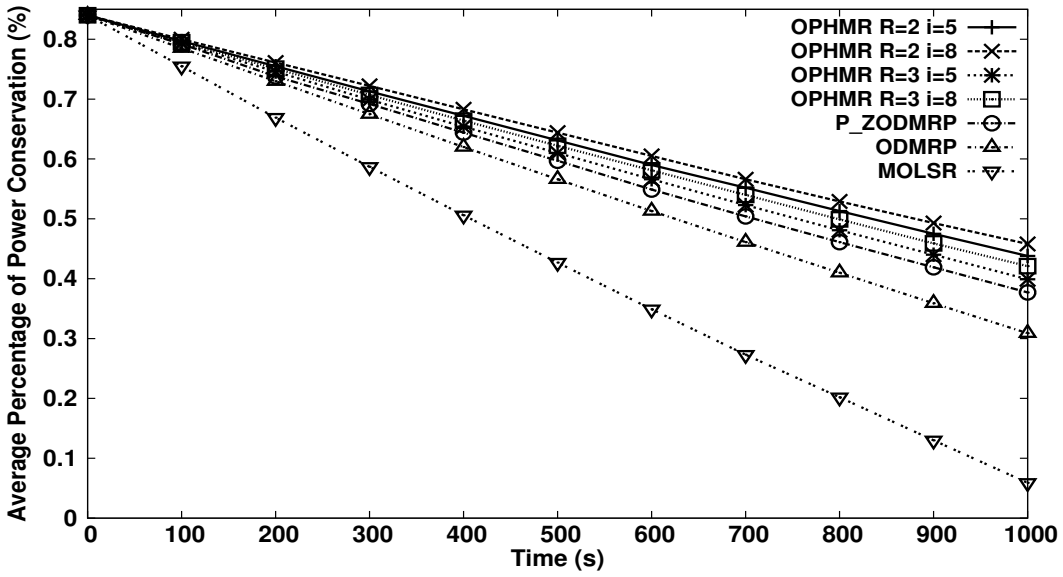


Figure 4.35: Average percentage of power conservation vs. time.

was distinguishable.

Fig. 4.35 depicts the average of power conservation over time. The plot confirms the beneficial effect of polymorphism in securing more power savings for the polymorphic protocols over the ODMRP and the MOLSR.

Another observation again is that the proactivity level didn't make a perceivable difference among the various settings of zone radius and update interval with regards to delivery ratios. However, on the power conservation side, the difference was clearer than its counterpart in the delivery ratio plot. Large gains of the OPHMR over the P_ZODMRP and the ODMRP were also observed. The MOLSR had the worst performance.

5) Performance Variation with Different Threshold Settings:

Experimental Scenario:

In this experiment, we evaluate the effect of the power threshold settings on the OPHMR protocol performance. In addition to the setting used in the above experiments, another setting is defined as follows: P_TH1 = 70%, P_TH2 = 40% and the threshold values for mobility and vicinity were set as above. In this scenario, 150 nodes were spread within the defined area. Node maximum mobility speed was varied from 0m/s to 60m/s. The traffic load was set to 20 packets per second, and 40 multicast members and 10 source nodes were considered.

Fig. 4.36, Fig. 4.38 and Fig. 4.37 depict the delivery ratio, the average end-to-end delay and the power conservation versus mobility speed, for the two power threshold settings considered. We have two parameters to consider here: the proactivity level translated in the setting of the zone radius R and update interval value i, and the conservativeness level implied by the threshold values of power. It can be noticed how high proactivity level ($R = 3$ and $i = 5$) improves protocol's

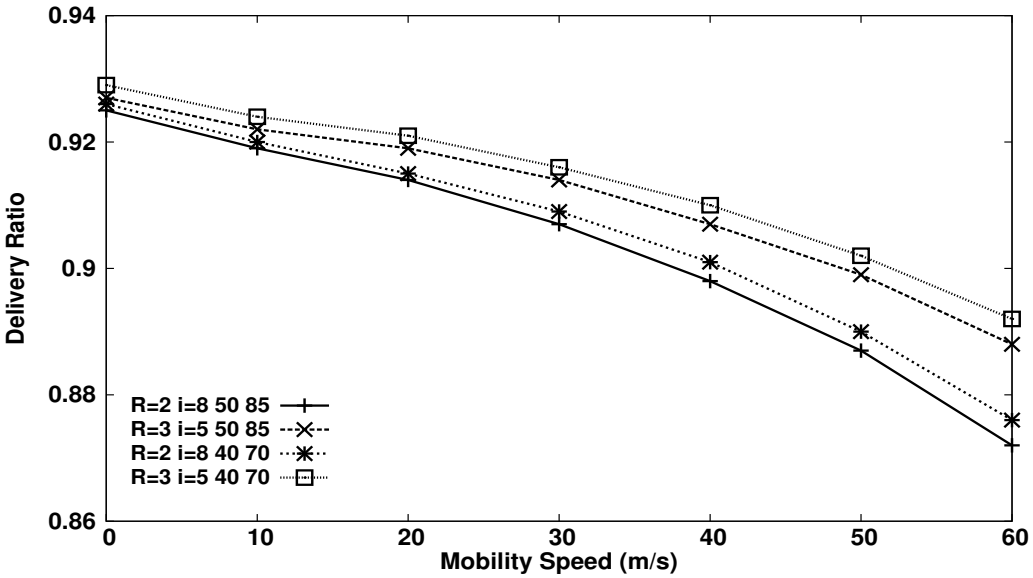


Figure 4.36: Delivery ratio vs. mobility speed.

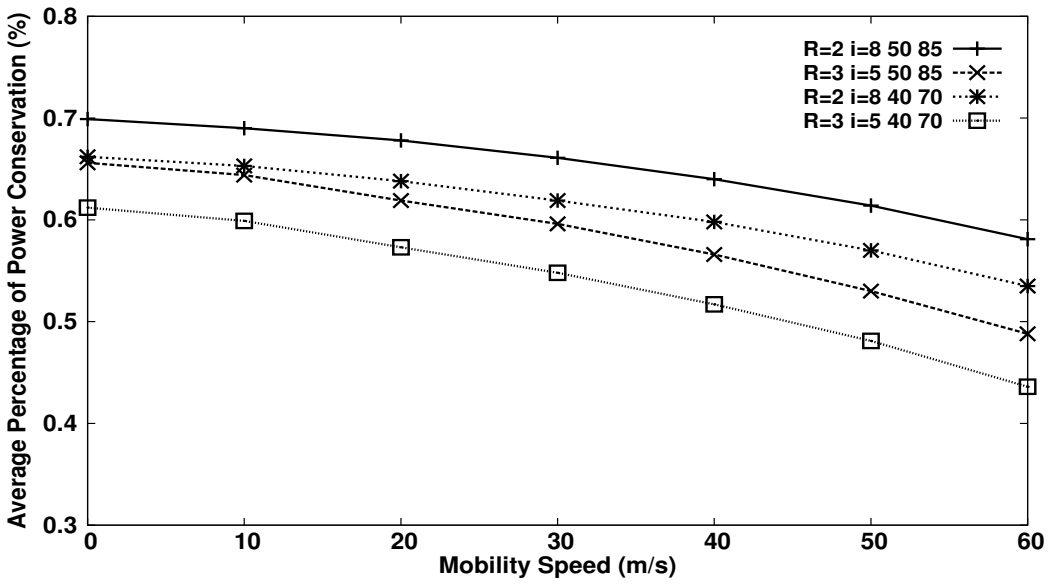


Figure 4.37: Average percentage of power conservation vs. mobility speed.

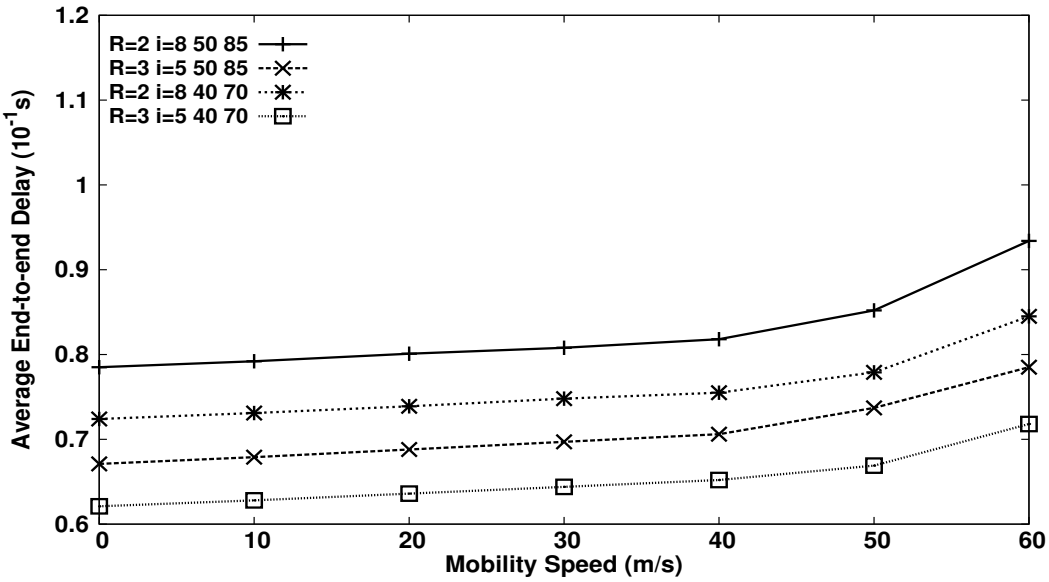


Figure 4.38: Average end-to-end delay vs. mobility speed.

performance at the cost of reduced power conservation gains, and viceversa (the setting of $R = 2$ and $i = 8$ decreases performance and increases power conservation gains). Then, at the same time, the conservativeness level can be used to tune or gauge the performance versus power conservation gain/loss. Here, setting the power threshold values low, increases performance and decreases power conservation gains, and vice versa.

4.6 Summary

We have proposed polymorphic routing protocol design concept. The protocol design is a novel way of combining three dimensions in protocol design, namely, *hybridity*, *adaptability* and *power awareness*. With regards to hybridity the protocol attempts to take benefits of the high efficiency of proactive routing in reducing

response time to transmission requests, and of the reduced control overhead offered by reactive routing.

To illustrate the polymorphic routing protocol design, we first introduced the P_ZODMRP protocol. P_ZODMRP is our first example of the polymorphic routing protocols. It defines a polymorphic algorithm to determine a node's behavior based on the node's power, mobility and vicinity conditions. The behaviors of nodes in P_ZODMRP are based on that of ZRP and ODMRP.

It was realized that P_ZODMRP can be further improved with an optimized forwarding mechanism borrowed from the OLSR protocol and thus were able to construct a better performing, polymorphic protocol named the OPHMR protocol. When compared to the ODMRP, the P_ZODMRP, and the MOLSR, the OPHMR clearly outperformed them in most situations. The superiority lies in the fact that on the long run, the protocol was able to extend battery life and enhance survivability of the nodes in MANETs. Hence, it has increased data deliverability ratio and decreased latency, while keeping the control overhead at acceptable levels.

The design approach we have adopted in this chapter is generic in nature and the choice of the right protocol (proactive or reactive) to use depends on its proven performance and on its applicability to the situation or environment where the protocol is deployed.

We think that this new concept of polymorphic protocols constitutes the next trend in the design of efficient multi-behavioral routing protocols for wireless, power-constrained networks such as MANETs.

Chapter 5

Proposed Polymorphic Unicast Routing Protocol in Vehicular Ad Hoc Networks

The good performance results reported for polymorphic routing in P_ZODMRP [10] and OPHMR [11] motivated us to capitalize on its suitability for routing in dynamic environments such as vehicular ad hoc networks (VANETs), and to propose for such networks a specifically tailored polymorphic routing protocol that we tag as the Polymorphic Unicast Routing protocol (PURP).

Some of the main hurdles in the deployment of efficient routing protocols for MANETs are the randomness of the mobility patterns of mobile nodes and their limited battery supplies. Another issue of concern is the power supply of the mobile nodes. Considering Vehicular Ad Hoc Networks (VANETs) these two hurdles are not critical, as the mobility patterns of nodes (mounted on vehicles) are more or less constrained and the power supply is not a problem.

As the issue of power is not of concerns in VANETs, it is illustrated that the PURP's parameters are adjusted adaptively according to different network conditions, namely, the mobility level, the vicinity density level, and the traffic load level. The resulting protocol gets vested with a polymorphic behavior that ensures dynamism and flexibility of operations to guarantee the best possible performance for the targeted criteria (such as lower latency and reduced overhead).

PURP's proactive mode of operation is driven by the Zone Routing protocol (ZRP) [14] its reactive behavioral mode is driven by the Ad hoc On-demand Distance Vector (AODV) routing protocol [15]. The choice of the ZRP and AODV protocols for the construction of the PURP comes from the fact that they are proven efficient, although not necessarily the best ones, as other extensions were proposed for them in the literature [60], [86]. Nevertheless, from its very nature the protocol design is generic and doesn't restrict the choice of the algorithms that will drive its reactive or proactive behavioral modes.

5.1 State of the Art Review of Routing Protocols for VANETs

A good coverage of some polymorphic and hybrid routing protocols reported in the literature for MANETs can be found in [11]. Additionally, we report a recent work of one of authors of this work on polymorphic protocols. Indeed, in a recent wok [87], Belghith *et al.*, proposed a new polymorphic routing protocol that combines the benefits of proactive routing and a (newly proposed [88]) probabilistic routing, *in a timely and periodic* manner in order to get the best of each approach

at the appropriate timing. The idea is based on the observation that proactive routing information gathered at the beginning of a routing period looses its accuracy rapidly in presence of high mobility and thus becomes more of burden than an asset.

However, probabilistic routing can remedy to this backwardness of the proactive routing and can take over the task of providing accurate routes (computed probabilistically) gradually in time, according to prescribed and carefully selected threshold values depending on the current mobility level. Very good results were reported with this new protocol that outperformed both proactive and probabilistic routings each considered alone [87]. This recent development again confirms the suitability of the polymorphic behavior approach for the design of routing protocols for dynamic networks.

In the remainder of this section, we will restrict our discussion to some of the serious works directed towards VANETs. In [89], the authors address a path finding procedure named Geographic Source Routing (GSR). They use the location information combined with the map information to calculate the route from a source to a destination, and use the source routing technique to determine the packet delivery path. With the map information, the path can avoid obstacles of transmission so that the route is guaranteed to be valid if there are no link breaks caused by mobility.

The authors of [90] define VANETs consisting of two components: vehicles and infostations (fixed access points connected to the Internet). Their proposed GeOpps protocol uses a combination of the concepts of geographical positioning and network delay tolerance. This protocol uses geographical information to calculate the route to the destination and uses the delay tolerant method to handle

data transmission. In the delay tolerant method data transmission is not much concerned about the end-to-end delay of one transmission. Thus, intermediate nodes can avoid unnecessary data packet transmissions by storing data packets and sending them later to the destination nodes once paths leading to them are found.

The concept of routing for Vehicle-Based Disruption-Tolerant Networks is proposed in [91]. The authors mainly deploy a new protocol, named MaxProp, in the disruption tolerant networks. Disruption tolerant networks allow for routing in networks where simultaneous end-to-end paths are unstable or unlikely. The disruption tolerant networks described in [91] are similar to MANETs and the delay tolerant networks are as described in [90]. The data transmission in disruption tolerant networks is the same as in the delay tolerant networks, that is, an intermediate node is allowed to store a data packet and deliver it to the destination later. The authors mainly focused on local buffer management of the node for the storage of data packets.

The authors of [92] presented a so called reliable geographical multicast routing protocol. In this protocol, the initiator node broadcasts geographical information within a limited zone area. The process is similar to that of the AODV, where a node reactively sends out request messages and wait for replies. Each node updates its routing information using the information extracted from the control packets received and determines the routing path.

In [93], the authors propose a position-based QoS routing protocol for VANETs. The core concept of this protocol is based on dividing the whole area into smaller uniform-size areas called grids. A source node chooses one node within every grid along the transmission path for packet forwarding. The length of the edge of

each grid must guarantee that two nodes within two consecutive grids are within their mutual transmission range. It is assumed that each node within the network is able to get the location information from a GPS device and can determine its grid. The protocol establishes routing paths on-demand. Nodes selects a specific forwarding node in its grid for any path finding process. Selected nodes in the grids constitute the routing path till the destination.

Finally the Connectivity-Aware Routing (CAR) protocol, proposed in [94], is a position-based routing protocol for VANETs. The CAR protocol uses a periodical updating HELLO message to maintain the neighbor's mobility and connectivity information. Each HELLO message contains the speed and the direction (named velocity vector) of the node. On receiving a HELLO message, the node uses the received information together with its own information to calculate the possible expiration time of a connection. Then the node puts such a connection pair into its neighborhood routing table.

When the source node requires a path to send the data to the destination, it generates a broadcast packet, with its coordinates and velocity vector, to look for the destination. On receiving the broadcast packet, a node places its own coordinates and velocity vector into the packet together with some other information to guarantee connectivity, then forwards the packet. The routing is based on using an anchor into the broadcasted packet that is defined when the angle between two velocity vectors is greater than a threshold value.

When the broadcasted packet finally reaches the destination node, this latter waits for a period of time for the other broadcasting packets from other possible routes. It then chooses the best route and replies to the source following the reverse path of the selected route. The reply message contains the anchor points along the

path. So that the route consists in a series of anchor points towards the destination.

The above presented protocols exhibit different forms of adaptive routing, but did not specifically emphasize those adaptive features. In fact, most of them have defined complex behaviors that can be formalized (or broken down) into simpler sub-behaviors to deal with different conditions. For example, in the case of the GSR protocol [90], we could define QoS requirement that may or may not use delay tolerance in the network (e.g., in case of real-time applications). This will lead to the definition of two types of behaviors: delay tolerant or non delay-tolerant.

Another example, is the case of the CAR protocol [94], where the periodical updating process can be made adaptive. The node uses the number of neighbors (its vicinity density information) to determine the updating interval time. This may also call for formalizing two types of behaviors based on a threshold value of vicinity density level.

With this line of thought, specifically dealing with situations of high vicinity density (busy neighborhood), high traffic intensity, or high mobility with appropriately defined behaviors would enrich these protocols with capabilities that will realize better performance and consequently would provide better QoS. This is what we propose to do in the present endeavor.

5.2 The PURP protocol

5.2.1 Polymorphic Algorithm in PURP

Targeting VANETs and adopting the same design approach as that of the OPHMR, we design the PURP protocol. Indeed, for this protocol, we identify the parame-

ters of interests as the mobility speed, the vicinity density and the traffic intensity. We define two traffic load level thresholds, one mobility level threshold and one vicinity density threshold. These inputs are easily obtainable from each MN. The mobility level may come directly from the vehicle's speedometer, the vicinity density level is computed from monitoring the exchanged control packets, and the traffic load is measured by periodical detection of channel usage.

The rationale for choosing two traffic intensity thresholds comes from the fact that in urban areas the likelihood of car traffic jams is high and its direct impact on data traffic jams is straightforward. Thus, we chose two levels of of thresholds to be able to tune to less proactive behaviors gradually in the presence of increasing data traffic.

Choosing different thresholds levels for mobility can have also its own rationale, especially, if we would like to have various behaviors in presence of different mobility levels. However, in our opinion, urban areas would have usually a high number of vehicles moving around, and even in high mobility, alternative routes for data transmission would be found easily. Consequently this will not warrant an increase of proactiveness.

As for vicinity density thresholds, we have chosen one single level, taking into account the results reported in [82, 83, 84] which concur on the fact that maintaining at least six neighbors leads to high reliability of a network in terms of connectivity for medium sized networks. Consequently, a single cutoff point around that connectivity level was chosen and adopted in the simulation.

The different threshold values for traffic load level are denoted (T_TH1 and T_TH2, where $T\_TH1 > T\_TH2$). The mobility speed threshold is denoted by (M_TH) and the vicinity density threshold value is denoted by (V_TH). These

will dictate the choice of the appropriate behavioral mode for an MN to engage into.

The four behavioral modes of operation are:

1. Proactive Mode 1 (*PM1*): when an MN is in *PM1*, it periodically updates its neighborhood topology and topology information by sending out an update packet with the Zone Radius R set as the Time-To-Live (TTL) and the update interval is set to a tunable parameter value i. The MN also maintains a Neighborhood Routing Table (NRT) which stores the topology and information saved in the received update packets.

2. Proactive Mode 2 (*PM2*): the behavior of an MN in *PM2*, is similar to the one of an MN in *PM1*, however, the Zone Radius is set to $R-1$. (a moderate proactive state).

3. Proactive Mode 3 (*PM3*): the behavior of an MN in *PM3*, is similar to the one of an MN in *PM2*, however, the update interval is set to $2\times i$ (the least proactive state).

4. Reactive Mode (*RM*): an MN in *RM* does not send out update packets but maintains the NRT table using information stored in the received packets.

The algorithm is shown as follows:

Our design of the PURP described in **Algorithm 3** aims at improving the overall protocol's performance. Effectively, when an MN is in a high traffic load scenario ($> T\_TH1$), the MN is forced to perform in RM. In contrast, when such MN is in a low traffic load scenario ($< T\_TH2$), the MN runs PM1.

Algorithm 3 Polymorphic Algorithm

if $TrafficLoad > T\_TH1$ **then**
  if the MN is not in *RM*, it switches to *RM*.
  then it notifies neighbors about the mode switch.
else
  if $TrafficLoad < T\_TH2$ **then**
    if the MN is not in *PM1*, it switches to *PM1*.
    then it notifies neighbors about the mode switch.
  else
    if $Vicinity < V\_TH\ OR\ Mobility > M\_TH$ **then**
      if the MN is not in *PM2*, it switches to *PM2*.
      then it notifies neighbors about the mode switch.
    else
      if the MN is not in *PM3*, it switches to *PM3*.
      then it notifies neighbors about the mode switch.
    end if
  end if
end if

In extremely high traffic load scenarios, massive data transmission could exhaust the channel capacity and cause high traffic jams (due to collisions). In such cases, the reactive mode is ideal since the PM usually aggravates the burden on the channel, and the on-demand characteristic of reactive mode could effectively reduce the generation of control packets transmission and mitigate the traffic channel usage so as to free more traffic channel capacity for data transmission. On the contrary, when the input traffic load is low, PM modes lead to faster response to transmission requests.

In case of medium traffic load situation, the MN's vicinity density level and mobility level are taken into consideration to determine the MN's behavior. The MN may operate in a moderate proactive mode (i.e., PM2), if its vicinity density level is low ($< V\_TH$) or if its mobility level is high. In contrast the MN may operate in a least proactive mode, that is PM3, when the vicinity density level is

high and mobility level is low.

When the vicinity density level is high, the MN has many neighbors around it. It switches to a RM to reduce congestion while still guaranteeing fast response to transmission requests as the on-demand building of transmission paths will converge quickly because of the availability of links.

For the consideration of node's mobility level, when the mobility level is high, topology changes occur more frequently, that is, link breaks and link repairs will happen more often. In such a case, more proactiveness leads to better performances since the MNs can be much more acquainted with topology information. Thus, with efficient and timely topology information, more accurate routing can be realized.

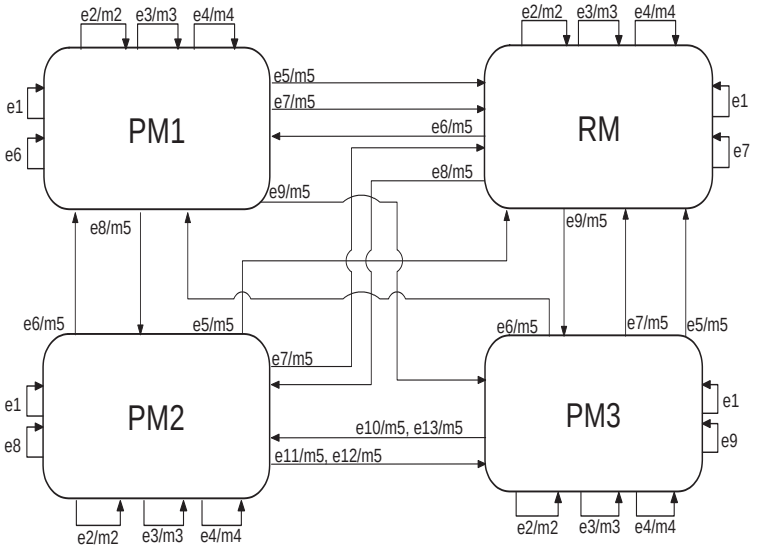
Based on the above considerations, we choose the least proactive mode PM3 in high vicinity density and low mobility level scenario. Otherwise, a moderate proactive mode PM2 is used.

5.2.2 The Finite State Machine Diagram of the PURP protocol

The state-transition diagram of the protocol is illustrated by the Finite State Machine depicted in Fig. 5.1.

5.2.3 Routing Issues

The detailed implementation of each of the four behavioral modes is given as follows. As stated above, the PURP is built using the proactive behavior of the ZRP and the reactive behavior of the AODV. In addition, the protocol is augmented with the Multipoint Relay (MPR) based optimization mechanism of the OLSR



e1: Route establishment request or a new/received packet ready to transmit/forward
e2: Neighbour's mode change indication packet received
e3: T3 expired
e4: Neighbour's routing table received
e5: T5 expired
e6: Traffic load changed to low level
e7: Traffic load changed to high level
e8: Traffic load changed to medium level when mobility level is high or vicinity level is low
e9: Traffic load changed to medium level when mobility level is low and vicinity level is high
e10: Vicinity level changed from high to low when mobility level is low and traffic load is medium
e11: Viciinity level changed from low to high when mobility level is low and traffic load is medium
e12: Mobility level changed from high to low when vicinity level is high and traffic load is medium
e13: Mobility level changed from low to high when vicinity level is high and traffic load is medium

m1: Establish the route or transmit the new/received packet
m2: Update routing table according to the received neighbour's mode change indication packet
m3: Reset T3 and broadcast its routing table
m4: Reset T5 and update routing table according to the received neighbour's routing table
m5: Send mode change indication packet

T3: Routing table broadcast timer
T5: Neighbour's routing table received timer

Figure 5.1: FSM diagram of the PURP routing protocol

[32]. This is done with the objective of reducing the amount of control packets forwarded. We describe next its proactive and reactive behaviors, as well as the adopted optimized forwarding mechanism.

Routing Tables and Identifiers:

Each MN maintains two routing tables. One routing table, which is called Forwarding Table (FT), contains entries about the route to each destination. Each entry in FT has the fields: destination address, destination sequence number, next hop address and lifetime.

The other routing table, which is called Reverse Path Table (RPT), maintains the reverse path information within the path finding procedure. Each entry in RPT contains the following fields: destination address, source address, source sequence number, broadcast id, expiration time for reverse path and last hop.

Each MN also maintains a monotonically increasing sequence number. Such a sequence number is used to extinguish the stale routing information in the routing table and to prevent loops. The detailed usage of the routing tables and identifiers are described below.

Proactive Behavior:

When an MN is in either of the three proactive modes, it periodically sends out update packets which has a TTL set to the Zone Radius. The update packets contain the FT of this MN. When an MN receives an update packet, it updates its FT. If the MN already has an entry that has the same destination address with the information from the update packets, it compares the destination sequence number in the two entries. If the sequence number in the update packet is greater than the one in the FT, the MN updates its corresponding entry, replaces the destination sequence number in the FT with the one received in the update packet, sets the

next hop address field to the address from which the update packet is received, and refreshes the lifetime. If the update packet contains an entry that does not exist in the FT, the MN inserts such entry into the FT and replaces the next hop address field to the address from which the update packet is received. After updating the FT, if the MN is in the MPR set, it reduces the TTL by 1 and forwards the packet.

Reactive Behavior:

The reactive behavior is similar to the behavior of AODV. When an MN has packets to send to a destination, it first looks up its routing table to see whether it already has a route to it. If found, it simply sends the data to the MN through the next hop field. If not, the source MN generates a route request (RREQ) packet with the following fields: source address, source sequence number, destination address, destination sequence number, broadcast id and hop count. The two fields, source address and source sequence number, uniquely identify the RREQ. The broadcast id is incremented whenever the source generates a new RREQ. The source MN broadcasts this RREQ to all its neighbors.

On receiving an RREQ, an MN first looks up in its RPT. If there is an entry that has the same source address and broadcast id, it means that the MN received a redundant RREQ and it drops it. Otherwise, the MN looks up in its FT to see whether it has some information to the destination. If it finds that it has an entry to the destination and the destination sequence number is greater than the one in the RREQ, the MN generates a route reply (RREP) to the source MN. If not, the MN updates its RPT using the information in the RREQ, sets the last hop field to the address from which the RREQ is received and then forwards the RREQ.

Eventually, a RREQ reaches an MN that possesses a route to the destination. The MN then generates a RREP with the following information: source address,

destination address, destination sequence number, hop count, and lifetime. The RREP is sent through the reverse path to reach the source MN that builds up the route.

On receiving the RREP, an MN updates its FT to insert the information received. The next hop field is set to the MN from which the RREP is received. An MN propagates the first RREP received for the same source MN. If multiple RREPs are received, the RREP with greatest destination sequence number is processed. If two or more have the same sequence number the one that has the smallest hop count will be used. The processing includes updating the FT and forwarding it according to the RPT. Then the source MN receives the RREP, it can begin data transmission. Later it can update its routing information if a better route is discovered.

When a link break is detected, the upstream MN of the break generates a unique RREP with a fresh destination sequence number and a hop count field that is set to unlimited. Then the MN propagates the RREP to all active upstream neighbors. Those MNs subsequently relay this message to all active source MNs. When the source MN receives such RREP, it can restart a path finding procedure if it still needs the route to the destination. The new path finding procedure is the same as described above.

In Fig. 5.2, we depict the flowchart summarizing the routing behavior specific of the PURP.

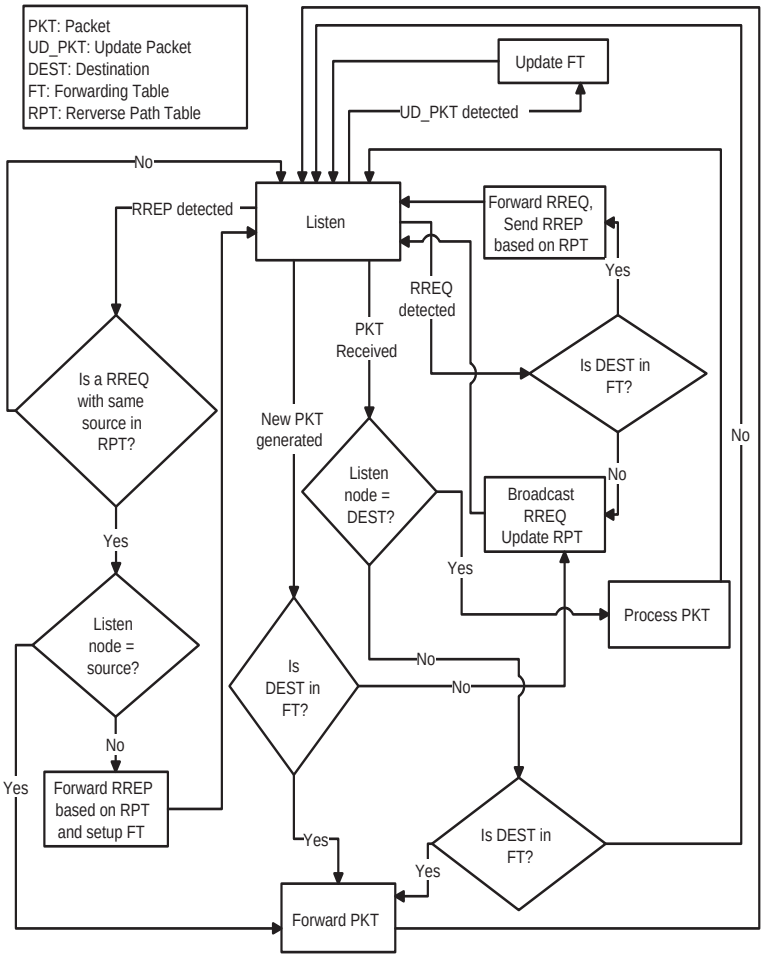


Figure 5.2: Flowchart of the PURP routing protocol

Simulation time	1000s
Simulation area	5000m×5000m
Propagation range	225m
Channel capacity	2Mbps
MAC protocol	The IEEE 802.11 MAC [1]
Traffic type	constant bit rate (CBR)
Power model	L. F. Feeney's work [81]
Zone update interval	5s
Zone radius	3
Zone lifetime	180s
Packet sending rate	10packets/s
Packet size	512 bytes

Table 5.1: The parameters for the simulation.

5.3 Performance Evaluation of PURP Protocol

We first validate our algorithm through simulation to evaluate the performance with fixed behaviors and scenarios. After that, we have performed a simulation based comparison of the performance of PURP against those of the AODV, the DSR the ZRP and the OLSR.

5.3.1 Simulation Scenarios

The simulation of the above protocols was implemented using the GlomoSim library [79]. Some common parameters are listed in Table.5.1.

Two parameters, R and i, were pre-configured for the PURP, where R denotes the Zone Radius (in number of hops) and i the tuning factor used for determining the update interval and table entries' lifetime.

The mobility model we chose in the simulation is Manhattan mobility model [95, 96, 97]. In our simulation experiments, the total area is divided into a 25x25 grid, and each MN can only move along grid lines. Each grid line runs two lanes in

different directions. At the intersection of the grid line, the direction goes either to the left or right. We use 0.5, 0.25 and 0.25 as the probabilities for an MN to move forward, turn left, or right respectively. When the node reaches the edge, the possibilities are 0.5 and 0.5 for an MN to turn left or turn right respectively. When the node reaches the corner, the node is forced to turn according to the border.

Each MN has a time interval with the length between 1 to 10 seconds, at the end of the time interval, the MN changes its speed. For example, if the maximum speed is n for the system, and the current speed is v, the speed of MN for the next time interval is taken randomly and uniformly from $v - 0.1n$ to $v + 0.1n$. In addition, the newly chosen speed is bounded by $(0, n]$. We have a pre-defined value of Safe_Distance, when the distance of two MNs, which are within the same lane and have the same direction, is smaller than the Safe_Distance, the latter MN should modify its speed so as not to exceed the foregoing MN. In other words, we do not consider the overtaking. We set the Safe_Distance value to 100 meters.

Fig. 5.3 depicts a layout of the grid according to which the mobile nodes move along the grid lines.

For the performance measures, we evaluate the delivery ratio, end-to-end delay and overhead against mobility speed, network traffic load and the total number of MNs.

5.3.2 Sensitivity Analysis

1)Protocol validation:

in the protocol validation simulation, we have defined two different simulation settings. In the first one, we force the mobile nodes to stay in either PM1 or RM

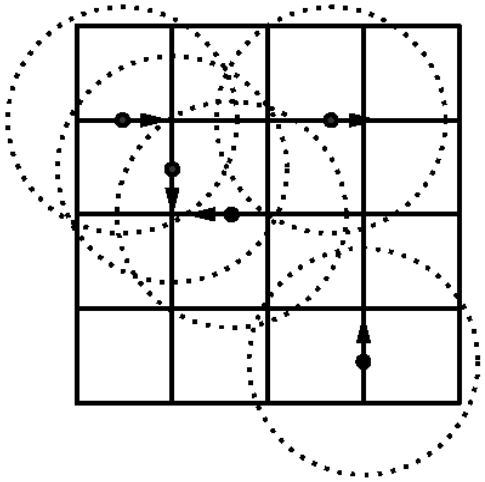


Figure 5.3: Grid layout for moving mobile nodes.

mode with the traffic load level set to high (50 packets/s), medium (30 packets/s) or low (10 packets/s) respectively. There are 1000 MNs uniformly spread within the defined area. The MN mobility speed varies from 20km/h to 80km/h. Fig. 5.4 and Fig. 5.5 show the result of this simulation. We can see that PM1 outperforms RM in terms of deliverability when the traffic load level is between low and medium. Whereas in high traffic load scenario, RM achieves better deliverability.

In terms of end-to-end delay, PM1 mode exhibits superior performance than that of the RM mode for low and medium traffic levels and looses the ground in case of high traffic loads. It can also be noticed that the mobility speed does not affect the performance of both types of modes considerably.

In the second simulation set, we force the mobile nodes to stay in PM1 or RM while maintaining a medium traffic load (30 packets/s), and varying the mobility

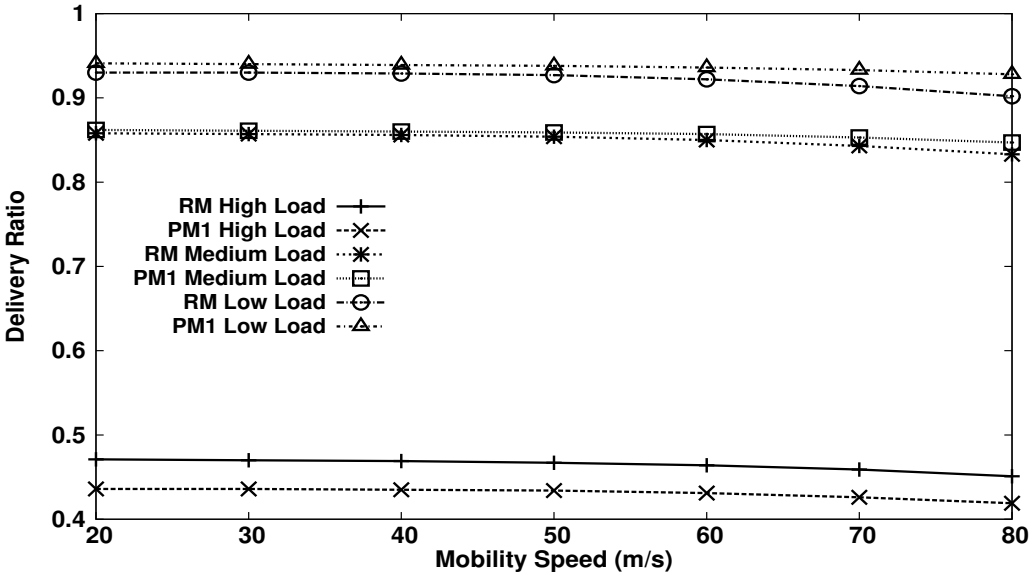


Figure 5.4: Delivery ratio for different traffic load models.

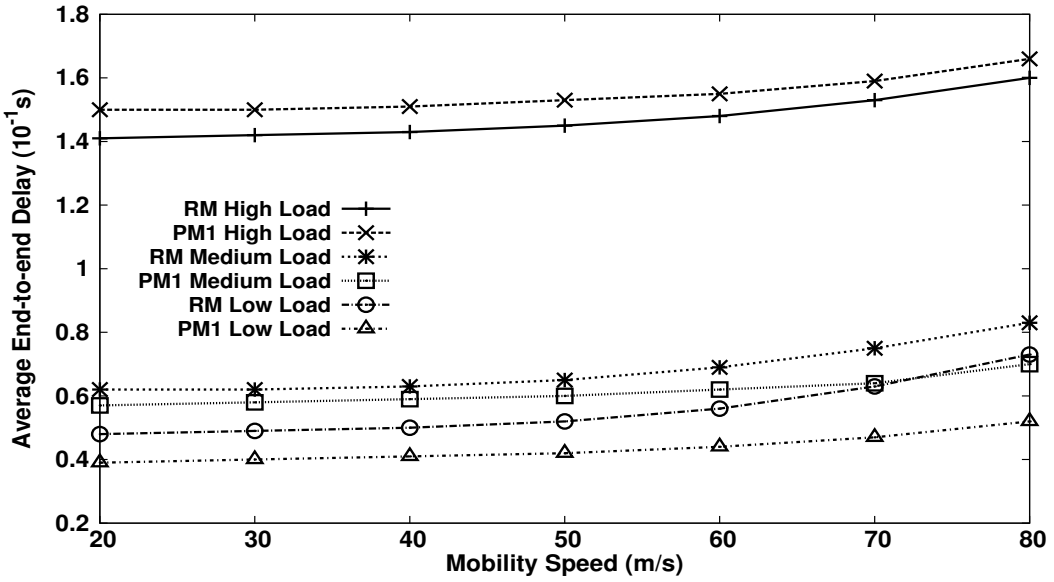


Figure 5.5: Average End-to-end delay for different traffic load level.

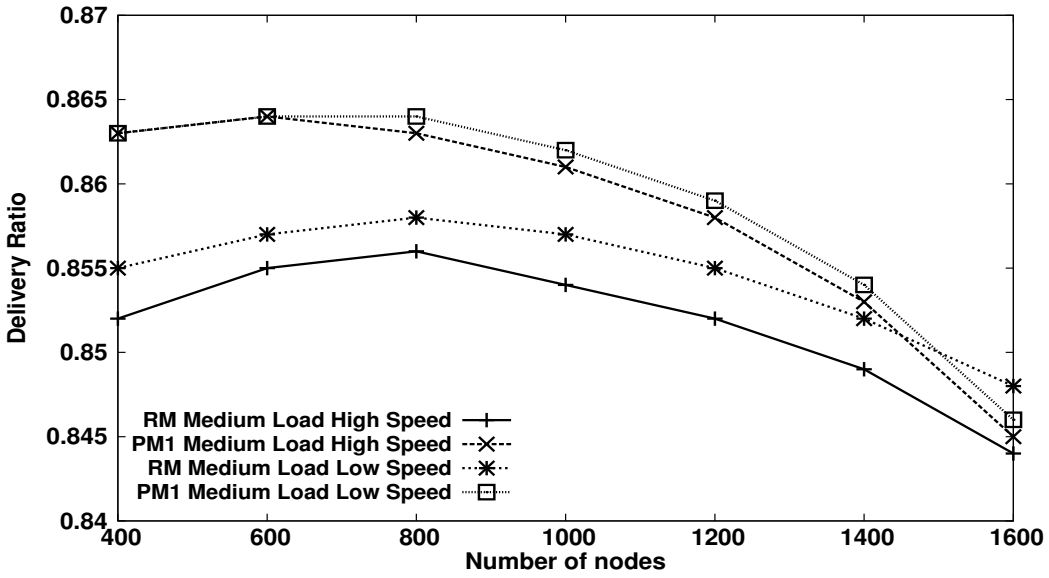


Figure 5.6: Delivery ratio for medium traffic & various mobility levels.

level between high (80km/h) and low (20km/h) respectively. The total number of MNs within the defined area varies from 400 to 1600. Fig. 5.6 and Fig. 5.7 show the results of this simulation. We can observe that in terms of deliverability PM1 mode outperforms that of RM, up until a high level of vicinity density. This superiority of the PM1 mode is also reflected on the results on the end-to-end delay (Fig. 5.7). In addition, a clear trend in the curves show an increase in good performance up to a certain level of vicinity density, above which a decrease in performance is noticed. This testifies that for each setting there is an optimal level of vicinity density that gives the best performance and suits the choice of the protocols parameters.

The last observation we can draw from these results is that high mobility speed affects negatively any operational mode of the protocol, and that is an expected result.

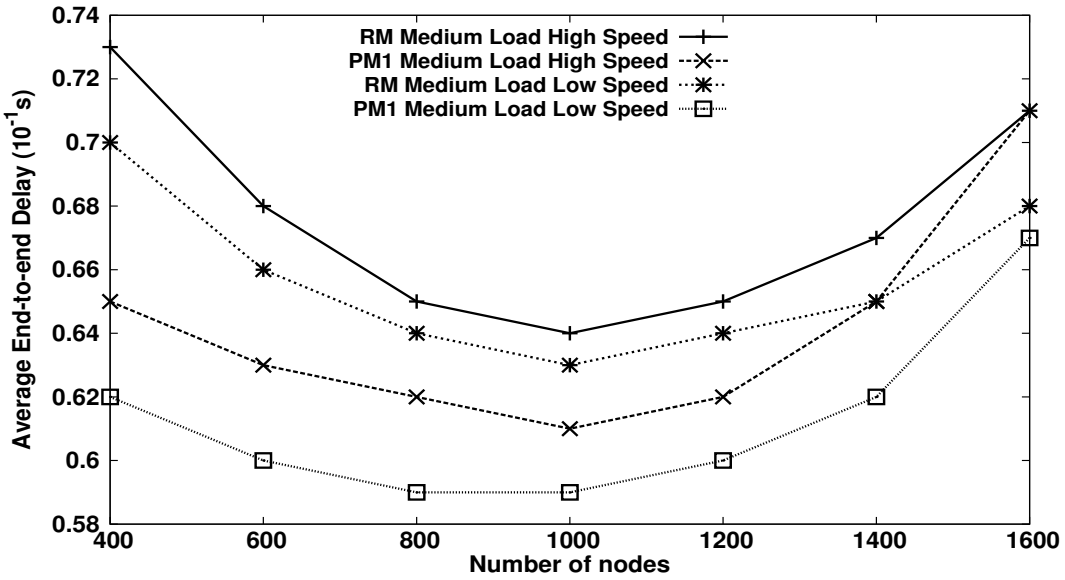


Figure 5.7: Average End-to-end delay for medium traffic & various mobility levels.

2) Mobility Speed:

Experimental Scenario:

In this scenario, 1000 nodes were uniformly spread within the defined area. The node mobility speed varies from 20km/h to 80km/h. The traffic load is set to 20 packets per second, and 100 source nodes are considered.

Our focus here is on the sensitivity of the mobility to various performance parameters. Figs. 5.8-5.10 respectively show the delivery ratio, the average end-to-end delay, and the control overhead packets versus the mobility speed. From the results, we see that the PURP protocol tops the other considered protocols for the delivery ratio with a relatively low end-to-end delay.

The polymorphic behavior of PURP is indicated in Figs. 5.10 where PURP adjusts its mode during the operation which generates control packets lesser than that of the proactive routing protocol but more than that of the reactive routing

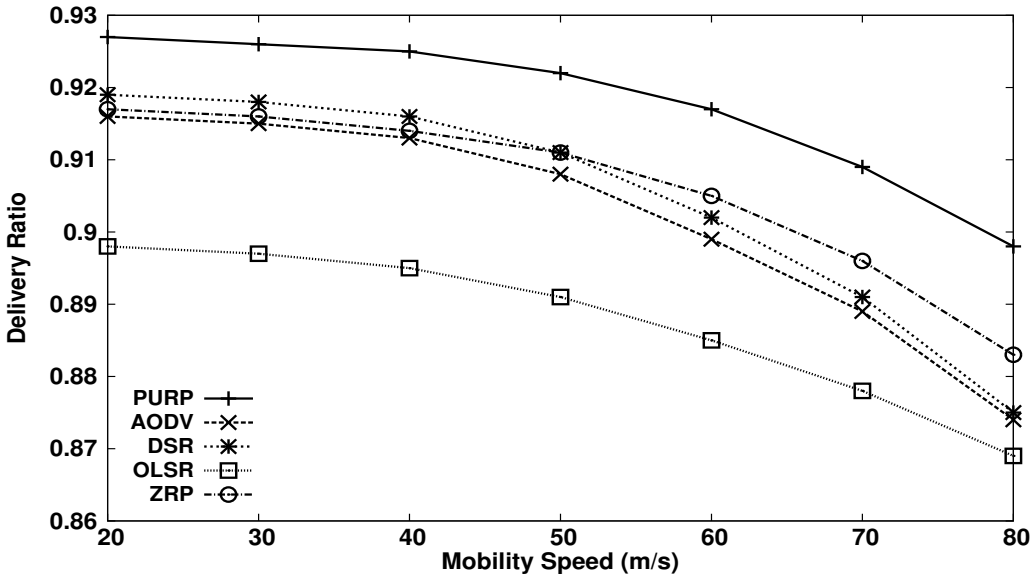


Figure 5.8: Delivery ratio vs. mobility speed.

protocols. Note that while the control packets per data packet of PURP is higher than those of reactive routing protocols due to its proactive mode, the proactive mode occurs when the channel is lightly loaded, hence it does not cause a negative impact on the performance which is clearly shown in Figs. 5.8-5.9.

3) nodes' Vicinity Density:

Experimental Scenario:

The total number of nodes within the defined area varies from 400 to 1600. Each node moves constantly with a predefined speed of 40km/h. The traffic load is 20 packets per second. Again in this scenario, we use 100 source nodes.

As shown in Figs. 5.11-5.13, again, PURP outperforms other protocols in almost all situations. Precisely, PURP gives the highest delivery ratios among all with low end-to-end delay. There is only a small range of vicinity density level where OLSR has slightly lower end-to-end delay. This is because OLSR is a pure

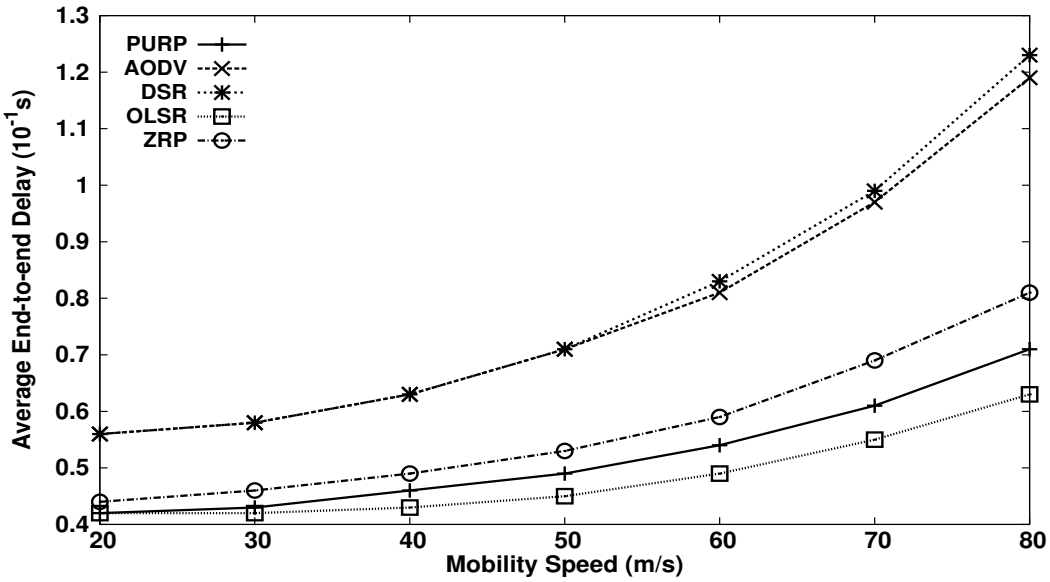


Figure 5.9: Average end-to-end delay vs. mobility speed.

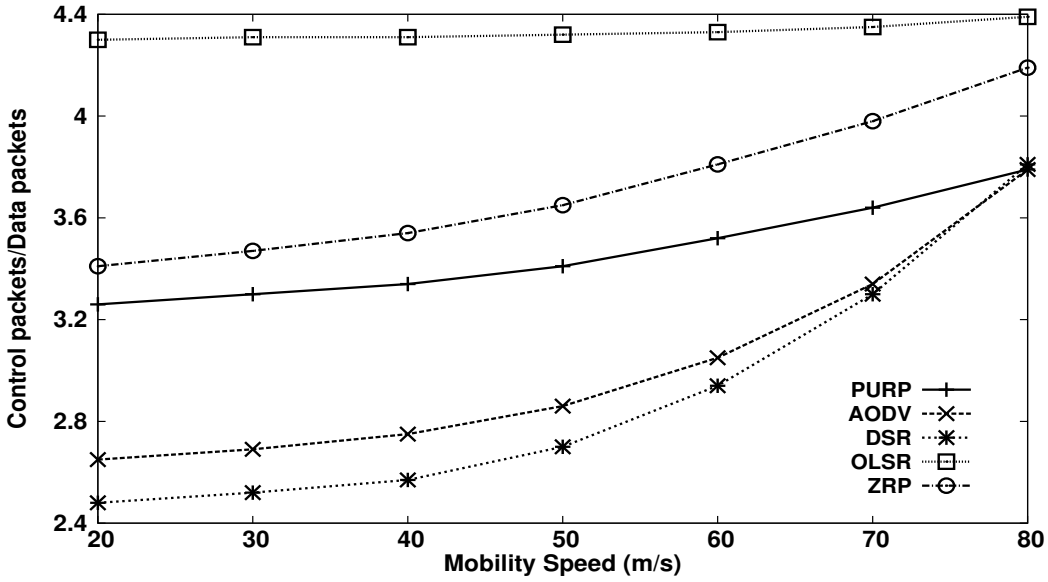


Figure 5.10: Control overhead vs. mobility speed.

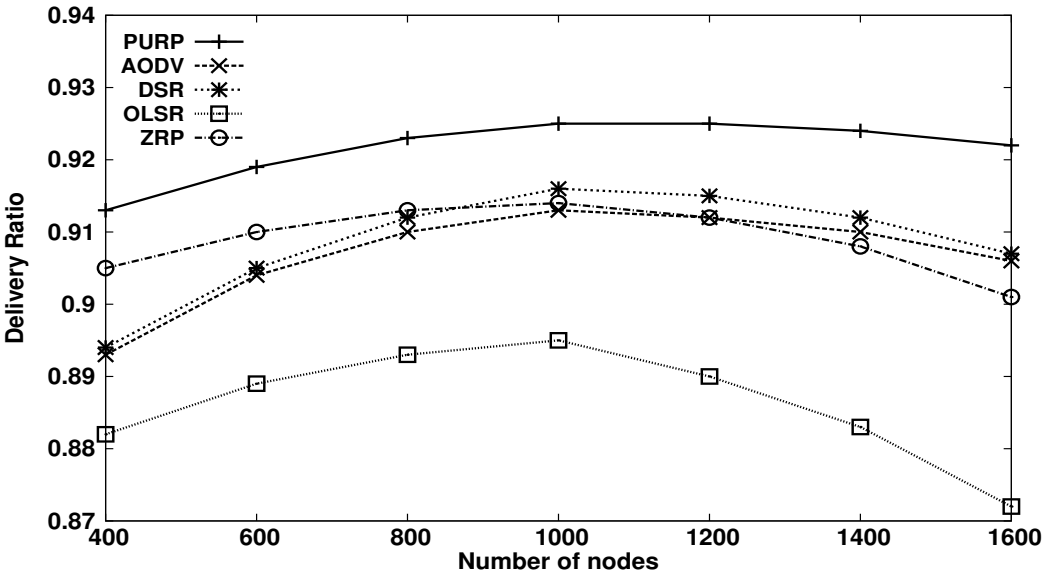


Figure 5.11: Delivery ratio vs. number of nodes.

proactive protocol, with a certain number of nodes when the routing information can be well maintained, the packet delivery can be performed with least time, hence OLSR gives the lowest end-to-end delay in that region. However, as the number of nodes increases, bandwidth usage for the control packet transmission increases dropping the data packet transmission opportunity, and thus the end-to-end delay for OLSR increases.

The switching between proactive and reactive modes results in a moderate control packet transmission as indicated in Fig. 5.13 where PURP transmits more (less) control packets than that of the reactive (proactive) routing protocols.

4) Network Traffic Load:

Experimental Scenario:

This experiment studies the effect of network traffic load on the performance. We consider 1000 nodes distributed within the defined area. The number of pack-

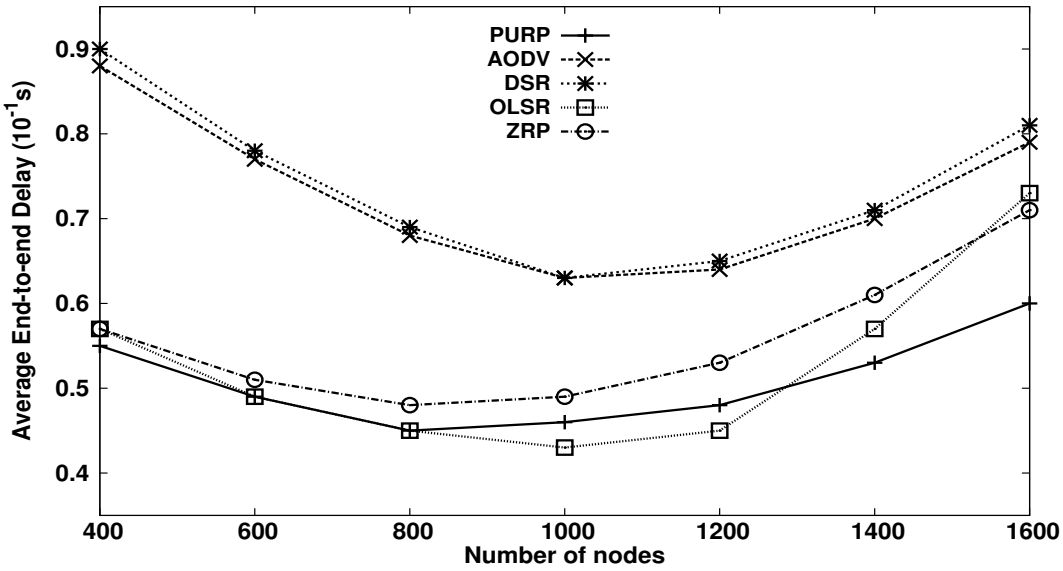


Figure 5.12: Average end-to-end delay vs. number of nodes.

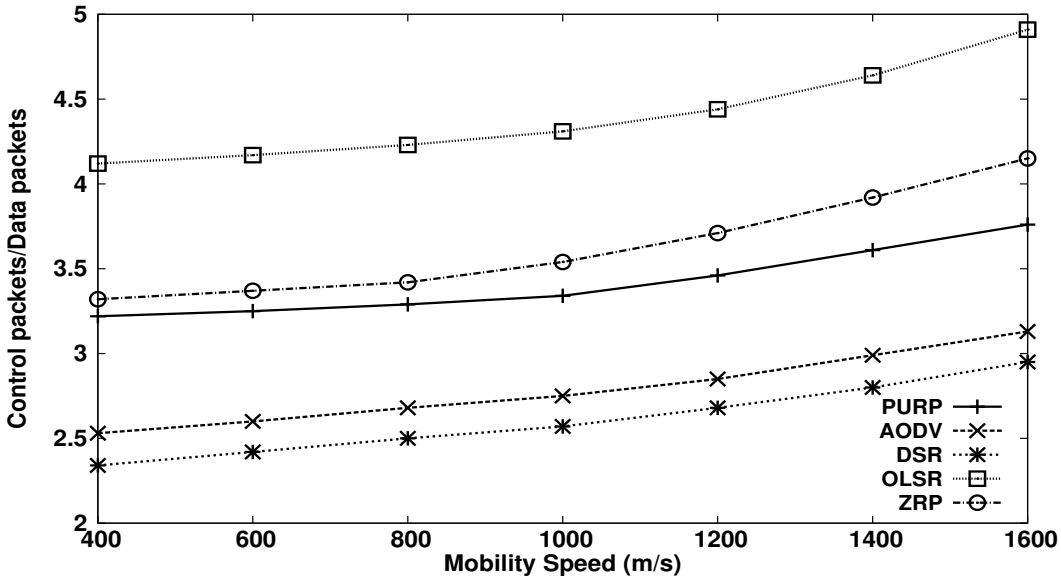


Figure 5.13: Control overhead vs. number of nodes.

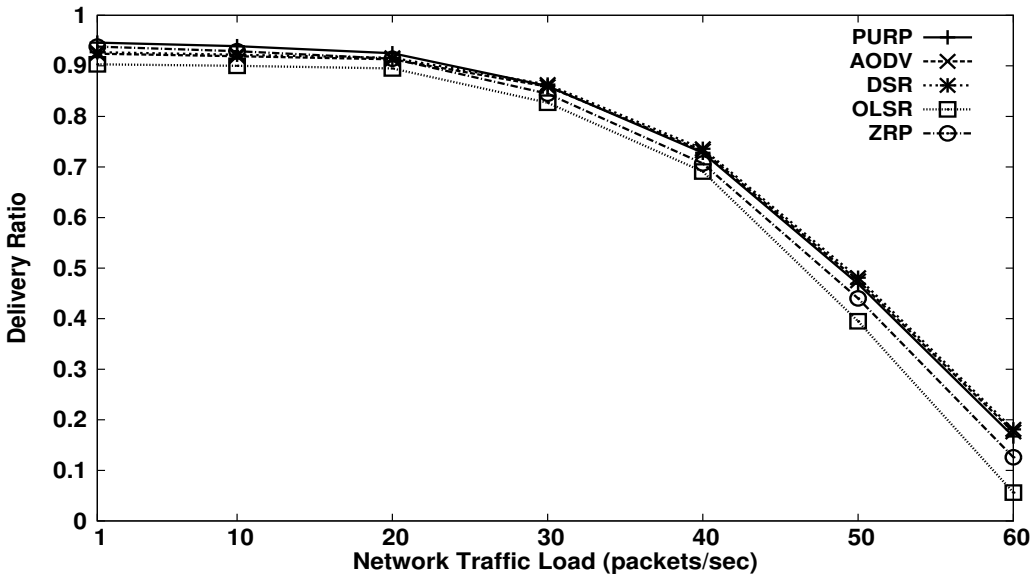


Figure 5.14: Delivery ratio vs. network traffic load.

ets sent ranges between 1 and 60 packets per second. Each node moves constantly with a predefined speed of 40km/h. The same setting of 100 source nodes applies in this scenario.

Fig. 5.14 shows the delivery ratio against traffic load and Fig. 5.15 shows the average end-to-end delay versus traffic load. As can be seen from the figures, all protocols perform similarly in this experiment. However, for comparison, PURP performs better than other protocols although slightly. This is because PURP is based on other protocols' implementations with slight modifications, hence PURP shares some performance limitation of those protocols. In this case, little performance gain is resulted from especially heavy network traffic load.

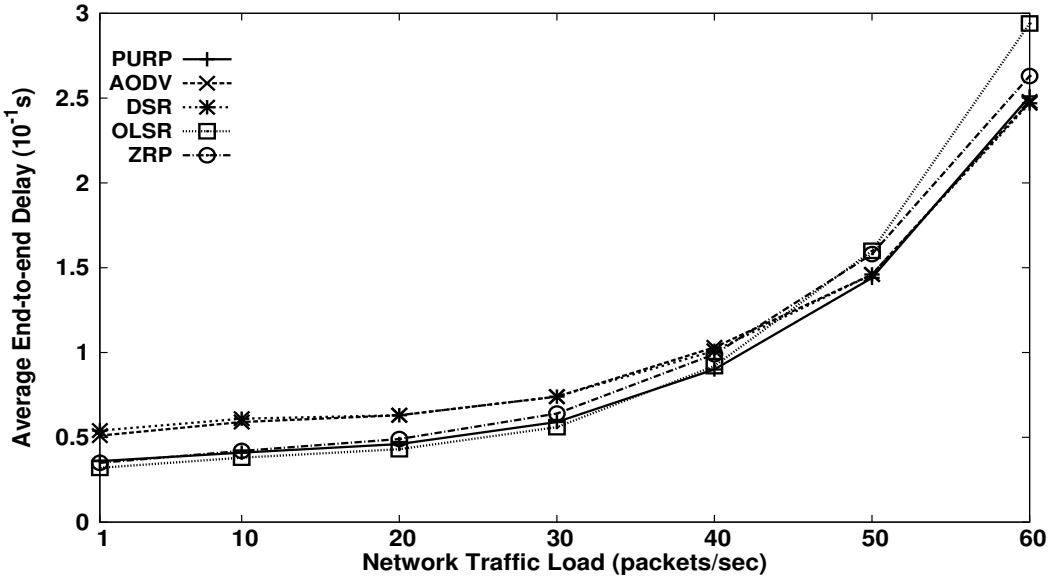


Figure 5.15: Average end-to-end delay vs. network traffic load.

5.4 Conclusion

We have presented a novel type of routing protocol design that is suitable for dynamic, multi-constrained Wireless Networks (mostly Ad hoc). In this design, we propose to empower the protocols with several carefully studied behavioral modes that enables the mobile nodes to choose the right mode of operation that suits the current environmental factors/situation that will help achieve the best performance with regards to specific metrics. This design results in what we have tagged polymorphic routing protocols, that is, protocols with multiple forms of operations that are dynamic, adaptive, and can react to multiple exterior or interior stimuli in order to do better its routing tasks.

We have discussed some design issues and presented some argumentations using two illustrative protocols. The OPHMR protocol that was designed earlier for Mobile Ad Hoc Networks (MANETs) and the PURP protocol designed for

Vehicular Ad Hoc Networks (VANETs).

The outcome of the simulation results of these protocols showed that both have outperformed their peers in most cases, thus confirming the effectiveness of the proposed design.

Chapter 6

Conclusion

In this thesis, we proposed several routing protocols for MANETs based on two approaches. We first adopted hybrid routing protocol design to propose two hybrid multicast routing protocols, namely ZMAODV and ZODMRP. Addressing the shortcoming of hybrid routing protocol design, we then proposed a novel design called the polymorphic routing protocol design concept. We demonstrated the polymorphic routing protocol design concept by proposing a number of polymorphic routing protocols, namely P_ZODMRP, OPHMR and PURP.

One main goal for designing a routing protocol for MANET is the data transmission performance. A routing protocol should at least be able to provide efficient routing paths to deliver data packets to destinations. The throughput performance and latency performance are two key measurements of data transmission performance. The throughput performance indicates the reliability and efficiency of the route selection and maintenance algorithm of a routing protocol. The latency performance reveals the efficiency of data transmission through selected routing paths of a routing protocol. In addition, a routing protocol may also con-

sider other performance achievements due to the characteristics of MANETs or the requirements of network applications. For example, power efficiency is a common consideration of MANET routing protocol design, and Quality of Service (QoS) support may also be required by some specific applications in the MANET environment.

Due to the characteristics of MANETs, different routing behavior can result in different performance in different network conditions. For example, proactive behavior usually yields better latency performance than reactive behavior, especially when the mobility is high which causes frequent topology changes. On the other hand, reactive behavior adopts a *lazy* approach to communication requirements where mobile nodes react only on-demand to data transmission requests and perform path finding operations only when needed. This operation of reactive behavior avoids the excessive bandwidth consumption of the control packets exchanging in proactive behavior so as to increase the throughput performance. Thus, a well designed routing protocol for MANET should balance between routing behaviors. Besides, if there are some other special requirements critical to the operation and performance of the MANET, the routing protocol should also be capable to take these requirements into account when designing its operational behaviors.

In our first contribution, we combined the zone concept from ZRP, which is a proactive behavior, with two well designed reactive routing protocols which are MAODV and ODMRP. The two new routing protocols are ZMAODV and ZODMRP. We evaluated their performance through simulations by comparing them with their predecessors. The results revealed that the new designed hybrid protocols outperform their predecessors. It means that the combination of

proactive and reactive behaviors can merge the benefits of the two behaviors, thus increases the performance of the newly designed protocols. The results also show that well designed proactive and reactive behaviors of a hybrid protocol can increase its performance both in throughput and latency aspects.

In our second contribution, we introduced our polymorphic routing protocol design concept. The main idea of the polymorphic concept is that a node can dynamically choose its routing behavior based on its environmental factors. The polymorphic routing protocols consist of a behavior selection algorithm based on environmental factors. To demonstrate our polymorphic routing protocol design concept, we proposed P_ZODMRP that uses a node's battery power, mobility speed and vicinity density level to determine the node's routing behavior. We examined the performance of P_ZODMRP through simulation by comparing it to other hybrid protocols. The results show that the dynamic behavior selection characteristic of polymorphic concept can effectively highlight the advantages of a routing behavior in various network environments, thus provides better performance than other hybrid protocols which merely combine routing behaviors. In addition, the polymorphic concept can easily accommodate other design requirements, for example, the power conservation in this case, into consideration.

Then we further modified P_ZODMRP by adding an optimized control packet forwarding mechanism into the proactive behavior of P_ZODMRP. The idea of optimized control packet forwarding mechanism is taken from OLSR which has shown to be effective in minimizing redundant control packet transmission. This new polymorphic routing protocol is named OPHMR. We have performed simulation to compare the performance of OPHMR with its predecessors. The results confirmed that OPHMR outperforms the other protocols.

Finally, we introduced the polymorphic concept into VANETs. We proposed a new polymorphic routing protocol, called PURP, for VANETs based on a new polymorphic behavior selection algorithm. This behavior selection algorithm considers the characteristics of VANETs and uses the traffic load, mobility speed and vicinity density level to decide its routing behavior. We evaluated the performance of PURP through simulation by comparing it to its peer protocols. The simulation evaluation again shows promising results for the polymorphic concept for VANETs.

6.1 Future Considerations

Polymorphic routing protocol design approach sets up a new trend in the research of routing protocol design. This thesis demonstrated three polymorphic routing protocols. However, the design approach is not limited to only described studies. Here, we list some potential issues for future polymorphic routing protocol design researches.

Firstly, different network conditions and requirements can be introduced into polymorphic concept. By examining the affect of different routing behaviors in different network conditions and requirements, polymorphic routing protocol can cover even larger range of routing protocol design area, and thus improve the performance of polymorphic routing protocol in many aspects.

Secondly, the routing behaviors of a polymorphic routing protocol can influence the performance of the protocol. Thus, selecting appropriately routing behaviors of polymorphic routing protocol, as well as fine-tuning the parameters of the selected routing behaviors are important issues in polymorphic routing proto-

col design.

Thirdly, based on the chosen scenarios and routing behaviors, the polymorphic algorithm design can significantly influence the performance of the routing protocol. To design a fine-tuned polymorphic algorithm with appropriate threshold values is an important research aspect in polymorphic routing protocol design area.

Bibliography

[1] I. . WG, "Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications: Higher-speed physical layer extension in the 2.4 GHz band," Sep. 1999.

[2] I. . W. Group, "Ieee 802.16 working group on broadband wireless access standards, http://grouper.ieee.org/groups/802/16/."

[3] "Internet engineering task force (IETF) mobile ad hoc networks (MANET) working group charter," http://www.ietf.org/html.charters/manet-charter.html.

[4] S. Corson and J. Macket, "Mobile ad hoc networking (MANET): Routing protocol performance issues and evaluation considerations," *RFC 2501*, Jan 1999.

[5] D. D. Perkins, H. D. Hughes, and C. B. Owen, "Factors affecting the performance of ad hoc networks," in *Proc. IEEE ICC02*, 2002.

[6] L. Chen and A. B. Mnaouer, "Performance evaluation of new hybrid multicast routing protocols for ad-hoc networks," in *Proc. 9th IEEE International Conference on Communications Systems*, 2004.

[7] V. Devarapalli, A. A. Selcuk, and D. Sidhu, "MZR: A multicast protocol for mobile ad hoc networks," in *Proc. ICC01*, June 2001.

[8] E. Royer and C. Perkins, "Multicast ad hoc on-demand distance vector (MAODV) routing," *IETF Internet Draft, draft-ietf-manet-maodv-00.txt*, 2000.

[9] Y. Yi, S.-J. Lee, W. Su, and M. Gerla, "On-demand multicast routing protocol (ODMRP) for ad hoc networks," *IETF Internet Draft, draft-ietf-manet-odmrp-04.txt*, 2002.

[10] A. B. Mnaouer, L. Chen, C. H. Foh, and J. W. Tantra, "A new polymorphic multicast routing protocol for MANET," in *Proc. IEEE ICC05*, 2005.

[11] ——, "The OPHMR: An optimized polymorphic hybrid multicast routing protocol for MANET," *IEEE Transactions of Mobile Computing*, vol. 6, no. 5, pp. 551–563, May 2007.

[12] P. Jacquet, P. Minet, A. Laouiti, L. Viennot, T. Clausen, and C. Adjih, "Multicast optimized link state routing," *IETF Internet Draft, draft-ietf-manet-olsr-molsr-01.txt*, 2001.

[13] L. Chen, A. B. Mnaouer, and C. H. Foh, "Polymorphic routing protocol design for dynamic multi-constrained wireless network," *submitted to IEEE Transactions on Vehicular Technology*.

[14] Z. J. Haas, M. R. Pearlman, and P. Samar, "The zone routing protocol (ZRP) for ad hoc networks," *IETF Internet Draft, draft-ietf-manet-zone-zrp-04.txt*, 2002.

[15] C. Perkins, E. Royer, and S. Das, "Ad hoc on-demand distance vector (aodv) routing," *RFC 3561*, 2003.

[16] L. Chen, C. H. Foh, and A. B. Mnaouer, "An optimized polymorphic hybrid multicast routing protocol (OPHMR) for ad hoc networks," in *Proc. IEEE ICC06, Wireless Ad Hoc and Sensor Network*, 2006.

[17] S. Murphy and J. J. Garcia-Luna-Aceves, "An efficient routing protocol for wireless networks," *ACM Journal on Mobile Networks and Application*, vol. 1, no. 2, pp. 183–197, 1996.

[18] C.-K. Toh, "Associativity-based routing for ad hoc mobile networks," *Wireless Personal Communications: An International Journal*, vol. 4, no. 2, pp. 103–139, 1997.

[19] V. D. Park and M. S. Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks," in *Proc. IEEE INFOCOM '97*, 1997.

[20] I. Chakeres and C. Perkins, "Dynamic MANET on-demand routing protocol (DYMO)," *IETF Internet Draft, draft-ietf-manet-dymo-04.txt*, 2006.

[21] S. Singh and C. S. Raghavendra, "PAMAS - power aware multi-access protocol with signalling for ad hoc networks," *ACM SIGCOMM Computer Communication Review*, vol. 28, no. 3, pp. 5–26, 1998.

[22] T. X. Brown, S. Doshi, and S. Bhandare, "The energy aware dynamic source routing protocol," *IETF Internet Draft, draft-brown-eadsr-00.txt*, 2003.

[23] J. Gomez, A. Z. Campbell, M. Naghshineh, and C. Bisdikian, "PARO: A power-aware routing optimization scheme for mobile ad hoc networks," *IETF Internet Draft, draft-gomez-paro-manet-00.txt*, 2001.

[24] D. Djenouri and N. Badache, "New power-aware routing protocol for mobile ad hoc networks," *The International Journal of Ad Hoc and Ubiquitous Computing (IJAHUC)*, vol. 1, no. 3, pp. 126–136, 2006.

[25] L. Barolli, A. Koyama, and N. Shiratori, "A QoS routing method for ad-hoc networks based on genetic algorithm," in *Proc. 14th International Workshop on Database and Expert Systems Applications*, 2003.

[26] C. Gomathy and S. Shanmugavel, "Supporting QoS in MANET by a fuzzy priority scheduler and performance analysis with multicast routing protocols," *EURASIP Journal on Wireless Communications and Networking*, vol. 5, no. 3, pp. 426–436, 2005.

[27] V. Kone and S. Nandi, "QoS constrained adapative routing protocol for mobile adhoc networks," in *Proc. 9th International Conference on Information Technology*, Dec. 2006.

[28] H. Liu and Y. Li, "A location based QoS routing protocol for ad hoc networks," in *Proc. 17th International Conference on Advanced Information Networking and Applications*, 2003.

[29] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing for mobile computers," in *Proc. ACM SIGCOMM*, Aug. 1994.

[30] J. Chroboczek, "The babel routing protocol, http://www.pps.jussieu.fr/˜jch/software/babel/."

[31] T. Clausen, P. Jacquet, A. Laouiti, P. Muhlethaler, A. Qayyum, and L. Viennot, "Optimized link state routing protocol," in *Proc. IEEE INMIC*, 2001.

[32] T. Clausen and P. Jacquet, "Optimized link state routing protocol (OLSR)," *RFC 3626*, 2003.

[33] T. Clausen, C. Dearlove, and P. Jacquet, "The optimized link state routing protocol version 2," *IETF Internet Draft, draft-ietf-manet-olsrv2-04.txt*, 2007.

[34] J. J. Garcia-Luna-Aceves, M. Spohn, and D. Beyer, "Source-tree routing in wireless networks," in *Proc. IEEE ICNP 99*, 1999.

[35] R. Ogier, F. Templin, and M. Lewis, "Topology dissemination based on reverse-path forwarding (TBRPF)," *RFC 3684*, 2004.

[36] Y.-Z. Lee, M. Gerla, J. Chen, J. Chen, B. Zhou, and A. Caruso, "Direction forward routing for highly mobile ad hov networks," *Ad Hoc and Sensor Wireless Networks*, vol. 2, no. 2, 2006.

[37] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark, "Scenario-based performance analysis of routing protocols for mobile ad-hoc networks," in *Proc. 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, 1999.

[38] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *Proc. the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networks*, 1998.

[39] B. Mans and N. Shrestha, "Performance evaluation of approximation algorithms for multipoint relay selection," in *Proc. the 3rd Annual Mediterranean Ad Hoc Networking Workshop*, 2004.

[40] M. Abolhasan, T. Wysocki, and J. Lipman, "Performance investigation on three-classes of MANET routing protocols," in *Proc. Asia-Pacific Conference on Communications*, 2005.

[41] T. Clausen, P. Jacquet, and L. Viennot, "Comparative study of CBR and TCP performance of MANET routing protocols," in *Workshop MESA*, 2002.

[42] P. Jacquet, A. Laouiti, P. Minet, and L. Viennot, "Performance analysis of OLSR multipoint relay flooding in two ad hoc wireless network models," *INRIA research report RR-4260*, 2001.

[43] D. Johnson, "Routing in ad hoc networks of mobile hosts," in *Proc. the IEEE Workshop on Mobile Computing Systems and Applications*, Dec. 1994.

[44] D. Johnson, Y.-C. Hu, and D. Maltz, "The dynamic source routing protocol for mobile ad hoc networks (dsr)," *RFC 4728*, 2007.

[45] Y.-C. Hu, D. Johnson, and D. Maltz, "Flow state in the dynamic source routing protocol," *IETF Internet-draft, draft-ietf-manet-dsrflow-00.txt*, 2001.

[46] C. Perkins and E. Royer, "Ad hoc on-demand distance vector routing," in *Proc. the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, Feb. 1999.

[47] M. Marina and S. Das, "On-demand multipath distance vector routing in ad hoc networks," in *Proc. IEEE ICNP'01*, 2001.

[48] S. Khurana, N. Gupta, and N. Aneja, "Reliable ad-hoc on-demand distance vector routing protocol," in *Proc. International Conference on Networks, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies*, 2006.

[49] S.-J. Lee, M. Gerla, and C.-C. Chiang, "On-demand multicast routing protocol," in *Proc. IEEE WCNC'99*, Sep. 1999.

[50] E. Yoneki and J. Bacon, "Content-based routing with on-demand multicast," in *Proc. the 24th International Conference on Distributed Computing Systems Workshops*, 2004.

[51] L. Klos and G. R. III, "Reliable group communication in an ad hoc network," in *Proc. Local Computer Networks*, 2002.

[52] N. Meghanathan, "Performance studies of MANET routing protocols in the presence of different broadcast routing discovery strategies," *Ubiquitous Computing and Communication Journal*, vol. 2, no. 4, Aug. 2007.

[53] S. R. Das, C. E. Perkins, and E. M. Royer, "Performance comparison of two on-demand routing protocols for ad hoc networks," in *Proc. IEEE INFOCOM00*, 2000.

[54] T. D. Dyer and R. V. Boppana, "A comparison of TCP performance over three routing protocols for mobile ad hoc networks," in *Proc. ACM MobiHoc'01)*, 2001.

[55] S. Giannoulis, C. Antonopoulos, E. Topalis, and S. Koubias, "ZRP versus DSR and TORA: A comprehensive survey on ZRP performance," in *Proc. IEEE Conference on Emerging Technologies and Factory Automation*, 2005.

[56] T. Kunz and E. Cheng, "Multicasting in ad-hoc networks: Comparing MAODV and ODMRP," in *Proc. Workshop on Ad Hoc Communications*, 2001.

[57] J. G. Jetcheva and D. B. Johnson, "A performance comparison of on-demand multicast routing protocols for ad hoc networks," *IEEE Personal Communications*, vol. 8, no. 1, pp. 16–28, 2002.

[58] S.-J. Lee and W. Su, "A performance comparison study of ad hoc wireless multicast protocols," in *Proc. IEEE INFOCOM00*, 2000.

[59] Z. J. Haas, "A new routing protocol for the reconfigurable wireless networks," in *Proc. 6th IEEE International Conference on Universal Personal Communications*, Oct. 1997.

[60] P. Samar, M. R. Pearlman, and Z. J. Haas, "Independent zone routing: an adaptive hybrid routing framework for ad hoc wireless networks," *IEEE/ACM Transactions on Networking*, vol. 12, no. 4, pp. 595–608, 2004.

[61] V. Ramasubramanian, Z. J. Haas, and E. G. Sirer, "SHARP: a hybrid adaptive routing protocol for mobile ad hoc networks," in *Proc. ACM MobiHoc'03*, 2003.

[62] B. An and S. Papavassiliou, "A mobility-based hybrid multicast routing in mobile ad-hoc wireless networks," in *Proc. IEEE MILCOM2001*, Oct. 2001.

[63] Z. J. Haas, M. R. Pearlman, and P. Samar, "The intrazone routing protocol (IARP) for ad hoc networks," *IETF Internet Draft, draft-ietf-manet-zone-iarp-02.txt*, 2002.

[64] ——, "The interzone routing protocol (IERP) for ad hoc networks," *IETF Internet Draft, draft-ietf-manet-zone-ierp-02.txt*, 2002.

[65] ——, "The bordercast resolution protocol (BRP) for ad hoc networks," *IETF Internet Draft, draft-ietf-manet-zone-brp-02.txt*, 2002.

[66] Z. J. Haas and marc R. Pearlman, "Evaluation of the ad-hoc connectivity with the zone routing protocols," *Wireless Personal Communications: Emerging Technologies for Enhanced Communication*, pp. 201–212, 1999.

[67] B. An and S. Papavassiliou, "A mobility-based clustering approach to support mobility manangement and multicast routing in mobile ad-hoc wireless networks," in *Proc. IEEE CISS2001*, 2001.

[68] K. Scott and N. Bambos, "Routing and channel assignment for low power transmission in PCS," in *Proc. IEEE ICUPC '96*, 1996.

[69] S. Singh, M. Woo, and C. S. Raghavendra, "Power-aware with routing in mobile ad hoc networks," in *Proc. IEEE MOBICOM*, 1998.

[70] A. Misra and S. Banerjee, "MRPC: Maximizing network lifetime for reliable routing inwireless environments," in *Proc. IEEE Wireless Communications and Networking Conference*, 2002.

[71] D. Kim, J. Garcia-Luna-Aceves, K. Obraczka, J.-C. Cano, and P. Manzoni, "Power-aware routing based on the energy drain rate for mobile ad hoc networks," in *Proc. 11th International Conference on Computer Communications and Networks*, 2002.

[72] S. Banerjee and A. Misra, "Minimum energy paths for reliable communication in multi-hop wireless networks," *CS-TR 4315, Technical Report, Department of Computer Science, University of Maryland, College Park*, December 2001.

[73] C.-K. Toh, G. Guichal, and S. Bunchua, "On-demand associativity-based multicast routing for ad hoc mobile networks (ABAM)," in *Proc. the 52nd IEEE VTS Vehicular Technology Conference*, 2000.

[74] J. G. Jetcheva and D. B. Johnson, "Adaptive demand-driven multicast routing in multi-hop wireless ad hoc networks," in *Proc. ACM MobiHoc'01*, 2001.

[75] S. K. Das, B. S. Manoj, and C. S. R. Murthy, "A dynamic core based multicast routing protocol for ad hoc wireless networks," in *Proc. ACM MobiHoc'02*, 2002.

[76] S. Park and D. Park, "Adaptive core multicast routing protocol," *ACM Wireless Networks*, vol. 10, no. 1, pp. 53–60, Jan. 2004.

[77] C.-C. Chiang, M. Gerla, and L. Zhang, "Forwarding group multicast protocol (FGMP) for multihop, mobile wireless networks," *Baltzer Cluster Computing*, vol. 1, no. 2, 1998.

[78] J. Garcia-Luna-Aceves and E. L. Madruga, "The core-assisted mesh protocol," *IEEE Journal on Selected Areas in Communication*, vol. 17, no. 8, Aug. 1999.

[79] "UCLA computer science department parallel computing laboratory and wireless adaptive mobility laboratory, GloMoSim: A scalable simulation environment for wireless and wired network systems." http://pcl.cs.ucla.edu/projects/glomosim/.

[80] C. Bettstetter, H. Hartenstein, and X. Perez-Costa, "Stochastic properties of the random waypoint mobility model," *ACM Wireless Networks*, vol. 10, no. 5, pp. 555–567, 2004.

[81] L. M. Feeney, "An energy consumption model for performance analysis of routing protocols for mobile ad hoc networks," *Mobile Networks and Applications*, vol. 6, no. 3, pp. 239–249, 2001.

[82] L. Kleinrock and J. A. Silvester, "Optimum transmission radio for packet radio networks or why six is a magic number," in *Proc. IEEE Nat. Telecommun. Conf.*, 1978.

[83] H. Takagi and L. Kleinrock, "Optimal transmission ranges for randomly distributed packet radio terminals," *IEEE Transactions of Communications*, vol. 32, pp. 246–257, 1984.

[84] T. Hou and V. Li, "Transmission range control in multihop packet radio networks," *IEEE Transactions of Communications*, vol. 34, pp. 38–44, 1986.

[85] ——, "Analyzing routing strategy NFP in multihop packet radio network on a line," *IEEE Transactions of Communications*, vol. 43, pp. 977–988, 1995.

[86] S.-J. Lee and M. Gerla, "AODV-BR: Backup routing in ad hoc networks," in *Proc. IEEE Wireless Communications and Networking Conference*, 2000.

[87] A. Belghith, A. B. Mnaouer, and H. Idoudi, "Polymorphic routing using proactive and probabilistic approaches for manets," in *Procs of the 1th Intl. Workshop on Future Trends on Design and Analysis of Dynamic Networks (FTDA-DN08), in conjunction with Qshine 2008*, July 31st 2008.

[88] A. Belghith, H. Idoudi, and M. Molnar, "Proactive probabilistic routing in mobile ad hoc networks," in *Accepted in the 2008 Intl. Conf. on Wireless Networks, ICWN08, July 14-17, Las Vegas, USA,*, 2008.

[89] C. Lochert, H. Hartenstein, J. Tian, H. Fubler, D. Herrmann, and M. Mauve, "A routing strategy for vehicular ad hoc networks in city environments," in *Proc. IEEE Intelligent Vehicles Symposium*, Columbus, Ohio, June 2003.

[90] I. Leontiadis and C. Mascolo, "GeOpps: Geographical opportunistic routing for vehicular networks," in *Proc. IEEE Workshop on Autonomic and Opportunistic Communications (Colocated With WOWMOM07)*, Helsinki, Finland, June 2007.

[91] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "MaxProp: Routing for vehicle-based disruption-tolerant networks," in *Proc. IEEE INFOCOM06*, Barcelona, Spain, April 2006.

[92] M. Kihl, M. Sichitiu, T. Ekeroth, and M. Rozenberg, "Reliable geographical multicast routing in vehicular ad-hoc networks," in *Proc. 5th International Conference on Wired/Wireless Internet Communications*, May 2007.

[93] W. Sun, H. Yamaguchi, K. Yukimasa, and S. Kusumoto, "Gvgrid: A QoS routing protocol for vehicular ad hoc networks," in *Proc. 14th IEEE Workshop on Quality of Service*, June 2006.

[94] V. Naumov and T. R. Gross, "Connectivity-aware routing (CAR) in vehicular ad-hoc networks," in *Proc. IEEE INFOCOM'07*, Anchorage, Alaska, May 2007.

[95] 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," in *Proc. IEEE INFOCOM03*, 2003.

[96] N. Sadagopan, F. Bai, B. Krishnamachari, and A. helmy, "PATHS: analysis of path duration statistics and their impact on reactive manet routing protocols," in *Proc. 4th ACM international symposium on Mobile ad hoc networking and computing*, 2003.

[97] F. Bai, N. Sadagopan, and A. helmy, "User manual for IMPORTANT mobility tool generators in ns-2 simulator, http://nile.usc.edu/important/mobility-user-manual.pdf."