

**ON-DEMAND MULTICAST  
ROUTING IN MOBILE AD HOC  
NETWORKS**

by

**Edward Cheng**

A thesis submitted to the Faculty of Graduate Studies in partial fulfillment of the  
requirements for the degree of

**Master of Engineering**

Ottawa-Carleton Institute of Electrical and Computer Engineering

Department of Systems and Computer Engineering

**Carleton University**

Ottawa, Ontario

CANADA, K1S 5B6

January, 2001.

© Copyright 2001, Edward Cheng

The undersigned recommend to the Faculty of Graduate Studies and Research  
the acceptance of the thesis

**ON-DEMAND MULTICAST  
ROUTING IN MOBILE AD HOC  
NETWORKS**

Submitted by Edward Cheng, M. Eng in partial fulfillment of the requirements  
for the degree of Master of Engineering

---

**Thesis Supervisor**

---

**Chair, Department of Systems and Computer Engineering**

## **ABSTRACT**

The advent of ubiquitous computing and the proliferation of portable computing devices have raised the importance of mobile and wireless networking. A major challenge lies in adapting multicast communication to environments where mobility is unlimited and outages/failures are frequent. This thesis investigates the performance of two prominent on demand multicast routing protocols aimed specifically at fully Mobile Ad Hoc Networks (MANET) – Multicast Ad hoc On-Demand Vector protocol (MAODV) and On-Demand Multicast Routing Protocol (ODMRP). We demonstrate that even though MAODV and ODMRP share similar on-demand behaviour, the differences in protocol mechanics can lead to performance differentials. Based on the observations, we make recommendations about how the performance of either protocol as well as future implementations can be improved.

## **ACKNOWLEDGMENTS**

I wish to extend my thanks to my supervisor, Professor T. Kunz, for his guidance and direction for this thesis. I would also like to acknowledge my family who have supported and challenged me to do well. Finally, my acknowledgements would not be complete without mention of my fiancée, Nathalie who has been a constant source of encouragement throughout my research process.

## TABLE OF CONTENTS

<b>1</b>	<b>INTRODUCTION.....</b>	<b>1</b>
1.1	MOTIVATION .....	3
1.2	RESEARCH OVERVIEW AND CONTRIBUTIONS .....	4
1.3	ORGANIZATION OF THESIS .....	5
<b>2</b>	<b>RELATED WORK .....</b>	<b>7</b>
2.1	MULTICASTING IN FIXED / WIRED NETWORKS .....	7
2.2	MULTICASTING IN FIXED INFRASTRUCTURE CELLULAR NETWORK....	9
2.3	MULTICASTING IN MOBILE AD HOC NETWORKS .....	13
2.3.1	<i>Adapting Traditional Wireline Multicast Protocols for MANET</i>	14
2.3.2	<i>MANET inspired Multicast Protocols .....</i>	16
<b>3</b>	<b>ON-DEMAND MULTICAST AD HOC ROUTING PROTOCOLS</b>	
	<b>19</b>	
3.1	MULTICAST AD HOC ON-DEMAND DISTANCE VECTOR PROTOCOL	
	(MAODV).....	19
3.1.1	<i>Multicast Route Discovery.....</i>	19
3.1.2	<i>Reverse Path Setup .....</i>	21
3.1.3	<i>Forward Path Setup.....</i>	22

3.1.4	<i>Multicast Route Activation</i> .....	23
3.1.5	<i>Group Hello Messages</i> .....	24
3.1.6	<i>Multicast Tree Maintenance</i> .....	25
3.1.7	<i>MAODV Multicast Parameters</i> .....	27
3.2	ON-DEMAND MULTICAST ROUTING PROTOCOL (ODMRP).....	27
3.2.1	<i>Mesh Creation</i> .....	28
3.2.2	<i>Routing Table</i> .....	31
3.2.3	<i>Forwarding Group Table</i> .....	31
3.2.4	<i>Data Forwarding</i> .....	31
3.2.5	<i>Soft State</i> .....	32
3.2.6	<i>Mobility Prediction</i> .....	32
3.3	QUALITATIVE SIDE BY SIDE PROTOCOL COMPARISON.....	33
3.4	CRITIQUE OF MAODV AND ODMRP .....	34
<b>4</b>	<b>SIMULATION</b> .....	<b>37</b>
4.1	SIMULATION ENVIRONMENT.....	37
4.1.1	<i>Random Way-point Mobility model</i> .....	38
4.1.2	<i>Traffic Pattern</i> .....	38
4.2	PERFORMANCE EVALUATION METRICS .....	39
4.2.1	<i>Packet Delivery Ratio</i> .....	39
4.2.2	<i>Number of data packets transmitted per data packet delivered</i> ..	40

4.2.3	<i>Number of control packets transmitted per data packet delivered</i>	40
4.2.4	<i>Number of control packets and data packets transmitted per data packet delivered</i>	40
4.3	NETWORK PARAMETERS	40
4.3.1	<i>Number of senders</i>	40
4.3.2	<i>Mobility</i>	41
4.3.3	<i>Multicast group size</i>	41
4.3.4	<i>Traffic Patterns</i>	41
4.4	VALIDATION OF MAODV AND ODMRP PROTOCOL IMPLEMENTATIONS	42
4.5	MULTICAST VERSUS UNICAST	45
<b>5</b>	<b>SIMULATION RESULTS</b>	<b>49</b>
5.1	MAODV ROUTE ACTIVATION	49
5.2	SENDERS	51
5.3	MOBILITY	55
5.4	MULTICAST GROUP	59
5.5	TRAFFIC SOURCE	63
5.6	PROTOCOL ANALYSIS	71
5.7	PROTOCOL IMPROVEMENTS	72

<b>6</b>	<b>CONCLUSIONS AND FUTURE WORK .....</b>	<b>75</b>
----------	------------------------------------------	-----------



## LIST OF FIGURES

<i>Number</i>	<i>Page</i>
FIGURE 1 MAODV MULTICAST ROUTE DISCOVERY .....	19
FIGURE 2 MAODV REVERSE PATH SETUP.....	21
FIGURE 3 MAODV FORWARD PATH SETUP.....	22
FIGURE 4 MAODV ROUTE ACTIVATION.....	23
FIGURE 5 MAODV GROUP LEADER HELLO MESSAGES .....	24
FIGURE 6 ODMRP MESH CREATION USING BROADCAST JOIN-QUERY .....	28
FIGURE 7 ODMRP JOIN-QUERY PROCESSING.....	29
FIGURE 8 ODMRP JOIN-REPLY PROCESS.....	30
FIGURE 9 AODV PACKET DELIVERY RATIO .....	43
FIGURE 10 ODMRP PACKET DELIVERY RATIO .....	45
FIGURE 11 PACKET TRANSMISSION RATIO AS A FUNCTION OF THE NUMBER OF SENDERS (UNICAST VS MULTICAST) .....	47
FIGURE 12 PACKET DELIVERY RATIO AS A FUNCTION OF THE NUMBER OF SENDERS (UNICAST VS MULTICAST) .....	48
FIGURE 13 PACKET DELIVERY RATIO AS A FUNCTION OF MOBILITY WITH DIFFERENT WAIT TIMES.....	50
FIGURE 14 PACKET DELIVERY RATIO AS A FUNCTION OF THE NUMBER OF SENDERS .....	52

FIGURE 15 PACKET TRANSMISSION RATIO AS A FUNCTION OF THE NUMBER OF SENDERS .....	53
FIGURE 16 CONTROL OVERHEAD PER DATA PACKET DELIVERED AS A FUNCTION OF THE NUMBER OF SENDERS .....	54
FIGURE 17 CONTROL AND DATA TRANSMISSIONS PER DATA PACKET DELIVERED AS A FUNCTION OF THE NUMBER OF SENDERS .....	55
FIGURE 18 PACKET DELIVERY RATIO AS A FUNCTION OF MOBILITY .....	56
FIGURE 19 PACKET TRANSMISSION RATIO AS A FUNCTION OF MOBILITY .....	57
FIGURE 20 CONTROL OVERHEAD PER DATA PACKET DELIVERED AS A FUNCTION OF MOBILITY.....	58
FIGURE 21 CONTROL AND DATA TRANSMISSIONS PER DATA PACKET DELIVERED AS A FUNCTION OF MOBILITY .....	59
FIGURE 22 DATA DELIVERY RATIO AS A FUNCTION OF MULTICAST GROUP SIZE .....	60
FIGURE 23 PACKET TRANSMISSION RATIO AS A FUNCTION OF MULTICAST GROUP SIZE .....	61
FIGURE 24 CONTROL OVERHEAD PER DATA PACKET DELIVERED AS A FUNCTION OF MULTICAST GROUP SIZE .....	62
FIGURE 25 CONTROL AND DATA TRANSMISSIONS PER DATA PACKET DELIVERED AS A FUNCTION OF MULTICAST GROUP SIZE.....	63

FIGURE 26 MAODV DATA DELIVERY RATIO AS A FUNCTION OF MULTICAST GROUP SIZE WITH DIFFERENT TRAFFIC SOURCES .....	64
FIGURE 27 ODMRP DATA DELIVERY RATIO AS A FUNCTION OF MULTICAST GROUP SIZE WITH DIFFERENT TRAFFIC SOURCES .....	65
FIGURE 28 MAODV PACKET TRANSMISSION RATIO AS A FUNCTION OF MULTICAST GROUP SIZE WITH DIFFERENT TRAFFIC SOURCES .....	66
FIGURE 29 ODMRP PACKET TRANSMISSION RATIO AS A FUNCTION OF MULTICAST GROUP SIZE WITH DIFFERENT TRAFFIC SOURCES .....	67
FIGURE 30 MAODV CONTROL AND DATA TRANSMISSIONS PER DATA PACKET DELIVERED AS A FUNCTION OF MULTICAST GROUP SIZE WITH DIFFERENT TRAFFIC SOURCES .....	68
FIGURE 31 ODMRP CONTROL AND DATA TRANSMISSIONS PER DATA PACKET DELIVERED AS A FUNCTION OF MULTICAST GROUP SIZE WITH DIFFERENT TRAFFIC SOURCES .....	69
FIGURE 32 MAODV CONTROL OVERHEAD PER DATA PACKET DELIVERED AS A FUNCTION OF MULTICAST GROUP SIZE WITH DIFFERENT TRAFFIC SOURCES.....	70
FIGURE 33 ODMRP CONTROL OVERHEAD PER DATA PACKET DELIVERED AS A FUNCTION OF MULTICAST GROUP SIZE WITH DIFFERENT TRAFFIC SOURCES.....	71
FIGURE 34 SELF-PRUNING FOR FLOODED PACKETS.....	73

FIGURE 35 NS-2 MOBILE NODE NETWORK COMPONENTS.....	88
----------------------------------------------------	----

## LIST OF TABLES

<i>Number</i>	<i>Page</i>
TABLE 1 QUALITATIVE CHARACTERISTICS OF MAODV VERSUS ODMRP .....	33
TABLE 2 SUMMARY OF SIMULATION PARAMETERS .....	39

## LIST OF ACRONYMS

### *Acronym*

AMRIS – Ad hoc Multicast Routing protocol utilizing Increasing Id numbers

AMROUTE – Ad hoc Multicast Routing

CAMP – Core Assisted Mesh Protocol

CBT – Core Based Tree

CBR – Constant Bit Rate

DVMRP – Distance Vector Multicast Routing Protocol

EXP – Exponential Distribution

GPS – Global Positioning System

GRPH – Group Hello

IETF – Internet Engineering Task Force

IP – Internet Protocol

Ipv6 – Internet Protocol Version 6

MACT – Multicast Route Activation

MANET – Mobile Ad Hoc Network

MAODV – Multicast Ad hoc On-Demand Vector routing protocol

MoM – Mobile Multicast

NSMP – Neighbour Supporting Multicast Protocol

ODMRP – On-Demand Multicast Routing Protocol

PIM-DM – Protocol Independent Multicast Dense Mode

PIM-SM – Protocol Independent Multicast Sparse Mode

RPF – Reverse Path Forwarding

RREQ – Route Request

RREP – Route Reply

ST-WIM – Shared Tree Wireless Multicast

TTL – Time To Live

## *Chapter 1*

### **1 Introduction**

A mobile ad hoc network (MANET) consists of a collection of dynamic nodes with sometimes rapidly changing multi-hop topologies that are composed of relatively bandwidth constrained wireless links. In MANETs, there is no assumption of an underlying fixed infrastructure. Nodes are free to move around arbitrarily. Each mobile node functions as a router to establish connections between any two nodes. Since each node has a limited transmission range, not all messages may reach all the intended hosts. To provide communication through the whole network, a source to destination path could be relayed through several intermediate neighbouring nodes.

Unlike typical wireline routing protocols, ad hoc routing protocols must address a diverse range of issues [CM99]. For instance, the network topology can change randomly and rapidly at unpredictable times. As well, since wireless links generally have lower capacity, congestion is typically the norm rather than the exception. The majority of nodes will rely on some exhaustible means for energy (i.e. batteries), thus routing protocols must limit the amount of control information that is passed between nodes. In summary, an ad hoc network routing protocol must be simple, robust, and minimize control message exchanges.



The goal of MANET is to extend mobility into the realm of autonomous, mobile, wireless domains, where a set of nodes form the network routing infrastructure in an ad hoc fashion. The majority of applications for the MANET technology are in areas where rapid deployment and dynamic reconfiguration is necessary and the wireline network is not available [CM99]. These include military battlefields, emergency search and rescue sites, classrooms, and conventions where participants share information dynamically using their mobile devices.

Alongside the growth in wireless applications, there has been a tremendous growth in the demand for group-oriented computing. There are more and more applications where one-to-many dissemination is necessary. The multicast service is critical in applications characterized by the close collaboration of teams (e.g. rescue patrol, battalion, scientists) with audio and video conferencing requirements and sharing of text and images. In general, wireless mobile multicasting poses several key challenges [Chi98]. Multicast sources move, making source oriented multicast protocols inefficient. Multicast group members move, thus precluding the use of a fixed multicast topology. Transient loops may form during tree reconfiguration. As well, tree reconfiguration schemes should be simple to keep channel overhead low. In multicasting, the key problem is to enable efficient routing of packets from a sender to multiple receivers. Now, coupled with the MANET characteristics described above, one can appreciate

that providing a suitable multicast service within an ad hoc network becomes extremely challenging.

## **1.1 Motivation**

The use of multicasting within a network has many benefits. Multicasting reduces the communication costs for applications that send the same data to multiple recipients. Instead of sending via multiple unicasts, multicasting minimizes the link bandwidth consumption, sender and router processing, and delivery delay [Pau98]. In addition, multicasting provides a simple yet robust communication mechanism whereby a receiver's individual address is unknown or changeable without any knowledge by the source. Within a wireless medium it is even more crucial to reduce the transmission overhead and power consumption. Multicasting can improve the utilization of the wireless link when sending multiple copies of messages by exploiting the inherent broadcast property of wireless transmission.

Current mobile IP routing protocols [Per96] developed for cellular networks cannot be applied to MANETs since there is no fixed home agent to serve as a routing reference. MANETs are inherently instant infrastructure multi hop networks. Traditional wire-line multicast protocols like Protocol Independent Multicast – Sparse Mode (PIM-SM) [DEF96], Core Based Tree (CBT) [BFC93], and Distance Vector Multicast Routing Protocol (DVMRP) [WPD88] do not work well in a MANET. In a wireless environment, since the nodes are mobile,

the frequent tree reorganization from these traditional multicast protocols can cause significant signalling overhead and frequent loss of datagrams. Tree reorganization is more frequent in MANET versus conventional static networks because the multicast protocols will need to deal with network changes as well as membership group changes.

## **1.2 Research Overview and Contributions**

This thesis concentrates on the current on-demand multicast routing protocols for ad hoc networks. The reason for this is that previous research [CGZ97, CG97, CGZ98, CGZ99] show that traditional wireline multicast protocols which have been modified for the wireless environment are not suitable in terms of scalability and performance with increased mobility. On-demand protocols do not need to keep an entry for each destination router in the routing table and maintain the information by periodic updates of the routing table. The overhead of storage and channel utilization limits the scalability of MANETs where each mobile node is a router. By maintaining only the active entries on an on-demand basis one can reduce overhead, thus improving performance and scalability. On-demand multicast is well suited to operate in an on-demand routing environment where routes are selectively computed as needed between communicating nodes instead of being maintained and updated globally by a routing infrastructure [CG98]. On-demand multicast is particularly attractive in mobile, rapidly

changing networks, where traffic overhead caused by routing updates and tree reconfigurations may become prohibitive beyond a critical speed.

There are currently two popular on-demand multicast routing protocols proposed for use in MANET. Namely, “Multicast Ad-Hoc On-Demand Vector” routing protocol (MAODV) [RP00,RP99], and “On-Demand Multicast Routing Protocol” (ODMRP) [GLS00,LGC99]. Up till now there has not been a side-to-side comparison of these protocols. This thesis provides the following contributions:

1. Support for developing Multicast Ad Hoc routing protocols within a standard network simulator, Ns-2. (Please see Appendices for more information)
2. An implementation of the MAODV and ODMRP protocols in NS-2.
3. A comprehensive evaluation of these two approaches in terms of a specific set of performance metrics.
4. Insight into where areas of improvement can be made in the area of on-demand multicast routing for MANET.

### **1.3 Organization of Thesis**

This thesis focuses on the on-demand multicast protocol design and performance evaluation for ad hoc networks.

**Chapter 2** walks the reader through multicasting in fixed networks (Internet), then through cellular networks and finally within the context of ad hoc networks.

**Chapter 3** presents the two on-demand multicast protocols ODMRP, and MAODV in detail, explaining detailed protocol operation and behaviour as well as a qualitative comparison of the merits of both protocols.

**Chapter 4** discusses the simulation environment and experimental parameters. Here we describe how the two protocols were validated, compared and evaluated.

**Chapter 5** provides a discussion on the simulation results and presents protocol enhancements.

**Chapter 6** contains the conclusions and directions for future research.

## *Chapter 2*

### **2 Related Work**

#### **2.1 Multicasting in Fixed / Wired Networks**

On the Internet, there are two popular wired network multicast schemes, namely, per-source shortest tree and shared tree.

The per-source tree scheme consists of broadcasting the packet from the source to all destinations along the source tree in a manner that avoids loops. This is accomplished by using “Reverse Path Forwarding” or RPF. In RPF, a router forwards a broadcast packet originated at a source out its other interfaces if and only if the packet is received on an interface that is on the shortest path from the router to the source. Thus, only those packets are forwarded that arrive on the reverse shortest path from the router to the sender. Examples of per-source tree commonly used in the Internet are DVMRP and PIM Dense Mode [DEF99]. In the wireline environment, per-source tree multicasting has many attractive properties. For example, the shortest tree from each source to all destinations is inherent in the routing protocol. Furthermore, source tree multicast distributes the traffic evenly in the network (assuming that the source and receivers are evenly distributed). As well, it does not rely on a control point (rendezvous point). In mobile networks, however, the per-source tree approach for multicasting presents a problem. Suppose a source moves faster than the routing

tables can track it. In this case, some of the nodes will have obsolete routing tables pointing in the “wrong direction”. Following the “reverse path forwarding” principle, multicast packets are dropped at such nodes, and may never reach some of the receivers. One way to alleviate this problem is to increase the routing update rate with mobility. However, the periodical full broadcast in implementations like DVMRP introduces costly control overhead on the low bandwidth wireless channel and is not suitable for sparse distributed membership and scaling the network size.

In the shared tree multicast scheme, each multicast group has a single tree rooted at a special router called the Rendezvous Point (RP). Each multicast group has its own RP, and “grows” its own shared tree. The intermediate routers in the tree are responsible for forwarding the multicast data to members. In this manner, all receivers join the multicast group by explicitly sending a JOIN message towards the RP. Senders send data to the RP, and the RP uses a single unidirectional shared tree to distribute the data to the receivers. Examples of shared-tree approaches are CBT and PIM Sparse Mode. The shared tree is less sensitive to source mobility and can in part overcome the fast moving source problem. Basically, a fast source will send its packet to the RP in unicast mode. Packets are correctly delivered to the RP on shortest paths, irrespective of the speed of the source. The RP will then multicast the packet on the shared tree to the intended destinations. This works as long as the shared tree is stable and the RP itself is

not fast moving. If ALL the nodes are moving fast (relative to the routing table updates), the shared tree solution fails. The shared tree also has some drawbacks with respect to the per-source scheme. First, traffic is concentrated on the backbone, rather than evenly distributed across the network and paths are often non optimal. This leads to lower throughput efficiency. Secondly, as the entire network moves and the membership changes dynamically, the RP may not be in the center aggravating the non-optimality of the paths.

## **2.2 Multicasting in Fixed Infrastructure Cellular Network**

Mobile networks with fixed infrastructure, or cellular networks, consist of stationary base stations and mobile endpoints. Each base stations is assigned a geographic area, or cell, and is responsible for connecting mobile endpoints to the wired portion of the network. Mobile users communicate via a single hop wireless channel with a base station, which is in turn connected to a wired backbone.

Mobile IP (Ipv4 and Ipv6) [Per 96, JP96] is the basic mechanism currently used to manage mobility to these end hosts. In Mobile IP, a mobile node may change its location without changing its IP address. The way this is achieved is through the use of a home agent and a foreign agent. A home agent represents a router on the mobile node's home network that is responsible for tunnelling datagrams for delivery to the mobile node when it is away from home. A foreign agent represents a router on a mobile node's visited network that provides routing



services to the mobile node while it is registered. The foreign agent detunnels and delivers datagrams to the mobile node that were tunnelled by the mobile node's home agent. While the mobile node is visiting a foreign network it is assigned a care-of address that represents the mobile node's current point of attachment. This care-of address is then registered with the home agent to allow the home agent to know where to tunnel datagrams to the mobile node. In the reverse direction, for datagrams sent by the mobile node, standard IP routing is used to deliver the datagrams to their respective destinations; it is not necessary to pass them through the home agent.

There are currently two basic approaches for supporting multicast service to mobile hosts in a fixed infrastructure cellular network by extending Mobile IP [Per96, XP97], foreign agent-based multicast (referred to as remote subscription) and home agent-based multicast (referred to as bi-directional tunnelling).

In foreign agent-based multicast, a mobile host has to subscribe to multicast groups whenever it moves to a foreign network. It is a very simple scheme and does not require any encapsulations (datagram tunnelling). This scheme has the advantage of offering an optimal routing path and non-existence of duplicate copies of datagrams. However, when a mobile host is highly mobile, its multicast service may be very expensive because of the difficulty in managing the multicast tree. Furthermore, the extra delay incurred when rebuilding a multicast tree can create the possibility of a disruption in multicast data delivery.

In home agent-based multicast, data delivery is achieved by unicast Mobile IP tunnelling via a home agent. When a home agent receives a multicast datagram destined for a mobile host, it encapsulates the datagram twice (with the mobile host address and the care-of-address of the mobile host) and then transmits the datagram to the mobile host as a unicast datagram. Consequently, if multiple mobile hosts that belong to the same home network visit the same foreign network, duplicate copies of multicast datagrams will arrive at the foreign network.

[BHW97] proposed a home agent-based multicast protocol called MoM (Mobile Multicast), where a home agent is responsible for tunnelling multicast datagrams to the mobile host. However, the home agent forwards only one copy of the multicast datagram to each foreign network that contains its mobile hosts. Upon receiving the multicast datagram, a foreign agent delivers it to mobile hosts using link level multicasting. The MoM protocol reduces multicast traffic by decreasing the number of duplicate copies of datagrams.

[SSK00] proposed a protocol using a multicast agent in wireless mobile networks. In this protocol, a mobile host receives a tunnelled multicast datagram from a multicast agent located in a network close to it or directly from the multicast router in the current network, which offers sub optimal multicast delivery to the mobile host. While receiving the tunnelled multicast datagram from a remote multicast agent, the local multicast agent starts the multicast join process, which

makes the multicast delivery route optimal. The protocol reduces network traffic load by decreasing the number of duplicate copies of datagrams and reduces the multicast data delivery path length since multicast agents that are located close to the current location of mobile hosts or located in the current network forward datagrams to mobile hosts.

[BR00] investigated four possible approaches to support PIM-DM multicast for multicast to mobile IPv6 hosts. The first approach, the local group membership on foreign link, is the simplest solution, and does not require any special encapsulation or decapsulation mechanisms. Moreover, routing of multicast packets is optimal. It is a good solution if processing resources on home agents and mobile hosts are very low. It is not a good solution for highly mobile hosts, both receivers and senders. Mobile receivers must re-subscribe to the multicast group after each movement to a new link. Each time they change a link, they experience quite a long join delay, and thus datagrams will be lost.

A bi-directional tunnel is interesting for highly mobile hosts, since no significant join and leave delay occurs. However, more processing and storage resources must be available in home agents and mobile hosts and the routing is sub optimal.

A uni-directional tunnel from the mobile host to the home agent is a combination of the approaches mentioned above. It preserves network and system resources

better than a bi-directional tunnel, routing to mobile receivers is optimal, and there is no additional bandwidth consumption due to mobile senders.

The last approach, the unidirectional tunnel from the home agent to the mobile host, seems to combine most disadvantages of the other approaches if the mobile host is both the sender and receiver for a multicast group.

However, all these schemes assume that the mobile host is the last hop in an infrastructure based cellular network. It is not able to handle truly ad hoc networks where intermediate nodes are mobile as well.

### **2.3 Multicasting in Mobile Ad hoc Networks**

In mobile ad hoc networks, there are three basic categories for multicast algorithms. The pro-active approach pre-computes paths to all possible destinations and stores this information in routing tables. To maintain an up-to-date database, routing information is periodically distributed throughout the network. The other approach is to create paths to other hosts on-demand. The idea is based on a query-response mechanism or reactive multicast. In the query phase, a node explores the environment. Once the query reaches the destination, the response phase is entered and establishes the path. The final approach is to simply flood the network. Every node receiving a message floods it to a list of neighbours. Flooding a network acts like a chain reaction that can result in exponential growth.

### 2.3.1 *Adapting Traditional Wireline Multicast Protocols for MANET*

[CGZ99] proposed wireless extensions to DVMRP, whereby each sender selectively “floods” multicast packets to all nodes within a specified range using RPF. However, this approach suffers from the periodic data flooding overhead incurred by the source in order to re-establish any new or lost connections. This periodic flooding causes considerable transmission overhead for the low bandwidth wireless channel. As well, with the RPF mechanism, if the shortest path changes and no multicast packets arrive on the new shortest path, the node becomes disconnected from the tree. Finally scalability to a large number of senders becomes problematic since each tree internal node stores the list of sources and associated timers. Storage and processing overhead grows linearly with the number of sources. The shared tree eliminates this problem.

[CGZ97] proposed a Shared Tree Wireless Network Multicast (ST-WIM) algorithm based on adapting PIM-SM to MANET. Several simulations were performed using the ST-WIM protocol measuring metrics like join latency, control packet overhead, throughput when varying multicast group size, and node mobility. ST-WIM’s results show that the performance of both hard and soft state multicast tree maintenance mechanisms degrade rapidly with increased mobility past 10m/s and increased number of mobile nodes.

[CG97] proposed a modified version of the CBT multicast algorithm. Each multicast group has a unique multicast identifier. Each multicast address

identifies a host group, the group of hosts that should receive a packet sent to that address. Each multicast group is initialized and maintained by a multicast server who becomes the core of the CBT for this multicast group. Initially the multicast server broadcasts the multicast identifier and its own node identifier using a flooding algorithm. When a node receives this information, it will use this when it needs to join or quit the multicast group. Simulations were performed to evaluate performance based on several criteria like control packet overhead, robustness to mobility, scaling properties with respect to multicast group membership, and response time to joining a group. Their simulation results show a rapid decrease in throughput, and increase in control packet overhead with increased mobility of the nodes.

[CGZ98] proposed an Adaptive Shared Tree multicast which attempted to reduce path costs and distribute traffic more evenly in the network by allowing a receiver to request, under certain circumstances, that a source deliver the multicast messages to it on the shortest path rather than on the shared tree path. Although this approach offers an improvement over ST-WIM proposed in [CGZ97] there is still a significant decrease in throughput as mobility in the nodes increase. As speed increases, throughput decreases, due to the inability of the routing and multicast protocols to keep up with node movements.

Results of the two approaches (Per Source and Shared Tree) show that these schemes scale well to large network size and can survive moderate speeds. In

comparison with the Per-Source Tree solution, the Share Tree scheme exhibits lower throughput at heavy load, as expected, due to higher traffic concentration on the common tree. It shows, however, much less control overhead than the Per-Source Tree, since the latter must constantly refresh separate trees rooted at different sources. It also offers better scalability to large network size. At high, uniform node mobility, both schemes perform rather poorly, indicating the need to explore non pro-active multicast strategies like on-demand multicasting.

### *2.3.2 MANET inspired Multicast Protocols*

[BGH00] simulated several multicast routing protocols developed specifically for MANET. Namely, Adhoc Multicast Routing (AMRoute) [BLM98], ODMRP, Adhoc Multicast Routing protocol utilizing Increasing Id numbers (AMRIS) [WTT98], and Core-Assisted Mesh Protocol (CAMP) [GM99] in diverse network scenarios using the GloMoSim library [UCLA]. AMRoute is a tree based protocol. It creates a bi-directional shared multicast tree using unicast tunnels to provide connections between multicast group members. Each group has at least one logical core that is responsible for member and tree maintenance. AMRIS establishes a shared tree for multicast data forwarding. Each node in the network is assigned a multicast session ID number. The ranking order of ID numbers is used to direct the flow of multicast data. CAMP supports multicasting by

creating a shared mesh structure. All nodes in the network maintain a set of tables with membership and routing information.

In their simulations the effect of mobility on the performance was measured by varying the speed of the network hosts. The number of data packets sent by senders was varied to emulate a variety of multicast applications. Different multicast group member sizes were simulated to investigate the impact of performance. Various traffic loads were also applied to study how traffic patterns influence multicast performance. Metrics were used to show the “efficiency” and “effectiveness” of the protocols. In their evaluation, they show that mesh protocols performed significantly better than the tree protocols in mobile scenarios.

[LK00] proposed a new ad hoc multicasting protocol called Neighbour Supporting Multicast Protocol (NSMP). NSMP uses a mesh infrastructure for resilience against link failures. As well, it attempts to minimize the frequency of control message broadcasts. For normal and periodic mesh maintenances, control messages reach only forwarding nodes and their neighbour nodes. In selecting a new route, NSMP prefers a path that contains existing forwarding nodes. Thus, NSMP enhances route efficiency by reducing the number of nodes. Through simulation in NS-2 [FV97], NSMP is compared with ODMRP. Simulation results show that NSMP substantially reduces control overhead and decreases data packet transmissions compared to ODMRP. Also, NSMP scales



well with increasing group size and sources do not show performance degradation in cases of high mobility. However, their simulation time only consists of a total of 300 seconds, with receivers and sources joining and leaving in the first 60 seconds and last 60 seconds respectively.

[HK00] evaluated multicast tree construction and proposed two new flooding methods that can improve the performance of the classic flooding method. They proposed the use of self pruning and dominant pruning to reduce the flooding cost, by utilizing neighbourhood information. While self pruning uses direct neighbour information only, dominant pruning uses neighbourhood information up to two hops apart. Based on extended neighbourhood information, each node decides the forward list for the next transmissions on the broadcast tree. The performance gain of dominant pruning is greater than that of self pruning. However, dominant pruning has larger overhead than self pruning and the overhead increases as the host mobility increases. Thus, the self pruning method could be more appropriate when the mobility of the host is high and the network is small. In contrast, the dominant pruning method could be the method of choice when the mobility is moderate and the network is large.

## Chapter 3

### 3 On-Demand Multicast Ad hoc Routing Protocols

#### 3.1 Multicast Ad Hoc On-Demand Distance Vector protocol (MAODV)

The MAODV protocol is capable of *multicast* communication. Multicast routes are discovered on-demand using a broadcast route discovery mechanism. MAODV creates bi-directional shared multicast trees connecting multicast sources and receivers. The operation of MAODV is loop-free (by using multicast group sequence numbers) and offers quick convergence when the ad hoc network topology changes.

##### 3.1.1 Multicast Route Discovery

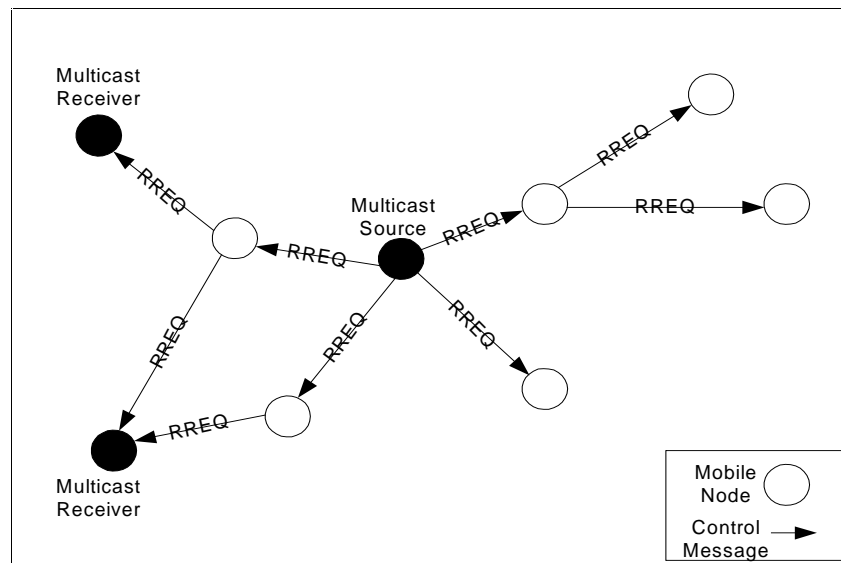


Figure 1 MAODV Multicast Route Discovery

A mobile node originates a Route Request (RREQ) message when it wishes to join a multicast group (sets a join flag in RREQ), or when it has data to send to a multicast group but it does not have a route to that group. The RREQ may either be broadcast or unicast depending on the information currently available at the source node. The source node checks to see if it has a record of the multicast group leader (first node to request a route to the multicast group) for that multicast in its request table. If it has the multicast group leader and the source node has a valid route to that node, it includes an extension field containing the group leader IP address and unicasts the RREQ along the known path to the group leader. Otherwise, if the source does not know who the group leader is, or if it does not have a valid route to the group leader, it broadcasts the request.

Only a member of the desired multicast group may respond to a join RREQ. If the RREQ is not a join request, any node with a fresh enough route (based on group sequence number) to the multicast group may respond. If an intermediate node receives a join RREQ for a multicast group of which it is not a member, or if it receives a RREQ and it does not have a route to that group, it rebroadcasts the RREQ to its neighbours.

Nodes receiving a join RREQ check their Request table for an entry for the requested multicast group. If there is no entry for the multicast group, the node enters the multicast group address, together with the IP address of the requesting node, in its request table. If there is no previous entry for the group, the

requesting node may become the group leader. A node wishing to join a multicast group consults its request table to determine the group leader.

### 3.1.2 Reverse Path Setup

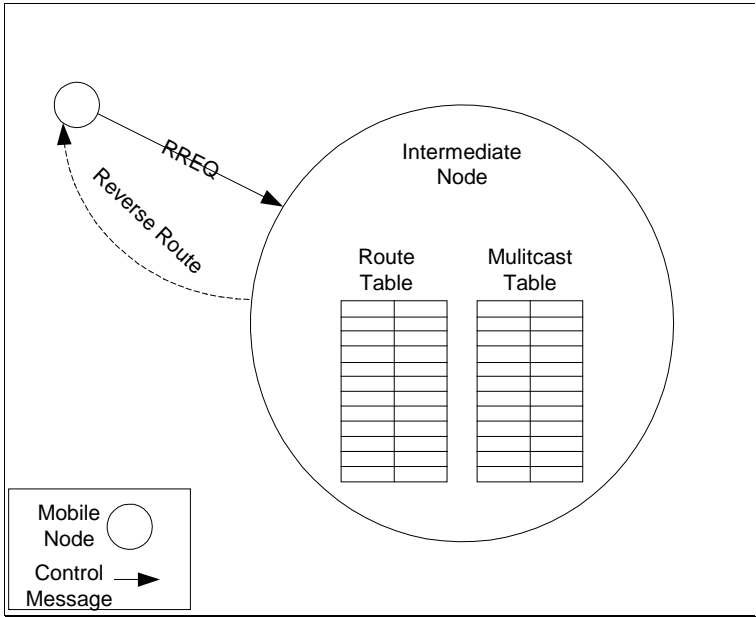


Figure 2 MAODV Reverse Path Setup

As the RREQ is broadcast across the network, nodes set up pointers to establish the reverse route in their route tables. A node receiving a RREQ first updates its route table to record the sequence number and the next hop information for the source node. This reverse route entry may later be used to relay a response back to the source. For join RREQs, an additional entry is added to the multicast route table. This entry is not activated unless the route is selected to be part of the multicast tree.

### 3.1.3 Forward Path Setup

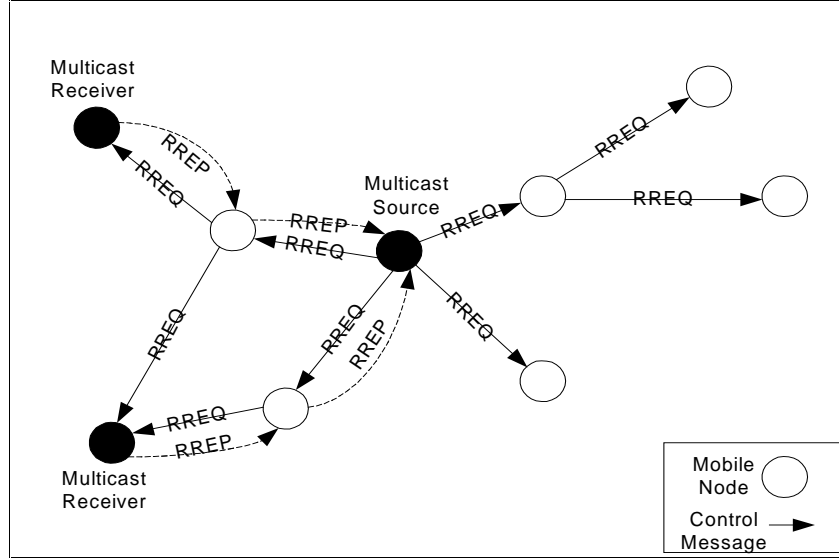


Figure 3 MAODV Forward Path Setup

If a node receives a join RREQ for a multicast group, it may reply if it is a member for the multicast group's tree and its recorded sequence number for the multicast group is at least as great as that contained in the RREQ. The responding node updates its route and multicast route tables by placing the requesting node's next hop information in the tables, and then unicasts a Request Response (RREP) back to the source node. As nodes along the path to the source node receive the RREP, they add both a route table and a multicast route table entry for the node from which they received the RREP, thereby creating the forward path.

### 3.1.4 Multicast Route Activation

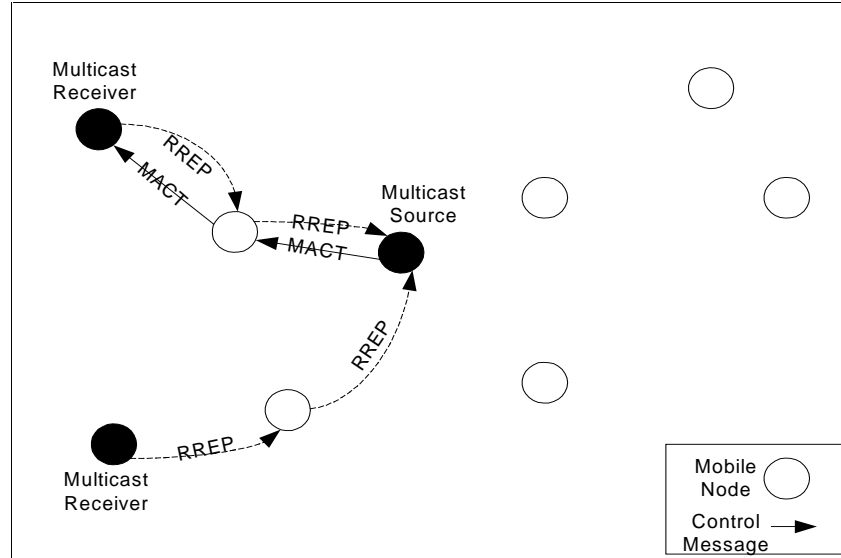


Figure 4 MAODV Route Activation

Multicast route activation deals with selecting and activating a link to be added to the tree when a new node joins the group. When a source node broadcasts a RREQ for a multicast group, it often receives more than one reply. The source node keeps the received route with the greatest sequence number and shortest hop count to the nearest member of the multicast tree for a specified period of time, and disregards other routes. At the end of this period, it enables the selected next hop in its multicast route table, and then unicasts a MACT message to this selected next hop. The next hop, on receiving the MACT message enables the entry for the source node in its multicast route table. If this node is a member of the multicast tree, it does not propagate the MACT any further.

However, if this node is not a member of the multicast tree, it will have received one or more RREPs from its neighbours. It keeps the best next hop for its route to the multicast group, unicasts a MACT to that next hop, and enables the corresponding entry in its multicast route table. This process continues until the node that originated the RREP (member of tree) is reached. The MACT message ensures that the multicast tree does not have multiple paths to any tree node. Nodes only forward data packets along activated routes in their multicast route tables.

### 3.1.5 Group Hello Messages

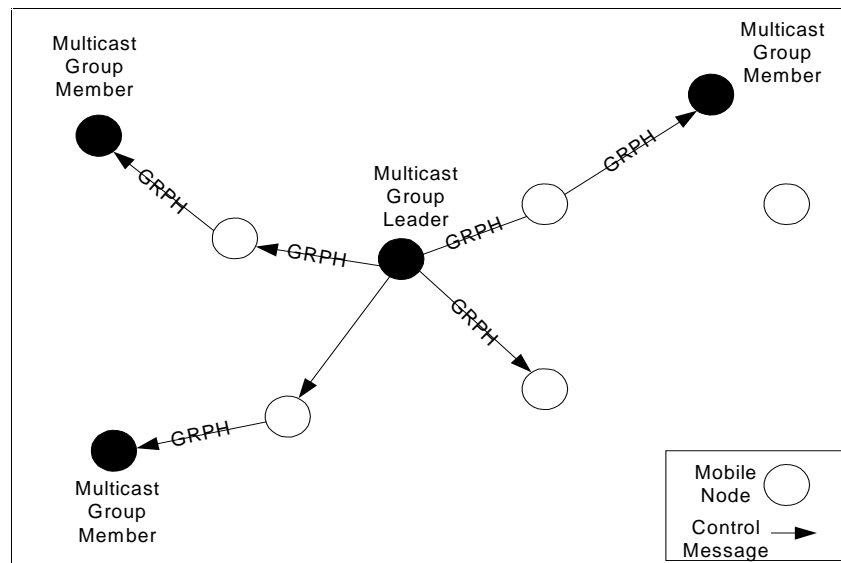


Figure 5 MAODV Group Leader Hello Messages

The first member of the multicast group becomes the leader for that group. This node maintains the group leader until it decides to leave the group, or until two

partitions of a multicast tree merge. The multicast group leader is responsible for maintaining the multicast group sequence number and broadcasting this number to the multicast group. This is done through a Group Hello message (GRPH), which is an unsolicited RREP. The Group Hello contains extensions that indicate the multicast group IP address and sequence numbers (incremented every Group Hello) of all multicast groups for which the node is the group leader. Nodes use the Group Hello information to update their request table.

### *3.1.6 Multicast Tree Maintenance*

#### *3.1.6.1 Pruning*

A multicast group member may decide to terminate its membership with the group, this requires pruning of the multicast tree. A leaf node may prune itself from the tree by setting the prune flag in the MACT message. A leaf node necessarily has only one next hop for the multicast group, so it unicasts the MACT message to that next hop. After sending the message, the node removes all information for the multicast group from its multicast route table. The next hop, on receiving the MACT, notes the prune flag, and consequently deletes the entry for the sender node from its multicast route table. Tree branch pruning ends when either a multicast group member or a non-leaf node is reached.

#### *3.1.6.2 Repairing Broken Links*

Nodes keep a record of the reception of any neighbour's transmission. A link breakage is detected if no packets are received from the neighbour after a



particular period. When a link breakage is detected, the node that is further from the multicast group leader (downstream of the break) is responsible for repairing the broken link. The downstream node initiates repair by broadcasting a RREQ with the Join flag set. Any node that is part of the multicast tree and that has a fresh enough multicast sequence number can respond to the RREQ by unicasting a RREP. If after a specific period the source node receives no RREP, it can be assumed that the tree cannot be reconnected. Thus the tree that is downstream from the break is left without a group leader. The group leader is chosen as follows. If the node that initiated the route rebuilding is a multicast group member, it becomes the new multicast group leader. On the other hand, if it was not a group member and has only one next hop for the tree, it prunes itself from the tree by sending its next hop a MACT message with the prune flag set. This continues until a group member is reached.

### *3.1.6.3 Reconnecting Partitioned Trees*

After the network multicast tree becomes disconnected due to network partition, there are two group leaders. If the partitions reconnect, a node eventually receives a Group Hello for the multicast group that contains group leader information that differs from the information it already has. If this node is a member of the multicast group, and if it is a member of the partition whose group leader has the lower IP address, it can initiate reconnection of the multicast tree. The node unicasts a RREQ with the repair flag set to its group leader. The

group leader, after receiving such a RREQ, grants the node permission to rebuild the tree by unicasting a RREP back to the node. After receiving the RREP granting it rebuilding permission, the node unicasts a RREQ to the other group leader, using the node from which it received the Group Hello as the next hop. When it receives the RREQ, the other group leader notes that the repair flag is set and takes the larger of its record of the group's sequence number and the received sequence number of the group. It then unicasts a RREP back to the source node. This group leader becomes the leader of the reconnected tree.

### *3.1.7 MAODV Multicast Parameters*

The following parameters are recommended by the MAODV draft.

GROUP\_HELLO\_INTERVAL = 5000 ms

RETRANSMIT\_TIME = 750 ms

## **3.2 On-demand Multicast Routing Protocol (ODMRP)**

The ODMRP protocol is mesh based, and uses a forwarding group concept (only a subset of nodes forwards the multicast packets). It applies on demand procedures to dynamically build routes and maintain multicast group membership. A soft-state approach is taken in ODMRP to maintain multicast group members. No explicit control message is required to leave the group.

### 3.2.1 Mesh Creation

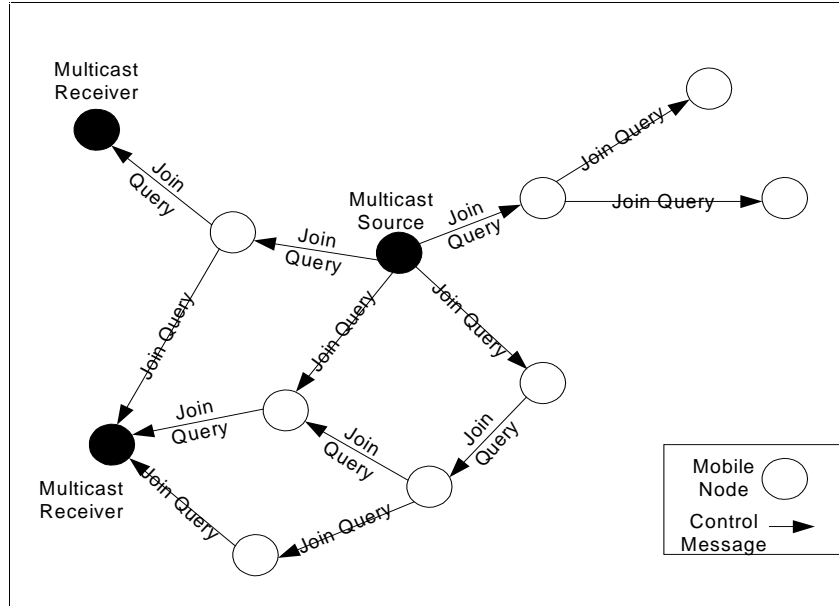


Figure 6 ODMRP Mesh Creation using broadcast Join-Query

In ODMRP, group membership and multicast routes are established and updated by the source on demand. When a multicast source has packets to send, but no route to the multicast group, it broadcasts a Join-Query control packet to the entire network. This Join-Query packet is periodically broadcast to refresh the membership information and update routes.

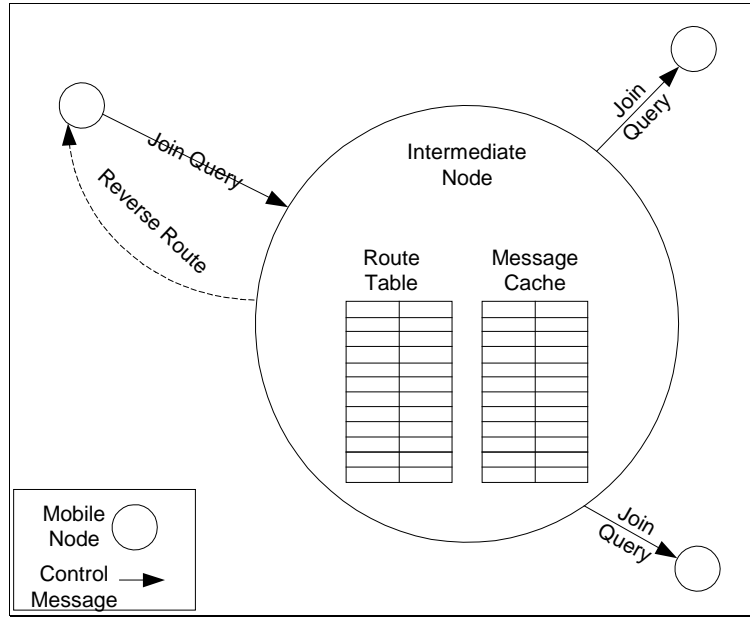


Figure 7 ODMRP Join-Query Processing

When an intermediate node receives the Join-Query packet, it stores the source ID and the sequence number in its message cache to detect any potential duplicates. The routing table is updated with the appropriate node ID (i.e. backward learning) from which the message was received for the reverse path back to the source node. If the message is not a duplicate and the Time-To-Live (TTL) is greater than zero, it is rebroadcast. By adjusting the TTL for broadcast messages, one can effectively limit the overhead through the network.

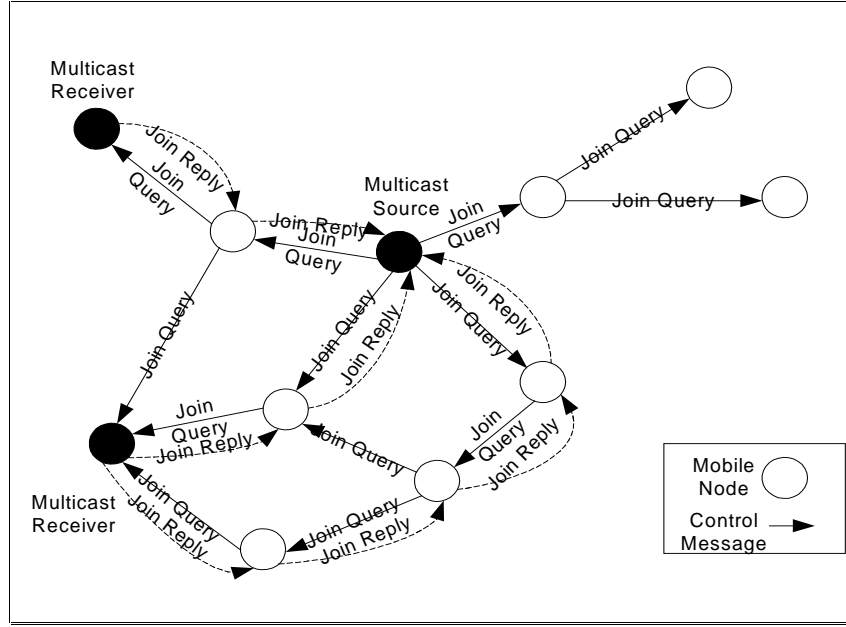


Figure 8 ODMRP Join-Reply Process

When the Join-Query packet reaches a multicast receiver, it creates and broadcasts a “Join Reply” to its neighbours. When a node receives a Join Reply, it checks if the next hop node ID of one of the entries matches its own ID. If it does, the node realizes that it is on the path to the source and thus is part of the forwarding group and sets the FG\_FLAG (Forwarding Group Flag). It then broadcasts its own Join Table built upon matched entries. The next hop node ID field is filled by extracting information from its routing table. In this way, each forward group member propagates the Join Reply until it reaches the multicast source via the selected path (shortest). This whole process constructs (or updates) the routes from sources to receivers and builds a mesh of nodes, the forwarding group.

### 3.2.2 *Routing Table*

A routing table is created on demand and is maintained by each node. An entry is inserted or updated when a non-duplicate Join-Query is received. The node stores the source node ID of the Join-Query and the next hop (i.e. the last node that propagated the Join-Query). The routing table provides the next hop information when transmitting Join Tables.

### 3.2.3 *Forwarding Group Table*

When a node is a forwarding group node of the multicast group, it maintains the group information in the forwarding group table. The multicast group ID and the time when the node was last refreshed are recorded.

### 3.2.4 *Data Forwarding*

After the forwarding group establishment and route construction process, sources can multicast packets to receivers via selected routes and forwarding groups. While it has data to send, the source periodically sends Join-Query packets to refresh the forwarding group and routes. When receiving the multicast data packet, a node forwards it only when it is not a duplicate and the setting of the FG\_FLAG for the multicast group has not expired. This procedure minimizes the traffic overhead and prevents sending packets through stale routes.

### 3.2.5 *Soft State*

In ODMRP, no explicit control packets need to be sent to join or leave the group. If a multicast source wants to leave the group, it simply stops sending Join-Query packets since it does not have any multicast data to send to the group. If a receiver no longer wants to receive from a particular multicast group, it does not send the Join Reply for that group. Nodes in the forwarding group are demoted to non-forwarding nodes if not refreshed (no Join Tables received) before they timeout.

### 3.2.6 *Mobility Prediction*

For highly mobile nodes equipped with Global Positioning System (GPS), ODRMP has the capability of adapting the refresh interval for the periodic flooding of the Join-Query messages. This is done in order to find the optimal flooding interval in order to reduce congestion, contention, and collisions caused by constantly flooding the network. By utilizing the location and movement information and a mobility prediction model, they predict the duration of time routes will remain valid. Using the predicted time before a route breaks, they can flood packets only when the predicted time expires.

With mobility prediction enabled in ODMRP, another feature in ODMRP that can be used to reduce route breaks, is to use a different route selection criteria. Instead of using the minimum delay path, the route that is the most stable is chosen. A multicast receiver must wait for a certain amount of time after

receiving a Join-Query so that it knows all the possible routes and route qualities.

The receiver then chooses the most stable route and broadcasts the Join Reply.

Please consult [GLS00] for more information on their mobility prediction model.

For the purposes of these simulations, for fairness in comparison, the mobility prediction feature in ODMRP was not used.

### 3.3 Qualitative Side by Side Protocol Comparison

The following table provides a side-by-side comparison of the two protocols.

Table 1 Qualitative characteristics of MAODV versus ODMRP

Characteristics	MAODV	ODMRP
Specific Unicast Protocol dependent	Yes - uses AODV unicast routing table	No
Multicast Support Required on Every Node	Yes - All nodes required to participate	Yes - All nodes required to participate
Distributed Operation	Yes	Yes
Loop Free	Yes	Yes
Demand Based / Reactive Operation	Yes - Source initiated route discovery	Yes - Source initiated route discovery
Proactive Operation	No	No
Periodic Messaging	Yes - Group Leader sends periodic: - Group Hello	Yes - Source node broadcasts periodic: - Join Query



Security	No	No
Sleep Period Operation	No	No
Control Packet Flood	Last Resort	Yes
Configuration	Tree	Mesh
Unidirectional Link Support	No -Assumes bi-directional links	No - Assumes bi-directional links
Link State Information	Hard - needs to detect broken links	Soft - lets broken links time out
Repair of Broken Links	Yes	No - Soft State
Network Partition Recovery	Yes	No - Soft State

### 3.4 Critique of MAODV and ODMRP

The two on-demand protocols share certain salient characteristics. In particular, they both discover multicast routes only in the presence of data packets in the need for a route to a multicast destination. Route discovery in either protocol is based on request and reply cycles where multicast route information is stored in all intermediate nodes on the multicast path. However, there are several important differences in the dynamics of the two protocols, which may give rise to significant performance differentials.

First, MAODV uses a shared bi-directional multicast tree for forwarding data packets while ODMRP maintains a mesh topology rooted from each source. In MAODV, any breaks in links may cause a partition in a multicast group; there are no alternative paths between source and destination. ODMRP provides

alternative paths and a link failure need not trigger the recomputation of the mesh from sources to receivers. However, a bi-directional tree is more efficient and would not result in duplicate sent to receivers.

Second, ODMRP broadcasts the reply back to the source while MAODV unicasts the reply back to the source. By using a broadcast mechanism, ODMRP allows for multiple possible paths from the multicast source back to the receiver. Since MAODV unicasts the reply back to the source, if an intermediate node on the path moves away, then the reply is lost, and the route is lost. However, a broadcasted reply requires intermediate nodes not interested in the multicast group to drop the control packets, resulting in extra processing overhead.

Third, MAODV does not activate a multicast route immediately while ODMRP does (unless mobility prediction is enabled). In MAODV, a potential multicast receiver must wait for a specified time allowing for multiple replies to be received before sending an activation message along the multicast route that it selects. Again, when an intermediate node on the chosen path moves away before a route activation is sent, the path is lost. On the flip side, waiting for a more stable route would be more advantageous than using one which will subsequently break right after the route has been activated.

Fourth, MAODV sends control messages to repair broken links and to manage network partitions. Since, there are no redundant links MAODV needs to

recover from breaks in links. ODMRP uses a soft state approach, and lets broken links timeout. Routes from multicast source to receivers in ODMRP are periodically refreshed by the source. However, depending on the refresh interval in ODMRP, the control overhead from sending route refreshes from every source could result in scalability issues.

Fifth, MAODV uses a multicast group leader to maintain up to date multicast tree information, while ODMRP source nodes periodically send request messages in order to refresh the multicast mesh. If two network partitions come together, MAODV requires explicit control to merge two network partitions.

Lastly, MAODV uses an expanded ring broadcast mechanism to broadcast control packets, while ODMRP sends all broadcast messages through the network. With the expanded ring broadcast technique, one limits the number of network wide broadcasts that are done. However, this is only used during the route discovery phase in MAODV, the periodic Hellos from the multicast group leader are broadcasted through the network.

## *Chapter 4*

### **4 Simulation**

#### **4.1 Simulation Environment**

The performance simulation environment used is based on *ns-2*, a network simulator that was originally developed by the University of California at Berkeley for the VINT project [FV99]. In previous work [BMJ98], the MONARCH research group in CMU developed support for simulating multi-hop wireless networks complete with physical, data link and IEEE 802.11 MAC layer models in *ns-2*.

The environment consists of 50 wireless mobile nodes roaming in a 1000 meters x 1000 meters flat space for 900 seconds of simulated time. The radio transmission range is 250 meters. A free space propagation channel is assumed. Group scenario files determine which nodes are receivers or sources and when they join or leave a group. A multicast member node joins the multicast group at the beginning of the simulation (first 30 seconds) and remains as a member throughout the whole simulation. Hence, the simulation experiments do not account for the overhead produced when a multicast member leaves a group. Multicast sources start and stop sending packets in the same fashion. Each data point represents an average of at least five runs with identical traffic models, but different randomly generated mobility scenarios. For fairness, identical mobility

and traffic scenarios are used across the compared protocols. Only one multicast group was used for all the experiments.

#### *4.1.1 Random Way-point Mobility model*

Each mobile node moves randomly at a preset average speed according to a “random waypoint model”. Here, each node starts its journey from a random location to a random destination with a randomly chosen speed (uniformly distributed between 0 – some maximum speed). Once the destination is reached, another random destination is targeted after a pause. By varying the pause time, the relative speeds of the mobiles are affected. For these set of experiments the pause time was always set to zero to create a harsher mobility environment. The maximum speeds used were chosen from between 1m/s to 20m/s.

#### *4.1.2 Traffic Pattern*

Different types of traffic generators were used in order to simulate multicast sources with different traffic characteristics. To simulate a uniform traffic pattern a constant bit rate (CBR) at four packets per second was used. For a non-uniform traffic pattern, an exponential distribution (EXP) was used, with the burst time set to 500ms. The size of generated multicast packets was kept constant at 512 bytes.

The following table provides a summary of all the simulation parameters:

Table 2 Summary of Simulation Parameters

Transmitter Range	250m
Bandwidth	2 Mbps
Simulation Time	900s
Number of Nodes	50
Pause Time	0
Maximum Mobility Speed	1m/s – 20 m/s
Environment Size	1000 x 1000m
Traffic Types	CBR and EXP
Packet Rate	4 packets/sec for CBR
Burst Time	500ms for EXP
Packet Size	512 bytes
Multicast Groups	1

## 4.2 Performance Evaluation Metrics

The following metrics were used in comparing the protocol performance. The metrics were derived from ones suggested by the IETF MANET working group for routing/multicast protocol evaluation [CM99].

### 4.2.1 *Packet Delivery Ratio*

The ratio of the number of packets actually delivered to the destinations versus the number of data packets supposed to be received. This number presents the effectiveness of a protocol in delivering data to the intended receivers within the network. The number of data packets supposed to be received is a theoretical

number projected from the multicast group member size and the number of packets sent from multicast sources.

#### *4.2.2 Number of data packets transmitted per data packet delivered*

“Data packets transmitted” is the count of every individual transmission of data by each node over the entire network. This count includes transmissions of packets that are eventually dropped and retransmitted by intermediate nodes. Note that in unicast protocols, this measure is always equal or greater than one. In multicast, since a single transmission can deliver data to multiple destinations, the measure may be less than one.

#### *4.2.3 Number of control packets transmitted per data packet delivered*

This measure shows the efficiency overhead in control packets expended in delivering a data packet to an intended receiver.

#### *4.2.4 Number of control packets and data packets transmitted per data packet delivered*

This measure tries to capture a protocol’s channel access efficiency, as the cost of channel access is high in contention-based link layers.

### **4.3 Network Parameters**

#### *4.3.1 Number of senders*

Vary the number of multicast senders in a given multicast group from one sender to twenty senders. The number of multicast group members, traffic source, and maximum speed were fixed at twenty, CBR, and 1m/s respectively.

#### 4.3.2 *Mobility*

Captures the relative motion of nodes in the network. Ad hoc multicast routing protocols must take action when the relative motion of nodes causes links to break or form, and a mobility metric should be proportional to the number of such events. By varying this parameter one tries to capture the robustness of the protocol to changes in network topology. With a pause time of zero the maximum mobility was varied from 1m/s to 20m/s. The multicast group members, traffic source, and number of senders were fixed at twenty, CBR, and five.

#### 4.3.3 *Multicast group size*

Vary the number of multicast members from ten multicast group members to fifty multicast group members to investigate the scalability of the protocol. The number of senders, traffic source, and maximum speed were fixed to five, CBR, and 1m/s respectively.

#### 4.3.4 *Traffic Patterns*

Vary the traffic source to investigate the response of the protocols to non-uniform (Exponential distribution) versus constant traffic patterns. The number of senders, multicast group members and maximum speed was fixed at five, twenty, and 1m/s respectively.



#### 4.4 Validation of MAODV and ODMRP Protocol Implementations

When implementing the multicast protocols, we followed the specification of each protocol as defined in the published literature. The MAODV implementation was an extension of the existing unicast implementation of AODV available from previous work done by [BMJ98]. The ODMRP implementation had to be developed from scratch since there was none available at the time in ns-2. To validate our implementations we performed numerous experiments within a less harsh environment than the one that will be used for performance evaluation. Basically, we used a 1000 x 1000 topology with a mobility speed of 1m/s, and a CBR traffic source. We kept the number of multicast sources to less than five senders and the multicast group size constant at twenty members. In this manner, we were able to validate the correct formation of a multicast group and the proper forwarding of packets to the multicast members. We also compared our validation results with those obtained from other implementations of the protocols.

In [RP99], the authors of MAODV implemented MAODV in the PARSEC [BM98] simulation development environment. Their simulations were conducted with a 50 node network in a 50m x 50m area for 300s. The transmission radius of the nodes was 10m. Their speed of the nodes in their simulations was varied from 0 m/s to 1m/s. Since they used a different simulator, with different scenarios it is hard to perform a one to one comparison with their results.

However, we can see that with two senders, our MAODV implementation exhibits a similar data delivery ratio (90%) to what they obtained for goodput ratio (~94%). They define the goodput ratio as the number of data packets received compared to the number of data packets sent. With between one to two senders we find that MAODV is quite effective in delivering packets to all twenty multicast group members. However with more senders, MAODV requires a period of time in order to elect a multicast group leader. If more than one multicast group leader was elected due to a network partition, a network partition merge is required when they come together. During this time, packets to multicast group members are lost, since the multicast tree is unstable.

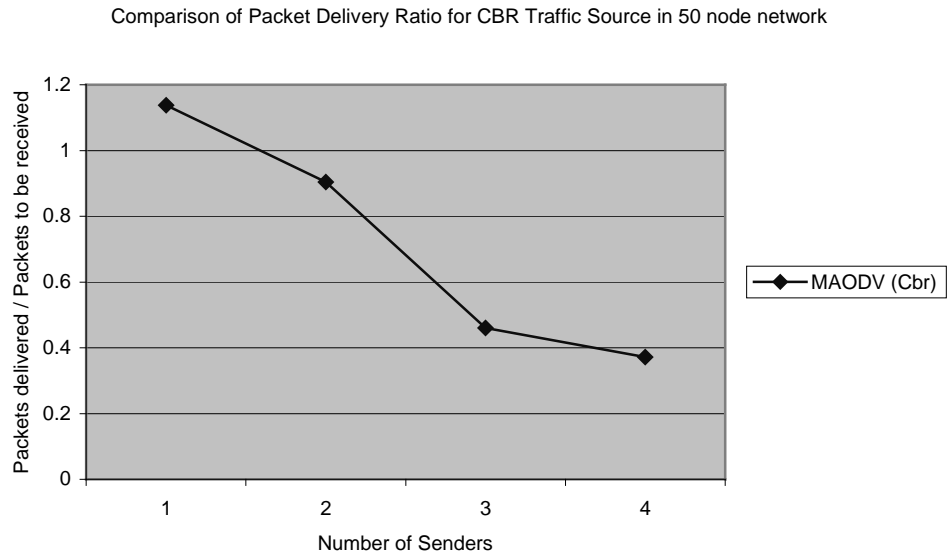


Figure 9 AODV Packet Delivery Ratio

In [BGH00], ODMRP was implemented with the GloMoSim library [UCLA]. Their simulation environment consists of a network of 50 nodes in a 1000m x 1000m area with a radio propagation range for each node of 250m. In one set of simulations they vary the number of senders from one to twenty with mobility set to 1m/s and the multicast group size set to twenty members. For four senders, our packet delivery ratio of 92% is close to their obtained results of  $\sim 95\%$ . We also notice a difference in their trend as the number of senders increase. Their packet delivery ratio remains the same as the number of senders increase. However it should be noted that their implementation of ODMRP limits the number of sources that can flood Join-Query messages at the same time. Whenever a source needs to flood a Join-Query, their implementation listens to see if any other source is flooding the packet. It then proceeds to send the Join-Query only if no flooded packets are received within a certain period. This has the effect of decreasing their collisions and congestion. This was not done in our implementation since we allowed sources to flood Join-Queries whenever they needed. Between one to four senders, we see that ODMRP is highly effective and actually delivers duplicate copies of packets at times due to the redundancy in the mesh.

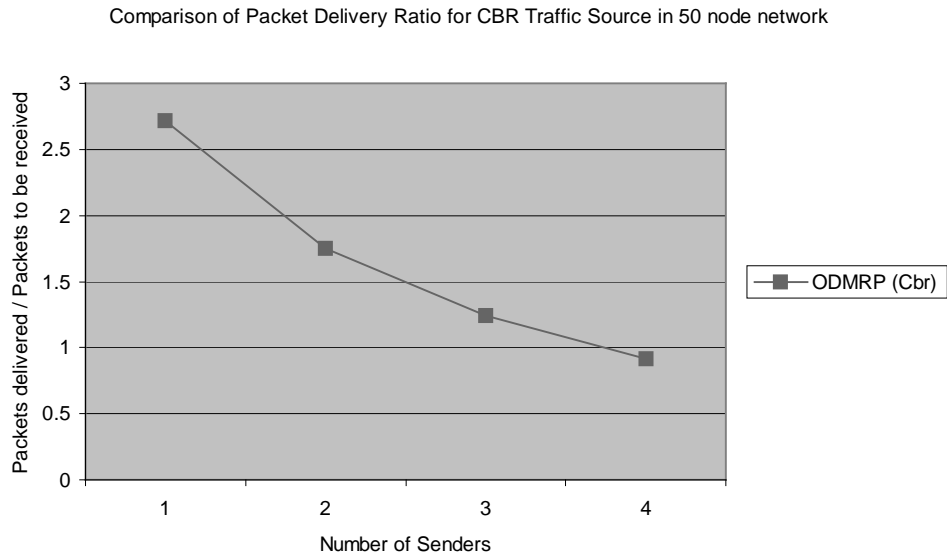


Figure 10 ODMRP Packet Delivery Ratio

#### 4.5 Multicast versus Unicast

At first glance, the of merits of using multicast routing protocols in a MANET may not appear advantageous given the decrease in performance when we have an increase in the number of senders. This is especially the case in MAODV for Figure 9 where the packet delivery ratio becomes less than 50% after two senders. However, if one were to look at the transmission overhead for every packet delivered when using unicast to group members the advantages of multicast become immediately obvious. Assuming a multicast group size of twenty members, in order to unicast to them all, each source will have to originate twenty packets in order to reach the multicast group. If all the members were one hop away then the transmission overhead would be at best one packet

transmitted for every packet delivered in order to unicast to all the members. In the case of a 1000x1000 m topology with a 250m transmission radius, the number of hops required in order to reach a multicast group member would be at most four hops. Thus, each unicast packet would have to be retransmitted at least three more times by intermediate nodes in order to reach a multicast group member. With twenty multicast group members, this could result in a transmission overhead of four data packet transmissions per data packet delivered. This does not take into account any congestion and loss of packets, which would result in, even more retransmissions.

We performed a comparison between multicast and unicast AODV using the basic 1000 x 1000 m topology with 50 mobile nodes, 20 multicast group members, mobility speed of 1 m/s, and a CBR traffic source. In this experiment, for the unicast simulation, each sender has 20 unicast connections to the appropriate multicast group members.

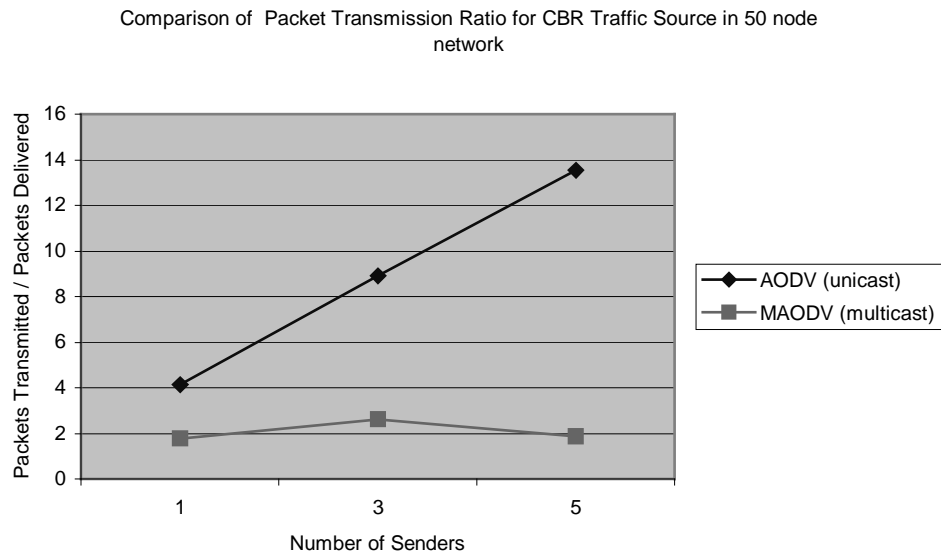


Figure 11 Packet Transmission Ratio as a Function of the Number of Senders (unicast vs. multicast)

In Figure 11, we see that the transmission cost of using unicast increases by over 500% as the number of senders reaches five. At five senders, in the unicast case, each sender needs to maintain twenty simultaneous unicast connections with twenty multicast group members for a total of one hundred unicast connections. As one can see, this results in a really inefficient use of the wireless medium as well as congestion in the network, resulting in a lot of retransmissions in order to deliver a packet. This is reflected in the packet delivery ratio in Figure 12, where the unicast is 45% less effective than the multicast at five senders.

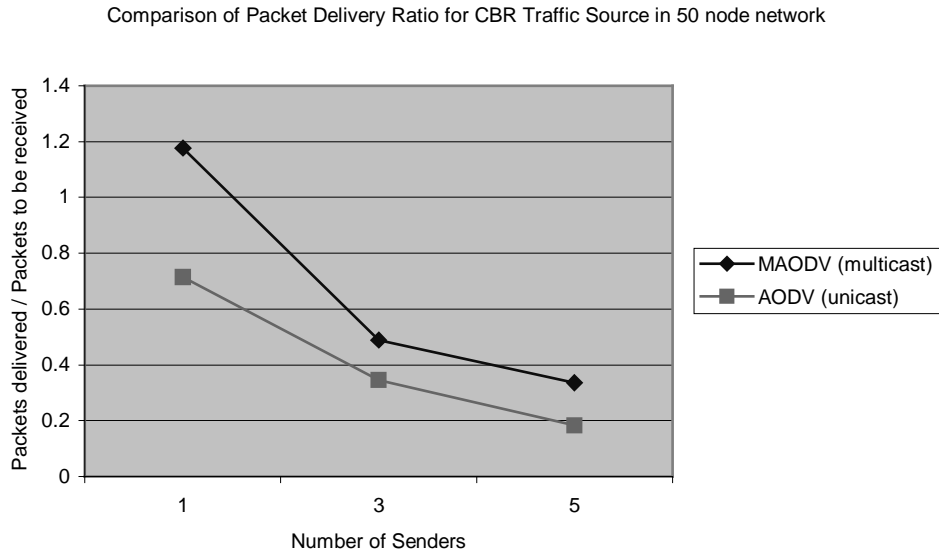


Figure 12 Packet Delivery Ratio as a Function of the Number of Senders (unicast vs. multicast)

We can see that the broadcast nature of the wireless medium is not taken advantage of when doing a single unicasts for multicast packets. As well, the control overhead would grow as the multicast group members became larger. This is because a route discovery would be required from every source to every multicast group member. As well, every source would need to keep track of all their multicast group members, placing extra processing overhead on the application.

## *Chapter 5*

### **5 Simulation Results**

#### **5.1 MAODV Route Activation**

We analyzed the effect of immediate route activation versus the delayed approach proposed in the AODV draft. In AODV, when a potential member node receives a response to its request to join a multicast group, it is required to wait a period of time before activating it. However, in many cases, the activation route spans multiple intermediate nodes that are mobile and moving around while the potential member is waiting to activate the route. If an intermediate node were to move out of the transmission range the potential multicast path would be lost. In this scenario, the potential member would need to reinitiate the route discovery process to discover another potential path to the multicast tree. We investigated the impact of using different periods for route activation on MAODV.

The mobility speed was varied from 1m/s to 20m/s, multicast group size was twenty members, the traffic source used was CBR at 4 packets/sec, and there were five senders.



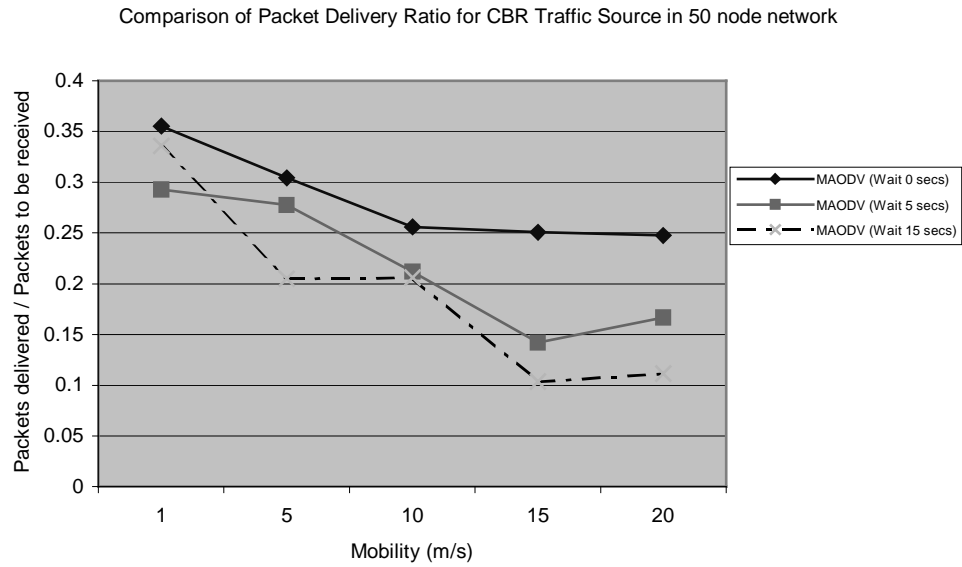


Figure 13 Packet Delivery Ratio as a Function of Mobility  
with Different Wait Times

From Figure 13, by using immediate route activation during high mobility (20m/s), MAODV exhibits at least a 49% increase in packet delivery ratio over having a wait time before activating a potential route. We see that this does not become a factor for low levels of mobility ( $<5\text{m/s}$ ) the difference in packet delivery ratio is less than 6%. For all subsequent experiments, MAODV will be configured with zero wait time before route activation.

## 5.2 Senders

For the first set of simulations, we varied the number of senders in the multicast group in order to evaluate the protocol scalability with respect to source nodes and the resulting effective traffic load. In Figure 14, ODMRP is over 53% more effective than MAODV in data delivery ratio as the number of senders increases from one to twenty. In terms of packet transmission ratio though, in Figure 15, at twenty senders, MAODV sends 75% fewer packets for each data packet delivered than ODMRP. As well, in Figure 16, MAODV sends 59% fewer control overhead packets than ODMRP for each data packet delivered as the number of senders reaches twenty. For both control and data transmissions, from Figure 17, MAODV sends 90% less packets than ODMRP for every packet delivered as the number of senders reaches twenty.

One can observe that both protocols do not scale well for packet delivery ratio as sender size increases along with the effective traffic load. In ODMRP, every source node will periodically send out route requests through the network. As can be seen from the results, when the number of source nodes becomes larger, the effect of this causes congestion in the network and the data delivery ratio drops significantly. MAODV, on the other hand, maintains only one group leader for the multicast group that will send periodic Group Hellos through the network. In this manner, it is much more scalable than ODMRP.

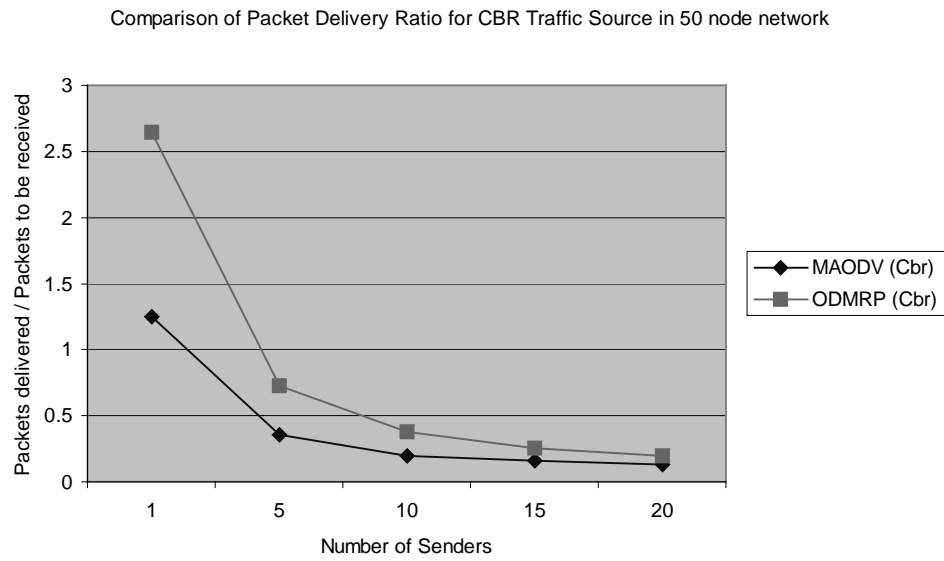


Figure 14 Packet Delivery Ratio as a Function of the Number of Senders

Comparison of Packet Transmission Ratio for CBR Traffic Source in 50 node network

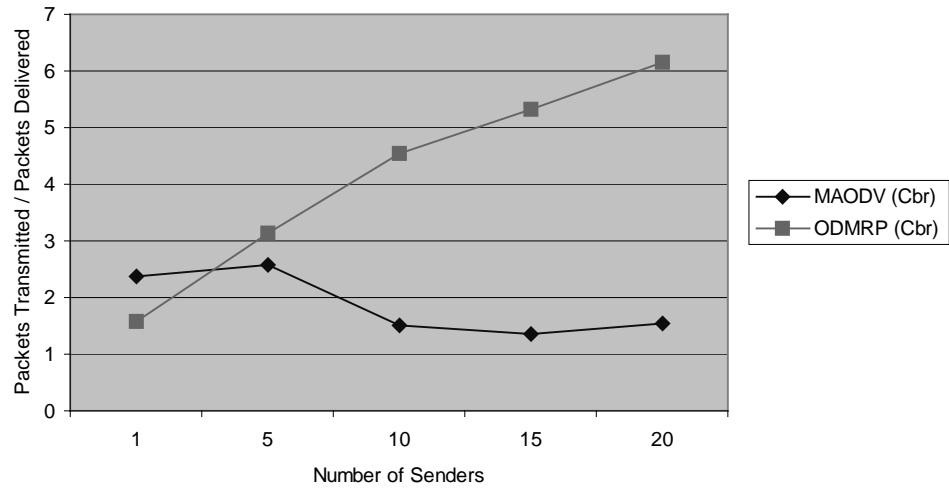


Figure 15 Packet Transmission Ratio as a Function of the Number of Senders

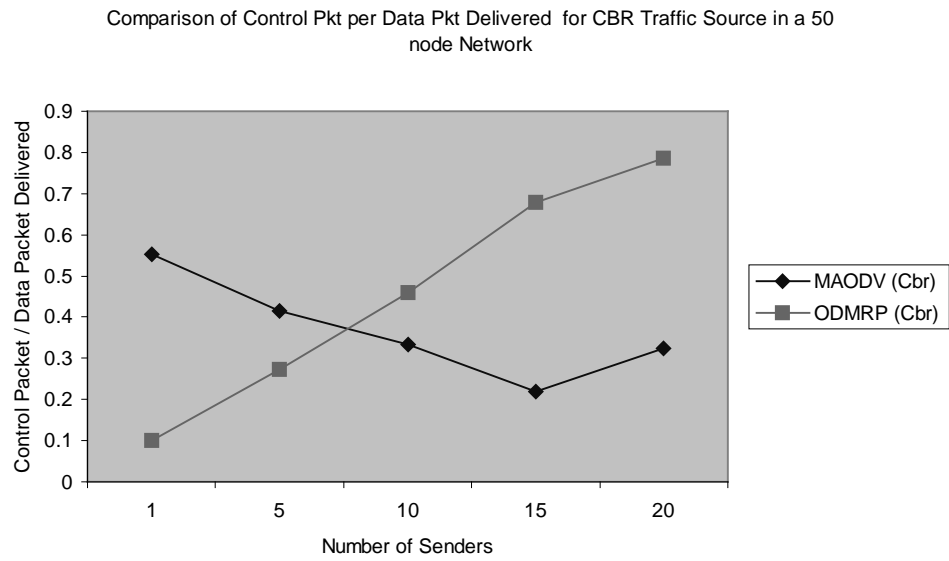


Figure 16 Control Overhead per Data Packet Delivered as a Function of the Number of Senders

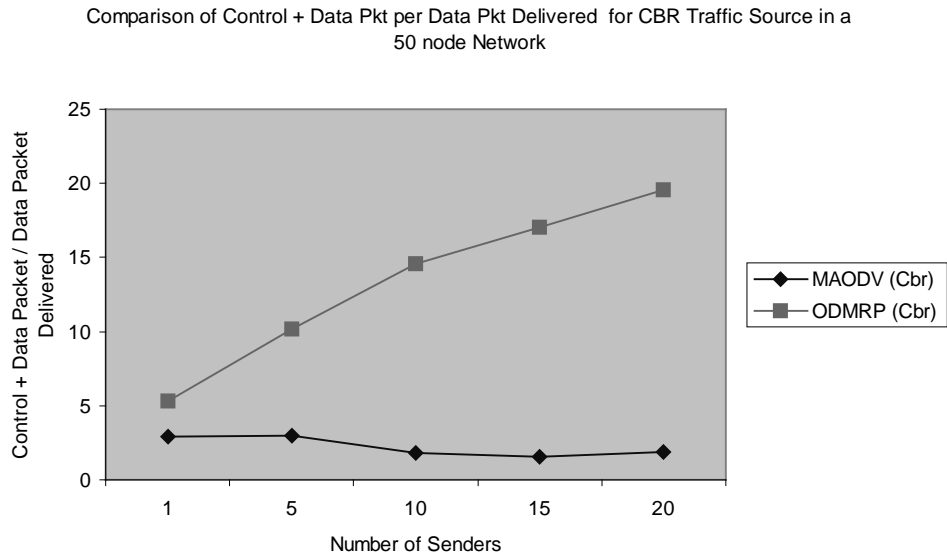


Figure 17 Control and Data Transmissions per Data Packet Delivered as a Function of the Number of Senders

### 5.3 Mobility

For the second set of simulations, we varied the mobility to evaluate the ability of the protocols to deal with route changes. From Figure 18, ODMRP is over 104% more effective than MAODV in data delivery ratio as the mobility is increased from 1m/s to 20m/s. In terms of packet transmission ratio, in Figure 19, ODMRP sends 40% less packets for each data packet delivered at high mobility ( $>15\text{m/s}$ ). As well, for control overhead, in Figure 20, ODMRP decreases by up to 74% less than MAODV for each data packet delivered as the mobility reaches 20m/s. For both control and data transmissions, in Figure 21, ODMRP sends 48% less packets than MAODV for every packet delivered. We

see that ODMRP is generally unaffected by increases in mobility, while MAODV is more sensitive to changes in mobility. The mesh topology of ODMRP allows for alternative paths thus making it more robust than MAODV. MAODV relies on a single path on its multicast tree, and must react to broken links, by initiating repairs.

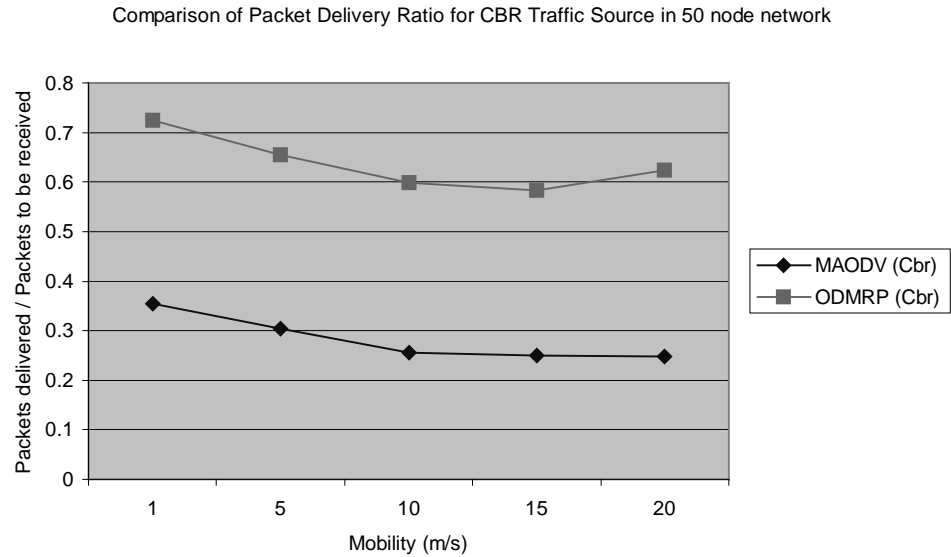


Figure 18 Packet Delivery Ratio as a Function of Mobility

Comparison of Packet Transmission Ratio for CBR Traffic Source in 50 node network

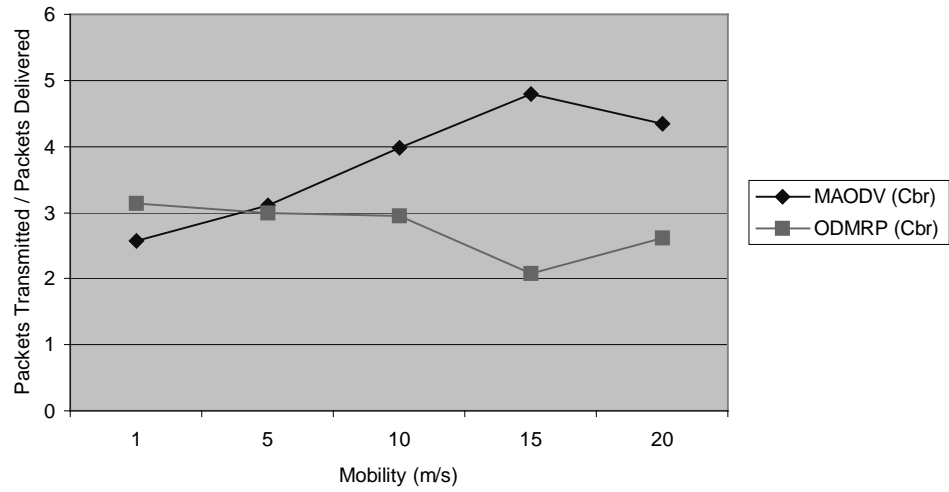


Figure 19 Packet Transmission Ratio as a Function of Mobility



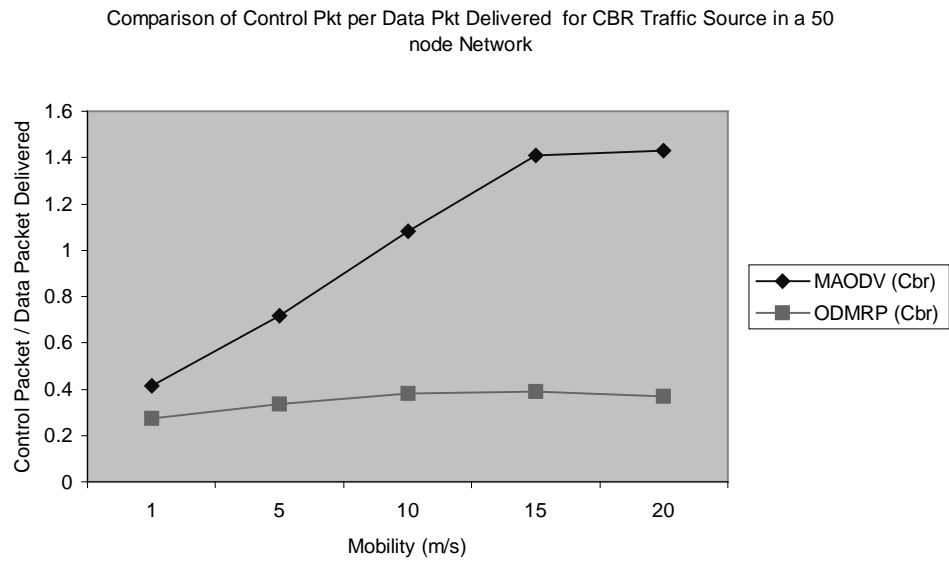


Figure 20 Control Overhead per Data Packet Delivered as a Function of Mobility

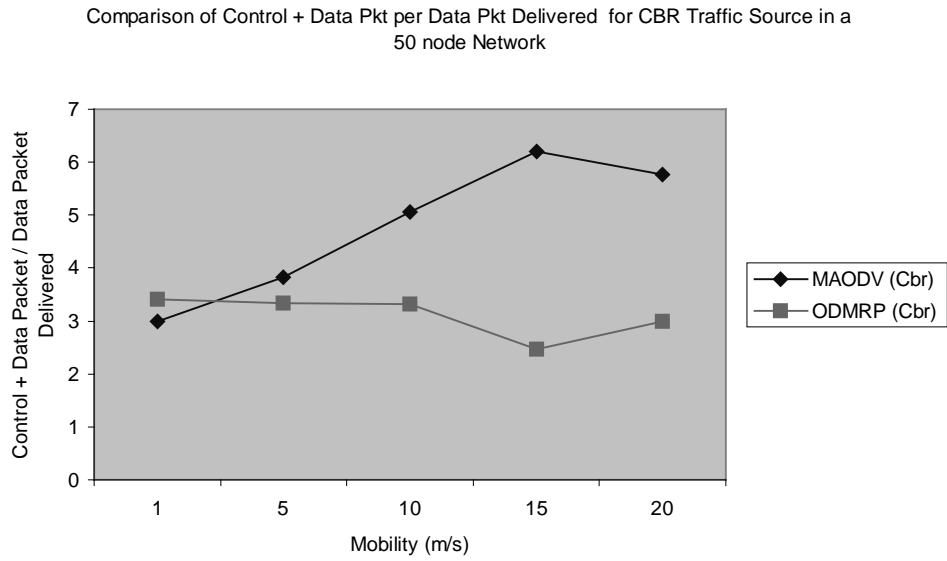


Figure 21 Control and Data Transmissions per Data Packet Delivered as a Function of Mobility

#### 5.4 Multicast Group

For the third set of simulations, we varied the number of members in the multicast group in order to evaluate the protocol scalability with respect to multicast group size. In Figure 22, ODMRP is 270% to 20% more effective than MAODV in data delivery ratio as the number of multicast group members is increased from ten to fifty. In terms of packet transmission ratio, in Figure 23, MAODV sends up to 48% less packets for each data packet delivered. As well, for control and data transmissions, from Figure 25, MAODV decreases by up to 46% less than ODMRP for each data packet delivered.

One can observe that ODMRP does not scale well with multicast group size. There is a drastic decline in packet delivery ratio as the multicast group increases to fifty members. This can be attributed to collisions that occur from the frequent broadcasts through the network. Despite the poor data delivery ratio, we see that MAODV scales better in terms of overall control and data transmissions for every packet delivered.

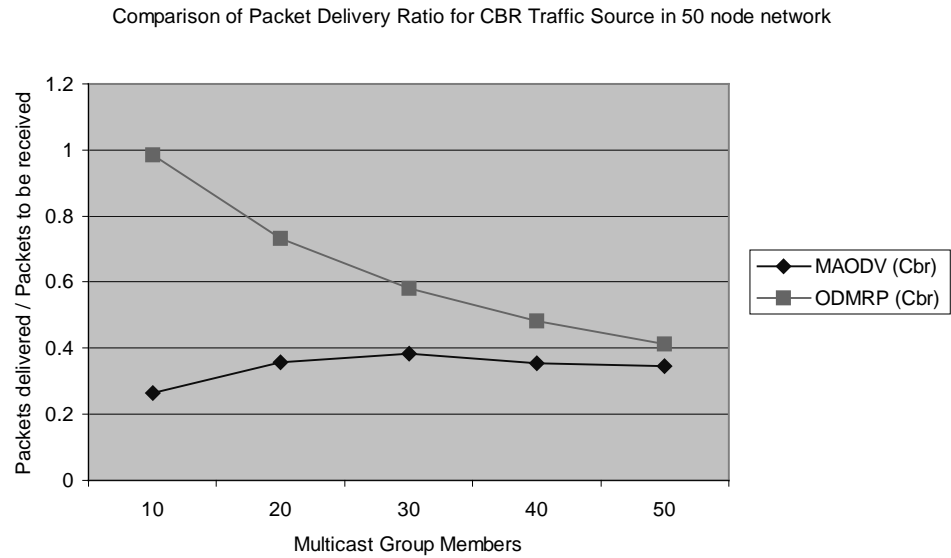


Figure 22 Data Delivery Ratio as a Function of Multicast Group Size

Comparison of Packet Transmission Ratio for CBR Traffic Source in 50 node network

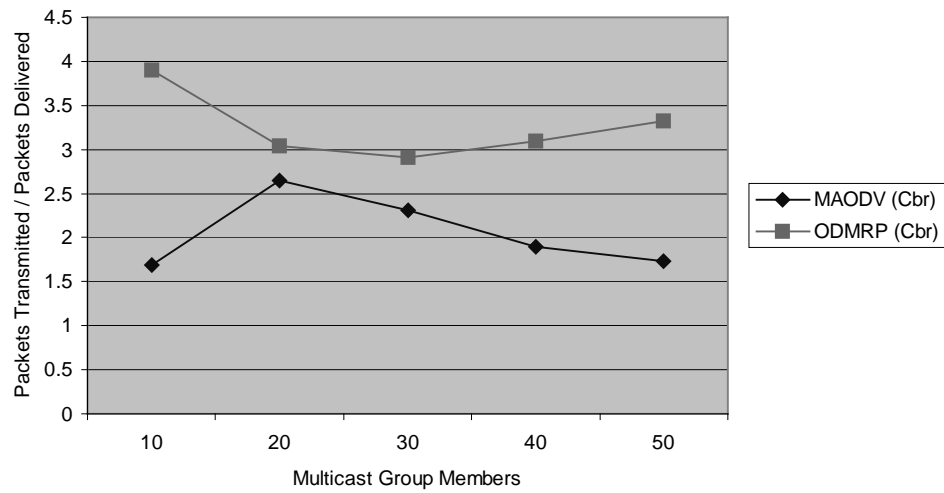


Figure 23 Packet Transmission Ratio as a Function of Multicast Group Size

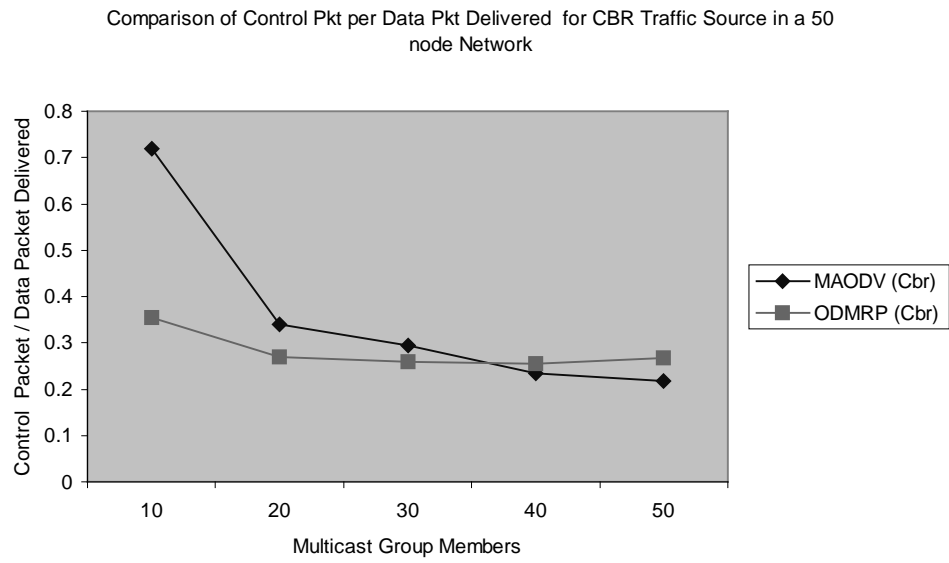


Figure 24 Control Overhead per Data Packet Delivered as a Function of Multicast Group Size

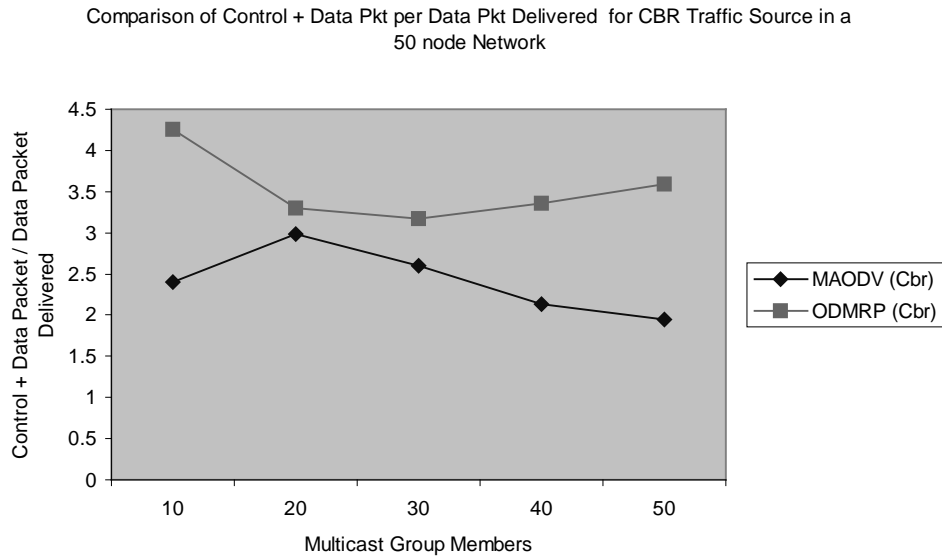


Figure 25 Control and Data Transmissions per Data Packet Delivered as a Function of Multicast Group Size

## 5.5 Traffic Source

For the final set of simulations, we varied the traffic source to evaluate the effect of a non-uniform traffic source on the two protocols. In Figure 26, MAODV is up to 57% less effective in data delivery with an EXP traffic source. For ODMRP, in Figure 27, an EXP traffic source is 10% less effective than CBR in data delivery ratio. In terms of packet transmission ratio, from Figure 28, MAODV sends 39% more packets for each data packet delivered using an EXP traffic source versus a CBR at fifty members. From Figure 29, ODMRP sends 37% less packets for each data packet delivered at fifty members with an EXP traffic source.

In general, we find that ODMRP is quite resistant to different traffic sources, and generally exhibits the same behaviour. MAODV on the other, hand is more sensitive to the type of traffic.

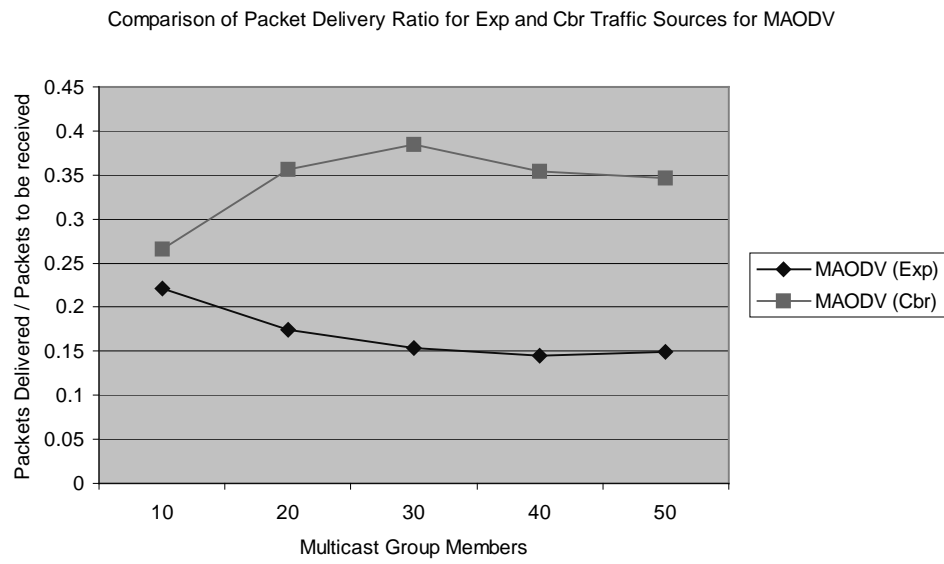


Figure 26 MAODV Data Delivery Ratio as a Function of Multicast Group Size with Different Traffic Sources

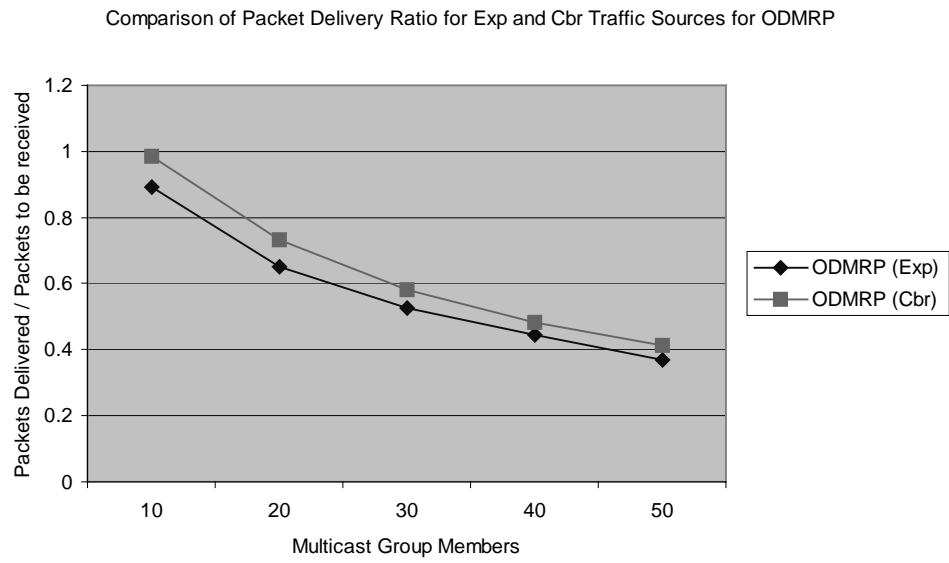


Figure 27 ODMRP Data Delivery Ratio as a Function of Multicast Group Size with Different Traffic Sources



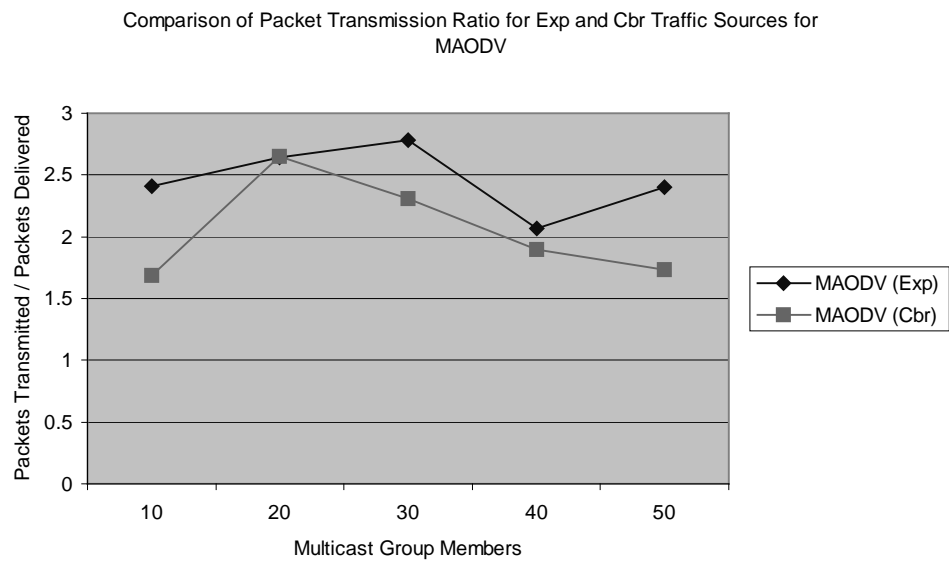


Figure 28 MAODV Packet Transmission Ratio as a Function of Multicast Group Size with Different Traffic Sources

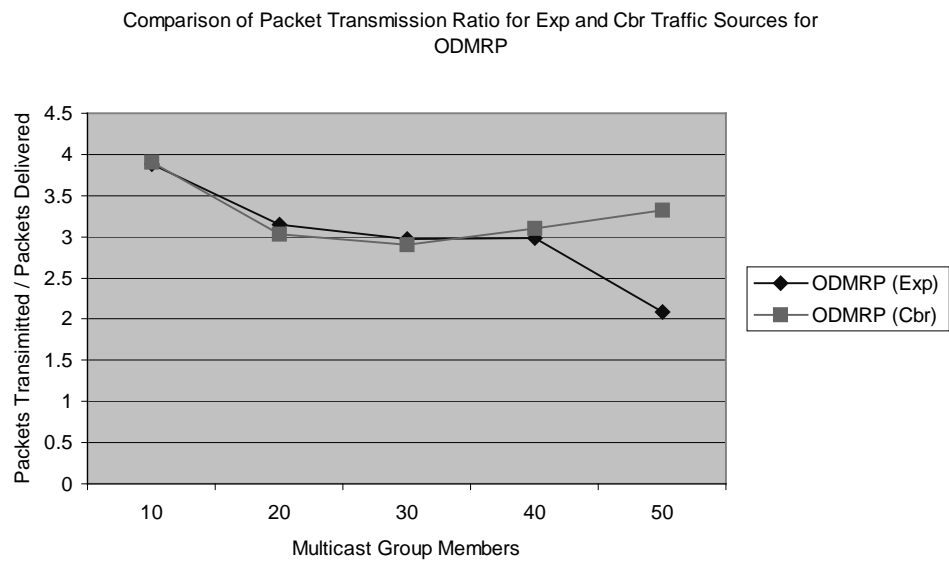


Figure 29 ODMRP Packet Transmission Ratio as a Function of Multicast Group Size with Different Traffic Sources

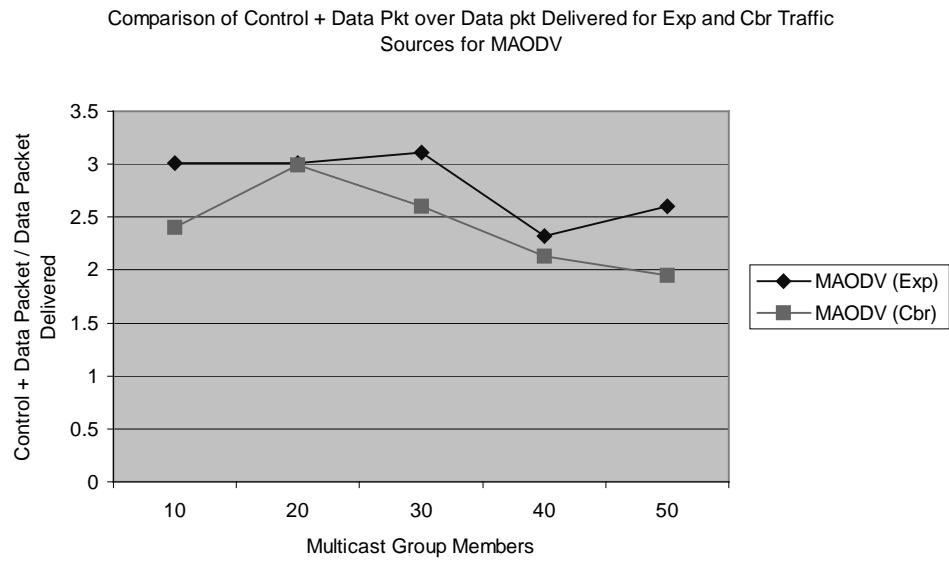


Figure 30 MAODV Control and Data Transmissions per Data Packet Delivered as a Function of Multicast Group Size with Different Traffic Sources

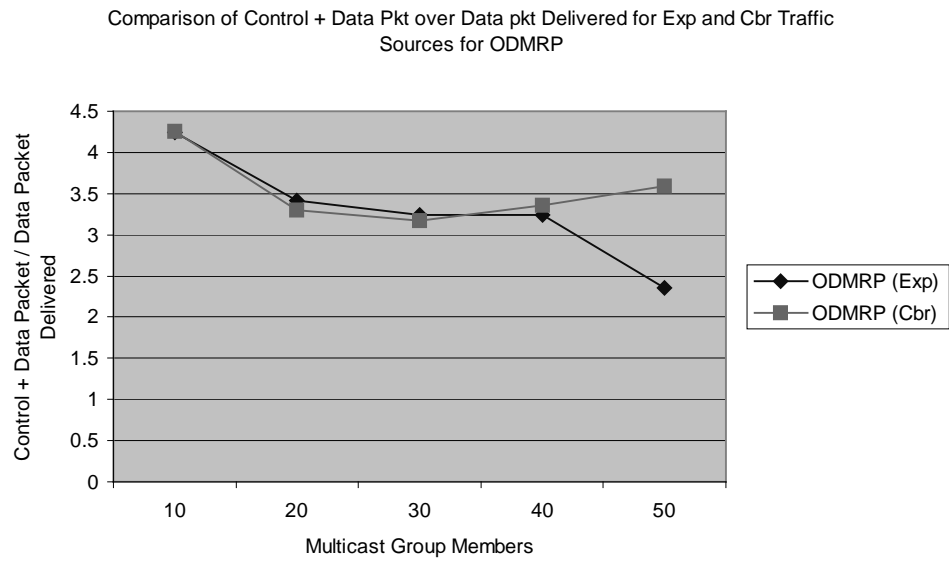


Figure 31 ODMRP Control and Data Transmissions per Data Packet Delivered as a Function of Multicast Group Size with Different Traffic Sources

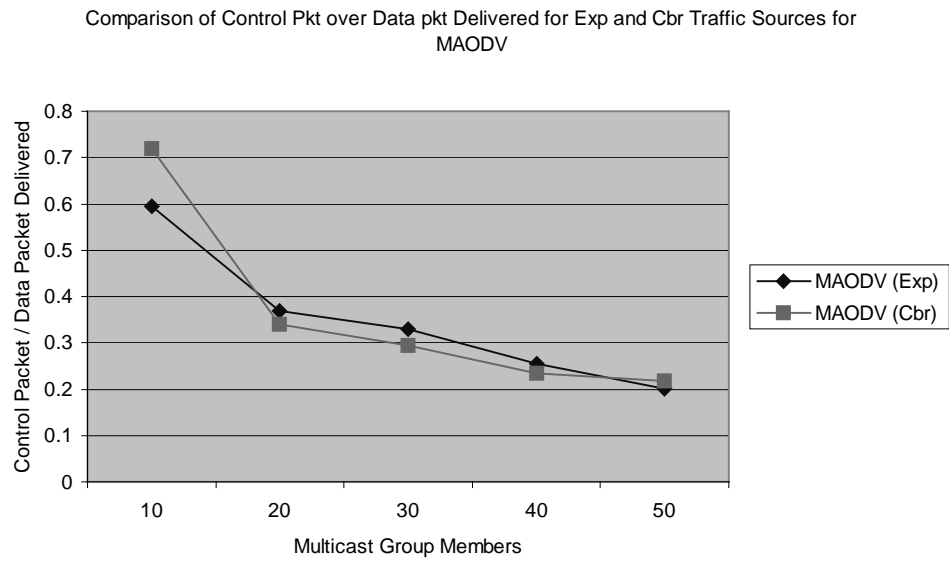


Figure 32 MAODV Control overhead per Data Packet Delivered as a Function of Multicast Group Size with Different Traffic Sources

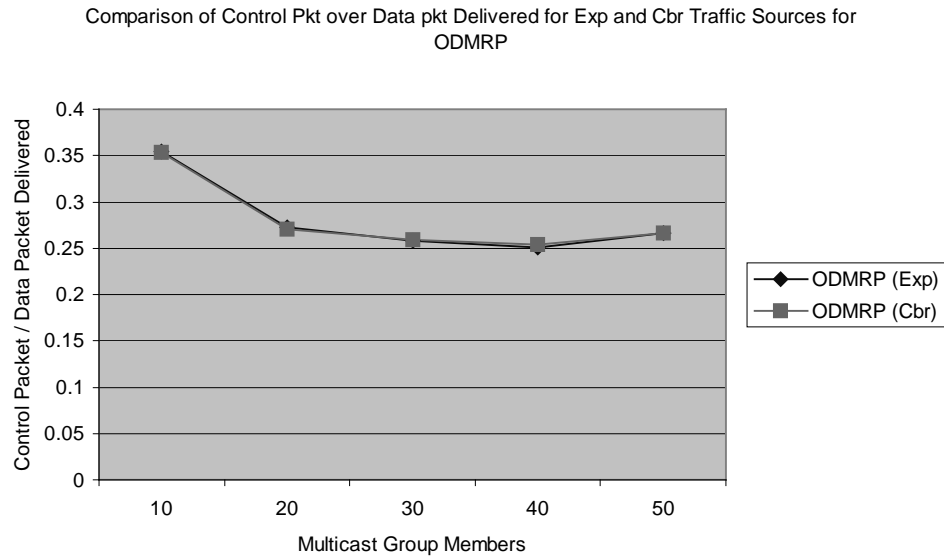


Figure 33 ODMRP Control overhead per Data Packet Delivered as a Function of Multicast Group Size with Different Traffic Sources

## 5.6 Protocol Analysis

MAODV shows a poor packet delivery ratio in comparison to ODMRP. Fundamentally, we can see that the use of a bi-directional shared tree strategy versus a mesh has an enormous impact on the protocol performance. Since MAODV uses a shared tree for data dissemination, there is only one path between member nodes. If a single tree link breaks because of node movement, packet collision, or congestion, destinations cannot receive packets. MAODV will then have to repair the link by broadcasting a Route Request where the link

failed. However, this requires a certain period of time, resulting in a network partition and ultimately data loss.

ODMRP provides redundant routes with a mesh topology and the chances of packet delivery to destinations remain high even when the primary routes are unavailable. The path redundancy enables ODMRP to suffer only minimal data loss and be robust to mobility.

ODMRP suffers from scalability issues as the multicast group increases or the sender size increases. This is because it maintains per-source meshes connecting receivers and senders. As the number of senders increase, the periodic Join Query packets that each source originates increases causing higher amounts of congestion and control overhead. We see a sharp degradation in packet delivery caused by larger number of collisions and buffer overflows.

MAODV, on the other hand scales much better since it uses a single multicast group leader to send out periodic Hellos through the network. Increasing the number of senders has minimal impact on the control overhead in MAODV.

## **5.7 Protocol Improvements**

In order to improve the scalability, one of the directions that can be taken is to implement the heuristic self-pruning approach suggested in [HK00] for improving the flooding of packets. In self-pruning, each node exchanges the list of its adjacent nodes with neighbours. Whenever a node wishes to forward a

packet it piggybacks an adjacent node list in the flooded packet. A node receiving the packet checks first if its adjacent neighbours are all in the list. If they are then, it refrains from forwarding the packet since it knows all its neighbours have already received the packet.

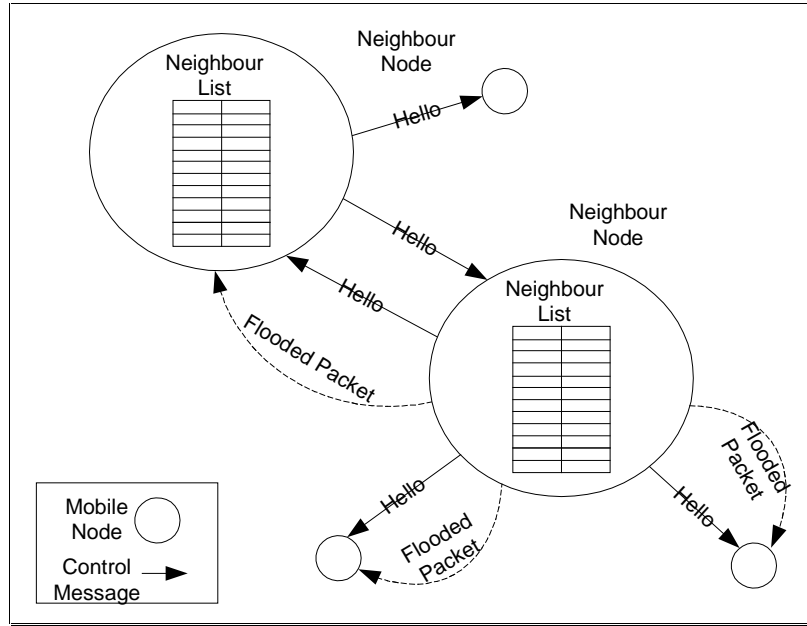


Figure 34 Self-Pruning for Flooded Packets

We implemented self-pruning for the Join-Query messages in ODMRP and for the Group Hello and Route Request messages in MAODV to evaluate whether there is control overhead decrease for flooded packets. Using a multicast group of twenty members, twenty senders, and a mobility speed of 1m/s, we find that the control overhead due to Join-Query in ODMRP was decreased by 17%. For this same scenario, MAODV Group Hello messages were decreased by 75% and



the Route Request messages were decreased by 90%. Looking at a different scenario with a multicast group of fifty members, five senders, and a mobility speed of 1m/s, we find the control overhead due to Join-Query for ODMRP decreases by 19%. For that scenario using MAODV, we find the Group-Hello messages decreasing by 35%, and the Route Request messages by 41%. However, in ODMRP, it should be noted that there is a control overhead introduced by using self-pruning in order to maintain neighbour information has not been accounted for. This is not the case in MAODV, since neighbour Hellos are already being used. As well, there is extra overhead in the flooded packets themselves in order to carry the adjacent node list information.

## *Chapter 6*

### **6 Conclusions and Future Work**

We have compared the performance of MAODV and ODMRP, two prominent on-demand multicast routing protocols for ad hoc networks. MAODV and ODMRP both use on-demand route discovery, but with different routing mechanisms. In particular, ODMRP uses a source based mesh topology while MAODV uses a shared bi-directional multicast tree. In general, we find that ODMRP out performs MAODV in terms of packet delivery. However, ODMRP suffers from scalability issues as the number of senders increase. In MAODV, we find that by using immediate route activation the packet delivery improves at high levels of mobility. As well, the use of self-pruning in ODMRP and MAODV decreases the control overhead for packets that are flooded through the network. We also experimented with a non-uniform traffic source versus the standard constant bit rate. We find that there is little effect on ODMRP while MAODV is more sensitive and performs worse in the presence of a non-uniform traffic source.

A future direction for improving scalability in ODMRP is to implement the dominant pruning approach suggested in [HK00] for improving the flooding of packets. In dominant pruning, the range of neighbourhood information is extended into two-hop apart nodes versus only directly connected nodes in self-

pruning. The sender selects adjacent nodes that should relay the packet to complete the broadcast. The Ids of selected adjacent nodes are recorded in the packet as a forward list. An adjacent node that is requested to relay the packet again determines the forward list. This process is iterated until broadcast is completed. By using techniques such as this, one could possibly decrease the congestion and control overhead caused by flooding packets.

As well, the mobility prediction feature in ODMRP was not evaluated. More investigation needs to be done to see whether this can improve the flooding of Join-Query packets. The route selection criteria should also be evaluated along with this to see if choosing more stable routes is a more advantageous criteria within a highly mobile environment.

The primary area of improvement for MAODV is the fragility of the bi-directional shared tree causing poor packet delivery ratios. A possible improvement can be to build redundant or backup links that can be used in the event of a primary link failure. Currently, when a node receives a RREP after it has already activated a route, it will discard the message. Instead of discarding it, this can be used as a redundant link in the event of a failure. This will save having to broadcast a Join RREQ in order to repair a broken link. That can be left as a last resort when there are no redundant links available.

Another method that can for improving the fragility of the bi-direction shared tree in MAODV is to utilize a feedback mechanism at the link layer in order to detect when the signal from a neighbour has degraded beyond a certain threshold. In this manner, one can detect possible link failures before they occur and begin route discovery of an alternate path through the network.

In this thesis we began experimenting with using a non-uniform traffic source. Although the impact was small on ODMRP, we find a significant performance hit in MAODV. This is another area that requires further investigation since multicast traffic comes in various types and the multicast protocols must be able to handle all types.

Up till now most evaluations have been done with a single multicast group. Since actual MANETs will have multiple multicast groups, evaluation needs to be done to look at the impact and interaction between multiple multicast groups in a network.

After implementing and evaluating the MAODV and ODMRP multicast protocols, we have learned some important lessons for future researchers in the area of ad hoc wireless multicast protocols. We find that although the two protocols used on-demand route discovery mechanisms, a mesh topology proves to be more robust than a tree based topology in the presence of high mobility. We also find that scalability can be an issue especially with source-oriented

protocols that utilize packet flooding. Techniques like self-pruning are relatively easy to implement and can be effective in decreasing the overhead due to flooding control packets. We also found Ns-2 to be an effective network simulator for wireless ad hoc networks. By using the existing physical, data link and IEEE 802.11 MAC layer models for wireless nodes, we were able to extend Ns-2 for implementing and evaluating multicast ad hoc protocols as well.

## BIBLIOGRAPHY

- [BFC93] Tony Ballardie, Paul Francis, and Jon Crowcroft. “Core Based Trees (CBT): An Architecture for Scalable Inter-Domain Multicast Routing”, In Proceedings of ACM SIGCOMM ‘93, San Francisco, CA, pp. 85-95, October 1993.
- [BHW97] Richard B. Bunt, Tim G. Harrison, Carey L. Williamson, Wayne L. Mackrell, “Mobile Multicast (MoM) Protocol: Multicast for Mobile Hosts”, In Proceedings of ACM Mobicom 97, Budapest, Hungary, pp. 151-160, 1997.
- [BM98] R. Bagrodia, R. Meyer, M. Takai, Y. Chen, X. Zeng, J. Martin, and H.Y. Song. “PARSEC: A Parallel Simulation Environment for Complex Systems”, IEEE Computer, vol. 31, no. 10, pp. 77-85, October 1998.
- [BR00] Christian Bettstetter, Anton Riedl, Gerhard Gebler. “Interoperation of Mobile Ipv6 and Protocol Independent Multicast Dense Mode”, Workshop on Wireless Networks and Mobile Computing (held in conjunction with ICPP2000, 29th International Conference on Parallel Processing), Toronto, Canada, pp. 531-539, August 2000.

[BGH00] Rajive Bagrodia, Mario Gerla, Julian Hsu, William Su, and Sung-Ju Lee.

“A Performance Comparison Study of Ad Hoc Wireless Multicast Protocols”, INFOCOM 2000, Tel Aviv, Israel, March 2000. Available at <http://www.cs.ucla.edu/NRL/wireless/PAPER/sj-infocom2k.ps.gz>

[BMJ98] Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu, and

Jorjeta Jetcheva. “A Performance comparison of multi-hop wireless ad hoc network routing protocols”, In Proceedings of the 4<sup>th</sup> International Conference of Mobile Computing and Networking (ACM MOBICOM ’98), pp. 85-97, October 1998. Available at <http://www.monarch.cs.cmu.edu/monarch-papers/mobicom98.ps>

[BLM98] E. Bommaiah, M. Liu, A. McAuley, and R. Talpade, “AMRoute: Ad-hoc Multicast Routing Protocol”, Internet-Draft, draft-talpade-manetamroute-00.txt,, Work in progress, Aug. 1998.

[CGL99] Ching-Chuan Chiang, Mario Gerla, and Sung-Ju Lee. “On-Demand Multicast Routing Protocol”, In Proceedings of IEEE WCNC’99 New Orleans, LA, pp. 1298-1302, September 1999. Available at <http://www.cs.ucla.edu/NRL/wireless/PAPER/odmrp-wcnc99.ps.gz>

[CG98] Ching-Chuan Chiang, Mario Gerla. "On-Demand Multicast in Mobile Wireless Networks", In Proceedings of IEEE ICNP'98, Austin, TX, pp. 260-270, October 1998. Available at <http://www.cs.ucla.edu/NRL/wireless/PAPER/icnp98-chiang.ps.gz>

[CG97] Ching-Chuan Chiang and Mario Gerla. "Routing and Multicast in Multihop, Mobile Wireless Networks", In IEEE International Conference on Universal Personal Communications (ICUPC'97), pp. 28-33, 1997. Available at [http://www.cs.ucla.edu/NRL/wireless/PAPER/Chiang\\_icupc97.ps.gz](http://www.cs.ucla.edu/NRL/wireless/PAPER/Chiang_icupc97.ps.gz)

[CGZ97] Ching-Chuan Chiang, Mario Gerla, and Lixia Zhang. "Shared Tree Wireless Network Multicast", In Proceedings of IEEE 6<sup>th</sup> International Conference on Computer Communications and Networks (ICCCN '97), September 1997. Available at [http://www.cs.ucla.edu/NRL/wireless/PAPER/Chiang\\_ic3n97.ps.gz](http://www.cs.ucla.edu/NRL/wireless/PAPER/Chiang_ic3n97.ps.gz)

[CGZ98] Ching-Chuan Chiang, Mario Gerla, and Lixia Zhang. "Adaptive Share Tree Multicast in Mobile Wireless Networks", In Proceedings of IEEE GLOBECOM'98, Sydney, Australia, pp. 1817-1822, November 1998. Available at [http://www.cs.ucla.edu/NRL/wireless/PAPER/Chiang\\_globecom98.ps.gz](http://www.cs.ucla.edu/NRL/wireless/PAPER/Chiang_globecom98.ps.gz)



- [CGZ99] Ching-Chuan Chiang, Mario Gerla, and Lixia Zhang. "Tree multicast strategies in mobile, multihop wireless networks", ACM/Baltzer Mobile Networks and Applications, vol. 4, no. 3, pp. 193-207, 1999. Available at [http://www.cs.ucla.edu/NRL/wireless/PAPER/Gerla\\_Monet.ps.gz](http://www.cs.ucla.edu/NRL/wireless/PAPER/Gerla_Monet.ps.gz)
- [CM99] S. Corson and J. Macker. "Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations", RFC 2501, IETF, January 1999. Available at <http://www.ietf.org/rfc/rfc2501.txt>
- [Chi98] Ching-Chuan Chiang. "Wireless Network Multicasting", PhD dissertation, University of California, Los Angeles, Department of Computing Science, May 1998. Available at <http://www.cs.ucla.edu/NRL/wireless/PAPER/thesis-ccchiang.ps.gz>
- [DRP00] Samir R. Das, Elizabeth M. Royer, Charles E. Perkins. "Ad Hoc On-Demand Distance Vector (AODV) Routing." IETF, Internet Draft: draft-ietf-manet-aodv-07.txt, 2000. Available at <http://www.ietf.org/internet-drafts/draft-ietf-manet-aodv-07.txt>
- [DEF96] S.E. Deering, D. Estrin, D. Farinacci, V. Jacobson, C.-G. Liu and L. Wei. "The PIM architecture for Wide-Area Multicast Routing", IEEE/ACM Transactions on Networking Vol. 4, No. 2, pp. 153-162, April 1996.

[DEF99] S. Deering, D. Estrin, D. Farinacci, V. Jacobson, "Protocol independent multicast version 2, dense mode specification", Internet Draft, 1997.

Available at <http://www.ietf.org/proceedings/99nov/I-D/draft-ietf-pim-v2-dm-03.txt>.

[FV99] Kevin Fall and Kannan Varadhan (Eds.). "Ns nodes and documentation", 1999, Available from <http://www.isi.edu/nsnam/ns/ns-documentation.html>

[GLS00] Mario Gerla, Sung-Ju Lee, William Su. "On-Demand Multicast Routing Protocol (ODMRP) for Ad Hoc Networks." IETF, Internet Draft: draft-ietf-manet-odmrp-02.txt, 2000. Available at <http://www.ietf.org/internet-drafts/draft-ietf-manet-odmrp-02.txt>.

[GM99] J.J. Gracia-Luna-Aceves and E.L. Madruga, "The Core-Assisted Mesh Protocol." IEEE Journal on Selected Areas in Communications, vol. 17, no. 8, pp. 1380-1394, Aug. 1999. Available at <http://www.cse.ucsc.edu/research/ccrg/publications/madruga.jsac.ps.gz>

[HK00] Hyojun Lim, Chongkwon Kim. "Multicast Tree Construction and Flooding in Wireless Ad Hoc Networks", ACM MSWIM'2000, Boston, MA,

pp. 61-68, August 2000. Available at  
[http://popeye.snu.ac.kr/bk21/file/2000/00\\_ic2.ps](http://popeye.snu.ac.kr/bk21/file/2000/00_ic2.ps)

[JP96] D. Johnson and C. Perkins. "Mobility support in Ipv6", In Proceedings of ACM Mobicom '96, November 1996.

[LGC99] S.-J. Lee, M. Gerla, and C.-C Chiang, "On-Demand Multicast Routing Protocol", In Proceedings of IEEE WCNC' 99, New Orleans, LA, , pp. 1298-1304, September 1999.

[LK00] Seungjoon Lee, Chongkwon Kim. "Neighbor Supporting Ad hoc Multicast Routing Protocol", MobiHoc: The First Annual Workshop on Mobile Ad Hoc Networking and Computing, pp. 37-44, August 2000.  
Available at <http://www.cs.umd.edu/~slee/nsmp-mobihoc.ps>

[Pau98] Sanjoy Paul. "Multicasting on the Internet and its Applications", Kluwer Academic Publishers, 1998.

[Per96] C. Perkins, "IP mobility support", RFC 2002, Mobile IP Networking Group, October 1996. Available at <http://www.ietf.org/rfc/rfc2002.txt>

- [RP99] Elizabeth Royer and Charles E. Perkins “Multicast Operation of the ad-hoc on-demand distance vector routing protocol”, Mobicom ’99, pp. 207-218, August 1999. Available at [http://alpha.ece.ucsb.edu/~eroyer/txt/aodv\\_Mobicom.ps](http://alpha.ece.ucsb.edu/~eroyer/txt/aodv_Mobicom.ps)
- [RP00] Elizabeth M. Royer and Charles E. Perkins. “Multicast Ad hoc On-Demand Distance Vector (MAODV) Routing.” IETF, Internet Draft: draft-ietf-manet-maodv-00.txt, 2000. Available at <http://www.ietf.org/internet-drafts/draft-ietf-manet-maodv-00.txt>
- [SSK00] Hee-Sook Shin, Young-Joo Suh, Dong-Hee Kwon. “Multicast Routing Protocol by Multicast Agent in Mobile Networks”, In Proceedings of International Conference on Parallel Processing, pp. 271-278, August 2000.
- [TO98] Gene Tsudik, Katia Obraczka. “Multicast Routing Issues in Ad Hoc Networks”, In IEEE International Conference on Universal Personal Communications, 1998. Available from <http://www.isi.edu/people/katia/mobile-mcast-routing-icupc98.ps.gz>
- [UCLA] 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/domans/glomosim.html>

[WPD88] D. Waitzman, C. Partridge, and S. Deering (editors), “Distance Vector Multicast Routing Protocol”, RFC 1075, BBN STC and Stanford University, November 1988.

[WTT98] C.W. Wu, Y.C. Tay, and C.-k. Toh, “Ad hoc Multicast Routing protocol utilizing Increasing id-numberS (AMRIS) Functional Specification”, Internet-Draft, draft-ietf-manet-amris-spec-00.txt, IETF, Work in progress, November 1998. Available from <http://www.ietf.org/proceedings/99mar/I-D/draft-ietf-manet-amris-spec-00.txt>

[XP97] G. Xylonmenos and G. Plyzos, “IP Multicast for Mobile Hosts”, IEEE Communications Magazine, pp. 54-58, January 1997. Available at <http://www.cs.ucsd.edu/groups/cs1/pubs/journal/commag97.ps>

## APPENDICES

### NS-2 CONFIGURATION FOR SUPPORTING MULTICAST AD-HOC ROUTING PROTOCOLS

#### **Configuration**

The following configuration was used for performing all the simulations described in this thesis.

Computer: Pentium III, 733 Mhz with 128 MB Ram

Operating System: Linux Mandrake 7.1

NS-2 Version: ns\_allinone-2.1b6

URL: <http://www.isi.edu/nsnam/ns>

To install ns-2 simply do a “./install” in the root directory of the downloaded installation.

#### **Network Components of a Mobile Node in NS-2**

The network stack for a mobile node consists of a Link Layer, an ARP module connected to the Link Layer, and Interface Priority Queue, a MAC layer, a Network Interface, all connected to the Wireless Channel. These network components are created and put together in Otcl, and was provided by work done in CMU for supporting wireless in Ns-2.

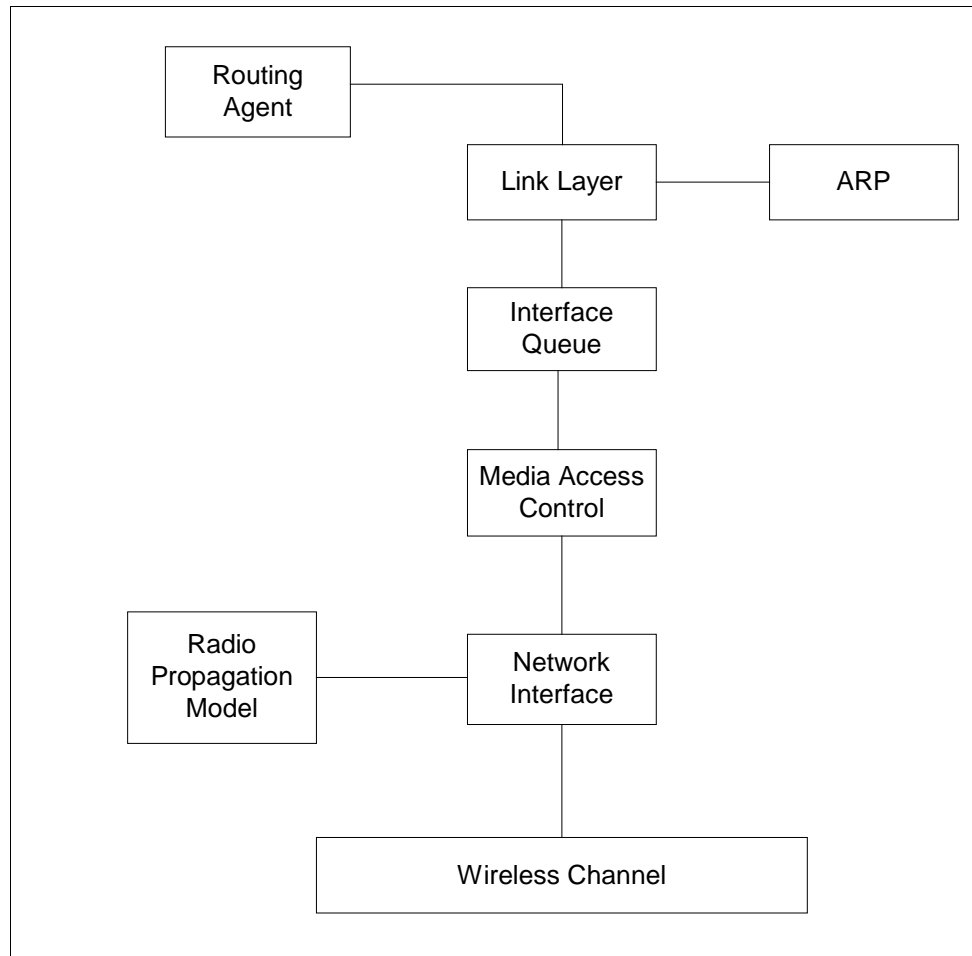


Figure 35 NS-2 Mobile Node Network Components

*Link Layer:*

The link layer is responsible for simulating the data link protocols. Many protocols can be implemented within this layer such as packet fragmentation and reassembly, and reliable link protocol. For a mobile node, the difference in the link layer is that it has an ARP module connected to it which resolves all IP to hardware (Mac) address conversions.

The packet flow through the Link Layer is as follows:

For all outgoing (into the channel) packets, they are handed down to the Link Layer by the Routing Agent. The Link Layer then hands down packets to the Interface Queue. For all incoming packets (out of the channel), the MAC layer hands up packets to the Link Layer, which then hands it off to the entry point for the Routing Agent.

Location: ~ns/ll.cc,h

### *ARP*

The Address Resolution Protocol module receives queries from the Link Layer. If ARP has the hardware address for destination, it writes it into the Mac header of the packet. Otherwise it broadcasts an ARP query, and caches the packet temporarily. For each unknown destination hardware address, there is a buffer for a single packet. If additional packets to the same destination are sent to ARP, the earlier buffered packet is dropped. Once the hardware address of the packet's next hop is known, the packet is inserted into the interface queue.

Location: ~ns/arp.cc,h



### *Interface Queue:*

The interface queue is implemented as a priority queue, which gives priority to routing protocol packets, inserting them at the head of the queue. It supports running a filter over all packets in the queue and removes those with a specified destination address.

Location: `~ns/priqueue.cc,h`

### *MAC Layer:*

The IEEE 802.11 distributed coordination function (DCF) Mac protocol has been implemented by CMU. It sends a RTS/CTS/DATA/ACK pattern for all unicast packets and simply sends out DATA for all broadcast packets. The implementation uses both physical and virtual carrier sense.

Location: `~ns/mac-802_11.cc,h`

### *Network Interfaces:*

The Network Interface layer serves as a hardware interface that is used by mobile node to access the channel. The wireless shared media interface is implemented as class `Phy/WirelessPhy`. This interface is subject to collisions and the radio propagation model receives packets transmitted by other node interfaces to the channel. The interface stamps each transmitted packet with the meta-data related

to the transmitting interface link the transmission power, wavelength. This meta-data in the packet header is used by the propagation model on the receiving network interface to determine if the packet has minimum power to be received and/or captured and/or detected (carrier sense) by the receiving node. The model approximates the DSSS radio interface (Lucent WaveLan direct-sequence spread-spectrum).

Location: `~ns/wireless-phy.cc,h`

*Radio Propagation Model:*

It uses Friss-space attenuation ( $1/(r^2)$ ) at near distances and an approximation to Two ray Ground ( $1/r^4$ ) at far distances. The approximation assumes specular reflection off a flat ground plane.

Location: `~ns/tworayground.cc,h`

*Antenna:*

An omni-directional antenna with unity gain is used by mobile nodes.

Location: `~ns/antenna.cc,h`

### *Wireless Channel:*

The wireless channel duplicates packets to all mobile nodes attached to the channel except the source itself. It is the receiver's (routing agent) responsibility to decide if it can receive the packet

## **Multicast Ad Hoc Networking Support in NS-2**

Up till now there had been no previous implementation of multicast ad-hoc routing protocols in the NS-2 simulation environment. However, since support was already available for unicast ad-hoc routing protocols from the work done at CMU, a framework was available for adding support for multicast ad-hoc routing protocols.

### **Multicast Group Addressing**

In NS-2, nodes are identified by an `ns_addr_t` type (basically a 32 bit integer), and unicast communication is done using this address type. For multicast communication, we set aside the upper space within the address space to be used for multicast communication, just like in IP addresses. Thus when a routing agent receives a packet with the destination address set to a value within the multicast address space, it passes it to the multicast routing functionality for processing or appropriate forwarding.

## Multicast Group Join and Leave

The multicast group join and leave, is implemented via tcl commands specified in separate configuration files. Thus, at specific times during the simulation, a user can specify when a particular mobile node is going to join or leave a particular multicast group. The join / leave command is implemented in the command handler of the multicast routing agent.

In the configuration tcl file the following is specified:

e.g.

```
$ns at <time in seconds> "$node $protocol-join-group <multicast address>"
```

or

```
$ns at <time in seconds> "$node $protocol-leave-group <multicast address>"
```

## Step-by-Step Changes to NS-2 for Adding a New Multicast Protocol

### 1. Add Creation of new Wireless Node Routing Agent

File: ~/ns-2/tcl/lib/ns-lib.tcl

Tcl procedure: Simulator instproc create-wireless-node

- add call to new tcl procedure for creation of the new protocol agent

e.g.

```

switch -exact $routingAgent_ {

AODV {

    set ragent [$self create-aodv-agent $node]

}

ODMRP {

    set ragent [$self create-odmrp-agent $node]

}

}

```

## 2. Implement create tcl procedure for new protocol agent

```

Simulator instproc create-mynew-agent { node } {

}

```

e.g.

```

Simulator instproc create-odmrp-agent { node } {

    set ragent [new Agent/ODMRP [$node id]]

    $self at 0.0 "$ragent start"      ;

    $node set ragent_ $ragent

    return $ragent

}

```

## 3. Hook into Agent class

File: ~/ns-2/tcl/lib/ns-agent.tcl

- Add instantiate OTcl methods for the Agent base class

e.g.

```
Agent/AODV instproc init args {
```

```
    $self next $args
```

```
}
```

```
Agent/AODV set sport_ 0
```

```
Agent/AODV set dport_ 0
```

```
Agent/ODMRP instproc init args {
```

```
    $self next $args
```

```
}
```

```
Agent/ODMRP set sport_ 0
```

```
Agent/ODMRP set dport_ 0
```

#### 4. Define Packet Types

File: ~/ns-2/tcl/lib/ns-packet.tcl

- set up the packet format for the simulation

e.g.

```
{ AODV off_AODV_ }
```

```
{ ODMRP off_ODMRP_ }
```

File: ~/ns-2/packet.h

e.g.

PT\_AODV

PT\_ODMRP

## 5. Add Trace support

File: ~/ns-2/cmu-trace.cc

- Add procedures for formatting trace of control packets for new routing agent

e.g.

```
void    format_aodv(Packet *p, int offset);  
void    format_odmrp(Packet *p, int offset);
```

## 6. Implement the new Routing Agent

### **MAODV Implementation**

Location: ~/ns/aodv

Files: aodv.cc, aodv.h, aodv-mcast.cc, aodv-packet.h

The MAODV implementation was based on the existing unicast AODV implementation provided in the release. The AODV routing agent inherits from

the standard Agent class. The following is a brief summary of the changes that were made:

- De-Multiplex between multicast control messages versus unicast control messages.
- Addition of a multicast routing table
- Addition of commands for join and leaving a multicast group
- Changes to control messages for multicast support per the draft
- Addition of support for multicast in AODV
- Recognition of multicast data packets and forwarding based on knowledge from the multicast routing table

### **ODMRP Implementation**

Location: ~ns/odmrp

Files: odmrp-packet.h, odmrp.cc, odmrp.h

The ODMRP implementation was developed from scratch since none was available. The ODMRP routing agent had to be newly defined in ns-2 so that it could be instantiated appropriately. The ODMRP routing agent inherits from the standard Agent class. As well the ODMRP control packet types had to be newly defined in order to send and receive control packets.



## Mobile Node Definition

In order to define a mobile node in ns-2 with various properties:

`$ns_ node-config - adhocRouting (ODMRP /AODV)`

- `llType LL`
- `ifqType Queue/DropTail/PriQueue`
- `antType Antenna/OmniAntenna`
- `propType Propagation/TwoRayGround`
- `phyType Phy/WirelessPhy`
- `channelType Channel/WirelessChannel`
- `topoInstance $topo`
- `agentTrace ON`
- `routerTrace on`
- `macTrace on`

## Trace Support

The trace support for wireless simulations is currently implemented in a `cmu-trace` object `CMUTrace` that derives from the base class `Trace`. The implementation of the trace functionality is in the file “`cmu-trace.cc`”. Changes were made to this file in order to recognize EXP traffic, as well as the control packets in MAODV and ODMRP.

## Mobile Node Movement

Scenario files were generated using the “setdest” scenario generation program for creating node movement files. It comes as part of the ns-2 installation under the following directory:

```
~ns/indep-utils/cmu-scen-gen/setdest
```

The command line to generate a scenario file is as follows:

```
./setdest -n <num of nodes> -p <pausetime> -s <max speed> -t <simulation  
time> -x <max x axis> -y <max y axis>
```

## Traffic Pattern / Connection Files

The connection/traffic files need to be created which specify the different traffic sources as well as when a particular mobile node joins or leaves a multicast group.

- Definition of CBR Traffic Sources

e.g.

```
set udp_(1) [new Agent/UDP]  
  
$udp_(1) set dst_addr_ 0xE000000  
  
$ns_ attach-agent $node_(0) $udp_(1)  
  
set cbr_(1) [new Application/Traffic/CBR]  
  
$cbr_(1) set packetSize_ 512
```

```

$cbr_(1) set interval_ 0.25

$cbr_(1) set random_ 1

$cbr_(1) set maxpkts_ 100000

$cbr_(1) set dst_ 0xE0000000

$cbr_(1) attach-agent $udp_(1)

$ns_ at 30.00000 "$cbr_(1) start"

```

#### - Definition of EXP Traffic Sources

e.g.

```

set udp_(1) [new Agent/UDP]

$udp_(1) set dst_addr_ 0xE0000000

$ns_ attach-agent $node_(0) $udp_(1)

set exp_(1) [new Application/Traffic/Exponential]

$exp_(1) set packetSize_ 512

$exp_(1) set burst_time_ 500ms

$exp_(1) set idle_time_ 500ms

$exp_(1) set rate_ 36k

$exp_(1) set dst_ 0xE0000000

$exp_(1) attach-agent $udp_(1)

$ns_ at 10.00000 "$exp_(1) start"

```

#### - Join Multicast Group

e.g.

```
$ns_ at 1.0000000000000000 "$node_(0) aadv-join-group 0xE000000"
```

```
$ns_ at 1.0000000000000000 "$node_(0) odmrp-join-group 0xE000000"
```

