PRO-ACTIVE CONNECTION MAINTENANCE IN AODV AND MAODV

by

Yufang Zhu

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

Master of Science
in Information and Systems Science

Department of Systems and Computer Engineering
Carleton University
Ottawa, Ontario
CANADA, K1S 5B6

August 2002

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

PRO-ACTIVE CONNECTION MAINTENANCE IN AODV AND MAODV

Submitted by **Yufang Zhu**, M.Sc.
in partial fulfillment of the requirements for
the degree of M. Sc. in Information & Systems Science

Thesis Supervisor

CHAIR, Department of Systems and Computer Engineering

Carleton University
August 2002

ABSTRACT

A major aspect of ad-hoc networks is that the nodes can move randomly, which requires the routing protocols in ad-hoc network to quickly respond to the network topology change in order to guarantee successful data packet delivery. A link state prediction method can predict the exact link breakage time of an active link before the breakage actually occurs. So by using link state prediction, a new route can be constructed before the old route becomes unavailable, thus avoiding data packet loss. In this thesis, we first added the link state prediction method to the reactive unicast protocol AODV. The source can smoothly update the currently used route to avoid any soon-to-be-broken link. Simulation results demonstrate that this pro-active route maintenance can significantly reduce packet loss (between 32% and 72%) with slight overhead increase (between 4% and 49%). We also examine the link state prediction method in the tree-based multicast protocol MAODV to maintain the multicast tree in advance and avoid any branch breakage. Simulation results show the throughput is greatly improved to above 85% from around 70% with overhead increase below 12%.

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Professor Thomas Kunz, for his comprehensive guidance for this thesis. His insight comments help me clarify my ideas and enlarge my visions.

I also appreciate the understanding and encouragement from my husband Shu and my family. They always be my strong and reliable support.

Table of Contents

List of Figures

List of Tables

List of Acronyms

ABR Associativity Based Routing

ACK Acknowledgement

AMRIS Adhoc Multicast Routing protocol utilizing Increasing Id numbers

AMRoute Adhoc Multicast Routing

AODV Ad-hoc On-demand Distance Vector Routing

ARP Address Resolution Protocol

CAMP Core-Assisted Mesh Protocol

CBT Core Based Tree

CBR Constant Bit Rate

CGSR Cluster Gateway Switching Routing

CSMA/CA Carrier Sense Multiple Access with Collision Avoidance

DAG Directed Acyclic Graph

DSDV Destination-Sequenced Distance-Vector routing

DSR Dynamic Source Routing

FORP Flow Oriented Routing Protocol

GPS Global Position System

GRPH Group Hello

IP Internet Protocol

LCC Least Cluster Change

LET Link Expiration Time

LL Link Layer

LMR Lightweight Mobile Routing

MAC Media Access Control

MACT Multicast Route Activation

MANET Mobile Ad Hoc Network

MAODV Multicast Adhoc On-Demand Vector routing protocol

ODMRP On-Demand Multicast Routing Protocol

RET Route Expiration Time

RREQ Route Request

RREP Route Reply

RTS/CTS Request To Send/Clear To Send

SSA Signal Stability-based Adaptive Routing

TCP Transmission Control Protocol

TORA Temporally Ordered Routing Algorithm

TTL Time To Live

Chapter 1 Introduction

A Mobile Ad-hoc Network (MANET) [21] is an autonomous system of mobile hosts connected by wireless links with no supporting fixed infrastructure or central administration. Due to the limited radio propagation, if two hosts are not within direct wireless transmission range of each other, the communication between them must pass through one or more other hosts. So a MANET is a multi-hop network, the hosts in which may serve as routers. These hosts are free to move randomly, which introduces several features for a MANET: (1) the network topology may change frequently and unpredictably; (2) the hosts only have limited battery power and need to contend for constrained bandwidth; (3) the wireless links between hosts may have variable capacity and the link direction may be bi-directional or unidirectional. All these characteristics make the routing in a MANET very challenging with a diverse set of performance issues [22].

Many different unicast routing protocols have been proposed with the goal to dynamically and efficiently create and maintain routes between two communicating hosts. Basically, these routing protocols can be classified into two categories: proactive (also called table-driven) and reactive (also called on-demand). Proactive protocols try to keep up-to-date routes between any host pairs in the network. Each host can have available routes to any other host at any time even if it may never use the routes to some destinations. To maintain such routes, the network suffers from substantial periodic routing-update control messages, which can waste the limited bandwidth. But the latency for data transmission may be minimal, as the route is always available before data transmission. DSDV [30] is an example of a proactive

protocol. In contrast, the reactive protocols only maintain the currently used routes between the host pairs. The route should be only created when a source needs to send data packets to a destination. So the routing-update control overhead is far less, thus increasing bandwidth utilization. But the source must wait for route discovery before sending data packets, which increases latency. DSR [15] and AODV [31] are examples of reactive protocols.

A MANET may operate in isolation, or may be connected to a fixed network as a stub. It is very useful in areas where the communication infrastructure is unavailable, or rapid deployment and dynamic reconfiguration is necessary. Examples include critical applications such as in military battlefields and civilian emergency disaster relief; and industrial, commercial, or educational applications involving cooperative mobile data exchange such as at conventions and in classrooms. Most of the applications are group-oriented, and multicasting can make the network hosts work in groups to carry out a given task, so multicasting is natural and typical in a MANET environment.

Multicasting in a MANET is more challenging in that all the group members keep moving, making reliable and efficient packet delivery to all members more difficult. Up to now, MANET multicast protocols can be basically divided into tree-based or mesh-based protocols according to how the data packet is delivered. Tree-based protocols propagate data over a spanning tree connecting all multicast group members, while mesh-based protocols forward data to all group members over a subset of the network. AMRoute [2] and MAODV [35] are examples of tree-based protocols. Flooding is the simplest mesh-based protocol, that is, when

receiving a non-duplicate data packet, every host re-broadcasts it to its neighbors in its transmission range. ODMRP [19] and CAMP [10] are other examples for mesh-based protocols. Like unicast routing protocols, multicast protocols can also be classified into proactive or reactive protocols. In tree-based protocols, AMRoute is proactive while MAODV is reactive. In mesh-based protocols, CAMP is proactive while ODMRP is reactive.

1.1 Motivation

In a MANET, mobility causes the network topology to change arbitrarily and frequently, but the scarcity of bandwidth does not allow substantial control message overhead for updating the topology change. Therefore, the reactive routing protocols seem to be more desirable than proactive strategies in such an environment. However, data packets can be lost in the middle of the routes constructed by reactive protocols, if an intermediate link on the route becomes broken as the result of the intermediate node moving out of range or suddenly switching off. This phenomenon has been described in [12] for TCP connections based on DSR. The loss of data packets may become even worse in multicast communications, as the multicast data packets is to be delivered to more than one receiver.

To improve routing reliability, several unicast routing protocols have been presented based on choosing routes composed of more reliable wireless links rather than those of shortest hops. ABR [40] and SSA [8] are examples. These protocols measure the link reliability based on past and current information on the link states.

However, as nodes move frequently, future link breakages cannot be avoided even for these more stable links. Therefore, to predict node mobility and the future states of the currently used wireless links is a reasonable approach for improving link availability and routing reliability. Several models [24] [11] [33] have been presented for measuring link availability or predicting link states, and protocols such as FORP [38], DSR [33], and ODMRP [18], have been proposed or enhanced with node mobility and link state prediction.

1.2 Research Overview and Contributions

This thesis concentrates on how to improve the reliability of routes in order to achieve better performance of the unicast protocol AODV and its multicast extension MAODV. We choose AODV and MADOV because AODV is a popular reactive routing protocol and also it suggests multicasting with MAODV, so that we can evaluate the improvement not only for unicasting but also for multicasting. Through studying related unicast and multicast protocols, and investigating the current methods for improving link reliability, the method for predicting link states used in [33] for DSR is selected to be implemented in AODV and MAODV, with the aim to detect link breakage and construct a new route in advance, thus reducing the packet loss.

This thesis provides the following contributions:

1. Added the link state prediction into the standard AODV protocol, and modified the standard implementation of AODV in the simulator NS2.
2. Improved the implementation of MAODV in NS2 [4] according to the

MAODV specification.

3. Added the link state prediction into the standard MAODV protocol, and implemented MAODV with prediction in NS2.
4. Compared AODV with prediction to standard AODV, and MAODV with prediction to standard MADOV.

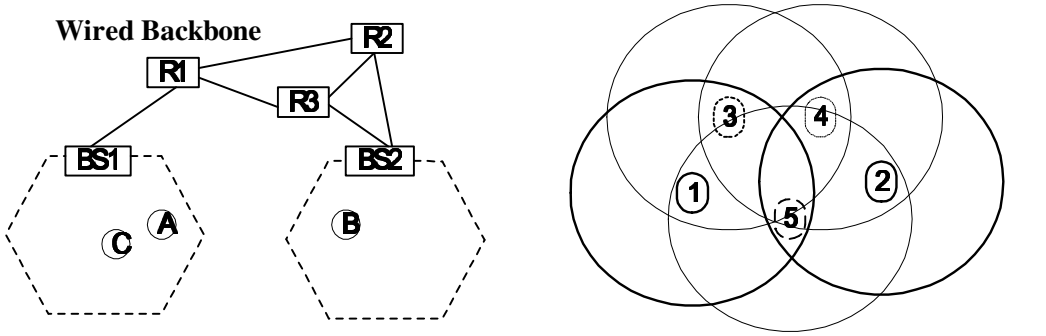
1.3 Organization of Thesis

This thesis begins with a brief introduction of basic MANET unicast and multicast protocols, and surveys the current research about link reliability and link state prediction in Chapter 2. With the aim to enhance AODV and MAODV with link state prediction, first AODV and MAODV are described and evaluated in Chapter 3 using the simulator NS2. Chapter 4 and Chapter 5 discuss the implementations and simulation analysis of AODV with prediction and MAODV with prediction. The comparison with standard AODV and MAODV are also presented. The thesis ends with conclusions and suggestions for future research in Chapter 6.

Chapter 2 Overview of Basic MANET Protocols

Wireless networks allow for more flexible communication since the nodes are not limited to a fixed physical location. There are two categories of mobile wireless networks: infrastructure networks; and infrastructureless networks.

Infrastructure networks, or cellular networks, consist of stationary base stations and mobile endpoints. Base stations are fixed and connected to the wired backbone, acting as gateways between mobile endpoints and the wired backbone. A mobile endpoint, or mobile end host, in the area of direct wireless transmission range covered by at least one of the base stations, communicates directly and only with the base station to exchange information with other fixed and mobile end hosts. Thus, wireless communication in such cellular networks is a single-hop communication. Figure 1(a) illustrates an example of an infrastructure network, in which BS means Base Station and R means backbone Router. A route between mobile end hosts A and B may be A-BS1-R1-R3-BS2-B, while the route between mobile end hosts A and C is A-BS1-C.



(a) Simple Example of Cellular Network (b) Ad-hoc Network with Transmission Range

Figure 1: Cellular Network and Ad-hoc Network

An infrastructureless network, or a MANET, consists of only mobile nodes,

with no base stations and wired backbone in charge of information exchange and network administration. Each mobile node not only operates as a host but also as a router, responsible for forwarding packets for other mobile nodes in the network that may not be within direct wireless transmission range of each other. Wireless communication in such a network is a multi-hop communication. Figure 1(b) gives an example of a MANET. The circle centered on a node number represents the transmission range of that node. Possible routes between node 1 and node 2 are 1-3-4-2 or 1-5-2.

Wireless links have significantly lower capacity than their wired counterparts. After the effects of multiple access, fading, noise, interference, etc., the capacity of a wireless link may be variable and the link direction may be unidirectional. In such an environment, congestion is prone to happen. Besides that, node mobility also challenges the multi-hop communication in a MANET. Typically, nodes in a MANET rely on battery with limited power during moving, and the network topology may change frequently, rapidly and unpredictably. All these features cause the routes between the communication pairs to fail easily, resulting in frequent route updates.

Traditional unicast and multicast routing protocols in wired networks rely on a static or quasi-static network topology and substantial control overhead exchange, which make them inapplicable in a MANET. Protocols proposed for cellular networks, such as Mobile IP [29], only consider the single-hop wireless case, in which the routing information and administration mainly depend on the stable wired backbone. Therefore, a MANET needs its own unicast and multicast routing

protocols. In this chapter, the basic unicast and multicast protocols for MANET are briefly described in Sections 2.1 and 2.2; related research on route reliability is presented in Section 2.3; and in Section 2.4, the approach pursued in this thesis is introduced.

2.1 Unicast Routing Protocols in MANET

Unicast routing protocols try to accomplish one-to-one communication in a network. Basically, as in Figure 2 [37], unicast ad hoc routing protocols can be generally summarized in two categories: proactive (also called table-driven); and reactive (also called on-demand).

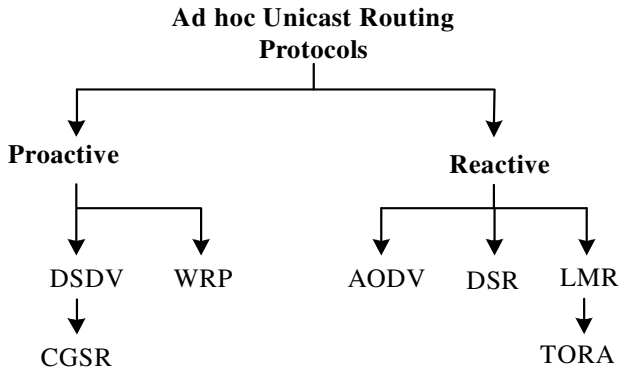


Figure 2: Categories of Ad hoc Unicast Routing Protocols

2.1.1 Proactive Protocols

The proactive routing protocols attempt to keep up-to-date routing information between any pair of mobile nodes. Routing-update messages are propagated throughout the whole network to get a consistent view of the network topology.

DSDV (Destination-Sequenced Distance-Vector Routing) [30] is a distance vector routing protocol based on the classical Bellman- Ford routing algorithm [16], which requires each node in the network to broadcast routing-update messages

periodically to update the routing table in which routes to all the possible destinations are recorded. The key design of DSDV is that, in addition to the routing table, each node also has a monotonically increasing even sequence number, which increments whenever a new routing-update message is sent out, thus letting other nodes know which routing information is fresher, avoiding routing loops. So in a routing table, in addition to the information about the destination node address, the hop count to the destination, and the next hop to that destination, the currently known largest sequence number of the destination is also contained.

CGSR (Cluster-Gateway Switching Routing) [6] uses DSDV as the underlying routing scheme, and modifies DSDV by using a hierarchical architecture and cluster-head-to-gateway routing. The mobile nodes form clusters by selecting one node as the cluster header and all other nodes in that cluster are in the transmission range of the cluster head. A gateway node is a node within the transmission range of two or more cluster heads. When a source generates data packets, it transmits the packet to its cluster head. If the destination is not in the same cluster, the head forwards the packets to the gateway node, thus into another cluster. This cluster-head-to-gateway step continues till the destination is reached. The LLC (Least Cluster Change) algorithm is used for keeping the cluster head unchanged as long as possible. Each node has a routing table that lists routes to other cluster heads. To map a destination node address to the destination cluster head address, an additional cluster member table is also included. In CGSR, although the routing table is smaller, the overhead of periodic broadcasting for maintaining the routing table and the cluster member table is as heavy as in DSDV.

WRP (Wireless Routing Protocol) [25] is another proactive distance vector routing protocol. Its main design is that each mobile node in the network keeps four tables (routing table; distance table; link cost table; and message retransmission list table) and broadcasts the information in the four tables periodically. The routing table contains the distance of each destination from the node, and the predecessor and the successor of the node on the route, with a tag to identify whether this route is a simple path, a loop or invalid. The distance table contains the distance of each destination via each neighbor of the node, and for the combination of each destination and each neighbor, the successor of that neighbor is also kept for accomplishing the route. The link cost table lists the cost of links to each neighbor and the number of timeouts since the last error-free routing-updates were received from that neighbor. The message transmission list table maintains information to trace the neighbors who have not acknowledged its routing-update message. The four tables together guarantee the routes to be optimal and fresh, accomplish fast routing convergence, and eliminate loops.

2.1.2 Reactive Protocols

Reactive routing creates routes only when desired by the source node. The route discovery follows a Request-Reply cycle and starts only on demand, that is, when a node requires a route to the destination and finds no existing route. In such a situation, the node initiates a route discovery process by broadcasting a Route Requests. This process is complete once one or more routes to the destination are found as Route Replies propagate back to the source. After the route is created, it is

maintained and updated till the destination is no longer accessible by any possible route or the source no longer needs that route.

This section introduces typical reactive unicast routing protocols: DSR, LMR and TORA. AODV is another typical protocol, but it will be described in Section 3.1, as it is the focus of this thesis.

DSR (Dynamic Source Routing) [15] is a reactive unicast protocol implementing source routing. Every node in the network maintains a route cache containing the complete and ordered list of nodes through which the packet must pass through to reach the destination. As the hop sequence is known to the source, any loop in routing can be excluded, and the routing decision is determined when sending out data packets. So data packets are appended with the same complete hop sequence in the packet header, intermediate nodes just forward the packet to the next hop along the hop sequence. Route discovery starts only on demand by broadcasting a new Route Request message tagged with a unique Request ID set by the source. The Request ID, with the source node address, helps nodes to be aware of and discard any duplicate Route Requests. When receiving a non-duplicate Route Request, if the node is neither the destination nor a node with a valid route to the destination, it appends its own address into the message and re-broadcasts it to its neighbors; otherwise, the node can send back a Route Reply with a complete and ordered list of intermediate nodes from the source to the destination. During propagation of the Route Reply back to the source, any intermediate node and the source can get the hop sequence, the complete route to the destination, and record it in one's route cache. No periodic routing-update messages are used in DSR. The

route is used till some link on that hop sequence breaks. The link breakage is detected by using a wireless MAC layer retransmission and acknowledgement mechanism or passive acknowledgements as described in [23]. Once a link breakage occurs at an intermediate node, the node sends a Route Error message back to the source node. Along the traverse of the Route Error, the broken link and the links after it are removed from any route cache that contains this hop. The source also removes any route containing that broken link. If the source still wants to send data packets to that destination, a new route discovery process is initiated; otherwise, there is no need to discover a new route. DSR also proposes several optimization options such as: (1) salvaging used for repairing a disconnected route locally; (2) promiscuous listening used for finding smaller hop-count route; and (3) piggybacking the bad link on its next Route Request, which can help remove the broken link in the caches of other nodes, and avoid other nodes generating Route Replies containing the bad link.

LMR (Lightweight Mobile Routing) [7] and TORA (Temporally-Ordered Routing Algorithm) [28] are on-demand unicast protocols based on the idea of the Gafni-Bertsekas (GB) [9] algorithm, which constructs a destination-oriented Directed Acyclic Graph (DAG), the multi-path to the desired destination. The DAG is rooted at the destination, so packets sent by the source travel along the route from upstream neighbor to downstream neighbor until the destination is reached. Only the destination has no downstream links, thus avoiding forming any loop. Both protocols have three similar functionalities: Route Construction; Route Maintenance; and Route Destruction. The Route Creation process is to construct

the DAG. A source assumes that it has a route to the destination as long as it has at least one downstream neighbor. Only when the node loses the last route to the desired destination, Route Maintenance is triggered if that node still needs a route to the destination. Route Destruction is used to erase invalid routes in the network. In LMR, the DAG is implied by the direction state of each network link, presented in a node's link state table. The disadvantage of LMR is that there is no time bound for terminating the search for a new path when a network partition occurs or when the destination permanently leaves the network. In TORA, nodes use a "height" metric to establish the DAG. The "height" is an ordered quintuple and the upstream or downstream direction of the link is assigned based on the relative height metric of the neighboring nodes. During routing, a node may only route data packets to a node with "lower height". This "height" metric can effectively detect a network partition, making TORA more favorable than LMR. This multi-path routing decouples the generation of control message overhead from the rate of the change of the network topology, and it also can alleviate congestion by using other available routes when one route suffers congestion.

2.2 Multicast Protocols in MANET

Multicasting plays an important role for communication in a MANET, where group tasks are often deployed. By sending the same data to multiple recipients, multicasting can reduce the consumptions of network bandwidth and host power.

For multicasting, a multicast group is constructed with one or more group members, which should receive and handle any information sent to that group. A

unique multicast identifier, namely its multicast address, is assigned to each group. At any time, each node may join a multicast group, and each group member may leave the multicast group. In a MANET, the group members randomly spread and frequently move in the whole network, which causes more difficulty in packet delivery and group maintenance. Many multicast protocols have been proposed for MANET, but here we only introduce basic MANET-inspired multicast protocols, which are summarized in Figure 3.

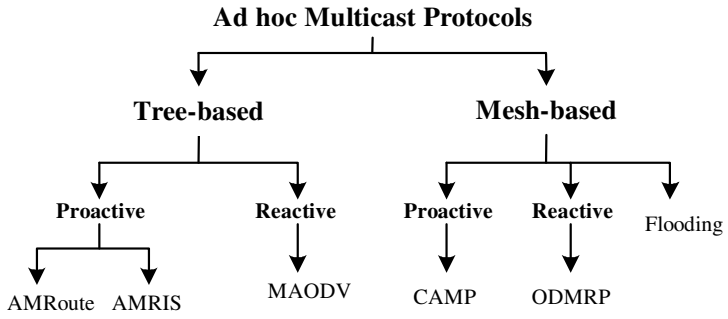


Figure 3: Categories of Ad-hoc Multicast Protocols

2.2.1 Tree-based Protocols

Tree-based protocols construct a tree structure to deliver data packets for one multicast group. There always is a core or leader on the tree, only responsible for maintaining the tree structure. This differs from the core in protocols like CBT [1], in which data packets are sent first to the core and then distributed from the core. MAODV is a reactive tree-based protocol. These multicast extensions of the unicast protocol AODV will be described in Section 3.2, for MAODV is the focus of this thesis.

AMRoute (Ad-hoc Multicast Routing) [2] is a proactive tree-based multicast protocol, using unicast tunnels to connect multicast group member pairs. There is at

least one core in each group, and initially each group member declares itself as a core. Each core periodically broadcasts Join-Reqs to discover other disjoint partitions for the group. When the Join-Req reaches a member in a different partition, that node responds with a Join-Ack and marks that node as its neighbor. The node that receives a Join-Ack also marks the sender of the Join-Ack message as its neighbor. Therefore, a mesh of tunnels is created between a pair of group members. A node wishing to leave the group sends Join-NAK to its neighbors and does not forward any data packets for the group. While a mesh is used for connecting group members, the data exchange in AMRoute is tree based. Once the mesh is created, the core periodically transmits Tree-Creates to its group neighbors through tunneling to build a shared tree. When a neighbor receives a non-duplicate Tree-Create, it forwards the message to all other neighbors. If a duplicate Tree-Create is received, a Tree-Create-NAK is sent back along the incoming tunnel. Then the node receiving Tree-Create-NAK marks the tunnel as not to be used for data transfer. Thus, a tree structure is established for data transfer by using a subset of the mesh structure. In addition, core nodes can use the reception of Tree-Create from other cores to decide whether to remain as a core. The key characteristic of AMRoute is the usage of mesh tunnels to establish the multicast tree. Therefore, as long as routes between tree members exist via the mesh, the tree can be formed even when the network topology changes. Also non-members need not support any multicast protocol. But AMRoute depends heavily on an underlying unicast protocol for keeping the tunnels among group members, although any unicast protocol can be used. Also, loops may be formed with several tunnels in AMRoute.

AMRIS [42] is another proactive tree-based protocol. Its key idea is that each node is tagged with a multicast session member ID (msm-ID), which provides a logical height and builds a DAG rooted from the Sid. Sid is a special node (usually a sender) that broadcasts a New-Session message including the Sid's msm-ID when a new multicast session begins. Neighbors, upon receiving a non-duplicate New-Session message, make their own msm-IDs larger than the one specified in the message, then rebroadcast the New-Session message with their own msm-IDs. Thus the msm-IDs increase as they radiate from the Sid, and except the Sid, every other node can have a potential parent whose msm-ID is smaller than its msm-ID. A node joins the session by sending a unicast Join-Req, traveling along the route to corresponding parents with smaller and smaller msm-IDs. If a group member is met, the member sends back a Join-Ack, so a registered parent/child relationship is created and a branch is grafted. If no Join-Ack is received, the node then broadcasts Join-Req searching for other potential parents. Link disconnection in AMRIS is detected by a multicast beaconing mechanism like neighbor HELLO messages in AODV (discussed in Section 3.1). After a link breakage occurs, a Join-Req is sent to potential parents. As nodes can only have at most one registered parent, the msm-IDs together establish a tree structure, and data packets are forwarded along tree paths. As AMRIS maintains the relationship between nodes of the whole network, its performance could suffer when traffic load or mobility rate increases.

2.2.2 Mesh-based Protocols

Mesh-based protocols provide route redundancy, as there may be more than

one route between any group member pairs. Flooding is a special example in that all nodes simply broadcast non-duplicate received data packets without any group structure construction and control overhead, but it evokes unnecessary data delivery as all nodes participate in forwarding data. Therefore, constructing a mesh with adequate size is the aim of mesh-based protocols.

Core-Assisted Mesh Protocol (CAMP) [10] is a proactive mesh-based multicast protocol. Group members construct a mesh of that group by sending Join Requests to a set of cores. The cores are used only for limiting the Join Request traffic and may not be part of the mesh. For each multicast group, there may be more than one core, and nodes can join a group even if all associated cores are unreachable. A node wishing to join a multicast group first determines if it has neighbors that are already mesh members. If so, the node announces its membership via a CAMP Update. Otherwise, the node either propagates a Join Request towards one of the cores, or attempts to reach a node in the mesh by broadcasting the requests. CAMP defines two types of members in the mesh: duplex member or simplex member. A duplex member can send and receive multicast data packets, while a simplex member can only send out data packets. So only duplex members can respond with a Join Ack, propagated back to the source of the request. CAMP maintains a mesh containing the shortest paths from each source to each receiver by periodically reviewing its packet cache to find the packets that arrive from nodes not on the current reverse shortest path given by the unicast routing table. If so, a heartbeat message is sent to the source along the new shortest path. If any node on the path is not a member of the mesh, then a push join

message forces the nodes to become mesh members. Therefore, CAMP must rely on an underlying unicast routing protocol like WRP [25], which guarantees correct distances to all the destinations within finite time.

ODMRP (On-demand Multicast Routing Protocol) [19] is a reactive mesh-based multicast protocol. It uses the forwarding group, proposed in [5], to construct the routes between any member pairs. When a source has packets to send to a group whose routing information is not maintained, it broadcasts a Join-Query message, piggybacking the data payload. Join-Queries are sent out periodically while the node keeps sending data packets. When a node receives a non-duplicate Join-Query, it establishes the reverse route to the source and rebroadcasts Join-Query again till a multicast receiver (a group member) is reached. The multicast receiver will then create and broadcast a Join-Reply message, which contains currently known routes to the sender, including the next hop of each route. When a neighbor receives a non-duplicate Join-Reply, it checks if it is on a route to the source by matching its own ID with the next hop recorded in any entry in that Join-Reply. If it does, it becomes part of the forwarding group, and broadcasts its own Join-Reply to the matched entries. Thus, Join-Reply is propagated back to the source along any possible route to that group member, building a mesh formed by the forwarding group. For ODMRP, no explicit control message needs to be sent for joining or leaving the group. As Join-Query is periodically broadcast by the source, starting or stopping sending Join-Query will automatically register or terminate the source's relationship with the group. The receiver not sending back Join-Reply implies it no longer wants to receive multicast data packets, thus ceases

to being a group member. The forwarding nodes do not forward packets to a timed-out route, so the mesh structure keeps updated. The main disadvantage of ODMRP is the excessive overhead for the periodic flooding of the Join-Query and Join-Reply messages.

2.3 Route Reliability and Link State Prediction

Several papers [37] [3] [20] [27] have evaluated the unicast and multicast protocols described above. For unicast protocols, simulation shows that reactive protocols have better performance than proactive protocols when node mobility rate increases, since they reduce the routing overhead and react quickly to topology changes. DSR and AODV are the two representatives. For multicast protocols, reactive protocols MAODV and ODMRP are more adaptive to mobility. As mobility is the key characteristic in a MANET, reactive protocols for MANET should be studied further.

Mobility makes communication more flexible, as nodes do not have to be limited to a fixed location, but it also brings the necessity to frequently maintain routes and group structure when a link on an active route becomes broken, especially when using reactive unicast protocols or tree-based multicast protocols. Therefore, improving the route reliability is a reasonable approach to improving the packet delivery throughput.

The first approach, presented in Section 2.3.1, is to construct routes based on link reliability. Another approach is to predict when the link on an active route will become broken and maintain the route in advance, described in Section 2.3.2.

2.3.1 Long-lived Route Protocols

In [24], a model is proposed to measure the probability that a wireless link exists between two mobile nodes at time $t_0 + t$, given that a link exists between them at time t_0. This model can provide the basis for route selection, choosing the route with the biggest minimum link probability, thus selecting the longest-lived route rather than the normal shortest hop route.

Associativity-Based Routing (ABR) [40] and Signal Stability-Based Adaptive Routing (SSA) [8] are reactive unicast routing protocols based on selecting long-lived routes. In both protocols, every node transmits beacons periodically to advertise its existence and neighbors can measure the distance and capacity of the link to it by receiving its beacons and learn the link's stability.

In ABR, the stability is recorded in form of "Associativity ticks", as a higher level of tick means more stable thus long-lived link. A Broadcast Query (BQ) message is initiated on demand for route discovery, and an intermediate node receiving BQ appends its identifier, the associativity ticks, the relaying load, the link propagation delay and the hop count of existing routes, to the BQ. So when the destination receives a BQ, it can choose the best route based on route stability and congestion information gathered in the BQ. Then the destination sends back a Reply to the source, establishing the most stable route.

In SSA, the link stability to neighbors is measured by classifying the neighbors as Strongly Connected (SC) or Weakly Connected (WC). When a source wants to send a packet to the destination and has no valid route, a Route Search (RS) is sent

out and only propagated further when being received from a SC neighbor, thus only reliable routes will be discovered. During the propagation, the node's address is appended to the RS message. So when the message reaches the destination, it contains the address sequence of intermediate nodes composing strong links. Thus, when the source receives a reply from the destination, a stable route is established.

A broken link is detected by not receiving a neighbor's beacon for a certain period. ABR will first try to repair the link locally with finding a route with shorter or equal hop count to the previous route, and then initiate a new route discovery when necessary. In SSA, the node detecting the failure sends an error packet to the source and the source will send an erase message to notify all nodes of the broken link and initiate a new route discovery when necessary.

2.3.2 Link State Prediction: Methods and Implementations

As the long-lived route cannot avoid future link breakage on it due to node mobility, another approach to improve route reliability is that with measuring node mobility or locating node position, the link breakage can be predicted and a new route can be constructed in advance if necessary.

2.3.2.1 Methods

The link availability prediction model in [11] assumes that if the distance between two nodes is less than the radius of their transmission range, they are able to communicate directly; otherwise, they cannot. Suppose the speeds (v_A, v_B), and directions (θ, *l,* and *m)* of the two nodes *A* and *B* are known and fixed, as shown in

Figure 4 [11],

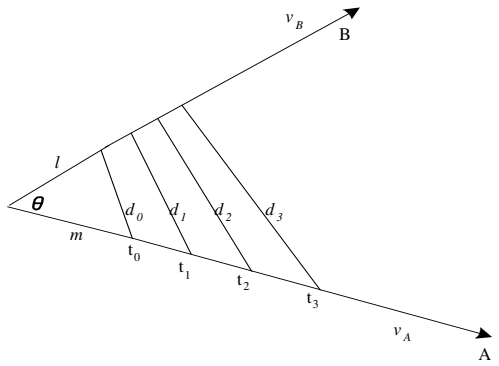


Figure 4: Schematic for Prediction Model

then the distance between them is:

$$d = \sqrt{(l + v_A t)^2 + (m + v_B t)^2 - 2\cos\theta(l + v_A t)(m + v_B t)} = \sqrt{at^2 + bt + c}\,, \qquad \textbf{(E1)}$$

with $a = v_A^2 + v_B^2 - 2v_A v_B \cos\theta$, $b = 2lv_A + 2mv_B - 2lv_B \cos\theta - 2mv_A \cos\theta$, $c = l^2 + m^2 - 2lm\cos\theta$.

Without learning speeds (v_A, v_B), and directions (θ, *l,* and *m)*, the distance can also be predicted if three distance values are known. Suppose Node A records the distances between itself and node B d_0, d_1, and d_2 at time t_0, t_1, and t_2. Node A can predict the distance d_3 at time t_3:

$$d_3 = \sqrt{at_3^{\,2} + bt_3 + c}\,,$$

with $a = \dfrac{(d_1^2 t_2 - d_2^2 t_1) - d_0^2 (t_2 - t_1)}{t_1 t_2 (t_1 - t_2)}, b = \dfrac{(d_1^2 t_2^2 - d_2^2 t_1^2) - d_0^2 (t_2^2 - t_1^2)}{t_1 t_2 (t_1 - t_2)}, c = d_0^2$.

In [38] [18], the same mechanism is used for calculating the amount of time that the link will stay connected with known transmission range *r*. Assuming two nodes *A* and *B* are within the transmission range *r* of each other, node A at (x_A, y_A)

moves with speed v_A at direction θ_A, and node B at (x_B, y_B) moves with speed v_B at direction θ_B. (θ_A and θ_B are in the range of 0 to 2π). So the remaining link time is:

$$T = \frac{-(ab+cd)+\sqrt{(a^2+c^2)r^2-(ad-bc)^2}}{a^2+c^2}, \qquad \textbf{(E2)}$$

with $a = v_A\cos\theta_A - v_B\cos\theta_B$, $b = x_A - x_B$, $c = v_A\sin\theta_A - v_B\sin\theta_B$, $d = y_A - y_B$. The two equations E1 and E2 are identical, and can be derived from each other.

Generally, speed, direction, and location (thus distance) of nodes can be provided by the Global Positioning System (GPS) [17]. In the absence of GPS, the distance of nodes can be obtained by measuring the received signal powers.

Several radio propagation models have been proposed to compute the received signal powers. The most popular model is the free space propagation model [34], in which a single line-of-sight path is considered: $P_r(d) = \frac{P_t G_t G_r}{L} * \frac{\lambda^2}{(4\pi)^2 d^2}$, where P_t is the transmitted signal power, G_t and G_r are the antenna gains of the transmitter and the receiver respectively, L is the system loss, and λ is the wavelength. The received signal power is inversely proportional to d^2, the square of the distance to the node that sent the signal. Another model, two-ray ground reflection model [34], considers both the direct line-of-sight path and a ground reflection path, giving more accurate prediction at a long distance than the free space propagation model: $P_r(d) = \frac{P_t G_t G_r}{L} * \frac{h_t^2 h_r^2}{d^4}$, where h_t and h_r are the heights of the transmit and receive antenna respectively. The received signal power

is inversely proportional to d^4, so we have a faster power loss as distance increases.

A method utilizing the signal power change for mobility prediction was proposed in [26]. Basically, the node's transmission power is constant. Received signal power samples are measured when receiving packets from a node's neighbor. So it is possible to compute the rate of change for a particular neighbor's signal power level and we can predict when the transmission power level will drop below the signal power reception threshold.

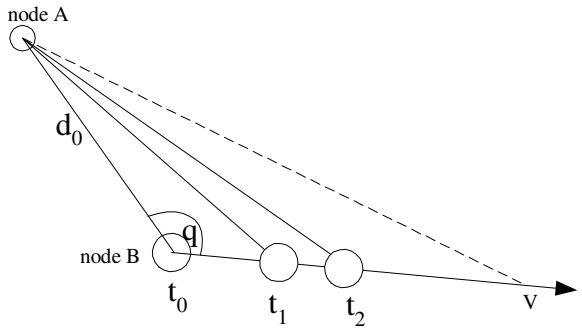


Figure 5: Movement of Node B in View of Node A

A more precise method to predict link breakage time by measuring received signal power is proposed in [33]. It uses computation of relative movements between two mobile nodes and the two-ray ground reflection radio propagation model. From the view of node *A*, the movement of node *B* can be viewed as in Figure 5 [33]. Assuming two nodes move with constant speeds and directions, as long as three received signal powers can be obtained, the time duration from when the third signal power was received to when the link will be broken is:

$$T = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \qquad \textbf{E3}$$

with $a = t_2\sqrt{P_1 P_s}\,\beta$, $b = \sqrt{P_s}\,((\sqrt{P_1} - \sqrt{P_2}) - t_2^2\sqrt{P_2}\,\beta)$, $c = t_2\sqrt{P_2 P_s} - t_2\sqrt{P_1 P_2}$,

$\beta = \dfrac{\sqrt{P_1 P_2}\,t_2 + \sqrt{P_2 P_3}\,t_3 - \sqrt{P_1 P_3}\,t_3 - \sqrt{P_2 P_3}\,t_2}{(t_2 t_3^2 - t_3 t_2^2)\sqrt{P_2 P_3}}$, where P_s is the signal power threshold, and P_1, P_2, and P_3 are the received signal powers at time t_1, t_2, and t_3.

In summary, all these equations assume that nodes are moving with constant speeds and constant directions during prediction. When the distance of nodes is obtained by measuring the received signal powers, constant transmission power is also assumed, as in IEEE 802.11 or Bluetooth.

2.3.2.2 Implementations

FORP (Flow Oriented Routing Protocol) [38] is a reactive unicast protocol with link state prediction mechanism. Route discovery is initiated on demand by broadcasting a Flow-REQ message. The node receiving the Flow-REQ will reply with a Flow-SETUP message if it has a fresh enough route to the destination; otherwise, it forwards the Flow-REQ appended with its own ID and the Link Expiration Time (LET) for that link, which can be computed by link state prediction equation E2. So when the Flow-REQ arrives at the destination, it contains the list of nodes along the route it has traveled and the LETs for each link along the route. The destination can then determine the Route Expiration Time (RET) by using the minimum of the set of the LETs in the Flow-REQ message. If the received route is more stable, that is with greater RET, than the one currently in use, the destination sends a Flow-SETUP message back to the source along the chosen route and activates the route. When forwarding the data flow, intermediate

nodes append LETs to each packet; so the destination can continue to compute the RET of that route. When the destination determines that the route is about to expire, a Flow-HANDOFF message is generated and propagated in the same manner as the flow-REQ message. After the source receives a Flow-HANDOFF message, it can determine the best route for the flow handoff based on the LETs contained in the Flow-HANDOFF. The source then sends a Flow-SETUP message along the new route.

Also, in [33], link state prediction is added to DSR. When a node receives a data packet, the received signal power is measured, and the time for link breakage is computed by using equation E3. If the link will break in a certain period of time, the node will inform the source node of the data packet and trigger the source node to find a new available route before that link finally becomes broken. Thus the communication from the source to the destination will be switched to the new route smoothly and packet loss at the broken link can be avoided. Simulation in [33] shows that by adding link state prediction in DSR, the packet loss is significantly reduced with a slight increase in control messages.

Another implementation of link state prediction is to limit control overhead, like reducing Join-Query flooding in multicast protocol ODMRP [19] [18]. With the prediction, Join-Queries are sent only when active routes will be disconnected. Join-Queries are broadcast with LETs like those in FORP for calculating the RET for the route. This RET is also included in the Join-Reply. If a forwarding group node receives multiple Join-Replies with different RET values, it selects the minimum RET and sends its own Join-Reply with this minimum RET. Thus, when

the source receives Join-Replies, it is able to know the minimal RET to the destination through the mesh. Therefore, instead of flooding Join-Query periodically, the source only floods a Join-Query before the minimum RET.

2.4 Thesis Approach

Research on routing and multicasting in MANET provides various solutions. To overcome the disadvantages introduced by mobility, efforts are put into estimating the distance of nodes and maintaining routes in advance. With GPS, the location, speed and direction of node movement is easy to obtain, thus the distance can be computed directly. Without GPS, the distance between nodes is obtained indirectly by measuring received signal powers based on a propagation model. In [33], such a method (equation E3) is proposed and examined for the reactive unicast protocol DSR. The conclusion is that for source routing, this method offers positive results. The method also needs to be explored in the context of another popular reactive unicast protocol AODV, since AODV accomplishes routing hop by hop with routing tables, not by source routing. More, AODV is extended with multicasting capabilities, the tree-based MAODV. As for multicasting, only one route exists between any member pairs in tree-based protocols, and the link breakage may result in packet loss more frequently than mesh-based protocols, so it is worthy to evaluate the method in a tree-based multicasting protocol.

The goal of this thesis is to implement the link state prediction method in equation E3 in AODV and MAODV, and analyze the simulation results. Therefore, in Chapter 3, AODV and MAODV with their simulations in NS2 are described.

The details of the prediction method implementation and simulation results for AODV and MAODV are provided in Chapter 4 and Chapter 5.

Chapter 3 AODV and MAODV: Protocol Specifications and Simulation in NS2

AODV (Ad-hoc On-demand Distance Vector Routing) [31] is a popular reactive unicast protocol, essentially a combination of both DSDV and DSR. It uses mechanisms of route maintenance from DSDV and route discovery from DSR. MAODV (Multicast Ad-hoc On-demand Distance Vector) [35] is the extension of AODV with multicasting capabilities. The evaluation of AODV and MAODV protocols is based on Network Simulator (NS2) [13], developed by the VINT project at the University of California at Berkeley and extended with the simulation of multi-hop wireless networks by the MONARCH research group at Carnegie-Mellon University. The AODV protocol was implemented in NS2, while we implemented MAODV protocol ourselves. All our simulations are based on NS2 version 2.1b8a on Linux 7.2.

3.1 AODV

3.1.1 Protocol Specifications

Each AODV node maintains a routing table, as nodes do in DSDV. Each route entry in the routing table contains the next hop information for the destination, currently known greatest sequence number of the destination, and the hop count to the destination. Associated with each route entry is a lifetime, indicating the length of time the route entry is valid. Routes are deleted from the table if they are not updated or used within the indicated lifetime. Like DSDV, each node maintains its own sequence number. Similar to the Request ID in DSR, another counter called

broadcast ID is also maintained. The broadcast ID, together with the source node's address, uniquely identifies each broadcast Route Request (RREQ).

The route discovery process in AODV is purely on-demand and follows a Route Request/Reply cycle like that in DSR. The source initiates route discovery by broadcasting a new RREQ with key fields <Source Address, Source Sequence Number, Broadcast Id, Destination Address, and Destination Sequence Number>. Source sequence number is its own sequence number; destination sequence number is the currently known destination sequence number by the source node. A node receiving RREQ discards duplicate RREQs by checking the pair of source node address and its broadcast ID. If it is a non-duplicate RREQ, the node updates its routing table to record the source sequence number and the next hop for the route to the source node. This reverse route may later be used to relay the Route Reply back to the source. Then, if the node is the destination or if it has an un-expired route to the destination with a destination sequence number at least as great as that indicated in the RREQ, a Route Reply (RREP) is generated. Otherwise, it rebroadcasts the RREQ to its neighbors.

The main fields of the RREP message are < destination address, destination sequence number, and hop count to destination>. Destination address is the destination address set in the corresponding RREQ; the destination sequence number is the sequence number recorded in the routing table of the responding node who generates the RREP; the hop count to destination is the hop distance from the responding node to the destination. The responding node unicasts the RREP back along the reverse route constructed during RREQ propagation. The

node receiving the RREP increments the hop count by one, updates routing information for the destination node in its routing table, thereby establishing the forwarding route to the destination. This unicast RREP continues to be relayed to the source node along the reverse route. Once the source node receives RREP, data packets can be sent along the forwarding route. It is likely that an intermediate node or the source node may receive more than one RREP for a given RREQ. In that case, the fresher RREP, either with greater destination sequence number, or with smaller hop count for the same destination sequence number, can be accepted. Otherwise, it will be discarded. RREPs received at intermediate nodes will be relayed to the source node. RREPs received at the source node update the routing information for the destination. Compared to the route discovery in DSR, in which the routing information is recorded in Route Requests and Route Replys, in AODV, the routing information is recorded in each related node. Therefore, for data packet delivery, in AODV the next hop routing decision is determined by each intermediate node as operated in DSDV, while in DSR the routing decision is made from the hop sequence determined by the source and appended in the packet header. Figure 6 illustrates the route discovery process in AODV. In the (_:_) notation, the first entry is the destination, the second entry is the next hop to the destination, which simply represents the reverse routes and the forwarding routes constructed during route discovery.

The detection of link breakage can be accomplished from wireless MAC layer retransmissions and acknowledgements, or by using periodic one-hop neighbor HELLO messages.

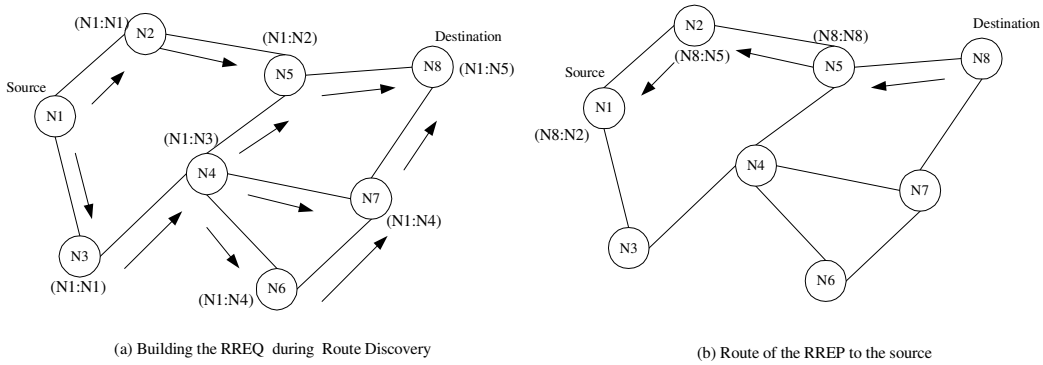


Figure 6: AODV Route Discovery

With MAC layer detection, when a node cannot send or relay data packets to the next hop along the forwarding route, the node propagates an unsolicited RREP with a greater sequence number and infinite hop count to the destination to all active upstream (nearer to the source) neighbors. The neighbors subsequently relay RREP to their active neighbors on active routes till all active source nodes are notified. Upon receiving notification of a bad link, source nodes can re-initiate the route discovery process if they still need a route to the destination.

One-hop neighbor HELLO message is another mechanism to detect link breakage. Each node periodically broadcasts a HELLO message only to its neighbors, indicating its existence. When HELLO message are not received from the next hop along an active route during a certain period, the active neighbors assume the next hop does not exist any more, thus triggering the process described above.

3.1.2 Simulation in NS2

For simulation in NS2, each mobile node on a flat ground uses an

omni-directional antenna with gains equal to 1 (G_t or G_r = 1), with the antenna height at 1.5m (h_t or h_r = 1.5). The wireless interface works like the 914MHz Lucent WaveLAN DSSS radio interface [41], with the system loss *L*=1. The power of signal attenuates based on the free space model at short distance and the two-ray ground reflection model at longer distance. The crossover point is around 86.14m. The transmitted signal power is around 0.2818 W and the correct-received signal threshold is around 3.652e-10 W, so the transmission range is about 250m.

The MAC layer protocol used in the simulation is the IEEE 802.11 Distributed Coordination Function (DCF) [14]. The transmission of each unicast packet is preceded by a Request-to-Send/Clear-to-Send (RTS/CTS) exchange to reserve the wireless channel for data transmission. When the neighbor correctly received the unicast packet, an Acknowledgement (ACK) is send to the sender of the packet. This mechanism reduces the potential collision by the hidden-terminal problem [39], thus improves the unicast transmission quality. The RTS control messages compete with broadcast data packets for the channel by using the unslotted CSMA/CA [14] mechanism, in which information can be transmitted after sensing an idle channel and may suffer from the collision by the hidden-terminal problem.

Nodes move on a flat ground, and the movement is modeled by the random waypoint model [15]. Each node moves from a random location to a random destination with a randomly chosen speed uniformly distributed between 0 and a given max speed. When the destination is reached, after remaining there for a given pause period, another movement with random speed and direction begins. This behavior repeats for the duration of the simulation. So node movement scenarios

can be characterized by maximum speed and pause time. Continuous movement is equivalent to 0 pause time, while no movement is obtained when the pause time is set to the duration of the simulation. In general, the unit of measuring pause time is second, and for max speed meter/second (m/s) is used.

With the goal to compare the performance of the protocols, a constant bit rate (CBR) traffic pattern is used. With CBR pattern, fixed size data packets will be sent at roughly the same time interval and not affected by the network flow control. So with CBR pattern, the performance of packet delivery is mostly determined by the performance of the routing protocols.

Every node in NS2 maintains a network interface transmit queue to queue all packets (including routing messages and data packets) till the MAC layer can transmit them. The interface queue is a priority queue with a maximum size of 50 packets. The routing messages are given higher priority than data packets, so the routing messages are queued at the head of the queue, whereas packets are inserted at the end of the queue.

An AODV implementation is provided in NS2. A send buffer is maintained by the sources for queuing data packets that need to be sent but do not have a valid route. Once the route is created, these packets can be transmitted. This buffer is FIFO with a maximum size of 64 packets. To prevent buffering of packets indefinitely, packets are dropped if they wait in the send buffer for more then 30 seconds.

Besides the protocol operation described in Section 3.1, several decisions were made to improve its performance [3] for AODV in NS2:

- To prevent synchronization and collision of broadcast messages, broadcast messages are propagated with using jitter with a random delay uniformly distributed between 0 and 10 milliseconds.
- To eliminate the overhead of the periodic HELLO messages, link breakage is detected only by using feedback from the 802.11 MAC layer. So only unicast packets can be used for link breakage detection and the link breakage can be detected only on-demand when a unicast packet needs to be routed over that link. When using the HELLO messages, the broken link can be detected at any time interval. [3] claims AODV with MAC layer feedback performs significantly better then AODV with periodic HELLO messages, so in our work, we use AODV with MAC layer feedback.

MAC layer link breakage detection	Yes
Lifetime for the forwarding route when RREP sent by a intermediate node	50 seconds
Lifetime for the forwarding route when RREP sent by destination node	60 seconds
Lifetime added when forwarding a packet along the route	50 seconds
Lifetime for the reverse route constructed by RREQ	10 seconds
Time for caching for the broadcast ID in RREQ	6 seconds
Number of times a RREQ is retried	3
TTL_START for first RREQ for an unknown route	1
TTL_INCREMENT for next time RREQ	2
TTL_THRESHOLD for the expanded ring search of RREQ	7
TTL for Network-wide Broadcast	30

Table 1: Parameters used for AODV Implementation

Table 1 lists the parameters used for the AODV protocol in NS2 simulation. To prevent unnecessary network-wide RREQ broadcasts, the expanded ring search is used to set the TTL value of a RREQ. The TTL of the first RREQ for an

unknown route is set to TTL_START. Then when a new RREQ is sent, its TTL is set to the TTL of the previous RREQ plus TTL_INCREMENT. If the previous TTL exceeds the TTL_THRESHOLD, TTL for the new RREQ is set to the TTL for network-wide broadcast.

Our aim is to predict the link state and maintain the route in advance. But the actual link breakage can be not only caused by movement, but also can be caused by congestion. So we need to measure the link prediction method with the traffic load generating little congestion. The traffic load can be changed by the number of connections in the network or the size of the data packet. In [3], packet sizes of 64 bytes and 1024 bytes are used and it is found that a packet size of 64 bytes can avoid congestion. In addition, our goal is to test the protocol's ability to determine routes to a destination, so frequently sending small size packets is preferred.

The space used in our simulation is rectangle field with size 1500m by 300m. The reason we use a rectangle field instead of a square field is that we would like to force the protocol to form longer routes.

There are 50 nodes in the simulation, in which 20 connections are established in the first 180 seconds. These 20 connections randomly select the source and the destination, so for one source, there may be more than one destination and vice versa. Once the connection is established, the source of each connection will send out 4 packets per second till the simulation ends.

To determine a reasonable simulation time, we simulate a scenario with the mobility speed for all the nodes uniformly distributed from 0 to max speed 20m/s, and the pause time set to 0. For this scenario, 10 cases with different mobility files

are generated to run 1200 seconds. At 100-second intervals, we collect data to calculate the average of the Packet Delivery Ratio and the Normalized Overhead. (The explanations of the Packet Delivery Ratio and The Normalize Overhead metrics are in Section 4.2.) As shown in Figure 7 and Figure 8, the metrics are rather stable after 600 seconds. With reference to the simulation time set in [33][3], we decide the simulation time for our unicast simulations to be 900 seconds for comparability.

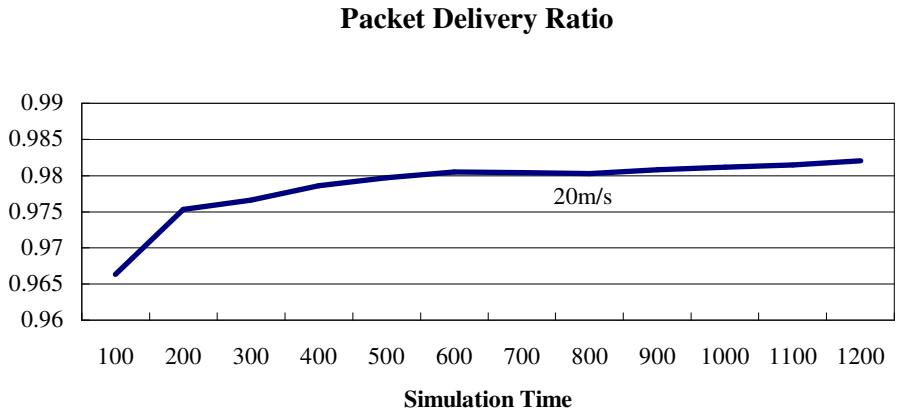


Figure 7: AODV Packet Delivery Ratio vs. Simulation Time

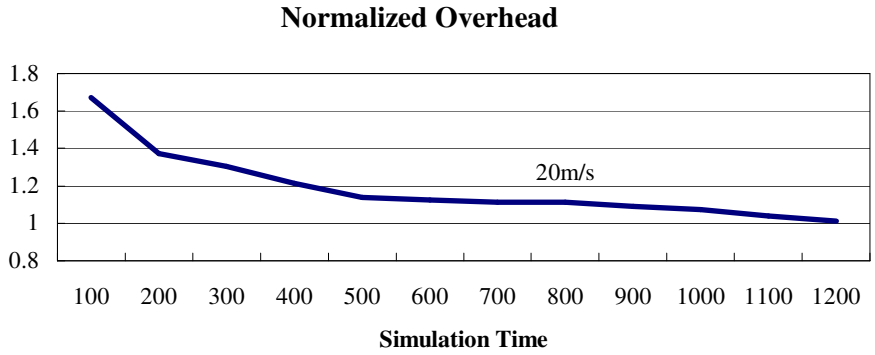


Figure 8: AODV Normalized Overhead vs. Simulation Time

3.2 MAODV

3.2.1 Protocol Specifications

Like unicast route discovery in AODV, in MAODV, multicast routes are discovered on demand, based on a broadcast route Request-Reply mechanism.

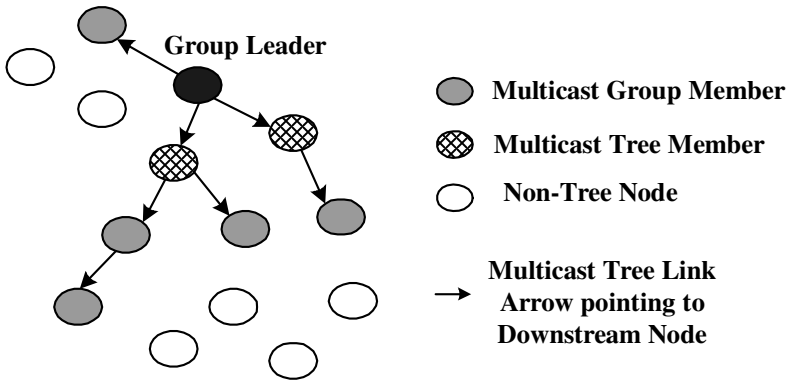


Figure 9: MAODV Multicast Tree

The multicast group is identified by the multicast group address, associated with group sequence numbers used for tracing the freshness of the group situation. Group members of the same multicast group compose a tree structure in MADOV with the help of those nodes that are not group member but must forward multicast information to group members, called non-group tree member. When a node wants to join a multicast group that does not currently exist (from its point of view) in the network, that node becomes the multicast group leader. The group leader is responsible for maintaining the multicast group sequence number and the tree structure. All the nodes on the tree can be organized as upstream node or downstream node from the view of the group leader. The group leader has no upstream node. For two nodes at the end of a link, measuring by the hop count to the leader, the node nearer the group leader is the upstream node compared to the

other node; whereas, the node farther from the group leader is the downstream node. Figure 9 [36] gives an example of a multicast group tree.

Besides the unicast routing table maintained for AODV, an MADOV node keeps a multicast routing table for the group tree structure. The multicast routing table entry contains fields such as: multicast group address, multicast group leader address, multicast group sequence number, hop count to multicast group leader, next hops information, and lifetime. Next hops information records the node's neighbors who are actually or potentially-will-be in the tree. An enabled flag, and a direction, are associated with each next hop. If the flag is enabled, the next hop is actually in the tree. Otherwise, the next hop with disabled flag cannot be used for forwarding or receiving any multicasting information. From now on, unless indicated as potential next hop, all next hops are actually in the tree. The direction can be upstream or downstream as explained earlier. At most one next hop can be indicated as the upstream node. The group leader can only have downstream nodes if there are any other group members. At every interior node in a multicast tree, the route entry for the multicast group should have multiple next hops: one upstream node and at least one downstream node. A leaf node only has one next hop: its upstream neighbor.

The same RREQ and RREP used in AODV are adapted to be used in MAODV. A node sends a RREQ message when it wishes to join a multicast group or when it has data to send to a multicast group, but has no route to that group. The Destination address in the RREQ is set to be the group address. The sequence number for the destination should be set to the currently known largest sequence

number of the corresponding group. If the RREQ is for joining the group, a join flag is set in the RREQ (J-RREQ); otherwise, leave the flag unset. Basically, the RREQ is broadcast in the network, but if the node has enough information about the group leader, RREQ may be sent as unicast to the leader. Here, we call the node that initiates RREQ the source node.

Only a member of the multicast tree, including the group members and the non-group tree members, may respond to a J-RREQ. Any node with fresh enough route to a multicast tree may respond to a RREQ without any flag. When receiving a non-duplicate RREQ, as in AODV, the unicast reverse route to the source is set up in the unicast routing table for future use when returning a corresponding RREP. The same criteria is used for responding with RREP as in AODV, checking the multicast routing table instead of the unicast routing table. If the node cannot respond to the RREQ with a RREP, it will rebroadcast it.

When the node can respond to the RREQ with RREP, it means the reverse unicast route may be the potential branch connecting the tree and the source. So in the multicast routing table, the next hop for the reverse route to the source should exist in the next hops information, but with disabled flag and downstream direction. The destination sequence number in RREP for multicasting is the group sequence number. An extension with information about the group leader address, the hop count to the group leader, and the hop count to the tree is also added to the RREP. If the node is a tree member, the hop count to the tree is set to 0; otherwise, it is set to the hop count to the tree in its multicast routing table. As the RREP is sent back to the source along the reverse route, the hop count to the group leader and the hop

count to the tree are incremented by 1. Two potential next hops will be added in any intermediate node, indicating the potential upstream node from which the RREP is received and the potential downstream node towards the source along the reverse route. Both next hops should be set with disabled flag. When the source receives RREP, only the potential upstream node will be added in the next hops information.

It is likely that intermediate nodes and the source may receive more than one RREPs, as explained in the AODV protocol, only the fresher route is forwarded to the source from intermediate nodes, while the source keeps the freshest route. The fresher route means greater group sequence number, or smaller hop count to the multicast tree when the group sequence numbers are equal. Unlike the RREP in AODV, which actually establishes the active route to the destination, the RREP in MAODV only provides the possible branch to the tree without activating it, because for all possible branches, only one branch can be grafted to the tree or only one route will reach to the tree for avoiding loops. So a new message, Multicast Route Activation (MACT), is used for grafting a branch or adding the route to the tree. Each MACT contains: flag, source address, source sequence number, and the multicast group address. If the source wants to join the multicast group, the join flag is set (J-MACT); otherwise, the flag is unset. MACT is initiated by the source waiting a certain period after sending the RREQ, as it hopes to receive more than one RREP and select the freshest one indicating the latest tree structure. The source sets the flag enabled for the potential upstream node indicated in the selected RREP and sends out MACT toward that upstream node. As MACT is forwarded by the

potential upstream node, both potential next hops added when handling the RREP become activated till the tree or the node generating the RREP is reached. Therefore, for J-RREQ, when the tree member generating the RREP activates the corresponding link to the downstream node, the branch is finally added to the tree. For RREQ with no flag, when the node generating the RREP activates the corresponding link to the downstream node, the route to the multicast tree is finally activated, and data packets can be sent. Figure 10 [35] illustrates the join procedure.

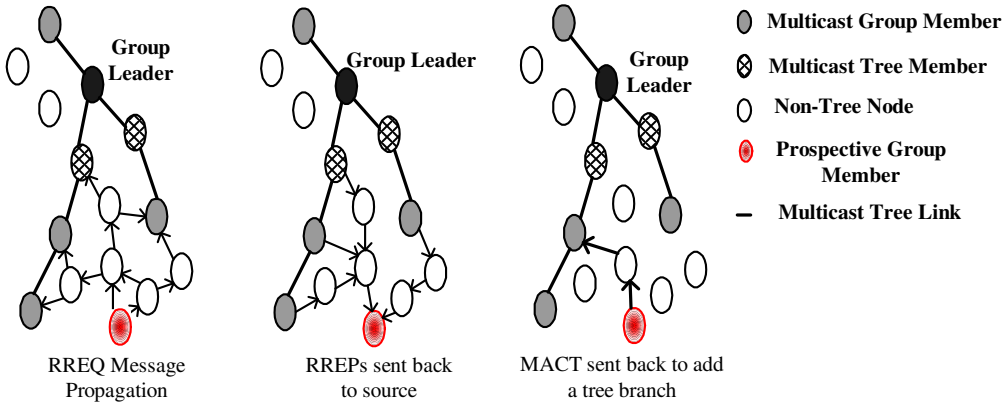


Figure 10: MAODV Multicast Join Operations

If a node tries to join a group tree, but has not received any RREP, after several attempts, it realizes that in the network there is no such group, or it cannot reach that group due to network partition. Then, as described earlier, it becomes the first node in that group and acts as the group leader to maintain the group sequence number and tree structure.

Multicast tree maintenance is much more complicated than unicast route maintenance. The group leader periodically broadcasts a Group Hello (GRPH) message throughout the whole network, to indicate the multicast group leader address and current group sequence number. The group sequence number is

incremented each time before the GRPH message is sent out. Each node receiving the GRPH message records a unicast reverse route to the group leader, while a tree member can update tree information in its multicast routing table.

A group member can leave the multicast group at any time. If the node is not a leaf node of the tree, it may discard its membership but needs to stay in the tree to serve as a router (a non-group tree member) for the tree. Otherwise, it may prune itself from the multicast tree by sending MACT with prune flag (P-MACT) to its upstream node. Figure 11 [36] gives an example of pruning. If receiving P-MACT makes the node a leaf and the node is not a group member, it can similarly prune itself from the tree. This procedure terminates when a group member or non-leaf tree member is met.

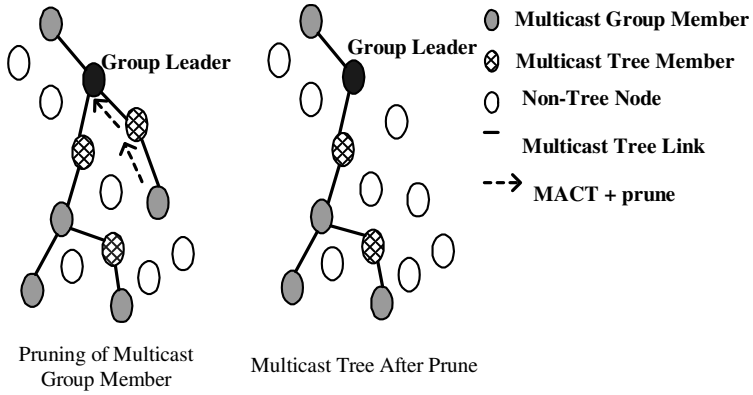


Figure 11: MAODV Group Member Pruning

Mobility can easily cause link breakage. Link breakage in a multicast tree can results in partial data delivery, as not all group member will receive multicast information. Link breakage can be detected by the detection methods used in AODV. After link breakage occurs, the link becomes disabled as the next hop indicating the link will be set with disabled flag. Unlike AODV, which may trigger

discarding the route and the sender searching a new route, in MAODV, the link repair procedure will be operated locally and only the downstream node for that link can initiate link repair. So the downstream node broadcasts a J-RREQ with additional information, which includes its own hop count to the group leader, in order to avoid that its own downstream nodes send back a RREP. Any node that is a tree member with smaller hop count to the group leader and fresh enough group sequence number can respond with RREP. After a certain period of time, if the source receives RREP, it will send J-MACT back to the selected upstream, and at the same time, MACT with update flag (U-MACT) will be sent down to all active downstream nodes, indicating them to update the new hop count to the group leader. Figure 12 [36] presents an example of repairing the tree.

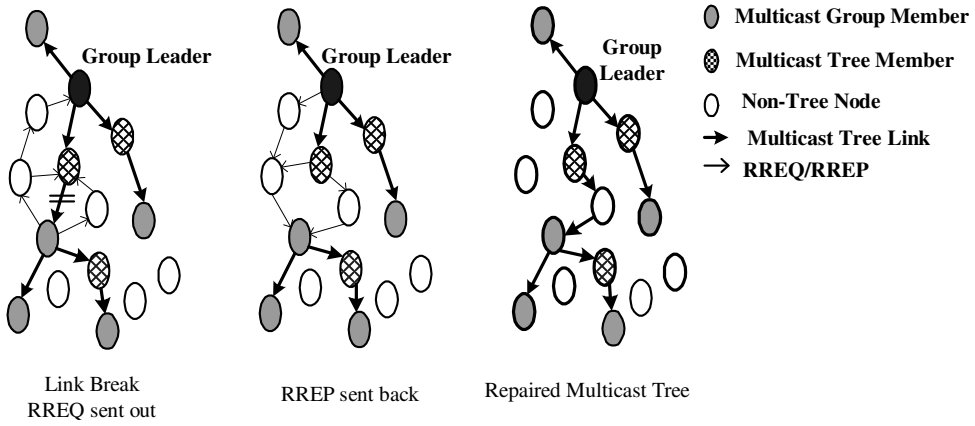


Figure 12: MAODV Repair of Multicast Tree

If no RREP is received after retrying several times, the source assumes that tree partitioning occurred and begins selecting a new group leader for the partitioned tree. If the source is a group member, it will become the new group leader. Otherwise, as it has no valid upstream node, it will force one of its downstream nodes to be the leader. So if it just has one downstream node, the

source will send out P-MACT to the downstream node, indicating that it will leave the tree and the tree needs a leader. If the source has more than one downstream nodes, it will select one and send MACT with group-leader flag (GL-MACT) to the selected downstream node, indicating it has another branch in the tree and the tree needs a leader. So the downstream node can receive either a P-MACT or a GL-MACT from its upstream node. When sending or propagating P-MACT, the node leaves the group with deleting the group information in its multicast group table. When receiving P-MACT from the upstream node, the node will disable the upstream node in its next hop information. When sending or propagating GL-MACT, the node changes the direction of the selected downstream node from downstream to upstream. When receiving GL-MACT from upstream, the node changes the upstream direction into downstream. P-MACT or GL-MACT will be propagated till a group member is reached. Once a group member is reached, it becomes the group leader and begins to broadcast GRPH periodically. If it has any downstream nodes, it will also send U-MACT to downstream nodes indicating the new leader and new hop count to the leader.

Mobility also can cause partitioned trees for the same group address to merge into one tree. Tree merges can be detected when a group leader receives a GRPH generated by another group leader for the same group address. Only one of the group leaders can initiate the merge process, so let it be the group leader with smaller address identifier. It unicasts a RREQ with repair flag (R-RREQ) to the group leader with greater address identifier along the unicast reverse route recorded when receiving the GRPH message. When the group leader with greater address

identifier receives R-RREQ, it sends back RREP with repair flag (R-RREP) and activates the next hop toward the source as downstream node at the same time. As the R-RREP travels back to the source, the intermediate node not only adds two next hops in its next hops information, similar to receiving a normal RREP, but also activates them. So when the source, the leader with smaller address identifier, receives the R-RREP, with activating the upstream node, the two partitioned trees are being connected. The group leader with greater address identifier becomes the new group leader for the merged tree. Figure 13 [36] illustrates the procedure. When the source receives R-RREP, it should also send out U-MACT to its downstream nodes, indicating the change of group leader and the hop count to the group leader.

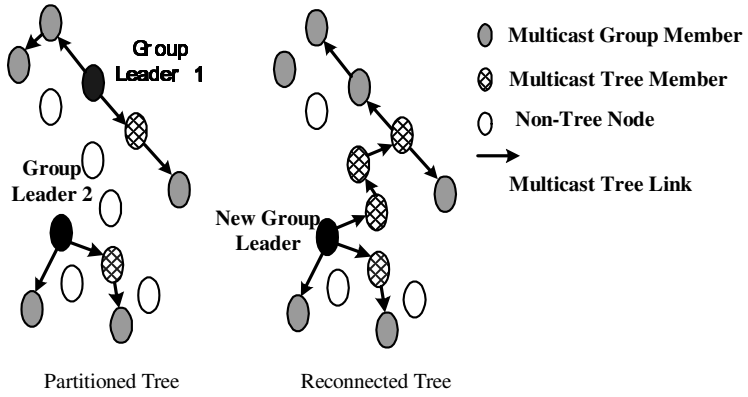


Figure 13: MAODV Tree Merge

3.2.2 Simulation in NS2

The physical features for MAODV are the same as those for AODV: flat ground, omni-directional antennas, Lucent WaveLAN DSSS radio interface, signal attenuation model, transmission power and range.

The same MAC mechanism is used. As MAODV must use the basic

mechanism in AODV, link breakage in MAODV is also detected by using MAC layer feedback, which requires the multicast data packets to be unicastly sent to every necessary neighbor. However, this method increases the bandwidth consumption and may easily result in congestion at a specific node. But unicast packet delivery is more reliable than broadcast packet delivery as it uses the RTS/CTS and ACK MAC layer messages.

The network interface queue is also the same. In addition to the send buffer for queuing unicast packets, an extra send buffer is maintained by the sources for queuing multicast data packets that need to be sent but the source currently has no valid route to the tree. Once the route is created, these packets can be sent. This buffer is also FIFO with the maximum size of 64 packets and retains packets only for less than 30 seconds. In our simulations for MAODV, only multicast data packet are sent out, excluding any influence by unicast traffic.

Besides the parameters used for AODV listed in Table 1, Table 2 lists the special parameters used for MAODV. In MAODV, not all group members, thus the receivers, are the senders. In the situation that a sender is at the upstream position, if a link breakage occurs, the link breakage can be only detected by the upstream node of that link. According to the MAODV protocol, only a downstream node can initiate local tree repair to avoid forming loops in the tree, so in this situation, the upstream node does not try to repair the tree and the downstream node does not know there is a link breakage. After not receiving any data packet for a certain period of time, the downstream node has to rely on the route in the multicasting route table to expire to recognize the link breakage and try to repair the tree locally.

So, we set the lifetime for multicast routes rather small compared to the lifetime selected for unicast routes.

Lifetime when activating the route for upstream node	5 seconds
Group Hello Interval	5 seconds
Time waiting for RREPs before sending MACT	2 seconds
Time waiting for unused branch to be pruned	3 seconds

Table 2: Extra Parameters for MAODV Implementation

We implemented MAODV in NS2 ourselves, based on the work in [4]. The validation of the MAODV implementation is provided by running different scenarios and acquiring similar results when compared to the results in other related papers [35] [20] [27] [36].

As mentioned above, all multicast data packets are sent via unicast to each tree neighbor. So, to avoid congestion, we use 64 bytes as data packet size as in AODV. The traffic pattern is also CBR, and the node movement also follows the random waypoint model.

We can use a square flat field for simulating MADOV, because the length of the route to deliver data mainly relies on the tree structure. The length for leaf nodes to the group leader is somewhat related to the simulation space, but since the source is not necessarily the leader, this results in longer or shorter routes. Simulations in [36] show that the packet delivery ratio is lower in a 1000m by 1000m area than in a 1500m by 300m area when mobility increases.

There are 50 nodes in our simulations. All the packet senders are group members, and a group member will join the group at the beginning of the simulation and keep its membership. The packets are sent out 30 seconds after the

simulation begins in order to let the members form an initial group tree. After that, each source sends out packets periodically. To avoid congestion, the total traffic load is 20 packets per second. So for 5 senders, each sender generates packets every 0.25 second. Only one group is involved in the simulation.

The simulation time is determined by the scenario with node max speed set to 20m/s and 0 pause time, in which there are 5 senders and 20 group members. 10 cases with different mobility files are generated to run 2000 seconds. At 100-second intervals, the average of the Packet Delivery Ratio and the Normalized Overhead are collected (the explanations of the Packet Delivery Ratio and the Normalize Overhead metrics are in Section 5.2.) The results are presented in Figure 14 and Figure 15. The simulation time of 1500 seconds is chosen with the overall consideration of appropriate results and simulation running time consumption. 1500 seconds is longer than the simulation times used in related papers. On a computer with Pentium III 731MHz, 512M Byte RAM, Linux 7.2, one simulation run takes about 50 minutes.

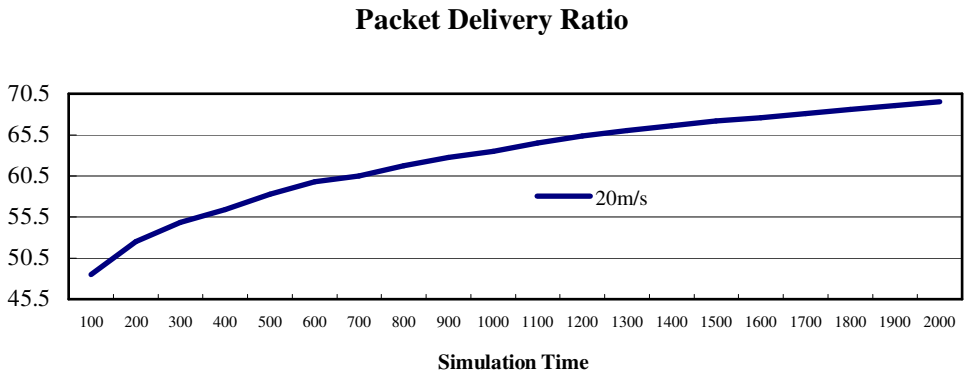


Figure 14: MAODV Packet Delivery Ratio vs. Simulation Time

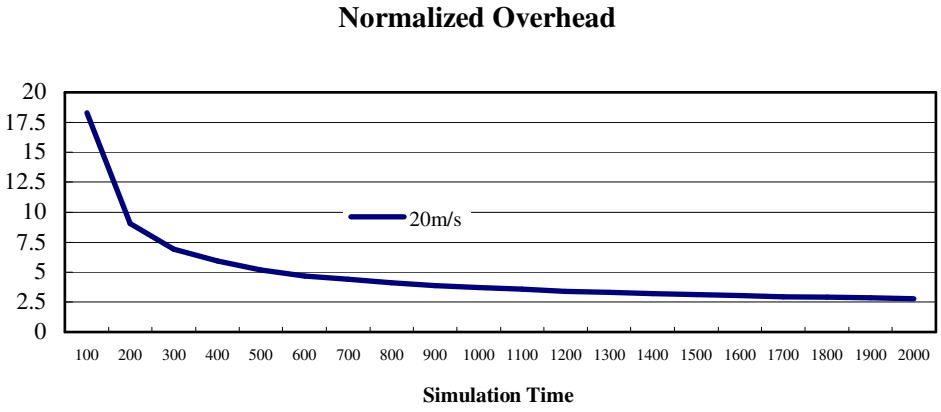


Figure 15: MAODV Normalized Overhead vs. Simulation Time

3.3 Validity of Link State Prediction in AODV and MAODV

As described in Section 2.4, the link state prediction equation E3 proposed in [33] will be used in AODV and MAODV to predict the breakage time and maintain routes in advance, which is called "proactive maintenance". Before the proactive maintenance methods for AODV and MAODV are implemented, it is necessary to examine the validity of the link state prediction equation E3 in AODV and MAODV protocols.

As proposed in [33], every mobile node maintains a link state table that contains its active neighbor mobile node addresses, the signal power value and the reception time of received data packets or control messages from these neighbor mobile nodes. The predicted link break time is calculated by using Equation E3 based on these data. A successful prediction is made only when the two nodes at the end of a link move constantly, that is, each node moves at the same speed and in the same direction during the prediction.

We choose the scenario with max speed 20m/s for our validity simulation. For AODV, pause time is set to 0, and parameters other than max speed and pause time

are the same as listed in Table 5 (on page 63). For MAODV, group size is set to 20; the number of senders is set to 5, and other parameters are kept the same as listed in Table 9 (on page 82). 10 cases are run for each scenario to get the average results.

Table 3 summarizes the average prediction results for AODV, and Table 4 lists the average prediction results for MAODV.

(T1 second,T2 second)	**(0.75,0.75)**	**(1,1)**	**(1.25,1.25)**	**(1.5,1.5)**
Average Number of Predictions	879.8	819.4	954.6	987
Average Number of Lost Packets	1220.6	1220.6	1220.6	1220.6
Average Number of Packets Dropped because of No Route	1132.4	1132.4	1132.4	1132.4
Loss: No-Route/Total	0.9259	0.9259	0.9259	0.9259
Average Number of No-Route Dropped Packets that can be avoided by a Successful Prediction	1005.7	1007.1	1007.2	1007.3
No-Route Drop: Predicted/Total	0.8878	0.8891	0.8892	0.8892
Average Number of Unused Predictions	79.1	102.4	126	146.3
Predictions: Unused/Total	0.0893	0.1108	0.1311	0.1470

Table 3: Prediction Results for AODV

(T1 second,T2 second)	**(0.75,0.75)**	**(1,1)**	**(1.25,1.25)**	**(1.5,1.5)**
Average Number of Predictions	661.2	680.8	698.7	714.5
Average Number of Lost Packets	2283.5	2283.5	2283.5	2283.5
Average Number of Packets Dropped because of No Route	2269.7	2269.7	2269.7	2269.7
Loss: NO-Route/Total	0.9939	0.9939	0.9939	0.9939
Average Number of No-Route Dropped Packets that can be avoided by a Successful Prediction	1685.3	1712.3	1729.3	1741.7
No-Route Drop: Predicted/Total	0.7460	0.7572	0.7641	0.7696
Average Number of Unused Predictions	35.8	47.1	57.9	67.2
Predictions: Unused/Total	0.05448	0.06956	0.0836	0.0947

Table 4: Prediction Results for MAODV

Explanations:

T1 is the threshold used to measure the difference of the estimated link breakage time and the current time when the unicast packet is received. T2 is the threshold used to measure the difference of the estimated link breakage time and the actual time when a packet is dropped because of link breakage.

As described above, when a unicast packet is received by a node, the node can estimate when the link will be broken. If the time difference is below T1, a prediction is made (i.e., we only care about links that we predict to break in the next T1 seconds). No duplicate prediction is included, as only one prediction is counted if the same prediction time is calculated when receiving another unicast packet later. Also if the same predictions are made from both directions of one link, only one is counted. From our observations, "The Average Number of Predictions" becomes larger when T1 becomes larger. It is because there are some false predictions made when nodes change either speed or direction during the next T1 seconds. If T1 is larger, more such false predictions will occur. A special case for AODV, the prediction that occurs on the link between the source and its next hop, is excluded. Because when the link between the source and its next hop becomes broken, instead of dropping data packets, the data packets are queued until a new route is found. So the prediction made for the link between a source and its next hop cannot be used for avoiding dropping packets, although it is used for queuing packets. For MAODV, this phenomenon is not significant because there may be more than one branch from a source, and the packet is queued only when no branch exists; otherwise, the packet is dropped.

When a link actually becomes broken, NS2 does not determine the exact breakage time. Rather, the link breakage is only detected when a node tries to send a unicast packet but cannot, and then the packet is dropped because of No-Route. As indicated in Table 3 and Table 4, about 92.59% of packet losses are No-Route drops for AODV, and about 99.39% for MAODV. There are other reasons for packet losses, such as the routing queue or the network interface queue is full, or the node address cannot be translated to a physical address by the ARP mechanism in time. Prediction aims to reduce the number of No-Route dropped packets, but it will also affect other kinds of packet loss when in congestion condition. As described in [33], there may be many No-Route dropped packets occurring at the same time. This phenomenon happens when congestion occurs around a specific node. "The Average Number of Packets Dropped because of No Route" represents how many packets are dropped with the reason that there is no route to the destination. Every No-Route dropped packet is counted.

No-Route dropping only occurs when the link becomes disconnected. So if the breakage time can be predicted, these losses can be avoided. T2 is used for computing "The Average Number of No-Route Dropped Packets that can be avoided by a Successful Prediction". It counts when No-Route dropping occurs before the estimated link breakage time plus T2. From Table 3 for AODV, we can see almost all No-Route Dropped Packets are dropped within 0.75 seconds after the estimated link breakage time. For MAODV, in Table 4, there is a slight different, but the difference is under 3.35% $((1741.7-1685.3)/1685.3)$. Not all No-Route packet losses can be avoided because prediction is made upon at least three unicast

packets received when each related node keeps the same direction and the same speed during prediction. As indicated in Table 3 and Table 4, at least 88% No-Route packet losses can be avoided for AODV, and at least 74% for MAODV, by using link state prediction method with Equation E3.

"Average Number of Unused Predictions" is also presented. The reason why the predictions are not used is mainly because false prediction occurs when nodes do not keep moving in the same direction or the same speed. Other reasons resulting in unused predictions includes more than one link breakage on one route, or packets being dropped beyond the estimated link breakage time plus T2. Table 3 and Table 4 indicate that unused predictions increase as T2 becomes larger.

Conclusion:

By using Equation E3, at least 88% No-Route packet losses for AODV, and at least 74% No-Route packet losses for MAODV can be avoided. But false prediction does occur, which will incur unnecessary route maintenance if link state prediction and proactive maintenance are implemented in AODV and MAODV.

A new parameter SETUP_TIME is introduced to indicate when the maintenance needs to be initiated before the link becomes broken. After comparing different T1 and T2 values, we choose 0.75 second as SETUP_TIME for AODV, and 1 second for MAODV. The next two chapters, Chapter 4 and Chapter 5, describe the proactive maintenance based on link state prediction in detail.

Chapter 4 Proactive Route Maintenance in AODV

In this chapter, a proactive route maintenance mechanism for AODV using the link state prediction method (equation E3) is proposed and analyzed with NS2. The modified protocol is called AODV-PRM (AODV with Prediction Route Maintenance).

4.1 AODV-PRM Description

The main difference between standard AODV and AODV-PRM is in route maintenance. In standard AODV, as described in Chapter 3, the link breakage is detected by MAC layer feedback when a unicast data packet cannot be successfully transmitted using the RTS/CTS mechanism. After the detection, the node that wants to transmit the packet, thus aware of the link breakage, will notify the upstream nodes till the source is reached. Then the source can initiate a route discovery process if necessary. The main idea of AODV-PRM is that when a node receives a unicast data packet, the estimated link breakage time of the link, from which the packet is received, is calculated and known in advance. If the link will become broken soon (determined by SETUP_TIME plus the current time), the node receiving the data packet will notify this situation to its upstream nodes till the source is reached. If the source still needs a route, it will initiate a route discovery process even though at the same time there is a valid route. Because the estimated soon-to-be-broken link is now actually connected, the data packets on the route can also pass through that link without loss. A new route without that soon-to-be-broken link is expected to be established before that link becomes

actually broken, so new data packets may go through the new route avoiding that bad link without queuing delay at the source node. Figure 16 illustrates the route maintenance procedure in AODV-PRM.

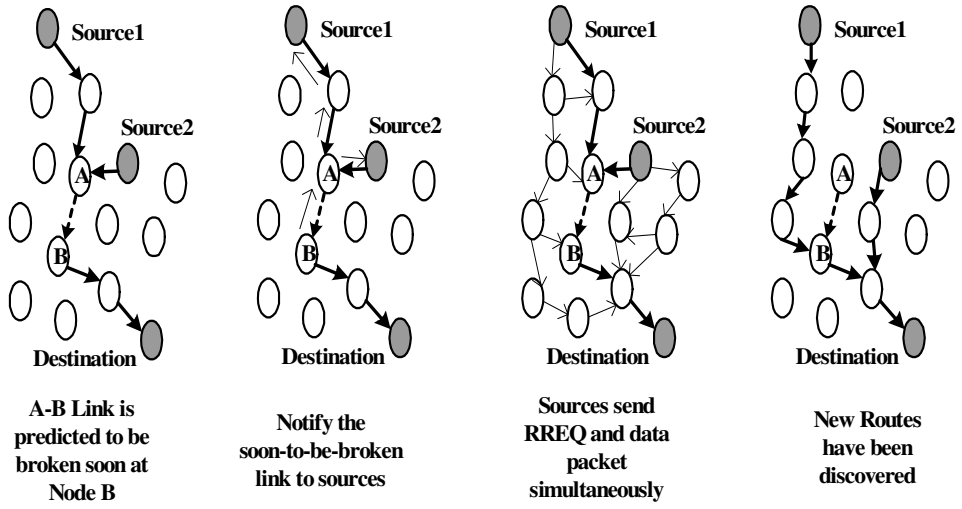


Figure 16: AODV-PRM Proactive Route Maintenance

Every time a mobile node receives a unicast data packet, it puts corresponding information (the previous hop mobile node address, the signal power and the reception time of the packet) into its link state table and updates the predicted link breakage time calculated by Equation E3. When the mobile nodes move at a fixed speed and toward a fixed direction, the predicted link breakage time should be the same when one mobile node continuously receives data packets from the other.

The AODV-PRM route maintenance contains two phases: route suspension and route rediscovery, which are described below.

4.1.1 Route Suspension

When a mobile node B receives a unicast data packet from its previous hop mobile node A, it will check the predicted link breakage time for this particular link.

If the predicted link break time indicates that this link will be broken in the near future (determined by SETUP_TIME plus current time), B will send a one-hop unicast control message back to A to indicate to A the predicted link breakage time of that link and the destination of the current data packet. This message is called LPW (Link Prediction Warning). At the same time, as the data packet already pass through the soon-to-be-broken link, mobile node B handles the packet as in standard AODV, routes this packet to the destination according to its routing table or processes this packet in case B is the destination.

When the mobile node A receives the LPW, with the destination of the data packet, node A can determine the route to the destination through node B, and set that route with state "RTF_P_LINK" to indicate the next hop is going to be broken. Meanwhile, the route expiration time is set to the predicted link breakage time, so that the route will be kept till that time, and after that time this route is going to be timed out from the routing table. Then mobile node A checks if there is an active upstream node using that route. If the mobile node A is an intermediate node in the route, a control message should be sent along the reverse route to inform those mobile nodes including the one who initiates the data packets used for prediction. So node A initiates a unicast control message to each active upstream node aiming to reach any related source node. This message is called RPE (Route Prediction Error).

Upon receiving a RPE, the related node sets the route state to "RTF_PREDICTION", in order to indicate that there is a soon-to-be-broken link along this route, but it is not the next hop. The expiration time of these routes is set

to the minimal value of the current expiration time and the predicted link break time. If there is any active upstream node for that route, the node should initiate its own RPE to upstream nodes. RPE propagation finishes when it reaches nodes with no active upstream node. Thus any related source node is notified of that link prediction, and all the routes stored in the routing tables of related mobile nodes are flagged by route state "RTF_PREDICTION".

If any mobile node wants to use one of these flagged routes to route its own data packet, the route rediscovery process will be triggered. But unlike the standard AODV, during the route rediscovery process, the data packet is sent immediately along the flagged route if the route expiration time is larger than the current time. If the flagged route expires when delivering a data packet in the middle of the route, the packet is dropped and standard AODV procedure after the detection of link breakage is triggered. If the flagged route expires when the source sends a data packet, the packet will be queued and the route discovery in standard AODV is initiated.

4.1.2 Route Rediscovery

Through route suspension, the source should be informed about the soon-to-be-broken link eventually. If the source has additional data packets to be sent along this route, the route rediscovery process is initiated to find another different route without any "RTF_P_LINK" link for delivering data packets. But at the same time, the data packet is sent out immediately along the soon-to-be-broken route if the route expiration time is larger than the current time.

Route rediscovery starts with broadcasting RREQ. If the current route state is "RTF_PREDICTION", which means there is a soon-to-be-broken link on this route but not the next hop, the RREQ is the same as standard AODV RREQ. If the route state is "RTF_P_LINK", which indicates the next link along this route will soon be broken, then the standard RREQ will be sent out with the mobile node address at the end of that link being attached, which includes the node itself and the current next hop. This RREQ is called P-RREQ (RREQ with Prediction). The mobile node addresses are provided in order to avoid the mobile nodes of the soon-to-be-broken link responding with RREP.

If, before the new route is established, more than one packet needs to be delivered by the source node, then RREQ will be triggered more than once as long as the old route state is "RTF_P_LINK" or "RTF_PREDICTION". This will induce unnecessary overhead, as one RREQ/RREP cycle normally is enough for building a new route. So when one RREQ is sent out, a RREQ expiration time is recorded in the routing table, only when the previous RREQ cannot return a valid RREP before that expiration time, a new RREQ will be broadcasted. This mechanism is also used in standard AODV for reducing unnecessary RREQs. The expanded ring search for broadcasting RREQ used in AODV is also used in AODV-PRM.

The main purpose of RREQ or P-RREQ is to find another route without any "RTF_P_LINK" link for future data packets. So when a mobile node receives a RREQ or P-RREQ, it will respond with RREP only if it has a valid route, not one with state "RTF_P_LINK" or "RTF_PREDICTION". An additional check must be used when a P-RREQ is received. If this P-RREQ indicates that the current node is

at the end of the soon-to-be-broken link by checking the additional node addresses in P-RREQ, this P-RREQ should be discarded to avoid constructing a route through the soon-to-be-broken link. Only the nodes not at the end of the soon-to-be-broken link can handle this P-RREQ. When an intermediate node receives a standard RREQ, or receives a P-RREQ but it is not included in that soon-to-be-broken link, it will check if there is any route with next hop in its own routing table that is in the state "RTF_P_LINK" before rebroadcasting the request If it has one, it will send out P-RREQ indicating the addresses of the nodes at the end of that particular link. In the current implementation, at most one soon-to-be-broken link can be attached in P-RREQ. So during route request propagation, RREQ may be replaced by P-RREQ, and P-RREQ may be replaced by RREQ as the node that has a route with state "RTF_P_LINK" will send out P-RREQ, while the nodes with other route states will send out RREQ. The reverse route to the source node is also set up in corresponding routing tables while propagating RREQ or P-RREQ. This reverse route is used for RREP to be routed back to the source node as used in standard AODV.

After checking the routing table and avoiding the "RTF_P_LINK" link, the RREQ or P-RREQ finally will reach the destination or some node with a fresh and stable route to the destination, where the route excludes any soon-to-be-broken link and the sequence number for the destination is equal to or larger than the destination sequence number in RREQ or P-RREQ. Then the node can initiate a RREP traveling back along the reverse route to the source node that initiates the RREP. This RREP is the same as the RREP in standard AODV.

When a mobile node receives RREP, it first will double check if the RREP is received from a soon-to-be-broken link. If so, the RREP is discarded; otherwise, the RREP can be used for updating its routing table to set up a new forward route for future data packet delivery. This check is necessary in situations where a RREQ has already been sent out, but a new soon-to-be-broken link is detected. After updating the routing table, the new forward route from the current node to the destination is established. If now this mobile node receives a data packet to that destination, no matter how the data packet arrives at this node, it will deliver the data packet to the destination through the new route. RREP finally will reach the source node that initiated the RREQ or P-RREQ. After the source node updates its routing table, the new route is completely built and the data packets will be delivered along this new route.

4.1.3 Control Message and Data Packet Delivery

This proactive route maintenance method introduces two new route states "RTF_P_LINK" and "RTF_PREDICTION". The propagation of the RREQ and the corresponding RREP messages avoid going through any "RTF_P_LINK" as described above. For unsolicited RREP, LPW and RPE messages, because their task is to notify related nodes of any possible or actual bad link, they treat routes with state "RTF_P_LINK" or "RTF_PREDCTION" as normal "RTF_UP" state, which indicates a valid route in standard AODV.

During data packet delivery, these new states are treated as the normal "RTF_UP" state. So if the state of the route for data packets is "RTF_P_LINK" or

"RTF_PREDICTION", as long as this route is not expired, the route is used just like an "RTF_UP" route.

More importantly, at any moment, at most one route to one destination is maintained in the node's routing table. So a special scenario will happen during the period after an old route is detected to be broken soon and before the new route is well established. Suppose there is a currently in-use route from S-> N1 -> N2 -> N3 -> N4 -> N5 -> D, and when data packets are delivered along this route, the N2 -> N3 link is predicted to be broken soon. So, in the routing table of node N2, the route to D with next hop N3 is set to "RTF_P_LINK"; in the routing table of node N1, the route to node D with next hop N2 is set to "RTF_PREDICTION"; and in the routing table of node S, the route to node D with next hop N1 is also set to "RTF_PREDICTION". As described before, if new data packets need to be sent from node S during the time of this route updating, RREQ is initiated, but the data packet is immediately routed if the route expiration time is larger than the current time. So the data packet will be routed along S -> N1 -> N2 -> N3 -> N4 -> N5 -> D. Suppose the new route will be S -> N9 -> N8 -> N7 -> N5 -> D. If, before the new route is built at the source node S, another data packet needs to be delivered, the data packet will also travel along the old soon-to-be-broken route. Only after the routing table in S is updated, the data packet will follow the new route. For example, if there is an already-sent packet at N4 before S constructed the new route, the packet will follow the route N4 -> N5 -> D. So there may be some data packets during a short time period that follow different routes to the destination. Because the route is constructed from the destination to the source, data packets can be

handed off smoothly to a different route.

4.2 Simulation Results and Analysis

AODV-PRM simulation is based on the same environment for AODV simulation in NS2. Because mobility is the key reason for packet losses, we design the scenarios for comparing the performance of AODV and AODV-PRM based on different mobility patterns. As indicated in Section 3.1, the mobility pattern can be determined by max movement speed and the pause time during simulation. So 9 scenarios combining three different max movement speeds and three different pause times will be simulated. The max movement speeds are 1m/s, 10m/s, and 20m/s; the pause times are: 0, 30 seconds, and 300 seconds. Table 5 summarizes other scenario parameters for AODV and AODV-PRM simulations.

Traffic Pattern	CBR
Simulation Area	1500m by 300m
Simulation Time	900 seconds
Total Nodes	50
Total Connections	20
The Data Packet Size	64 bytes
Traffic Load	4 packet/second for each connection

Table 5: AODV Scenario Parameters

With the AODV implementation parameters in Table 1 and SETUP_TIME set to 0.75 second, and the simulation parameters in Table 5 plus different pause times and max speeds, the simulation results for AODV and AODV-PRM are presented below.

4.2.1 Performance Metrics

Four metrics are used for measuring performance:

- *Data Packet Delivery Ratio*: Number of Data Packets Delivered over Number of Data Packets Generated. “Number of Data Packets Delivered” is the total number of received data packets by destinations; “Number of Data Packets Generated” is the total number of generated data packets by sources. This metric can measure the delivery reliability, the throughput of the protocol.
- *Normalized Routing Overhead*: Number of Routing Messages Transmitted divided by Number of Data Packets Delivered. “Number of Routing Messages Transmitted” is the total hops of transmitting routing control messages (RREQ, P-RREQ, RREP, RPE and LPW). So we can estimate how many transmitted routing messages are used for one successful data packet delivery by this metric to determine the efficiency and scalability of the protocol.
- *Route Optimality Ratio*: Total Number of Hops on Shortest Routes relative to Total Number of Hops on Actual Routes. With mechanisms in NS2, when a data packet is received by the destination, the hop number on an optimal route to the destination is provided, so we can use it to measure the optimality of the actual route by this metric.
- *Average End-to-End Delay*: average packet delivery time from a source to a destination. First for each source-destination pair, an average delay for packet delivery is computed. Then the whole average delay is computed

from each pair average delay. End-to-end delay includes the delay in the send buffer, the delay in the interface queue, the bandwidth contention delay at the MAC, and the propagation delay.

These metrics are not completely independent. For example, a lower packet delivery ratio means that the delay metric is evaluated with fewer samples; and the probability of a packet loss may be higher when using a longer route. The hop count is not necessarily proportional to the delay, as delay also includes delays other than propagation delay.

4.2.2 Performance Comparison when Varying Mobility

In this section, first, the four performance metrics are used to compare the average performance of AODV and AODV-PRM, resulting from the average value of 10 cases for each scenario. To get more precise results of the 10 cases, the difference of the same case for each AODV and AODV-PRM run is calculated and the average difference values are analyzed.

4.2.2.1 Average Performance for AODV and AODV-PRM

Figure 17 shows that in all scenarios, AODV-PRM has a better packet delivery ratio than AODV. The ratio of AODV-PRM is parallel to that of AODV, which indicates the correctness of the AODV-PRM implementation. For max speed set to 1m/s or 10m/s, the ratio is higher when pause time is greater, because greater pause time means more stable links. One interesting observation is the ratio at 20m/s speed and 300-second pause time, in that the ratio is lower than that of the 0 pause

time and 30-second pause time scenarios. Also the improvement of AODV-PRM for that scenario is lower when compared to the improvement for 0 pause time and 30-second pause time scenarios. The explanation is that with high-speed movement, the route cannot be maintained quickly enough compared to the sudden and quick position change, therefore resulting in more packet losses. The link state prediction method also cannot predict such sudden and quick link state change as during that change it needs to receive at least three unicast packets before predicting the link state correctly.

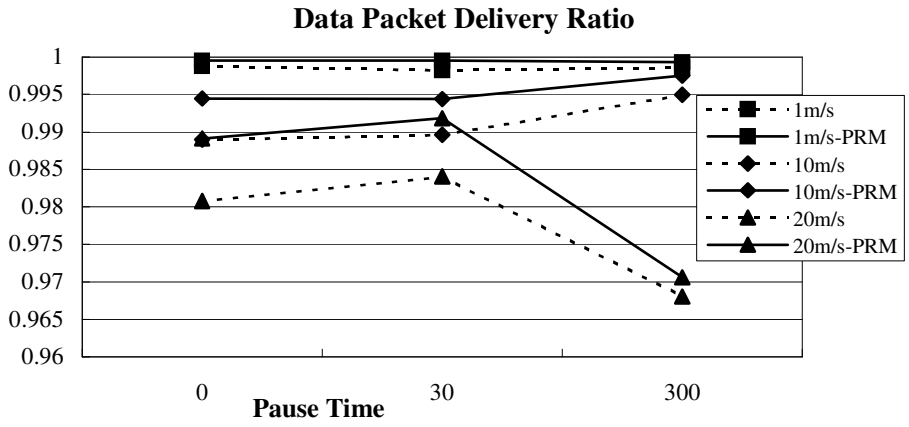


Figure 17: AODV and AODV-PRM: Packet Delivery Ratio vs. Mobility

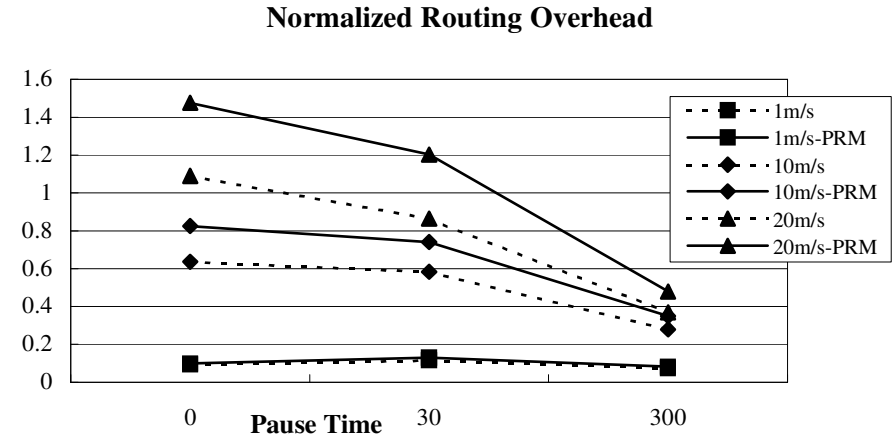


Figure 18: AODV and AODV-PRM: Normalized Overhead vs. Mobility

Figure 18 illustrates that AODV-PRM produces more control overhead than AODV in all scenarios. The difference is very small in 1m/s max speed scenarios, and becomes larger when max speed increases. The main reason is that AODV-PRM introduces more control messages to rediscover a new route even while the current route is in use. To analyze the overhead more precisely, the overhead in AODV is classified into RREQ overhead, RREP overhead (responding to the RREQs), and unsolicited RREP overhead (after link breakage detection). In AODV-PRM, P-RREQ belongs to RREQ overhead, RPE belongs to unsolicited RREP overhead, and LPW overhead is a new overhead.

	Different Overheads	RREQ (Including P-RREQ for AODV-PRM)	RREP (Only responding to RREQ)	Unsolicited RREP (Including RPE for AODV-PRM)	LPW (only for AODV-PRM)
1m/s, 300s pause time	AODV	3882.8	521.7	231.4	
	AODV-PRM	3639.1	1137.6	250.9	114.8
1m/s, 30s pause time	AODV	6091.2	911.3	406.8	
	AODV-PRM	5589.8	2095.5	476.2	187.4
1m/s, 0 pause time	AODV	5037.8	605.4	264.9	
	AODV-PRM	4480.3	1315.7	279.4	139
10m/s, 300s pause time	AODV	14511.2	2034.5	1177.7	
	AODV-PRM	14480.4	5674.5	1650.9	514.1
10m/s, 30s pause time	AODV	30386.7	4002.9	2436.6	
	AODV-PRM	30515.5	11824.4	3629.7	1089.3
10m/s, 0 pause time	AODV	33385.7	4224.9	2615.6	
	AODV-PRM	33879.1	13401.2	3924.8	1194.9
20m/s, 300s pause time	AODV	18753.9	2462.6	1398.3	
	AODV-PRM	19562.7	6987.9	2150	639.1
20m/s, 30s pause time	AODV	46888.5	6323.1	4095.4	
	AODV-PRM	48309	19676.6	6423.2	1704.1
20m/s, 0 pause time	AODV	56088.1	7308.8	4843.3	
	AODV-PRM	59325.6	23943.3	7938.8	2036.7

Table 6: AODV and AODV-PRM: Overhead Breakdown

Table 6 summarizes the average number of those overheads, which show the main increase of overhead in AODV-PRM is the RREP overhead, while RREQ is kept at the same level for AODV and AODV-PRM. One reason for the RREP overhead increase in AODV-PRM is that more nodes have routes to the destination, as all the downstream nodes after the soon-to-be-broken link along the previous route have a valid route to the destination. Another reason involves the protocol implementation, as in current standard AODV implementation, all RREP message are forwarded to the source no matter if they are useful for updating the routing table. So the RREP overhead can be reduced by only forwarding useful RREPs towards the source. Also the LPW overhead can be reduced by sending back fewer LPWs. In the current implementation, whenever a prediction occurs, a LPW is sent back. So if there are continuous predictions made when continuously receiving data packets, more than one LPW is transmitted.

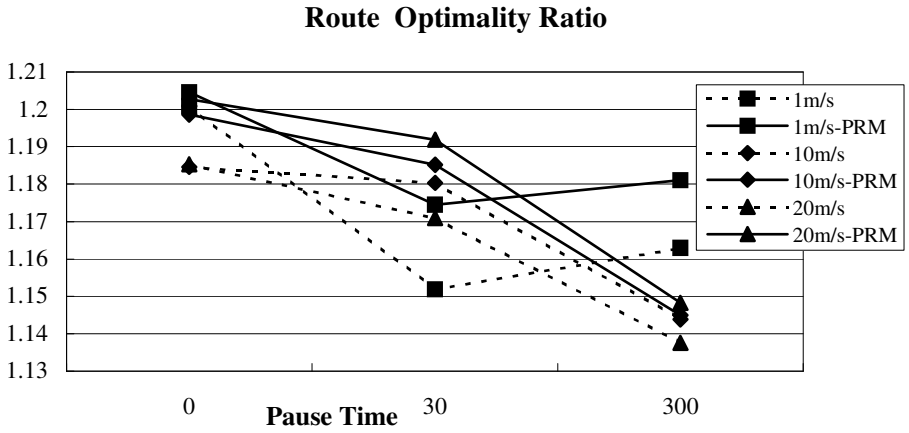


Figure 19: AODV and AODV-PRM: Route Optimality Ratio vs. Mobility

Figure 19 indicates that AODV-PRM has higher route optimality ratio than AODV in all scenarios, although the difference is rather small. When in route

rediscovery phase in AODV-PRM, for the nodes along the soon-to-be-broken route, the nodes after the link that will soon be broken all have a valid route to the destination, while the nodes before that link cannot respond to the new RREQ with prediction. Other unrelated nodes may have a route to the destination as they participate in the discovery of the current used route, but most of them are expired. So it is likely that the nodes on the current used route but after that soon-to-be-broken link will respond to the new route rediscovery by sending back RREP, which always results in non-optimal routes with longer hop count compared to the soon-to-be-broken route. Because the route discovery not only depends on hop counts from the source to the destination, but also on network traffic load at that moment. Also, when you find a route, you use it, and there is no way of knowing if a shorter route has become available. So the Route Optimality Ratio cannot equal to 1 even for standard AODV.

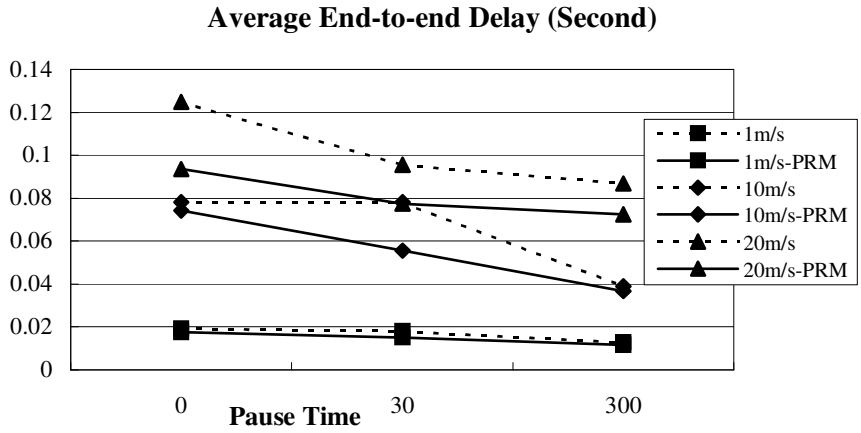


Figure 20: AODV and AODV-PRM: Average End-to-end Delay vs. Mobility

Figure 20 presents the packet end-to-end delay for AODV and AODV-PRM. In all scenarios, the end-to-end delay in AODV-PRM is smaller than that in AODV.

For lower speed, the difference is rather small. As the end-to-end delay includes delay in queues while discovering a route, and the transmission delay, which mainly depends on the traffic load, so although the Route Optimality Ratio is higher in AODV-PRM than in AODV, AODV-PRM outperforms AODV in terms of the delay. All delays are under 0.14 second, so for SETUP_TIME set to 0.75 second, there is enough time for LPW and RPE propagating back to the source and the source finding a new route.

4.2.2.2 Differences for Performance of AODV and AODV-PRM

In this section, the difference between the performances of AODV and AODV-PRM is presented, which is calculated based on each case for each scenario. While the results in Section 4.2.2.1 give a general performance comparison, Table 7 and Table 8 list the average value and 95% confidence interval for relative differences computed for each case for each scenarios. Performance differences between AODV and AODV-PRM are presented as percentage decrease of packet loss and end-to-end delay, and percentage increase of normalized overhead and route optimality ratio. We use the improvement of packet loss rather than the improvement of packet delivery ratio because the delivery ratios of both AODV-PRM and AODV are above 96% and very close. All the percentages are calculated based on the standard AODV simulation results in the form of $\frac{AODVPRM - AODV}{AODV}$.

In Table 7, the average results are consistent with the results for the general performance comparison of AODV and AODV-PRM except the end-to-end delay

when in scenarios with 1m/s max speed and 300-second pause time, and 10m/s max speed and 300-second pause time. For the general performance, the end-to-end delay is smaller in AODV-PRM than that in AODV for all scenarios. But in Table 7, for 300-second pause time, and max speeds of 1m/s and 10m/s, the end-to-end delay in AODV-PRM is greater than that in AODV, even though the difference is very small.

		1m/s	10m/s	20m/s
		Average	Average	Average
0 pause time	Packet Loss Decrease	59.83%	50.23%	43.72%
	Normalized Overhead Increase	4.20%	29.41%	35.19%
	Route Optimality Increase	0.36%	1.20%	1.46%
	End-to-end Delay Decrease	4.12%	1.96%	22.26%
30-second pause time	Packet Loss Decrease	71.34%	46.04%	48.77%
	Normalized Overhead Increase	11.79%	27.18%	48.64%
	Route Optimality Increase	1.98%	0.41%	1.80%
	End-to-end Delay Decrease	13.45%	28.72%	18.05%
300-second pause time	Packet Loss Decrease	65.11%	50.28%	32.79%
	Normalized Overhead Increase	12.47%	26.46%	29.88%
	Route Optimality Increase	1.58%	0.12%	0.94%
	End-to-end Delay Decrease	-2.66%	-0.46%	15.14%

Table 7: Summary of AODV-PRM Performance: Average Values

		1m/s	10m/s	20m/s
		95% Confidence Interval	95% Confidence Interval	95% Confidence Interval
0-second pause time	Packet Loss Decrease	(53.19%, 66.47%)	(46.46%, 53.99%)	(39.06%, 48.39%)
	Normalized Overhead Increase	(-5.71%, 14.13%)	(25.14%, 33.69%)	(29.46%, 40.92%)
	Route Optimality Ratio Increase	(-1.34%, 2.07%)	(0.46%, 1.94%)	(0.75%, 2.17)
	End-to-end Delay Increase	(-14.71%, 22.95%)	(-9.53, 13.45%)	(12.27%, 32.25%)
30-second pause time	Packet Loss Decrease	(67.77%, 74.91%)	(36.43%, 57.45%)	(44.99%, 52.57%)
	Normalized Overhead Increase	(5.17%, 18.40%)	(20.43%, 33.92%)	(13.87%, 83.40%)
	Route Optimality Ratio Increase	(0.71%, 3.25%)	(-0.26%, 1.08%)	(1.07%, 2.53%)
	End-to-end Delay Decrement	(-0.26%, 27.15%)	(20.23%, 37.22%)	(13.87%, 22.23%)
300-second pause time	Packet Loss Decrement	(54.22%, 75.99%)	(39.38%, 61.17%)	(18.52%, 47.05%)
	Normalized Overhead Increment	(3.16%, 21.77%)	(16.41%, 36.52%)	(24.92%, 34.84%)
	Route Optimality Ratio Increment	(0.29%, 2.86%)	(-0.78%, 1.03%)	(0.42%, 1.46%)
	End-to-end Delay Decrement	(-23.40%, 18.09%)	(-22.87%, 21.95%)	(7.83%, 22.44%)

Table 8: Summary of AODV-PRM Performance: Confidence Intervals

From Table 8, we can see the difference from case to case for each scenario. For packet loss, no matter what scenario, all cases present that AODV-PRM has less losses than AODV. For normalized overhead, some cases in scenario with 0

pause time and 1m/s max speed, even show overhead decrease in AODV-PRM, although the average differences always present that AODV-RPM has more overheads than AODV. For route optimality ratio, although AODV-RPM has greater average ratio than AODV, some cases produce smaller ratio in AODV-PRM in scenarios: 1m/s max speed and 0 pause time, 10m/s max speed and 30-second or 300-second pause times. For end-to-end delay, the differences from case to case in one scenario are relatively large. Only in scenarios with 20m/s max speed and all different pause times, and 10m/s max speed and 30-second pause time, do all cases result in smaller delay in AODV-PRM than in AODV. In other scenarios, except 30-second pause time and 1m/s max speed, the delay improvements are not stable and show no clear trend.

Summary:

From the simulation results, AODV-PRM does significantly reduce No-Route packet losses and improves the data throughput, while the control message overhead in AODV-PRM increases. The packet loss improvement is between 32% and 72%, and the increase of overhead is between 4% and 49%. The route optimality ratio of AODV-PRM is slightly greater than that of AODV with the increase below 2%. At fast speed, the end-to-end delay in AODV-PRM is smaller than that in AODV, but at low speed, the end-to-end delays in AODV-PRM keeps the same level as those in AODV, indicating no big change and no clear trend. In addition, the results show that the number of hops is not necessarily proportional to the end-to-end delay, which is also mentioned in [32] when comparing DSR and

AODV.

DSR is another reactive unicast protocol, but implements source routing and has a salvaging mechanism to repair routes locally. It also can have better data throughput when using this link state prediction method [33], in which the decrease of packet loss is between 12% and 34%, while the overhead increases between 18% and 24%.

In the next chapter, the same link state prediction method will be implemented for multicast protocol MAODV and the simulation results will be analyzed.

Chapter 5 Proactive Tree Maintenance in MAODV

In this chapter, a proactive tree maintenance mechanism for MAODV by using the link state prediction method (equation E3) is proposed and analyzed with NS2. The modified protocol is called MAODV-PTM (MAODV with Proactive Tree Maintenance).

5.1 MAODV-PTM Description

The main difference between standard MAODV and MAODV-PTM is in the maintenance of the multicast tree. In standard MAODV, as described in Chapter 3, the link breakage is detected by MAC layer feedback when trying to send out unicast packets, not using the mechanism of periodic neighbor HELLO messages. After the detection, the node that wants to transmit the packet notices the link breakage, and the tree is maintained locally if the node is the downstream node of that link. In MAODV and MAODV-PTM, the link direction downstream or upstream is measured according to the node's hop count to the multicast group leader. If the node at the end of the link has a greater hop count to the multicast group leader, the node is the downstream node to the other node at the other end of the link, and the other node acts as the upstream node. This upstream/downstream concept is quite different to the upstream/downstream concept used in AODV and AODV-PRM, where is refers to the data traffic direction as the data packets generated at the source node travel from upstream nodes to downstream nodes and reach the destination. In a tree structure, as the group leader is the root of the tree, any node can only have at most one active upstream neighbor node. If the node is

the upstream node, the link breakage is just used to update its multicasting routing table to indicate the next hop through the broken link becomes unavailable. If the node is the downstream node, besides updating the next hop to be unavailable, the node initiates tree repair by sending out RREQ with its own hop count to the group leader being attached, which is necessary to avoid its own downstream nodes responding to the RREQ. The main idea for tree maintenance in MAODV-PTM is to keep the branch being connected to the tree while the link breakage occurs. The link breakage is detected by using link state prediction Equation E3, thus before the tree branch becomes disconnected, a new route to the group leader can be discovered in advance. Figure 21 illustrates the tree maintenance procedure in MAODV-PTM.

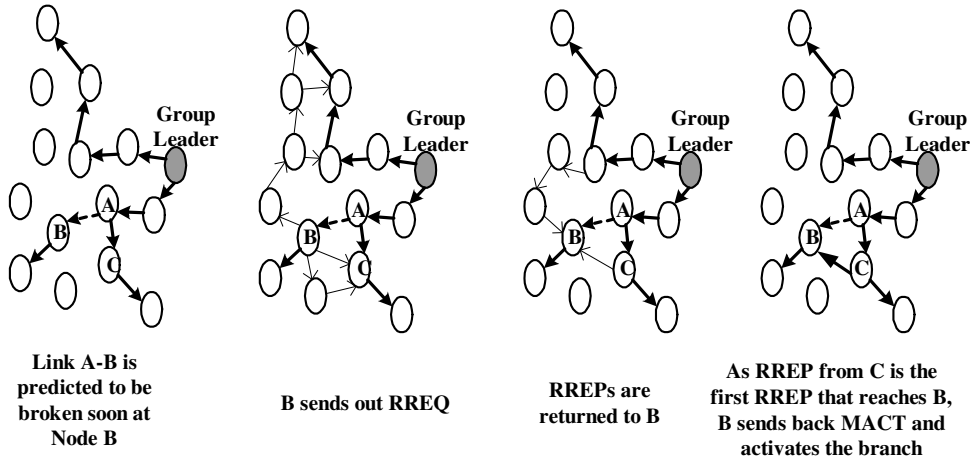


Figure 21: MAODV-PTM Proactive Tree Maintenance

The implementation of link state prediction method is the same as that in AODV-PRM. The tree maintenance of MAODV-PTM includes two phases: local suspension and tree branch reconnection, which are described below.

5.1.1 Local Suspension

Local Suspension is triggered when a node on the multicast tree detects that the link from which it receives the data packet will become broken soon (determined by SETUP_TIME plus the current time). If the node is the upstream node of that soon-to-be-broken link, that is, if the node is closer from the group leader than the other node at the other end of the link, it first will update that link in the group tree with state "NH_DUP", and then notify the downstream node by sending a new one-hop message DD-MACT. If the node is the downstream node of that link, it will notify the upstream node by sending a new one-hop message DU-MACT. If the upstream node receives DU-MACT, it also will update the next hop through the soon-to-be-broken link with state "NH_DUP". No matter whether this node is upstream or downstream, in the multicast routing table, the next hop thought that soon-to-be-broken link is set to "NH_DUP" state for that multicast group tree.

By using the notification from both nodes at the end of the link, no matter the direction of the data traffic, the downstream node will always be notified of the bad link. That is much better than standard MAODV, in the situation that the data traffic always flows from upstream node to downstream node. If that situation occurs in MAODV, the link breakage cannot initiate local repair for the tree and actually causes tree partition unknown by the downstream node, which results in data packets always being partially delivered, thus not all group members can receive data packets.

In MAODV-PTM, except for the two nodes at the end of the

soon-to-be-broken link, other nodes on the tree do not realize that link state change, unlike the route suspension in AODV-PRM, in which all related routes know about that link and the sources initiates route rediscovery. Because for MADOV, the route construction is based on a tree rooted at the group leader, not simply combined by the routes from all senders to all receivers. The group leader is responsible for managing multicast group sequence number, and only initiates the tree structure maintenance when a tree merger occurs. When link breakage occurs, in MAODV, the broken branch is maintained locally if the downstream node needs a route to the group tree. So, in MAODV-PTM, only the nodes at the end of the soon-to-be-broken link need to know the link state change and the downstream node initiates branch reconnection if necessary.

When a node has a next hop with state "NH_DUP", the data packets can be delivered through that next hop as the state indicates that now the next hop is still available but will be broken in the near future.

5.1.2 Branch Reconnection

After the downstream node of that soon-to-be-broken link knows the link state change, if it still needs a route to the multicast group tree, it will initiate a broadcast route request to find a new route for the branch reconnection. In general, all nodes on the tree should maintain their routes, because the leaf nodes of the tree are always group members.

To limit unnecessary broadcast route requests initiated at the downstream, like the RREQ broadcast in AODV, an expiration time is set for every request. Only

when the previous request cannot invoke any valid reply, a new request will be sent out. The expanded ring search is also used for reducing broadcast traffic.

Branch Reconnection makes use of the local tree repair method in MAODV. The route request for the branch reconnection is the same as the J-RREQ used in MAODV when a link breakage actually happens, in which its own hop count to the group leader is attached to normal J-RREQ, in order to prevent its own downstream nodes from sending back a RREP. Like the tree repair in MAODV, any node that is a tree member with smaller hop count to the group leader and fresh enough route can respond with RREP; otherwise, the J-RREQ is broadcast again. But before sending a RREP or a J-RREQ, the node will check if the next hop to the reverse unicast route is a "NH_DUP" link. If it is, the J-RREQ will be discarded, the RREP cannot be sent back.

The available RREP is propagated back along the reverse unicast route to the downstream node that initiates the J-RREQ. During the propagation, the branch to the group leader is formed by recording corresponding upstream/downstream relationships between intermediate nodes, but the branch is not activated until a MACT is received. More importantly, at every intermediate node, the state of the link on the reverse route will be checked again to avoid any "NH_DUP" link. If any "NH_DUP" link is found, the RREP is discarded. Thus when the downstream node of that soon-to-be-broken link receives a RREP, a potential available branch is formed. Unlike the tree repair in MAODV, once that downstream node receives a RREP, it will immediately send J-MACT to the potential upstream to activate that branch and send U-MACT to its active downstream nodes to update the hop count

to the group leader. If other RREPs are received later, those RREPs are discarded as now the downstream node already has one active upstream neighbor.

Because the procedure of branch construction in MAODV is from the downstream node to the upstream node, when the link at the downstream node becomes active, the corresponding link at the upstream node is still unavailable. At that point, if the old upstream neighbor (containing that soon-to-be-broken but currently being active link) is set to be unavailable, the data packet may not be delivered between the nodes at the downstream of that link and the nodes at the upstream of that link. So when the new upstream hop is activated, the old upstream hop is also kept. The old upstream link with state "NH_DUP" becomes unavailable when the link actually becomes broken, which is detected by the link layer when the node tries to send data packets along that hop but cannot.

5.1.3 Control Message and Data Packet Delivery

The RREQ/RREP cycle tries to avoid any "NH_DUP" link to construct a stable route. But for J-MACT messages used for activating the route, they pass through the "NH_DUP" link as a normal valid link. Because when the upstream node receives a MACT from a downstream node, the downstream node already activates that branch. An alternative solution is that the upstream node sends back a control message to cancel that activation, but this method is not used in the current implementation. A "NH_DUP" link is also treated as a normal valid link when propagating other control messages, such as U-MACT used for updating the group leader address and the hop count to group leader, P-MACT and GL-MACT used

for pruning tree and finding a new group leader, and GRPH used for broadcasting group information.

For data delivery, during the period that we try to reconnect the branch, it is necessary to send and receive any valid data packet through a "NH_DUP" link. Because when the new route is formed, the old route also remains, so during the period where the two upstream hops are both active, duplicate data packets may be received. To exclude the duplicate packets, a cache is used for recording the IDs of the received data packets. When receiving a data packet, the cache is checked first. If the packet ID is not in that cache, then the packet can be handled by the node, and the packet ID is recorded in that cache; otherwise, the packet is discarded.

5.2 Simulation Results and Analysis

MAODV-PTM simulation is based on the same environment for MAODV simulation in NS2. For multicast protocols, the main factors that affect the protocol performance are the size of the multicast group, the number of data packet senders, and the node movement pattern. So we compare MAODV and MAODV-PTM with regard to those three factors. The implementation parameters are listed in Table 1 and Table 2. The basic scenario parameters are listed in Table 9 with the size of the group set to 20 nodes, the number of data packet senders set to 5 nodes (all are group members), and the node movement pattern set to 20m/s with 0 pause time. When varying the size of group members, other scenario parameters are kept fixed. The same applies for varying the number of data packet senders, and varying the node mobility patterns. The SETUP_TIME for initiating link suspension is set to 1

second as discussed in Section 3.3.

Traffic Pattern	CBR
Simulation Area	1000m by 1000m
Simulation Time	1500 seconds
Total Nodes	50
Total Groups	1
Total Group Members	5 ～ 20
Total Senders	1 ～ 20 Note: All senders must be members
Pause Time	0 second
Node Mobility Max Speed	0 ～ 20 meter/second
Traffic Load	20 packet/second
The Number of Cases for one Scenario	10

Table 9: MAODV Scenario Parameters

5.2.1 Performance Metrics

Similar to the metrics for AODV and AODV-PRM, four metrics are used for measuring MAODV and MAODV-PTM performances:

- *Data Packet Delivery Ratio*: Number of Data Packets Delivered to the Receivers divided by the result of the Number of Data Packets Generated multiplied by the Number of Receivers. This metric measures the throughput of the protocol, in which a successful delivery requires that all receivers receive that packet.
- *Normalized Routing Overhead*: Number of Control Messages Transmitted divided by the result of the Number of Data Packets Delivered to the Receivers divided by the Number of Receivers. “The Number of Data Packets Delivered to the Receivers divided by the Number of Receivers” is a metric to show how many packets have been successfully received.
- *Average Hop Count*: average hop count of packet delivery for

sender-receiver pairs. For example, assume there are 5 senders, all of which are members, and there are 20 group members. So for each packet, there are 19 receivers excluding the sender themselves. So there are $5 \times 19 = 95$ pairs. And for each pair, an average hop count is computed. Then the whole average hop count is calculated among the average hop count for each pair.

- *Average End-to-End Delay:* average end-to-end delay of packet delivery for sender-receiver pairs. The calculation method is the same as the method calculating average end-to-end delay in AODV.

As mentioned for metrics used by AODV and AODV-PRM, the metrics used for evaluating MAODV and MAODV-PTM are also not completely independent. For multicasting protocol, other factors also exist. For example, for metrics on the average hop count and the average end-to-end delay, if a tree partition occurs, for the same packet, some receivers receiving the packet will have more samples, other receivers that did not receive that packet will have less samples.

5.2.2 Performance Comparison when Varying Mobility

Node mobility is a key reason resulting in packet losses and partial packet delivery, so in this section, the performance under different mobility patterns is presented. There are 2 factors for designing mobility pattern: pause time and max speed of node movement. Here, we set the pause time to the fixed value 0 second and choose the max speeds as 1m/s, 5m/s, 10m/s, and 20m/s.

Figure 22 shows the packet delivery ratio comparison of MAODV and

MAODV-PTM under different max speeds. As node movement results in tree disconnection and repair, both ratios reduce when the max speed increases. In all scenarios, MAODV-PTM outperforms MAODV and keeps the ratio above 90%. But the ratio for MAODV drops sharply to 67% in the 20m/s max speed scenario, which is much worse than the unicast protocol AODV. So the improvement by using MAODV-PTM is quite significant when the nodes move faster.

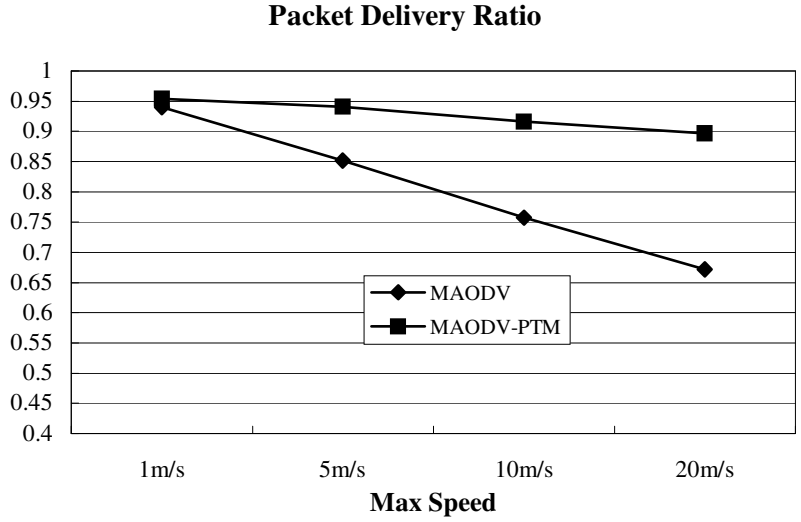


Figure 22: MAODV and MAODV-PTM: Packet Delivery Ratio vs. Max Speeds

Figure 23 demonstrates that the normalized overhead of MAODV-PTM is less than that of MAODV no matter the choices of max speed. The normalized overhead increases when max speed increases for both MAODV and MAODV-PTM. The normalized overhead is a relative metric based on the number of successful data packet delivery, which can be measured by packet delivery ratio. If the packet delivery ratio is quite different as shown in Figure 22, the absolute value of the overhead also needs to be considered. Table 10 lists the average absolute overheads of MAODV and MAODV-PTM under different max speeds,

which indicates the overheads for both are at the same level with slight difference, so the difference between the normalized overheads of MAODV and MAODV-PTM is mainly due to the difference of their packet delivery ratios.

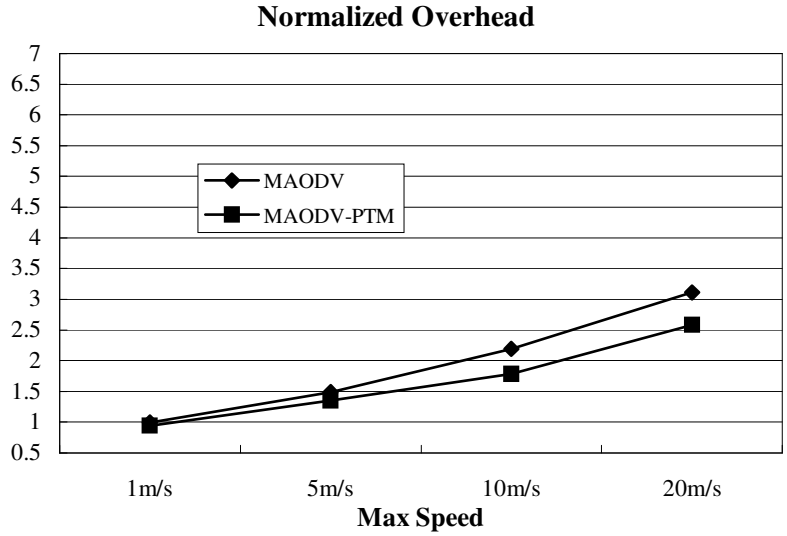


Figure 23: MAODV and MAODV-PRM: Normalized Overhead vs. Max Speeds

		1m/s	5m/s	10m/s	20m/s
Absolute Overhead	MAODV	27372.2	37211.9	48440.8	61171.1
	MAODV-PTM	26367.6	37241.7	47964.5	67857.8
Hops on Tree	MAODV	24.9594	22.8540	20.5646	17.8919
	MAODV-PTM	25.3660	26.8514	27.0690	27.4408
One-hop Delay (second)	MAODV	0.00659	0.00663	0.00647	0.00643
	MAODV-PTM	0.00697	0.00801	0.00923	0.01470

Table 10: MAODV and MAODV-PTM: Other Results under Max Speeds

The average hop count as a function of max speed is illustrated in Figure 24. The hop counts for both MAODV and MAODV-PTM are rather stable, not significantly affected by different max speeds. Except for the 1m/s scenario, the hop count of MAODV-PTM is larger than that of MAODV, and as the max speed increases, the difference becomes bigger. It is because in MAODV-PTM, as more successful data deliveries are accomplished, the multicast tree may be larger to

reach the members that cannot be reached by MAODV. The average hops on the multicast tree can measure how large the tree is. Two reasons can cause the hop count on the tree to become bigger. One is more branches, the other is longer branch. From Table 10, we can see that the tree in MAODV-PTM is larger than the tree in MAODV, and the difference between them is quite big when in a scenario with 20m/s max speed.

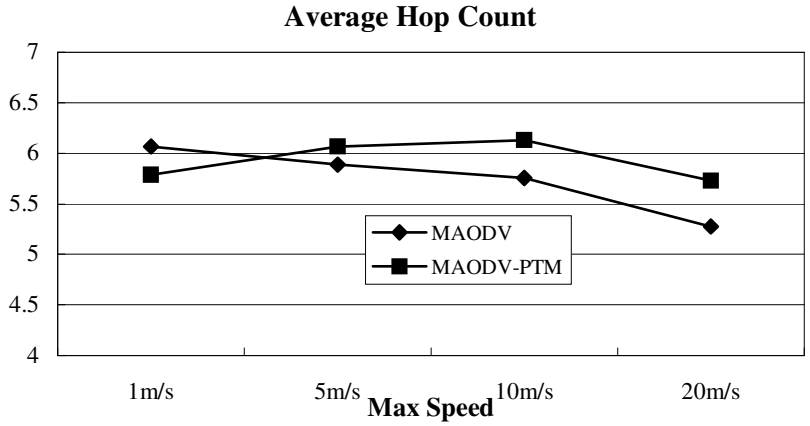


Figure 24: MAODV and MAODV-PTM: Average Hop Count vs. Max Speed

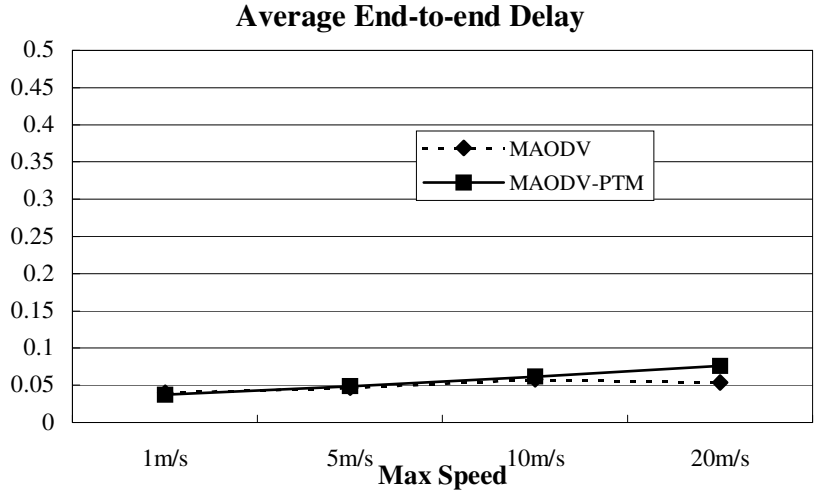


Figure 25: MAODV and MAODV-PTM: End-to-end Delay vs. Max Speed

Figure 25 illustrates the average end-to-end delay for MAODV and MAODV-PTM in terms of different max speeds. The delays of both protocols increase as the max speed increases, while the delay in MAODV-PTM increases much faster. Like the hop count comparison in Figure 24, except in the 1m/s scenario, MAODV-PTM needs a longer time to deliver data packets than MAODV. One reason for longer delay is that the hop count becomes larger. Another reason is that while more data packets are delivered in MAODV-PTM, there may be more active neighbors at one specific intermediate node. As all packets are delivered via unicast, this may incur longer delays in queuing and transmission contention. The average one-hop delays listed in Table 10 prove that the one-hop delay in MAODV-PTM is longer than that in MAODV, and increases faster when nodes move faster.

		1m/s	5m/s	10m/s	20m/s
Packet Delivery Ratio Increase	Average	1.42%	8.87%	15.95%	22.42%
	Confidence Interval	(0.22%, 2.62%)	(7.33%, 10.38%)	(13.87%, 18.03%)	(20.32%, 24.53%)
Absolute Overhead Decrease	Average	3.45%	-0.21%	0.76%	-11.17%
	Confidence Interval	(-0.90%, 7.80%)	(-2.99%, 2.56%)	(-2.31%, 3.82%)	(-15.87%, -6.47%)
Hop Count Decrease	Average	5.18%	-2.87%	-6.07%	-7.93%
	Confidence Interval	(0.89%, 9.47%)	(-6.30%, 0.56%)	(-9.71%, -2.44%)	(-9.81%, -6.06%)
End-to-end Delay Decrease	Average	11.68%	-3.76%	-4.87%	-28.21%
	Confidence Interval	(-12.44%, 35.79%)	(-11.21%, 3.7%)	(-17.01%, 7.27%)	(-34.18%, -22.23%)

Table 11: Average and 95% Confidence Interval for Performance Changes of MADOV-PTM based on MAODV under Different Max Speeds

Table 11 summarizes the differences of the delivery ratio in the form of *MAODVPTM–MAODV*, and the other three metrics in the form of

$\frac{MAODVPTM - MAODV}{MAODV}$ under different max speeds between MAODV-PTM and MAODV by comparing them case-by-case, which shows the packet delivery ratio always is improved in MAODV-PTM. Here, unlike comparing the decrease of packet losses in AODV-PRM, we use the improvement of packet delivery ratio instead, because the increase of packet delivery ratio is significant. Also, the absolute overhead difference is used instead of the normalized overhead difference when comparing AODV and AODV-PRM, because as the improvement of packet delivery ratio is very large, the comparison of normalized overheads in MAODV and MAODV-PTM is not important. All the results are consistent with the general performance results of MAODV-PTM and MAODV.

5.2.3 Performance Comparison when Varying Group Size

By varying the multicast group size, the traffic becomes more and more intense in the network. If all the nodes in the network are group members, we effectively have a broadcast scheme. As in our simulation, all senders should be group members, so we choose scenarios with the number of group members set to 5,10, 20, and 50.

Figure 26 illustrates the packet delivery ratio of MAODV and MAODV-PTM under different group size. As the number of group members increases, and the multicast tree becomes larger, the performances of both protocols degrades. In all scenarios, MAODV-PTM outperforms MAODV. In the scenarios of 5, 10, or 20 group members, the ratio is about 90% ~ 95% for MAODV-PTM, and about 67% ~ 76% for MAODV. For the 50 group members scenario, both ratios reduce

significantly, especially in MAODV-PTM. It is mainly because all the nodes are on the multicast tree and more tree branches are formed. In this situation, node movement may result in difficulty to update the tree structure and thus produces a lot of control overhead. Also the unicast data packet delivery may incur congestion at a particular node, because of more downstream branches.

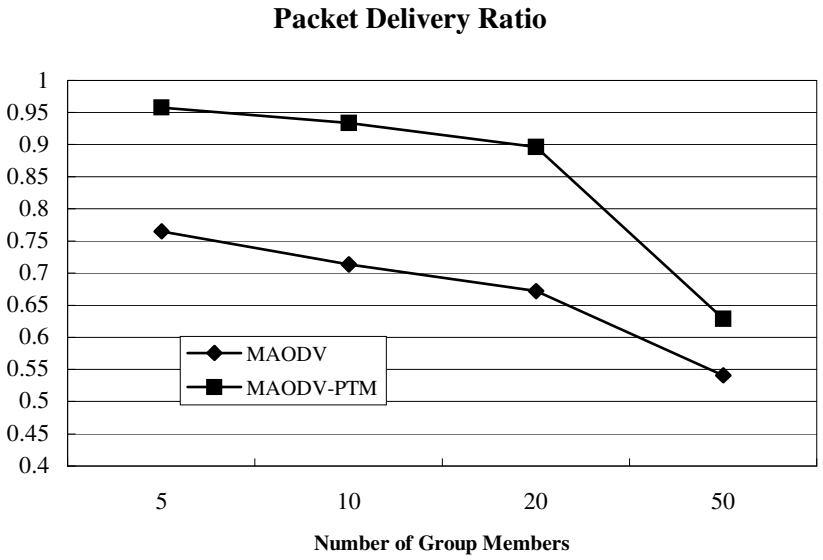


Figure 26: MAODV and MAODV-PTM: Packet Delivery Ratio vs. Number of Group Members

The normalized overheads of MAODV and MAODV-PRM as a function of the number of group members are given by Figure 27, in which MAODV-PTM has lower normalized overhead than MAODV in all scenarios. But the normalized overheads of both protocols increase as the number of group members increases, especially for the scenario with 50 group members, in which the overhead increases sharply. The reason is mentioned before, which includes congestion and more absolute overhead. Additionally, Table 12 lists the absolute value of overhead in MAODV and MAODV-PTM. The absolute value of overhead in MAODV-PTM is

slightly greater than the value in MAODV. Because MAODV-PTM has higher packet delivery ratio, the normalized overhead of MAODV-PTM is less than that of MAODV.

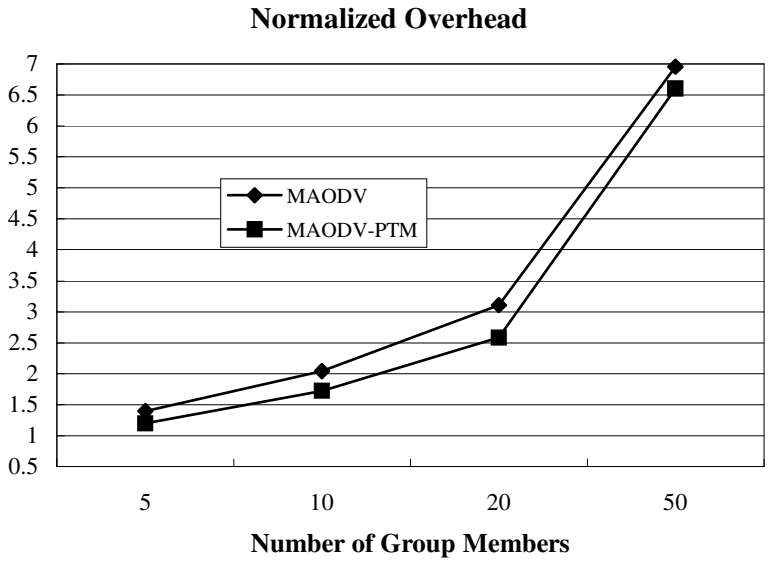


Figure 27: AODV and AODV-PTM: Normalized Overhead vs. Number of Group Members

		5 members	10 members	20 members	50 members
Absolute Overhead	MAODV	31206.4	42728.2	61171.1	110400
	MAODV-PTM	33812.7	47262.4	67857.8	121965.5
Hops on Tree	MAODV	8.6308	12.4211	17.2919	26.2683
	MAODV-PTM	13.0433	19.7380	27.4408	33.4371
One-hop Delay (second)	MAODV	0.002927	0.004192	0.006426	0.026132
	MAODV-PTM	0.004250	0.006608	0.01470	0.080325

Table 12: MAODV and MAODV-PTM: Other Results under Different Number of Group Members

Figure 28 presents the average hop count for MAODV and MAODV-PTM according to multicast group size. Although the difference of the hop counts in MAODV-PTM and MAODV is at most 1 hop, MAODV-PTM has a larger average hop count than MAODV. As explained above, MAODV-PTM has large hop count

because it may maintain longer branches than MAODV if it has a better packet delivery ratio, which is demonstrated by the hops on the tree in Table 12. In the 50 group members scenario, the difference of both hop counts is relatively small. It is because for both protocols the tree is not well maintained in such a broadcasting environment.

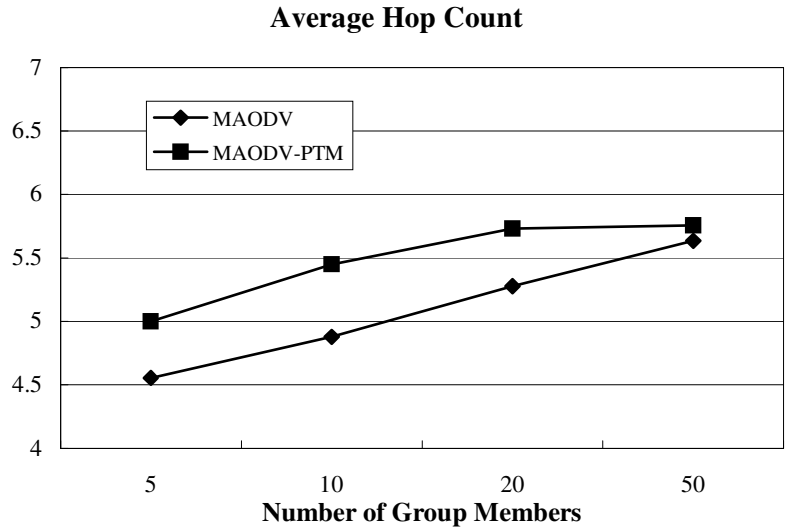


Figure 28: MAODV and MAODV-PTM: Average Hop Count vs. Number of Group Members

The average end-to-end delays as a function of the number of group members are shown in Figure 29. As more and more nodes become group members, the end-to-end delay increases for both MAODV and MAODV-PTM. The sharp increase of the delay for MAODV-PTM in the 50 group members scenario indicates the unicast data packet delivery is not suitable for delivering packets a in broadcast environment.

Table 13 lists the differences of the four metrics under different number of group members between MAODV-PTM and MAODV by comparing them case-by-case, which shows consistency with the general performances of

MAODV-PTM and MAODV. If 0 is not in the confidence interval, this indicates that the results from all cases show a statistically significant trend. If 0 is included in the interval, the case-by-case comparison is not conclusive and shows no clear trend.

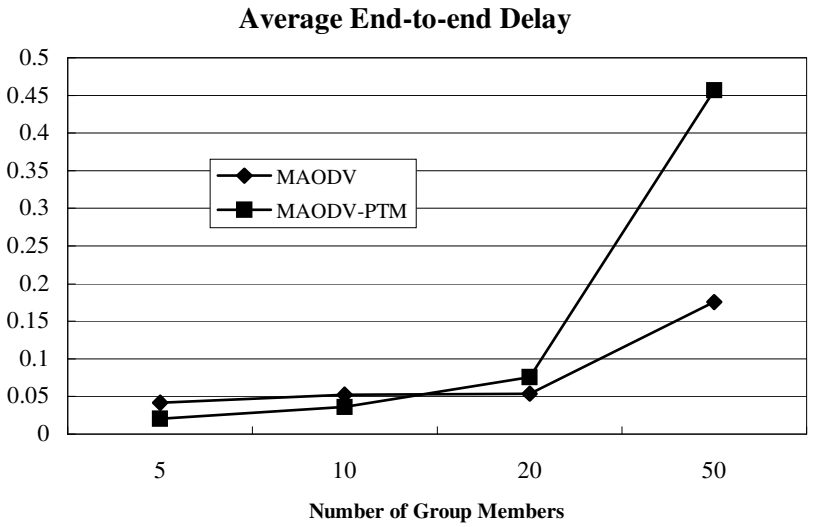


Figure 29: MAODV and MAODV-PTM: End-to-end Delay vs. Number of Group Members

		5-member	10-member	20-member	50-member
Packet Delivery Ratio Increase	Average	19.26%	21.94%	22.42%	8.76%
	Confidence Interval	(17.60%, 20.93%)	(20.06%, 23.82%)	(20.32%, 24.53%)	(7.37%, 10.15%)
Absolute Overhead Decrease	Average	-8.31%	-10.64%	-11.17%	-10.51%
	Confidence Interval	(-11.20%, -5.41%)	(-15.73%, -5.56%)	(-15.86%, -6.47%)	(-13.39%, -7.62%)
Hop Count Decrease	Average	-8.53%	-10.50%	-7.93%	-2.06%
	Confidence Interval	(-13.84%, -3.21%)	(-12.13%, -8.86%)	(-9.81%, -6.06%)	(-5.06%, 0.93%)
End-to-end Delay Decrease	Average	111.18%	45.21%	-28.21%	-61.41%
	Confidence Interval	(63.21%, 159.25%)	(21.23%, 69.19%)	(-34.18%, -22.23%)	(-63.45%, -59.36%)

Table 13: Average and 95% Confidence Interval for Performance Change of MADOV-PTM based on MAODV under Different Group Size

5.2.4 Performance Comparison when Varying Number of Senders

By increasing the number of senders, the traffic becomes more and more decentralized. As in our simulation, all senders should be group members, so we choose scenarios with the number of senders set to 1, 5, 10, and 20. As the traffic load is fixed at 20 packets per second, for scenarios with different number of senders, the traffic from a sender should be 20/(Number of Senders) packets per second.

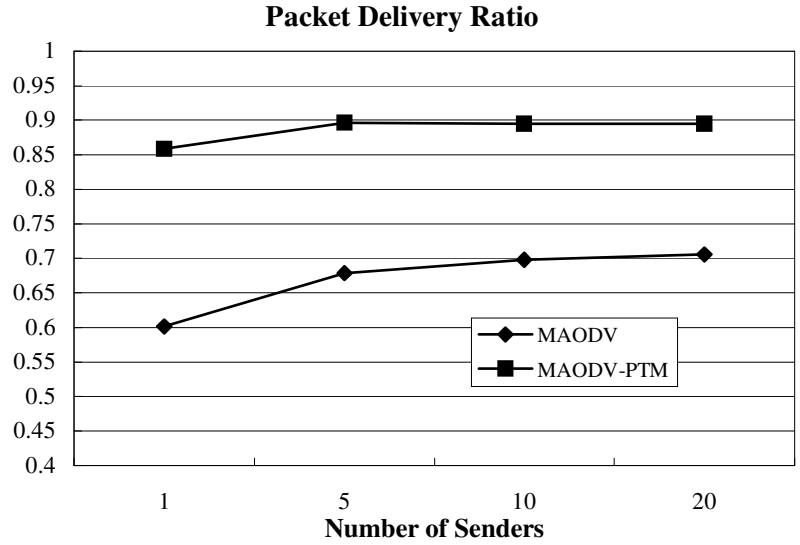


Figure 30: AODV and AODV-PRM: Packet Delivery Ratio vs. Number of Senders

Figure 30 illustrates the packet delivery ratio of MAODV and MAODV-PTM as a function of the number of multicast senders. In the scenarios of 5, 10, or 20 sources, the performance is rather stable, with MAODV-PTM at 90%, and MAODV around 70%. For the 1-sender scenario, the performance degrades as the traffic direction is centralized from one node. For the current implementation, we avoid the sender being the group leader in the 1-sender scenario, which may

significantly reduce the performance of MAODV, because the local tree repair is initiated at the downstream node. If only the upstream node knows about a link breakage, the multicast tree is just partitioned and may not be repaired for a long time (until the link times out).

The comparison of normalized overheads of MAODV and MAODV-PRM is given by Figure 31, in which MAODV-PTM outperforms MAODV. For the 1-sender scenario, the improvement is significant. Table 14 lists the absolute value of overheads in MAODV and MAODV-PTM, from which we can see that, except for the 1-sender scenario, the absolute value of overheads in MAODV-PTM is slightly greater than the value in MAODV. With consideration of packet delivery ratio, the 1-sender scenario in MAODV-PTM achieves the significant improvement due to the great packet delivery ratio improvement and lower overheads.

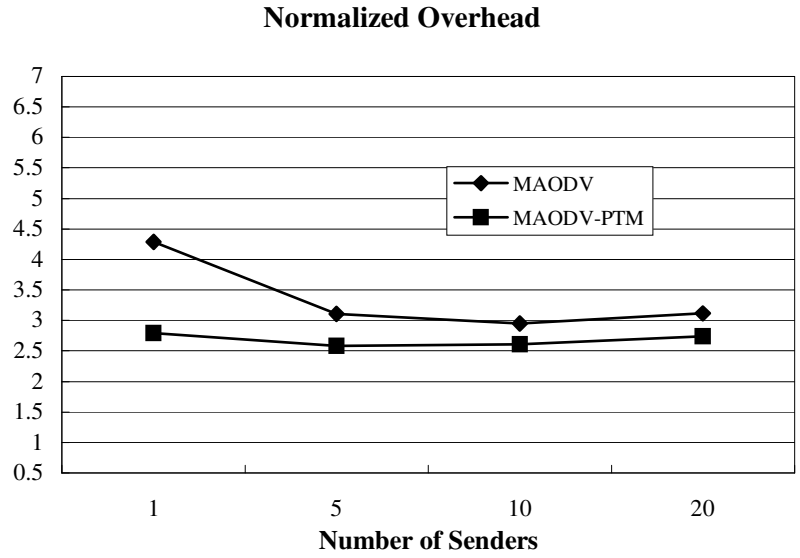


Figure 31: AODV and AODV-PRM: Normalized Overhead vs. Number of Senders

		1 sender	5 sender	10 sender	20 sender
Absolute Overhead	MAODV	74075	61171.1	60182.9	64178.5
	MAODV-PTM	70080.9	67857.8	68356	71839.1
Hops on Tree	MAODV	15.1680	17.8919	18.9579	19.9293
	MAODV-PTM	25.7090	27.4408	27.5211	27.5665
One-hop Delay (second)	MAODV	0.005898	0.006426	0.007209	0.008078
	MAODV-PTM	0.017117	0.014695	0.016358	0.017985

Table 14: MAODV and MAODV-PTM: Other Results under Different Number of Senders

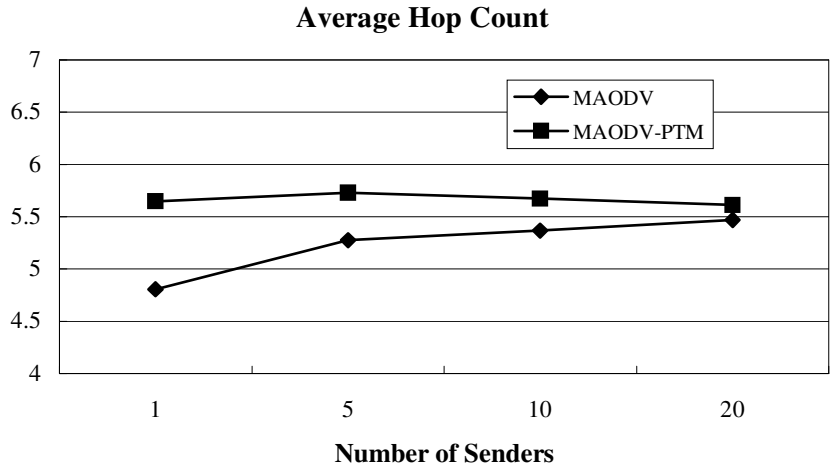


Figure 32: MAODV and MAODV-PTM: Average Hop Count vs. Number of Senders

Figure 32 presents the average hop count for MAODV and MAODV-PTM at different numbers of senders. Although the difference of the hop counts in MAODV-PTM and MAODV is at most 1 hop, MAODV-PTM has larger hop count than MAODV. As explained above, MAODV-PTM may have a large hop count because it may maintain longer branches than MAODV if it has a better packet delivery ratio. But the difference is reduced as the number of senders increases. In the 1-sender scenario, the difference is the largest. It is because centralized data traffic in MAODV may easily cause tree partition that is not known by the

downstream node, as explained before, thus, making branches shorter and resulting in lower packet delivery ratio. But the tree partition occurring during centralized traffic in MAODV-PTM can always be known by the downstream node, so the packet delivery ratio and the average hop count are rather stable.

As shown in Figure 33, the average end-to-end delays for MAODV and MAODV-PTM increase with the number of senders. In all scenarios, MAODV-PTM has more delay than MAODV. Referenced to the packet delivery ratio in Figure 31, the extra delay for MAODV-PTM is used for reaching further nodes to accomplish high data packet delivery ratio. In the 1-sender scenario, the increment of delay is relatively large, because the centralized traffic may result in longer branch length in MAODV-PTM than the branch length in MAODV.

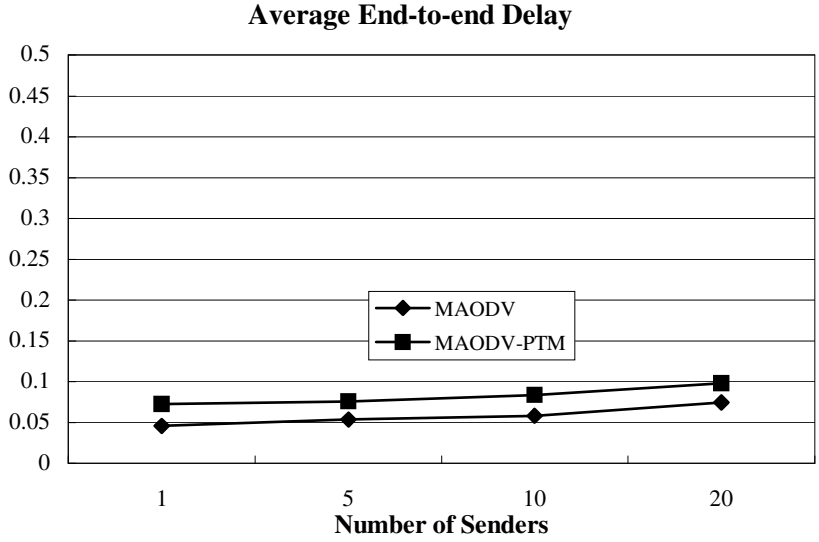


Figure 33: MAODV and MAODV-PTM: Average End-to-end Delay vs. Number of Senders

The differences of the four metrics under different numbers of senders between MAODV-PTM and MAODV are presented in Table 15, which are

achieved by comparing the results case by case. The results are consistent to the general performance of MAODV-PTM and MAODV.

		1-sender	5-sender	10-sender	20-sender
Packet Delivery Ratio Increment	Average	25.73%	22.42%	19.72%	18.95%
	Confidence Interval	(22.83%, 28.63%)	(20.32%, 24.53%)	(17.78%, 21.66%)	(17.47%, 20.44%)
Absolute Overhead Decrement	Average	5.14%	-11.17%	-11.24%	-11.84%
	Confidence Interval	(2.27%, 8.00%)	(-15.87%, -6.47%)	(-16.38%, -6.10%)	(-14.45%, -9.23%)
Hop Count Decrement	Average	-14.92%	-7.93%	-5.31%	-2.52%
	Confidence Interval	(-16.89%, -12.95%)	(-9.81%, -6.06%)	(-7.47%, -3.16%)	(-5.21%, -0.17%)
End-to-end Delay Decrement	Average	-34.28%	-28.21%	-29.37%	-24.37%
	Confidence Interval	(-46.59%, -21.97%)	(-34.18%, -22.23%)	(-35.97%, -22.78%)	(-30.38%, -18.36%)

Table 15: Average and 95% Confidence Interval for Performance Changes of MADOV-PTM based on MAODV

5.2.5 Observation and Summary

The performance of MAODV and MAODV-PTM, in terms of different mobility max speeds, different group sizes, and different numbers of senders, are evaluated. When varying node mobility, the packet delivery ratio of MAODV degrades sharply, but the ratio for MAODV-PTM remains above 90%. When varying group size, the number of packets delivered to destinations increases, and the packet delivery ratio of both protocols reduces slightly except the 50 group member scenario. The delivery ratio of MAODV-PTM is above 90%. When changing the number of senders, because the traffic load in the network is fixed, the packet delivery ratio for MAODV-PTM is rather stable. In terms of packet delivery ratio, MAODV-PTM performs much better than MAODV, with the improvement about 20%. To achieve the high delivery ratio in MAODV-PTM, the overheads are

slightly increased. For the average hop count of all sender-receiver pairs, packets in MAODV-PTM travel at most 1 hop more than the data in MAODV. The end-to-end delay in the multicast case is quite different from the unicast case, in that the multicast data packet delivered by unicast can cause more congestion than delivering unicast packet. This phenomenon is well observed when all the nodes become group members.

Chapter 6 Conclusions and Future Work

In this thesis, the link state prediction method equation E3 is implemented in standard AODV and MAODV, which accomplish packet delivery hop by hop by checking a node's routing table. By using this link state prediction method, the link breakage time can be known before the links actually become broken, so that the unicast route and the multicast tree can be updated in time to avoid packet loss. In this thesis, a proactive route maintenance mechanism (AODV-PRM) is proposed for improving the performance of AODV, and a proactive tree maintenance mechanism (MAODV-PTM) is proposed for improving the performance of MAODV. Simulations show both unicast and multicast protocols achieve higher packet deliver ratio by using proactive maintenance.

For AODV-PRM, when varying the max speed from 1m/s to 20m/s, the packet loss improvement is between 32% and 72%, and the increase of overhead is between 4% and 49%. The end-to-end delay is improved in some scenarios, and at least keeps at the same level as the delay in standard AODV. The route construction in both AODV and AODV-PRM are near optimal, and AODV-PRM has a slightly larger (below 2%) optimality ratio than standard AODV.

For MAODV-PTM, the scenarios with different max speeds, different group sizes and different number of senders are examined. In all scenarios except for broadcasting, the packet delivery ratio increases (about 20%) significantly with slightly more overhead (below 12%). Due to the unicast data delivery method used for delivering multicast data, which can cause congestion around a single node, the average end-to-end delay of different sender-receiver pairs in MAODV-PTM is

larger than those in standard MAODV. As MAODV-PTM achieves much better throughput, the multicast tree becomes larger to reach some group members that can not be reached in MAODV, thus the average hop count of all sender-receiver pairs becomes a little bit larger (about 1 hop).

AODV-PRM can be further improved by limiting overhead of unnecessary RREP and LPW messages as described before. To further improve the performance of MAODV-PTM, the unicast delivery method can be replaced by broadcasting. When in standard MAODV, if broadcasting is used for data delivery, the neighbor HELLO message should be added to detect link breakage. But by predicting the link state when a node receiving a packet, the neighbor HELLO message can be completely taken out. Thus, the end-to-end delay may be much shorter. One drawback of using broadcast is that broadcast cannot guarantee that the packet is successful transmitted to active neighbors, because the collision occurring at neighbors cannot be acknowledged by the node that sends the packet.

Up to now, two reactive unicast protocols, DSR and AODV, have been enhanced with the link state prediction method, and achieved better performance. But DSR and AODV make use of different routing methods to maintain routes and delivery packets, as DSR implements source routing and AODV applies hop-by-hop routing. So the link state prediction method is a general solution to improve route reliability for reactive protocols.

The proactive unicast protocols such as DSDV, as described in Chapter 2, rely on the substantial periodic routing-update overhead to try to keep up-to-date routing information between any pair of nodes. And the routing-updates are propagated

throughout the whole network to get a consistent view of the network topology. The link state prediction can also be adapted by proactive unicast protocols to limit the overhead to save the scarce bandwidth. The basic idea is to predict the link state when receiving packets and send out routing-update messages only when there is a link that is predicted to be broken soon.

As for multicast, because the tree structure only maintains one route between any two nodes in the tree, one link breakage will cut the routes to more than one receiver. This thesis has demonstrated that the throughput of tree-based protocol MAODV has been significantly improved by using link state prediction As mentioned in Chapter 2, [19] [18] make use of the link state prediction obtained by GPS to limit the Join-Query flooding for mesh-based protocol ODMRP. But another direction for implementing link state prediction in mesh-based protocols is to construct more stable and smaller mesh by knowing the link state in advance.

Finally, it is possible to adapt the link state prediction to enhance any QoS service for real time communications.

References

[1] Ballardie, T.; Francis, P.; and Crowcroft, J.; “Core based trees (CBT): An architecture for scalable inter-domain multicast routing”, Proceedings of the ACM SIGCOMM’93 Conference on Communications Architectures, Protocols and Applications, San Francisco, CA, USA, September 1993, pages: 85–95.

[2] Bommaiah, E.; Liu, M.; McAuley, A.; and Talpade, R.; "AMRoute: Ad-hoc Multicast Routing Protocol", Internet Draft, draft-talpade-manetamroute-00.txt, August 1998, work in progress.

[3] Broch, J.; Maltz, D. A.; Johnson, D. B.; Hu, Y.-C. and Jetcheva. J.; “A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols”, Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM’98), Dallas, TX, USA, October 1998, pages 85-97.

[4] Cheng. E.; “On-Demand Multicast Routing in Mobile Ad Hoc Networks”, M. Eng. Thesis, School of Computer Science, Carleton University, January 2001.

[5] Chiang, C.-C.; Gerlar, M. and Zhang, L.; “Forwarding Group Multicast Protocol (FGMP) for Multihop, Mobile Wireless Networks”, ACM/Baltzer Journal of Cluster Computing: Special Issue on Mobile Computing, December 1998, Vol. 1, No. 2, pages 187-196.

[6] Chiang, C.-C.; Wu, H.-K.; Liu, W. and Gerla, M.; “Routing in Clustered Multihop Mobile Wireless Networks with Fading Channel”, Proceedings of IEEE Singapore International Conference on Networks (SICON'97),

Singapore, April 1997, pages 197-211.

[7] Corson, S. and Ephremides. A.; “A distributed routing algorithm for mobile wireless networks”, ACM/Baltzer Journal of Wireless Networks, February 1995, Vol. 1, No. 1, pages 61-81.

[8] Dube, R.; Rais, C. D.; Wang, K.-Y. and Tripathi, S. K.; “Signal Stability-Based Adaptive Routing (SSA) for Ad Hoc Mobile Networks”, IEEE Personal Communications Magazine, February 1997, Vol. 4, No. 1, pages 36-45.

[9] Gafni, E. and Bertsekas, D.; “Distributed algorithms for generating loop-free routes in networks with frequently changing topology”, IEEE Transactions on Communications, January 1981, Vol. 29. No. 1, pages 11--18.

[10] Garcia-Luna-Aceves, J. J. and Madruge, E.L.; "A Multicast Routing Protocol for Ad-Hoc Networks", Proceedings of the18th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'99), New York, NY, USA, March 1999, pages 784-792.

[11] He, D.; Jiang, S. and Rao, J.; “A Link Availability Prediction Model for Wireless Ad Hoc Networks”, Proceedings of the International Workshop on Wireless Networks and Mobile Computing, Taipei, Taiwan, April 2000, pages D7-D11.

[12] Holland, G. and Vaidya, N.; "Analysis of TCP Performance Over Mobile Ad Hoc Netowrks", Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM’99), Seattle, WA, USA, August 1999, pages 219-230.

[13] http://www.isi.edu/nsnam/ns/

[14] IEEE Computer Society LAN MAN Standards Committee. “Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications”, IEEE Standard 802.11-1997, The Institute of Electronics Engineers, New York, NY, USA, 1997.

[15] Johnson, D. B. and Maltz, D. A.; "Dynamic Source Routing in Ad Hoc Wireless Networks", Mobile Computing, edited by Tomas Imielinski and Hank Korth, Kluwer Academic Publishers, ISBN: 0792396979, 1996, Chapter 5, pages 153-181.

[16] Jubin, J. and Tornow, J. D.; “The DARPA Packet Radio Network Protocols”, Proceedings of the IEEE, January 1987, Vol. 75, No. 1, pages 21-32.

[17] E. D. Kaplan (Editor). “Understanding the GPS: Principles and Applications”, Artech house, ISBN: 0890067937, 1996.

[18] Lee, S.-J.; Su, W. and Gerla, M.; “Wireless Ad Hoc Multicast Routing with Mobility Prediction”, ACM Mobile Networks and Applications Journal (MONET), August 2001, Vol. 6, No. 4, pages 351-360.

[19] Lee, S.-J.; Su, W. and Gerla, M.; "On-Demand Multicast Routing Protocol (ODMRP) for Ad Hoc Networks", Internet Draft, draft-ietf-manet-odmrp-02.txt, January 2000, work in progress.

[20] Lee, S.-J.; Su, W.; Hsu, J.; Gerla, M. and Bagrodia, R.; “A Performance Comparison Study of Ad Hoc Wireless Multicast Protocols”, Proceedings of 19th Annual Joint Conference of the IEEE Computer and Communicastions Societies (INFOCOM’00), Tel Aviv, Israel, March 2000, pages 565-574.

[21] Macker, J. and Corson, S.; IETF Mobile Ad Hoc Networks (MANET)

Working Group Charter; http://www.ietf.org/html.charters/manet-charter.html

[22] Macker, J. and Corson, S.; "Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations", RFC2501, January 1999, available at http://www.ietf.org/rfc/rfc2501.txt.

[23] Maltz, D. A.; Broch, J.; Jetcheva, J. and Johnson, D. B.; "The Effects of On-Demand Behavior in Routing Protocols for Multihop Wireless Ad Hoc Networks", IEEE Journal on Selected Areas in Communications Special Issue on Mobile and Wireless Networks, August 1999, Vol. 17, No. 8, pages 1439-1453.

[24] McDonald, A. B.; and Znabi, T.; " A Path Availability Model for Wireless Ad Hoc Networks", Proceedings of IEEE Wireless Communications and Networking Conference 1999 (WCNC'99), New Orleans, LA, USA, September 1999, pages 35-40.

[25] Murthy, S.; and Garcia-Luna-Aceves, J. J.; "An Efficient Routing Protocol for Wireless Networks", ACM Mobile Networks and Applications Journal (MONET), Special Issue on Routing in Mobile Communication Networks, October 1996, Vol.1, No. 2, pages 183-197.

[26] Narendran, B.; Agrawal, P. and Anvekar, D. K.; "Minimizing Cellular Handover Failures without Channel Utilization Loss", Proceedings of IEEE Global Communications Conference (GLOBECOM'94), San Francisco, CA, USA, December 1994, Vol. 3, pages 1679-1685.

[27] Obraczka, K.; Tsudik, G.; and Viswanath, K.; "Pushing the Limits of Multicast in Ad Hoc Networks", The 21st International Conference on Distributed

Computing Systems (ICDCS'2001), Phoenix, AZ, USA, April 2001, Pages 719-722.

[28] Park, V. D.; and Corson, M. S.; "A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks", Proceedings of the16th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'97), Kobe, Japan, April 1997, Vol. 3, pages 1405-1413.

[29] Perkins, C. E.; "IP mobility support", RFC 2002, October 1996, available at http://www.ietf.org/rfc/rfc2002.txt.

[30] Perkins, C. E. and Bhagwat, P.; "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers", Proceedings of the ACM SIGCOMM'94 Conference on Communications Architectures, Protocols and Applications, London, UK, August 1994, pages 234-244.

[31] Perkins, C. E. and Royer, E. M.; "Ad-hoc On-Demand Distance Vector Routing", Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'99), New Orleans, LA, USA, February 1999, pages 90-100.

[32] Perkins, C. E.; Royer, E. M.; Das, S. R. and Marina, M. K.; "Performance Comparison of Two On-demand Routing Protocols for Ad Hoc Networks", IEEE Personal Communications Magazine Special Issue on Mobile Ad Hoc Networks, Feb 2001, Vol. 8, No. 1, pages 16-29.

[33] Qin, L.; "Pro-active Route Maintenance in DSR", M. Sc. Thesis, School of Computer Science, Carleton University, August 2001.

[34] Rappaport, T. S.; "Wireless Communications: Principles and Practice (2nd

Edition)", Prentice Hall, ISBN: 0130422320, 2002.

[35] Royer, E. M. and Perkins, C. E.; "Multicast Operation of the Ad-hoc On-Demand Distance Vector Routing Protocol", Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM'99), Seattle, WA, USA, August 1999, pages 207-218.

[36] Royer, E. M. and Perkins, C. E.; "Transmission Range Effects on AODV Multicast Communication", To Appear in ACM Mobile Networks and Applications (MONET) Special Issue on Multipoint Communication in Wireless Mobile Networks, 2002.

[37] Royer, E. M. and Toh, C.-K.; "A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks", IEEE Personal Communications Magazine, April 1999, pages 46-55.

[38] Su, W. and Gerla, M.; "IpV6 Flow Handoff in Ad-Hoc Wireless Networks Using Mobility Prediction", Proceedings of IEEE Global Communications Conference, Rio de Janeiro, Brazil, December 1999, pages 271-275.

[39] Tobagi, F. A. and Kleinrock, L.; "Packet Switching in Radio Channels: Part-II The Hidden Terminal Problem in Carrier Sense Multiple Access Models and the Busy Tone Solution", IEEE Transactions on Communications, December 1975, Vol.COM-23, No.12, pages1417-1433.

[40] Toh, C.-K.; "Associativity-Based Routing for Ad-Hoc Networks". Wireless Personal Communications Journal, Special Issue on Mobile Networking and Computing Systems, March 1997, Vol. 4, No. 2, pages 103-139.

[41] Tuch, B.; “Development of WaveLAN, an ISM Band Wireless LAN”, Lucent Technical Journal, July/August 1993, pages 27-33.

[42] Wu, C. W.; Tay, Y. C. and Toh, C.-K.; “Ad hoc Multicast Routing protocol utilizing Increasing id-numberS (AMRIS) Functional Specification”, Internet Draft, draft-manet-amris-spec-00.txt, November 1998, Work in progress.

Appendix

1. Implementation Environment

Computer Pentium III 731MHz, 512M RAM

Operating System: Linux Red Hat 7.2

NS2 version: ns-allinone-2.1b8

To install NS2, download from http://www.isi.edu/nsnam/ns/, and type "./install" in the directory of extraction folder: ns-allinone-2.1b8.

2. Network Components in a Mobile Node in NS2

Figure A-1 [13] demonstrate the network components in a NS2 mobile node.

Wireless Channel simulates the wireless media by duplicating packets to all mobile nodes attached to the channel except the source itself. The signal strength received at the receiver is computed by using radio propagation model, and the receiver determines whether or not the packet can be detected.

Radio Propagation Model uses Friss-space attenuation ($1/r^2$) at near distances and an approximation to Two Ray Ground ($1/r^4$) at far distance. The antenna used for receiving the signal is an omni-directional with unity gain.

NetIF (Network Interface) simulates the hardware interface to the wireless channel. The interface stamps each transmitted packet with meta-data such as the transmission power, wavelength, which is used by the receiving network interface to determine if the packet has minimum power to be received and/or captured and/or detected. Lucent WaveLen DSSS radio interface is employed.

MAC (Media Access Control) uses IEEE 802.11 distributed coordination function

(DCF), which uses the RTS/CTS/DATA/ACK pattern for all unicast packets and simply CSMA/CA for all broadcast packets.

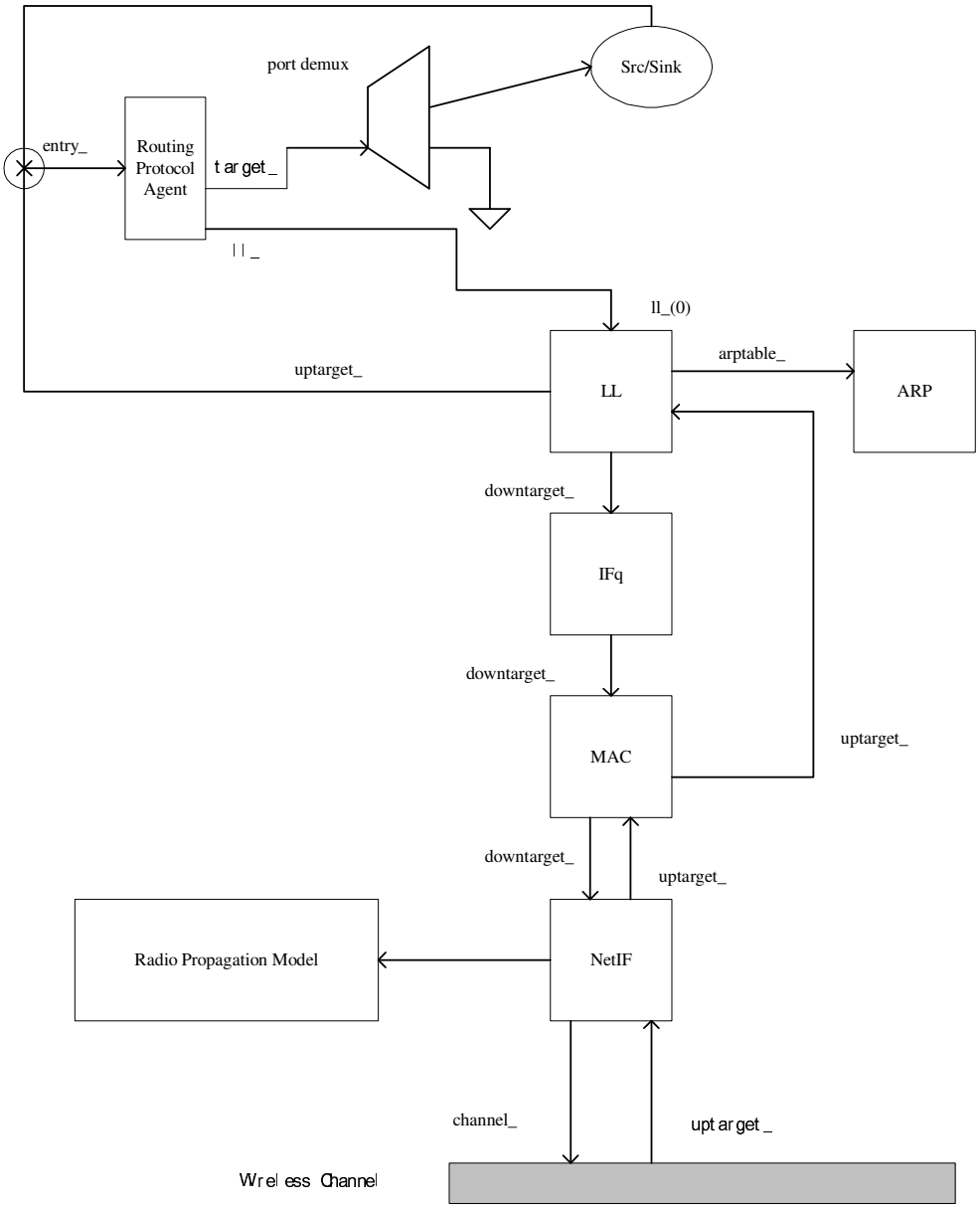


Figure 34: NS2 Mobile Node Network Components

Ifq (Interface Queue) queues outgoing packets before actually sending them out. It is priority queue that gives higher priority to routing protocol packets by inserting them at the head of the queue.

LL (Link Layer) simulates the data link protocols, which implements functions such as packet fragmentation and reassembly.

ARP is the Address Resolution Protocol. LL queries ARP about the address of the next hop for outgoing packets. If ARP has the address for the next hop, it writes the address into the packet header. Otherwise it broadcasts an ARP query, and caches the packet temporarily. For each unknown address, there is a buffer for only single packet. In case additional packets to the address need LL to query ARP, the earlier buffered packet is dropped. Once the address is known, the packet is inserted into the interface queue.

3. The Prediction Algorithm Implementation

Related files: ns-allinone-2.1b8/ns-2.1b8/ node.{h, cc}; wireless-phy.cc.

- In wirless-phy.cc, when a node receives a unicast packet, and it is the next hop of the packet, it then adds (previous hop address, receive time, signal power strength) to the node's relevant table in order to help nods predict the link breakage time.
- In node.{h, cc}, the link breakage prediction equation E3 is implemented to calculate the link breakage time.

4. AODV Modification

Direct related files: ns-allinone-2.1b8/ns-2.1b8/aodv/aodv.{h, cc}; aodv_packet.h

The modification is corresponding to the AODV-PRM description in Chapter 4.

Other related files:

- ns-allinone-2.1b8/ns-2.1b8/cmu-trace.cc: to add new defined messages (P-RREQ, RPE, LPW) to be displayed in trace-files. Also, when receiving a unicast data packet, the previous hop and the predicted link breakage time are added to be displayed in trace-files.
- ns-allinone-2.1b8/ns-2.1b8/rttable.{h,cc}: to include new route states (RTF_P_LINK, and RTF_PREDICTION).

5. MAODV and MAODV-PTM Implementation

Direct related files: ns-allinone-2.1b8/ns-2.1b8/aodv/aodv.{h, cc}; aodv_mcast.cc; aodv_packet.h. In aodv.h, a flag "PREDICTION" is defined. If this flag is valid as "#define PREDICTION", MAODV-PTM is implemented; otherwise, MAODV is implemented. The modification of MAODV to MAODV-PTM is corresponding to the MAODV-PTM description in Chapter 5.

Other related files:

- ns-allinone-2.1b8/ns-2.1b8/cmu-trace.cc: to add the new defined message (MACT) for multicasting to be displayed in trace-files. Also, when receiving a unicast data packet, the previous hop and the predicted link breakage time are added to be displayed in trace-files.
- ns-allinone-2.1b8/ns-2.1b8/ll.cc: to include broadcast packet with destination address set to the multicast group address. (This function is not used in current implementation.)
- ns-allinone-2.1b8/ns-2.1b8/Makefile: to include new files: aodv_mcast.cc; mttable.{h, cc}; and nhlist.{h, cc} into compilation.

- ns-allinone-2.1b8/ns-2.1b8/aodv/mttable.{h,cc}: new files to handle multicast routing table.
- ns-allinone-2.1b8/ns-2.1b8/aodv/nhlist.{h,cc}: new files to handle the list of next hops in multicast routing table entry.
- ns-allinone-2.1b8/ns-2.1b8/rtqueue.{h,cc}: to add new functions for handling the send buffer at source nodes.
- ns-allinone-2.1b8/ns-2.1b8/tcl/mcast/ns_mcast.tcl: to hook the multicast join and leave commands into agent class.

6. Creating Mobile Node Movement Scenario Files

Under directory: ns-allinone-2.1b8/ns-2.1b8/indep-utils/cmu-scen-gen/setdest, run:

```
./setdest [-n num_of_nodes] [-p pausetime] [-s maxspeed] [-t simtime]  [-x maxx]
[-y maxy] > [output-file]
```

7. Creating CBR Traffic Pattern Scenario Files

Under directory: ns-allinone-2.1b8/ns-2.1b8/indep-utils/cmu-scen-gen, run:

```
ns cbrgen.tcl [-type cbr|tcp] [-nn nodes] [-seed seed] [-mc connections] [-rate
packet/second for one connection]>[output-file]
```

As default packet size is 512 bytes, we must change it to 64 bytes, so modify the parameter in traffic pattern scenario file:

```
$cbr_(0) set packetSize_ 64
```

For multicast, the traffic pattern scenario is like:

1. set the traffic from a source to a multicast group address

```
set udp_(0) [new Agent/UDP]
$udp_(0) set dst_addr_ 0xE000000
```

```
$ns_ attach-agent $node_(1) $udp_(0)
set cbr_(0) [new Application/Traffic/CBR]
$cbr_(0) set packetSize_ 64
$cbr_(0) set interval_ 0.25
$cbr_(0) set random_ 1
$cbr_(0) set maxpkts_ 10000
$cbr_(0) attach-agent $udp_(0)
$cbr_(0) set dst_ 0xE000000
$ns_ at 30.00000000000000 "$cbr_(0) start"
```

2. set the group member into a multicast group

```
$ns_ at 0.010000000000000 "$node_(1) aodv-join-group 0xE000000"
```