PERFORMANCE EVALUATION OF AODV AND OLSR UNDER MOBILITY

BY TANUJA KUMAR

A thesis submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Master of Science

Graduate Program in Electrical and Computer Engineering

Written under the direction of

Professor Yanyong Zhang

and approved by

New Brunswick, New Jersey

January, 2009

ABSTRACT OF THE THESIS

Performance Evaluation of AODV and OLSR Under Mobility

by Tanuja Kumar

Thesis Director: Professor Yanyong Zhang

Wireless mobile ad hoc network is a infrastructureless network where each network node not only acts as a host but also acts as a router. Since the nodes are mobile, the environment is highly dynamic. For these networks to function properly a routing protocol is required that can respond to the rapid changes in the topology. Many routing protocols have been developed for accomplishing this task. The objective of this thesis is to study the impact of mobility on the performance of two mobile routing protocols, AODV, which is reactive routing protocol and OLSR, which is proactive routing protocol. Since not many MANETs have been deployed, most of the studies are simulation based. But for this thesis, experiments were conducted on national Open Access Research Testbed (ORBIT) for Next Generation Wireless Networks. We developed a basic framework to analyze the performance of routing protocols. We firstly evaluated the performance in a static environment where nodes are arranged in static linear topology and concluded that OLSR outperformed AODV. To study the mobility, we used Reference Point Group Mobility model that generates real life scenarios. It is clear that there is considerable cost associated with mobility. Both the protocols show decrease in throughput, higher standard deviation, more dead links and higher overhead when compared to their respective performance in static environment. However, the

relative performance of AODV and OLSR depends on the mobility scenario. AODV performed better than OLSR for discrete scenario when time snapshots were taken at a lower frequency i.e. every 30 seconds. On the other hand, OLSR performed better in pseudo-continuous scenario when time snapshots were taken at higher frequency i.e. every 5 seconds.

Acknowledgements

I thank my advisor, Prof. Yanyong Zhang for giving me the opportunity to work with her and for her attention, insight and support during this research. In addition, special thanks to Prof. Wade Trappe for his guidance and suggestions throughout this research and the preparation of this thesis.

I would also like to thank Prof. D.Raychaudhuri for his valuable guidance and time that he spent with me. I thank ORBIT facility in-charge Ivan Seskar for helping me out to conduct the experiments on the testbed.

I have been fortunate to have good friends and would like to thank all of them. For my master research, I thank Gautam for rescuing me whenever I was stuck with a problem. I thank Ayesha, Sumit and Amrita and others for brainstorming discussions that we have had.

Finally, I would like to thank those closest to me, whose presence helped make the completion of my graduate work possible. I am thankful to my husband, Sandeep Kumar, for reinforcing the fact that he is always there to help me out of any problem. Given the amount of effort he put in me, I think along with me he also earned a graduate degree in electrical engineering. I would like to thank my family and my parents for their confidence in me.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

An ad hoc wireless network consists of a collection of geographically distributed nodes that communicate over wireless links without the aid of any fixed infrastructure or central administration. Mobile Ad hoc NETwork(MANET) is a rapidly deployable, self configuring network of mobile nodes. There is no need for existing infrastructure like base station or access point to function properly. The nodes are connected via wireless links to form an arbitrary topology. As the nodes are mobile, the network environment is highly dynamic.

One of the original motivations for MANET is found in military activities like a group of soldiers moving towards a target. The distinct network properties of MANET, namely, infrastructureless, self-forming and self-healing makes it ideally suitable for tactical networking applications [1]. Apart from military, other potential applications, where ad hoc networks can be a good solution, include scenarios where it is expensive to set up an infrastructure, for example, in remote areas. Furthermore, MANET techmology, when properly combined with satellite-based information delivery, can provide an extremely flexible solution for establishing communications for fire/safety/rescue operations or other scenarios requiring rapidly-deployable communications [2]. In office environment, ad hoc network may offer a spontaneous inter-personal communication in case of an unplanned meeting.

The main characteristics of MANET as defined by MANET working group in RFC 2501 are mentioned below:

- Dynamic topologies: The nodes are mobile and can move randomly thus causing the network topology to change rapidly at unpredictable times.

- Bandwidth constrained: In wireless communications, the links have very low capacity as compared to hard-wired links. Practically, the realized throughput of a wireless network is less than radio's maximum transmission rate.

- Energy constrained operation: The mobile nodes in the network rely on batteries for their operation. Thus, the most important criteria when designing a system for MANET may be energy conservation.

- Limited physical security: In general, radio networks are more vulnerable to security attack as compared to fixed networks. The possibility of eavesdropping, spoofing and denial-of-service attacks is higher. In addition to this, there is no centralized administration in MANET, thus offering robustness against single point of failure.

Since there is no infrastructure support for MANET and a destination node may be out of range of a source node transmitting packets, a routing procedure is always needed to find an optimal path to forward the packets between source and destination. In ad hoc network, a node not only acts as a host but also as routers that route data to an intended destination. This causes additional problems along with highly dynamic environment, which leads to unpredictive changes in topology. The traditional IP routing protocols can not be used for MANET because IP protocols are designed to support routing in a network with a fixed infrastructure. The routing environment for ad hoc network is different because nodes are mobile and there is a greater probability of link failure that can cause frequent route changes. The characteristic of self-organization and self-configuration demands that the routing protocol should also be self-starting and self-organizing.

The major challenges with routing in MANET are listed below:

- While designing routing algorithm, it is generally assumed that all nodes have same transmission range, or in other words, links are symmetric. But in mobile ad hoc networks, the nodes are constantly changing their location and therefore, the concept of symmetric links does not apply here. Infact, the links are asymmetric, for example, a node B receives a hello packet from node A does not tell anything about the quality of connection in the reverse direction.

- As the topology of MANET is highly dynamic, some stale routes are generated in routing table, adding to the routing overhead of the protocol.
- The mobile nodes operate on batteries, therefore, to conserve energy, routing protocol should be able to optimize its operations.

The main tasks of routing protocol in MANET are topology discovery and topology maintenance. The topology discovery phase is usually periodic advertisements about a node's whereabouts as well as exchange of certain request-reply messages. On the other hand, topology maintenance is detecting link connectivity breaks and fix the broken link, if possible. Routing protocol in MANET can be classified in many ways depending on routing algorithm and network organization. Depending on the network structure, classification can be flat, hierarchical routing and geographic position assisted, while based on routing strategy used, they can categorized as table driven or on-demand as shown in figure below 1.1.

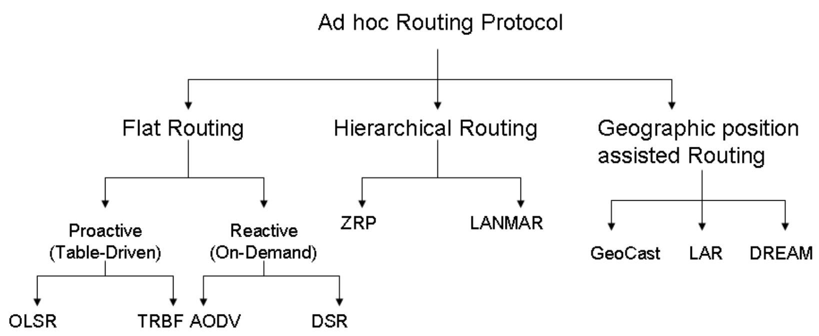


Figure 1.1: Classification of routing protocols in MANET

Out of the above classified routing protocols, AODV and OLSR are the most popular among the research community and therefore, this thesis will concentrate on these two. There are many evaluations available for these protocols in simulation environment [3] [4] but very few studies have been done that involve software implementation of these protocols on real network testbeds. Therefore, it is important to understand the functionality and performance of these protocols in real network environment. In this thesis, we used ORBIT testbed to conduct all the experiments. ORBIT testbed is a facility that consists of 20-by-20 grid of wireless radio nodes.

As is clearly seen from classification that AODV is an on-demand or reactive routing protocol where as OLSR is a proactive routing protocol. AODV builds routes between nodes only when needed by the source node. The routes are maintained as long as they are required by the source node. On the other hand in OLSR, each node exchanges topology information with other nodes in the network regularly. OLSR uses the concept of multi-point relays to forward the link state information with in the network, thus by reducing the overall protocol overhead involved in routing. Loop-free functioning of AODV is ensured by the use of sequence numbers. Furthermore, it keeps track of only the next hop in the route instead of the entire route.

The main objective is to study the impact of mobility on performance of MANET routing protocols. To get a better understanding of how mobility affects the routing mechanism of these routing protocols, it is important to model the mobility pattern appropriately. Extensive research material is available on modeling the mobility to evaluate MANET routing protocols. The most researched ones are Random Way Point, Random Walk, Reference Point Group Mobility. Random way point is a simple model to characterize mobility but at same time it does not capture certain characteristics, like mobility correlation between new time interval and previous time intervals, that are important to model real life situations. Therefore, this framework focuses on reference point group mobility model to evaluate the performance of AODV and OLSR. Before conducting the experiments for emulating mobility on ORBIT testbed, a few baseline experiments were done in static environment to develop understanding of routing mechanism of these protocols.

The main challenge faced while conducting the experiments on ORBIT testbed was the wireless nodes are static. To evaluate the performance of routing protocols on the testbed, there was a need to develop a framework that could emulate mobility on static nodes. This can be done in two ways. Firstly, varying the Packet Error Rate(PER) between the links as the nodes move apart or come closer. Changing PER emulates the mobility on static nodes. Second method is by injecting noise on the wireless links. Orbit testbed is equipped with Centralized Arbitrary Waveform Injection Subsystem(CAWIS). Using CAWIS, one can inject AWGN signals into the main ORBIT grid and create different topologies. For this thesis, we used the first method because it offers much better control over the experiment set up.

The rest of this thesis is organized as follows. Section 2 presents a survey of ad hoc routing protocols. This section discusses the pros and cons of proactive and reactive routing protocols along with a detailed description of routing algorithm of AODV and OLSR. Section 3 outlines the topologies used for conducting experiments in the static environment and the results of the baseline experiments are discussed. Section 4 provides introduction to various mobility models and justified the use of RPGM for this study. Also, this section describes the steps and process involved in emulating mobility on static ORBIT nodes. Furthermore, this section presents the results obtained from the experiments conducted in mobile environment. Finally, conclusion and future work are listed in section 5.

Chapter 2

Survey of Ad hoc Routing Protocols

2.1 Introduction

The two main classes of the routing protocol in traditional packet switched network are distance vector and link state. Both classes of the protocol use shortest path algorithm to find the best next hop neighbor. In link state routing, each node has connectivity graph that shows which nodes are connected to what other nodes whereas in distance vector routing, a node has only information about the next hop. Open Shortest Path First (OSPF) used in wired internet routing is based on link state algorithm and Routing Information Protocol (RIP) uses distance vector algorithm. These protocols, theoretically, can be employed in ad hoc networks but a number of specialized protocols have been developed for ad hoc networks. The main motivation behind developing a separate class of ad hoc routing protocols is that the shortest path protocols have high convergence time and high message complexity [5]. Because the wireless links have limited bandwidth and ad hoc networks have highly dynamic topology, the message complexity should be kept low. The two main classes of ad hoc routing protocol are proactive routing and reactive routing.

2.2 Types of Ad hoc Routing Protocol

Proactive routing is also known as table driven routing. This class of routing protocol keeps track of routes from a source to all the destinations whether or not the routes are required. To maintain the routes, periodic routing updates are exchanged between the nodes in the network. The main advantage of such an algorithm is that there is no delay in establishing a communication session and routing table is updated as soon as there is a change in topology. Disadvantages are additional control traffic to keep the routing table up to date irrespective of whether all the routes are used in a session or not. Example of proactive algorithm is Optimized Link State Routing (OLSR).

Reactive routing is also called on-demand routing as the routes are established only when needed to forward the data packets. This algorithm has significantly low routing overhead when the traffic is light and network is less dynamic, since there is no need maintain the routes when there is no data traffic. The major disadvantages are longer delay in establishing the routes for forwarding the data and excessive flooding of the control messages that may lead to network clogging. Example of reactive routing are Ad hoc On-demand Distance Vector (AODV) and DSR (Distance Source Routing).

Since AODV and OLSR are the most researched protocol in research community, this thesis will concentrate on these two protocols only. In order to better understand the mechanism and implementation of these protocols, below is the detailed description of each.

2.3 AODV: Ad hoc On-demand Distance Vector

2.3.1 Introduction

Ad hoc On-demand Distance Vector (AODV), as the name suggests is an on-demand protocol designed for mobile ad hoc networks [6]. This protocol responds quickly to changing link conditions and link breakages. The nodes mark the routes as invalid whenever there is a link breakage. AODV does not require a node to maintain routes to destinations that are not in active communication. Loop freedom in AODV is ensured by using destination sequence numbers. These also allow nodes to use the most recent route to a destination. The routing table information includes the destination address and the next hop address with the number of hops required to reach the destination. Also, the most recent destination sequence number associated with destination and lifetime of the route is stored in the table. If during the lifetime, the route is not used, the routing table entry is discarded.

2.3.2 Overview

The message types defined by AODV are Route Request(RREQ), Route Reply(RREP) and Route Error(RERR). AODV does not play any role as long as the endpoints in the communication link have valid routes to each other. When a route to a new destination is required, a node broadcasts the RREQ message to find a route. A route is found when the RREQ reaches the destination itself, or an intermediate node that has a 'fresh enough' route to the destination. The route is made available by unicasting the RREP message back to the source of the destination. Since, each node that receives the RREQ caches the route back to the source, the RREP can be unicasted to the origination of the RREQ. The link status of active routes is continuously monitored for any link breakage. When a link breaks, RRER message is propagated down the route to notify the affected nodes about the loss of link. The purpose of RRER message is to indicate which destinations are now unreachable because of the link breakage. Each node keeps a 'precursor list' that contains the IP address for each of its neighbors that are likely to use it as a next hop towards each destination.

2.3.3 AODV Operation

The basic operation of AODV can be divided into three phases:

- Route discovery
- Route maintenance
- Hello messages

2.3.4 Route Discovery

2.3.4.1 Generating and Forwarding a RREQ

When a destination is previously unknown to the node or the route to the destination is no longer valid, the node disseminates a RREQ. The Destination Sequence Number(DSN) is the last known DSN for this destination. If no sequence number is known, unknown sequence number flag is set. After broadcasting a RREQ, the node waits for a RREP. If the RREP is not received with NET_TRAVERSAL_TIME ms, the node rebroadcast the RREQ, upto a maximum of RREQ_RETRIES times, which is set to 2. When a node receives a RREQ, it creates or updates the route to the previous hop without valid sequence number and then checks if it has received the RREQ with same originator IP address and RREQ ID. If such a RREQ has been received, the node silently discards the newly received RREQ. If the RREQ received is not be discarded, the intermediate node searches for a reverse route to the Originator IP address. If the reverse route already exists, it is updated only if the originator sequence number is higher than the destination sequence number of the originator IP address in the routing table or if the sequence numbers are equal but the hop count in RREQ is smaller than the existing hop count in the routing table. The RREQ is rebroadcasted if the active route does not exist in its routing table or if the existing DSN is smaller than the DSN field of the RREQ. The Time to Live(TTL)in the outgoing RREQ is decremented by one and the hop count field is incremented by one to account for the new hop through the intermediate node.

2.3.4.2 Generating a RREP

A node generates a RREP if

- it is itself the destination.
- it has a fresh enough route to satisfy the request, i.e., the DSN in the route table entry for the destination is greater than or equal to the DSN of RREQ.

The RREP is unicast to the next hop towards the originator of the RREQ. As the RREP is propagated, the hop count field in RREP is incremented by one at each hop.

2.3.4.3 Generating Gratuitous RREP

When a node receives a RREQ and responds with a RREP, it discards the RREQ. If the RREQ has 'G' flag set, and the intermediate node replies to the RREQ, it unicasts a gratuitous RREP to the destination node.

2.3.4.4 Receiving and Forwarding RREP

When a node receives a RREP, it searches for a route to the previous hop. Once the route to the destination is created or updated in the route table, the route is marked active and the next hop is assigned to the node from which RREP was received. Consequently, the node can use this route to forward the data to the destination. If the node is not the node indicated by the Originator IP address in RREP, then the RREP packet is forwarded towards the next hop, selected based on the route table entry. When a node forwards a RREP, the precursor list for the corresponding destination node is updated by adding to it the next hop node to which the RREP is forwarded.

2.3.5 Hello Messages

AODV maintains network connectivity by reception of broadcast hello messages on the active routes. A node that is a part of an active route periodically broadcasts hello messages, which are RREP messages with TTL=1, to announce its presence. If a node does not receive a hello message with in a specified interval, then it is assumed that the neighbor node is no longer in transmission range and the connectivity to this node has been lost. Whenever a node receives a hello message from a neighbor, the node checks if it has an active route to the neighbor and if not, it creates one. If the route already exists, the TTL for the route is increased.

2.3.6 Route Maintenance

A RERR message is generated in following three scenarios:

1. While transmitting data, if a node detects a link break for the next hop of an active route. In this case, the node makes a list of unreachable destinations comprising of unreachable neighbors and other destinations that use the unreachable neighbor as next hop.

2. A node receives a packet for the node it does not have an active route in its routing table. For this, there is only one unreachable destination.

3. If a node receives a RERR from a neighbor for one or more active routes, the list consists of destinations in RERR for which there is corresponding entry in the local routing table that has the transmitter of the received RERR as the next hop.

For cases 1 and 2, the DSN(s) in the routing table for the unreachable destinations are incremented by one. Then a RERR is broadcast with the unreachable destination(s) and their incremented DSN(s) included in the packet. For case 3, the node updates the DSN and invalidates the route for the destination. A RERR message is then broadcasted to the neighbor nodes in the precursor list of the destination.

AODV has a mechanism called Local Repair by which the upstream node of the broken link attempts to repair the link locally instead of sending RERR. The node initiating the local repair follows the route discovery phase. If the node does not receive a RREP, it then transmits a RERR message for that destination. The process of local repair may result in greater path lengths to the destinations for which local repair was initiated.

2.4 OLSR: Optimized Link State Routing

2.4.1 Introduction

The Optimized Link State Routing (OLSR) is proactive table driven protocol for mobile ad hoc networks [7]. It facilitates efficient flooding of control messages throughout the network by using selected nodes called MultiPoint Relays(MPRs). MPRs are selected by each node and are used to forward control messages resulting in a distributed operation of the protocol. In addition to this, a node continuously maintains routes to all destinations in the network, thus making the protocol suited for traffic pattern that is random and sporadic. Furthermore, the proactive nature makes OLSR suitable for networks where communicating pairs change over time.

2.4.2 Overview

The protocol is an optimization of classical link state routing algorithm and uses the concept of MultiPoint Relays(MPRs). The problem of flooding the network with control messages is overcome by the MPR nodes. A node periodically exchanges hello messages to discover its two-hop neighbor. Using this information, each node selects a set of MPRs, which are one-hop neighbors. A node selects MPR such that there exists a path to each of its two-hop neighbors via a node selected as a MPR. The main responsibility of MPR is to forward the control messages throughout the network, thus minimizing the number of transmissions. MPRs periodically broadcast the control information, thereby announcing the reachability to the nodes that have selected it as a MPR. A node uses this information to determine next hop destinations for all the nodes in the network using the shortest path algorithm. This way routes to all nodes are known before hand that leads to no route discovery delay as is encountered in AODV. Because of its proactive nature, routing overhead is generally greater than that of a reactive protocol.

2.4.3 OLSR Operation

The core functioning of OLSR can be divided into three processes namely:

1. Neighbor/Link Sensing
2. Efficient control flooding using MPR
3. Optimal path calculation using shortest path algorithm.

2.4.4 Neighbor/Link Sensing

Each node periodically exchanges HELLO message with each other. A hello message mainly consists of link information and neighborhood information, i.e., two-hop neighbors, MPRs and MPR selector. A MPR selector set of a node is a set that has selected it as its MPR. The three important tasks performed by hello message exchange are namely link sensing, neighbor detection and MPR selection signaling. For neighbor and link sensing, a hello message typically comprised of list of links and list of one-hop symmetric neighbors. A hello message is broadcasted by a node to its neighbors and is never forwarded by other nodes. On reception of a hello message, a node performs a link sensing, neighbor detection and MPR selection set population. Each node in the network selects its MPR set. MPR set is elected based on the rule: For all two-hop neighbors n there must exist a MPR m so that n can be reached by m [8]. Smaller the MPR set, minimum is its protocol overhead.

2.4.5 Efficient Control Flooding Using MPRs

Due to the proactive nature of OLSR, each node maintains the partial topology graph of the network. This information is extracted from Topology Control(TC) messages and is then used for calculating the shortest paths to destinations. A MPR node broadcasts a TC message periodically that is disseminated across the network using the other MPR nodes. A TC message contains MPR selector set of the source of the message and is forwarded by MPR if and only if it received the message for the first time by that node and it is in the MPR set of the previous hop node. This controlled flooding results in minimized retransmissions.

2.4.6 Optimal Path Calculation Using Shortest Path Algorithm

A routing table is maintained by every node, which is then refreshed and updated whenever a change in the topology is detected. To populate a routing table, shortest path algorithm is used on the partial topology graph obtained from TC messages. It is important to note that OLSR is not involved in forwarding of data packets.

Chapter 3

Baseline Experiment Evaluation Results

We conducted some baseline experiments in static environment to understand the functioning of AODV and OLSR. A linear topology of two to six nodes was considered for the experiments. Various performance metrics considered for baseline evaluation are average and aggregate throughput and protocol overhead. Parameters like hop count and packet size were varied during the experiments to observe the behavior of the protocols. In addition to linear topology, a complex topology consisting of 10 nodes with three simultaneous flows was also considered to understand the interaction between medium-access layer and routing layer. Furthermore, before we could start the mobility experiments, we also evaluated how the protocols respond to a node, which previously not in transmission range comes into transmission range and offers an alternate shorter hop path to the destination. From these baseline experiments some interesting results were obtained which are presented in sections below.

3.1 Orbit Testbed

The experiments were conducted on ORBIT(Open Access Research Testbed for Next-Generation Wireless Networks) [9] system hosted by WINLAB (Wireless Information Network Laboratory), an industry-university research center at Rutgers University. ORBIT is a laboratory-based wireless network emulator. The testbed is a two-dimensional grid of 400 802.11a/b/g radio nodes that can be dynamically interconnected into specified topologies. Each node on the grid is a PC with a 1 GHz VIA C3 processor with 512MB RAM, 20GB local disk, two wireless mini-PCI 802.11a/b/g cards and two 100BaseT Ethernet ports for transfer of experimental data, control and management information. All the nodes run Debian GNU/Linux with 2.6 kernel.

The testbed provides an Experiment Controller called the Node Handler and local client called the Node Agent that runs on each node. The node handler multicasts the commands to the nodes and keeps track of their execution. On the other hand, the node agent software component executes the command received from node handler and reports the information back to the node handler.

The experimental measurements are collected using ORBIT Measurement Framework and Library(OML). The framework is based on client/server architecture. An OML collection server collects the data from all the nodes involved in the experiment. The results are archived in SQL database that can be accessed later on. Also, an OML collection client is associated with each node that collects the data, does pre-processing, if required, and then sends them to OML collection server.

3.2 Experiment Parameters and Setup

The experiments were performed using IPERF traffic generator to generate UDP traffic flows. IPERF [10] is a network performance tool that measures TCP and UDP bandwidth performances. The UDP traffic was used with varying characteristics like packet size, offered load and traffic distribution. Measurements during the experiment were collected using log files of IPERF and then parsing them with AWK scripts, a programming language designed for processing text-based data.

AODV implementation developed by Upsala University called AODV-UU [11] version number 0.9.5 was used. For OLSR, OLSRd [12] implementation version number 0.4.1 was used. Both these implementations are freely available on the internet. The implementations were integrated into ORBIT framework to conduct the experiments.

Since the nodes in ORBIT environment are in single collision domain, which means that a node can listen to any node. Therefore, to emulate a multi-hop behavior, a MAC filter called MACKill was used. MACKill is a software based MAC filter that filters the packet at layer two or MAC layer. The module at a node is provided with a list of MAC addresses of the nodes that should not be heard. The filter rejects all the packet from the nodes whose MAC address is specified in the list. MACKill can filter any percentage of packet specified in the list, for example, it can drop 100% of packets or 10% of packets received from a node, which is specified in the filter list.

The experimental configurations that were kept same throughout the experiments are:

- Wireless card in 802.11a mode using channel 48.
- Implementation for AODV used was AODV-UU 0.9.5 and for OLSR was OLSRd 0.4.1.
- Topology creation using MACKill.
- All the baseline experiments were conducted with a physical layer rate of 36Mbps and offered load of 20Mbps.

3.2.1 Baseline Experiment Results for Static Topology

We started with a basic two-hop topology as shown in figure 3.1 and evaluated the performance of protocols under link saturation as well as when links are not saturated.

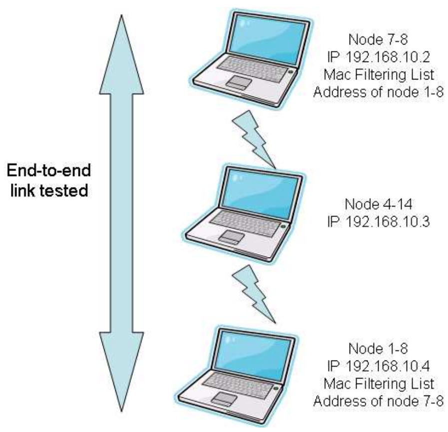
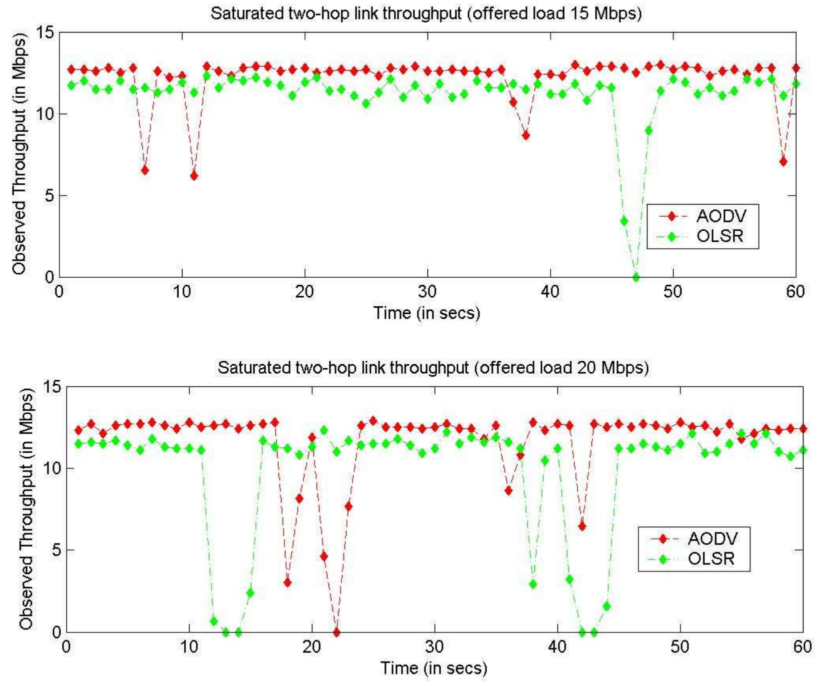
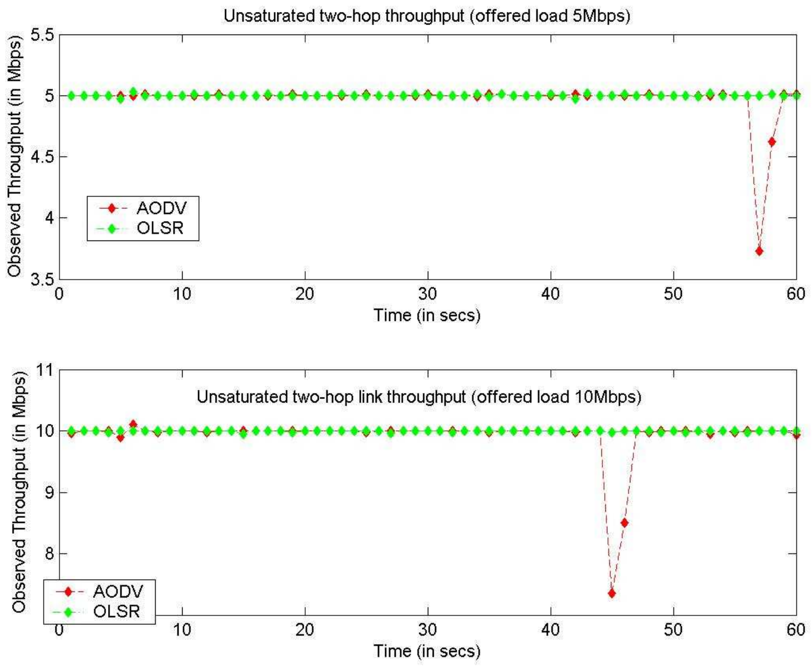


Figure 3.1: Two-hop linear topology

The physical layer rate for the experiment was 36Mbps. For unsaturated scenario, both the protocols have comparable throughput except occasional drop in case of AODV (figure 3.2(b)). The cause for the drop in throughput for AODV may be it tries to repair routes when it found them missing. For saturated scenario, the variation in throughput



(a) Saturated two-hop link throughput



(b) Unsaturated two-hop link throughput

Figure 3.2: Throughput measurements for a two-hop link

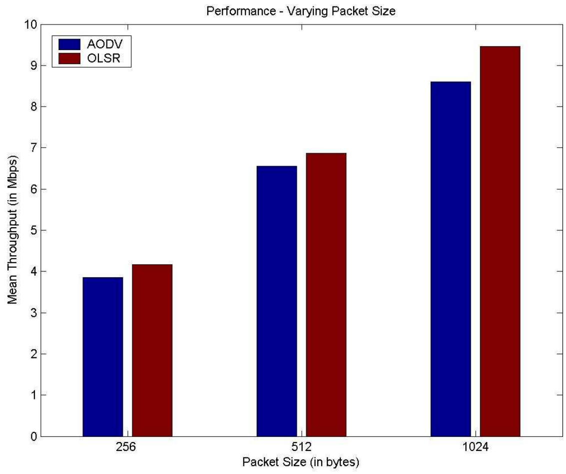
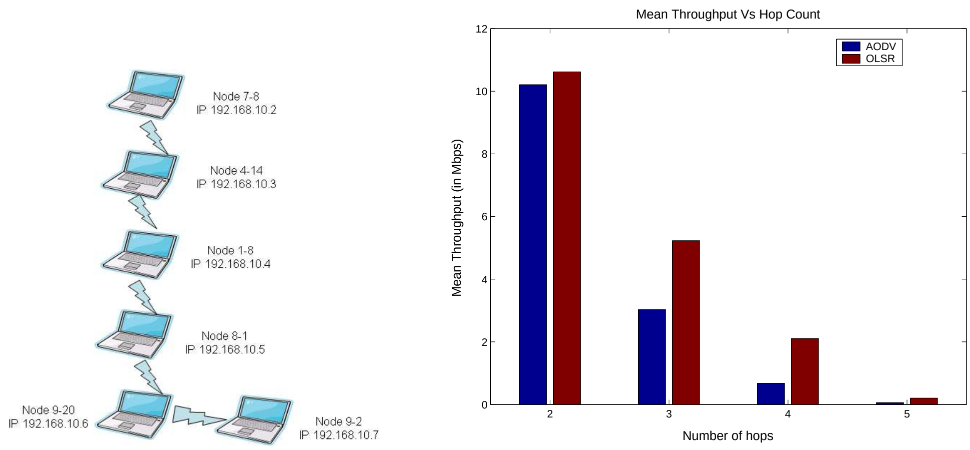


Figure 3.3: Performance-Varying packet sizes

with time is clearly visible in figure 3.2(a). OLSR showed link outage for longer duration than AODV, especially for offered load of 20 Mbps.

The second experiment was conducted using the same two-hop topology, as shown in figure 3.1 and measuring the throughput while varying the data packet sizes from 256, 512 and 1024 bytes. The main objective behind this was to understand if there was a correlation between packet size and observed throughput. Figure 3.3 shows the mean throughput over an experimental duration of 600 seconds. As the size of data packet is increased, mean throughput also increases. Both the protocols show comparable increment in mean throughput as the size of data packets goes up. From the behavior of the routing protocols it can be concluded that control packets, which are smaller in size, requires same channel arbitration time as that of data packets irrespective of size of data packets.

The third experiment in the baseline experiment list was to vary the hop count from two to five to observe how fast the routing table is populated and study the capacity in terms of number of hops a route can take. Figure 3.4(a) shows the topology used and figure 3.4(b) shows the results. It can be seen from the results that the throughput scales down as the number of nodes go up. Adding more hops increases the overhead per flow and hence, the deteriorated performance. Note that the throughput for AODV decreases at a faster rate compared to OLSR, as hop count is increased.



(a) Topology - Varying hop count (b) Performance - Varying hop count

Figure 3.4: Experimental set up for varying the hop count

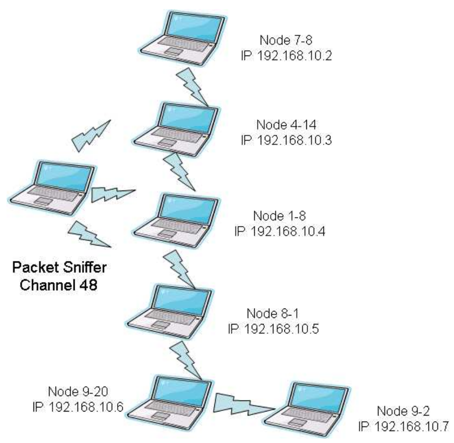
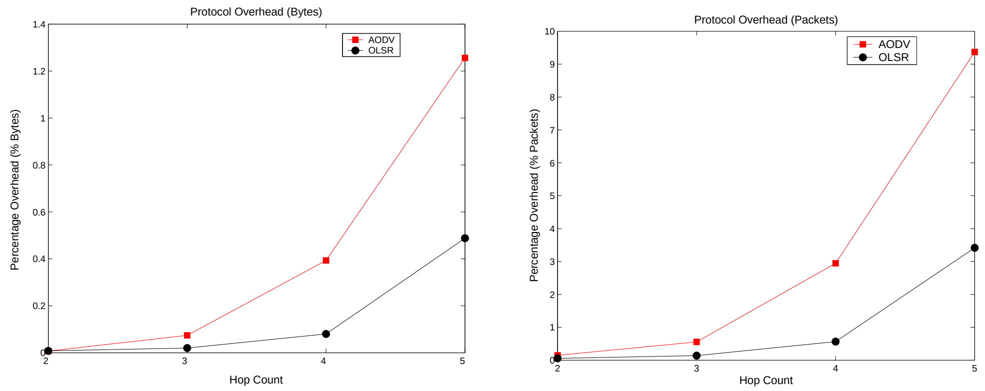


Figure 3.5: Topology - Protocol overhead

In addition to measuring the throughput, protocol overhead was also observed. An extra node was designated as a sniffer node as shown in figure 3.5 that sniffed the packets from the network. Later on, packets captured were analyzed using a popular network protocol analyzer called Wireshark [13] that is freely available for public. It is easily deducible from the figures 3.6(a) and 3.6(b) that protocol overhead for AODV is more than OLSR and the difference goes up exponentially as hop count increases.

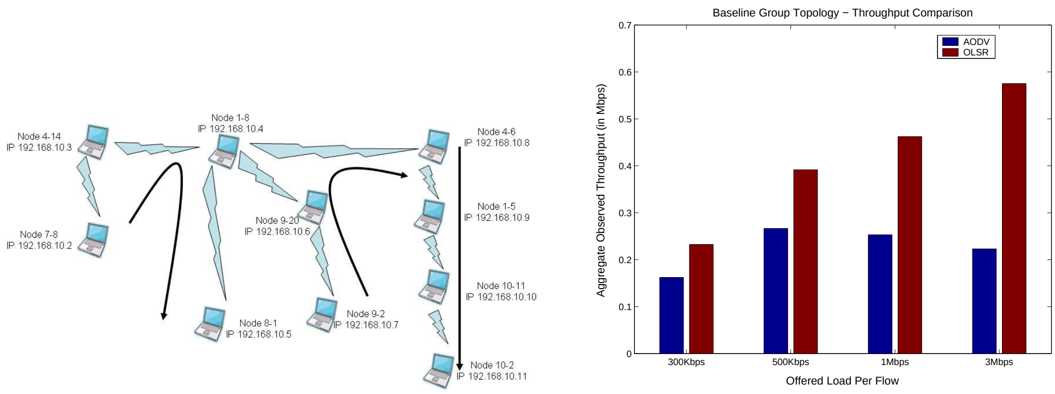
Last experiment in baseline list was to involve 10 nodes and test the interaction between MAC and routing layer. For this setup, the physical layer rate was set at 6Mbps and three simultaneous data flows were configured as shown in figure 3.7(a). During



(a) Protocol overhead (in percent bytes) (b) Protocol overhead (in percent packets)

Figure 3.6: Experimental measurements for protocol overhead

experiment, the load on the network was increased on a per flow basis from 300Kbps to 3Mbps and aggregate throughput of the network was measured. The results of the experiment are shown in figure 3.7(b). While the aggregate throughput of the network for OLSR goes up as the offered load is increased but for AODV the throughput tapers off for offered load higher than 500Kbps.



(a) Group Topology (b) Performance - Group topology

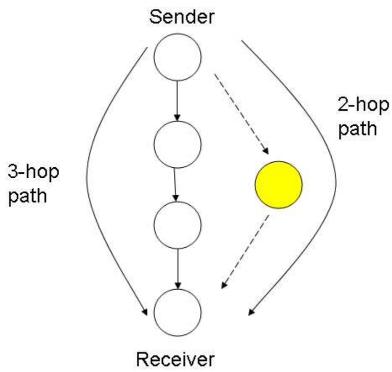
Figure 3.7: Experimental set up and results for group topology

3.2.1.1 Experiment to Observe Frequency of Route Switch

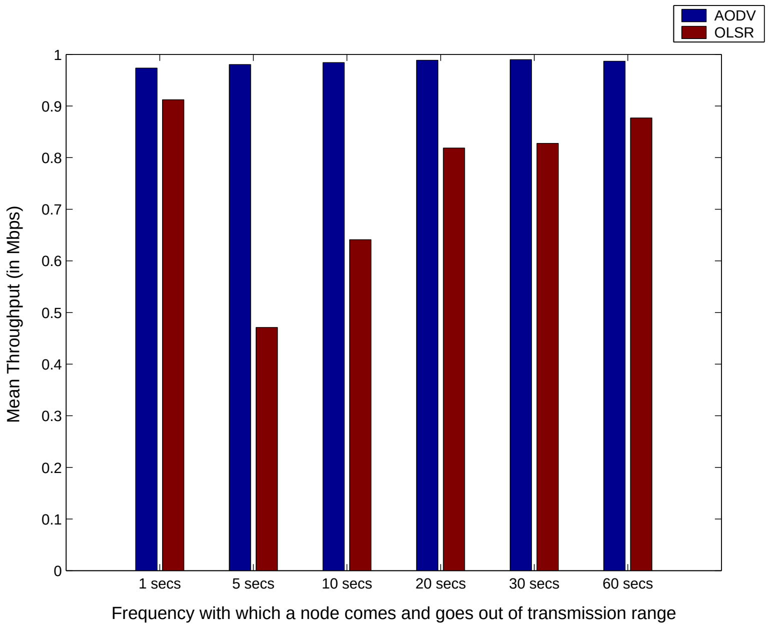
The main objective of this experiment was to study how fast a protocol switched to an optimal route (shortest path) if a neighboring node in transmission range moved away and lost connection. The frequency of movement of the node, previously in transmission range, moving away and coming back into range was varied from 1, 5, 10, 20, 30 and 60 seconds respectively. The initial topology used was a three-hop linear topology, as shown in figure 3.8(a). Periodically, a node appears in the neighbor and disappears. It is seen that if the frequency of movement of a node is 1 second, neither of the protocols switch route and continue to follow a three-hop path. For 30 and 60 seconds experiment, AODV chooses shorter two-hop path for fewer times as compared to OLSR (table 3.1). And for frequency of 10 and 5 seconds, AODV never chooses two-hop path. As shown in figure 3.8(b) the mean throughput observed is lower for OLSR especially at higher frequency because of excessive path switching.

Table 3.1: Percentage of the time protocol switched to a two-hop path

Protocol	1 sec	5 secs	10 secs	20 secs	30 secs	60 secs
AODV	0	0	0	1.8	3.2	17.4
OLSR	0	37	39.5	50	53.3	61.5



(a) Topology - Route change from three-hop to two-hop



(b) Performance - Route change from three-hop to two-hop

Figure 3.8: Experimental set up and results for route change

Chapter 4

Mobility Models and their impact on MANET

4.1 Mobility Models

A mobile ad hoc wireless network is a multi-hop wireless network consisting of mobile nodes communicating through wireless links, without any infrastructure or central entity. Mobility creates a highly dynamic environment that is one of the major challenges in ad hoc networks. Since the nodes are mobile, the relative movement between nodes creates or breaks the wireless connection, thus resulting in network topology change. The routing protocol should be intelligent enough to react to these dynamics. The primary goal of a routing protocol is to maintain a correct and efficient route between communicating nodes without deteriorating the performance of a network. Hence, mobility model plays a very crucial role in evaluation and study of an ad hoc routing protocol.

A mobility model defines dynamic characteristics of a node movement. It is important to use a mobility model that can emulate the movements close to real life applications because the performance of routing protocol greatly depends on the mobility pattern. There are many mobility models available for ad hoc networks like Random Walk, Random Way Point(RWP) and Reference Point Group Mobility(RPGM) to study the performance of routing protocol. RWP and RPGM are the most studied mobility models by the researchers, therefore we would focus on these two.

4.1.1 Random Way Point

Random Way Point is a simple, commonly-used, synthetic model used in simulation studies of ad hoc routing protocols. In this elementary mobility model, each node independently chooses its destination randomly within the network boundaries and move towards the destination with a constant speed. The speed is chosen randomly from a uniform distribution between 0 and V_{max}. After a node reaches its destination it pause for a specified pause time and then again selects a random destination and random speed to move towards a new destination. The process is repeated for the entire duration of simulation. On the other hand, if the mobility model is Random Walk, the node selects its new destination by randomly choosing speed and direction from a predefined ranges $[V_{min}, V_{max}]$ and $[0, 2\pi]$ respectively.

In RWP, nodes move independently of each other, however, there can be scenarios in ad hoc networks where it is important to model the mobile nodes as they move together. For example, a group of soldiers in a military scenario assigned a task to achieve a common goal. Another example can be during disaster relief rescue crew forms different groups and work cooperatively. These applications are the motivation behind studying a group mobility model. One of most studied group mobility model is Reference Point Group Mobility Model(RPGM)[14].

4.1.2 Reference Point Group Mobility

RPGM represents the random motion of a group of nodes as well as random motion of individual nodes within group. Each group has a center, which is either the logical center of the group or the leader of the group. Thus, each group is composed of a one group center and number of members. For this framework, we assumed the center to be group of the leader. The group movement is defined by the path traveled be the leader of group. The group leader motion is used to calculate group motion via a group motion vector, $\vec{V^t_{group}}$. The vector not only defines the motion of group leader but also the general motion trend of the group. Each member deviates from general group motion vector $\vec{V^t_{group}}$ by some degree. The group motion vector $\vec{V^t_{group}}$ can be randomly chosen or can be designed based on predefine path of the group.

The movement of group members significantly depends on the movement of their group leader. Each node moves randomly about their own pre-defined reference point that follows group movement. Based on this pre-defined reference point, group member

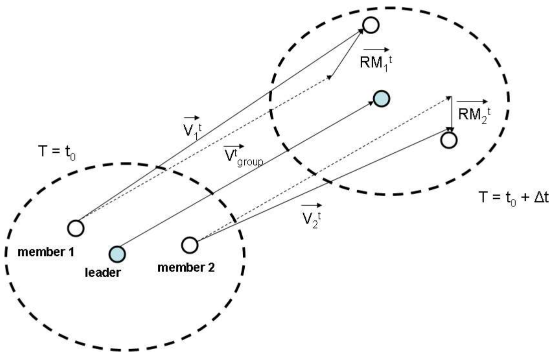


Figure 4.1: Movement of nodes using RPGM model

can be placed randomly in the neighborhood. The motion vector of group member i at time t, $\vec{V_i^t}$ can be described as

$$\vec{V_i^t} = \vec{V_{group}^t} + \vec{RM_i^t}$$

where $\vec{RM_i^t}$ is a random vector deviated by group member i from its reference point. The vector $\vec{RM_i^t}$ is an independent identically distributed (i.i.d) random process whose length is uniformally distributed in the interval $[0,\, r_{max}]$. r_{max} is the maximum allowed distance deviation and the direction of vector is uniformly distributed between $[0,\, 2\pi)$.

A path that the group follows is given by defining a sequence of checkpoints along the path corresponding to given time intervals. If the checkpoints are properly selected, many realistic situations can be modeled such as movement of a group that has to reach a predefined destination within given time intervals to accomplish its task.

Figure 4.1 gives an illustration of movement of group using RPGM mobility model. In figure 4.1, $\vec{V_{group}^t}$ is the motion vector of the group leader as well as of the whole group. $\vec{RM_i^t}$ is the random deviation vector of group member i and final group member motion vector of group member i is represented by vector $\vec{V_i^t}$.

In RWP, initially, nodes are randomly distributed in the simulation area that does not represent the manner in which the nodes distribute themselves when moving. Also, due to its simplicity and elementary behavior, it is not able to emulate real life situations. Moreover, the major drawback of this model is that it does not consider the mobility correlation between new time intervals and previous time intervals. Furthermore, the model does not take into account the relative positions of other nodes in the network. The problem with Random Walk mobility model is its memory-less mobility pattern, which means it can generate unrealistic movements such as sudden stops and sharp turns. In contrast to RWP and Random Walk, RPGM can generate realistic mobility patterns. RPGM provides a general and flexible framework by use of checkpoints, thereby providing a solution to model many realistic situations where a group must reach a pre-defined destination to accomplish a task.

4.1.3 Modeling mobility on Orbit

The main challenge faced while conducting the experiments on ORBIT testbed was the static nodes. The nodes in the testbed are static and are in single collision domain, which means that every node can listen to every other node in the testbed. Therefore, to create a multi-hop network, a software based MAC level filter called MACKill was used. MACKill can filter the packets received from a node depending whether the node is added in the filter list or not. The filter can be configured to filter out 100% packets or any other number of packets as desired. To evaluate the performance of the protocol under mobility, there was a need of a framework that could emulate mobility on static ORBIT nodes. A model that can generate the conditions such as changing link error rate, increase or decrease in noise etc, a node experience when it is mobile. We had two options to emulate mobility on the node. First option was to inject Additive White Gaussian Noise (AWGN) noise using the Orbit testbed noise generator. The testbed is equipped with Centralized Arbitrary Waveform Injection Subsystem(CAWIS). Using CAWIS, one can inject AWGN signals into the main ORBIT grid and create different topologies. Second option was to change the link PER as the nodes move away or come closer to each other. For this thesis, we used the second method because it offers much better control over the experiment set up. Also, using MACKill filter to change link PER was more reliable as compared to injecting noise into the system. In addition to this, it was difficult to vary the noise on a per link basis as a node moves away or comes closer to another node. The next most important step was to derive a relationship between distance and PER that can be mapped onto MACKill filter.

4.1.3.1 PER vs Distance Relationship

For deriving the relation between PER and distance, we assumed Log-distance path loss model, the simplest propagation model. The propagation model indicates that average received signal power decreases logarithmically with distance, whether in outdoor or indoor channels. The average large-scale path loss for an arbitrary transmitter-receiver is expressed as a function of distance by using a path loss exponent, n

$$PL(d) \propto (\frac{d}{d_0})^n$$

or

$$PL(dB) = PL(d_0) + 10nlog(\frac{d}{d_0})$$

where n is the path loss exponent that indicates the rate at which the path loss increases with distance. d_0 is the close-in reference distance which is determined from measurements close to the transmitter, and d is the transmitter-receiver distance [15]. The parameters in the path-loss model can be found either through measurements or standard path-loss values available in literature. The reference path loss $PL(d_0)$ for this thesis was assumed to be 51.7dB with path loss exponent n=2 and reference distance d_0 to be 1m [16]. We know that received SNR (Signal to Noise Ratio) is the difference between transmitted SNR and path loss, which is given by

$$SNR_{received} = P_{transmitted} - PL(d) - N_{power}$$

where $P_{transmitted}$ is the transmitted power, N_{power} is the noise power and PL(d) is the path loss. We assumed $P_{transmitted} = 200mW$ which in dB is -6.98dB. Substituting the expression of $PL(dB)$ in above equation and further manipulation yields

$$log_{10}d = \frac{PL(d) - PL(d_0)}{10n}$$

The above equation gives a relation between distance and path loss or SNR. To get a relation between distance and PER, we need to obtain physical layer performance of 802.11a standard, i.e. a relation between SNR and PER. The relation between SNR and

PER was derived from physical layer simulation results in [16](see table 4.1 for values). The simulation model used in [16] was Incisive SPW 802.11a model. AWGN model was used to represent the channel. To see the effect of fading, additional loss factor was introduced in the simulator. The simulations were carried out using a packet size of 1500 bytes for practical system. The waterfall curves obtained as a result of simulations define the relation between SNR and PER. This relation was then used in getting a matrix between distance and PER. The distances, in meters, corresponding to PER are given in table 4.1.

Table 4.1: Relationship Between Distance and PER

SNR(in dB)	PER	PL(in dB)	Distance(in m)
10.9	0.6	103.1	371.53
11.3	0.3	102.7	354.81
11.9	0.1	102.1	331.13
12.1	0.05	101.9	323.59
12.3	0.03	101.7	316.22

The figure 4.2 is a plot between $log_{10}(PER)$ and distance. We run a linear regression between the two to interpolate and extrapolate. The line of fit is

$$log_{10}(PER) = 0.0232d - 8.7731$$

where PER is the packet error rate and d is the distance between transmitter and receiver. The slope and the intercept are statistically significant as shown in table 4.2. In addition, $R^2 = 0.9822$ indicates goodness of fit. For any distance if PER comes out be greater than one, we assume it to be equal to one. This assumption is reasonable because after a certain distance the link completely breaks indicating PER to be 100% or equal to one.

Table 4.2: Regression Analysis (log(PER) Vs Distance)

	Coefficients	Std Err	t stat	P-value
Intercept	-8.7731	0.6122	-14.3314	0.0007
Slope	0.0232	0.0018	12.8624	0.0010

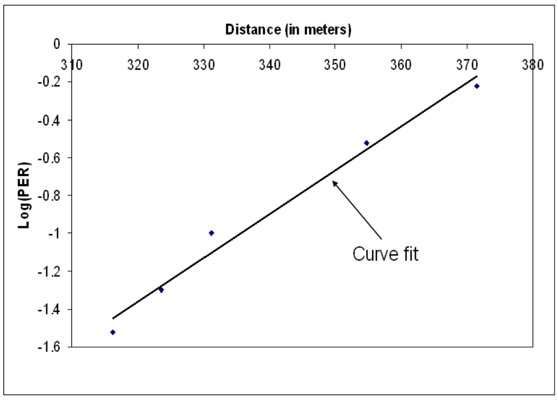


Figure 4.2: Distance vs. PER

4.2 Mobility Scenarios and Performance Metrics

In this thesis, to have a better understanding of routing protocol functioning, we first used a very basic straight line mobility model. We assumed that a group of five nodes move in straight lines towards a target destination. With this baseline, we considered RPGM model to be the most suitable model to emulate real life scenarios. As described in section 4.1.2, the important characteristic of RPGM model is that each node in the group deviates speed and direction randomly from that of a group leader. The movement of the group is defined by the motion of the group leader. In general, movement of a group member can be characterized as follows [17]:

1. $V_{member}(t) = V_{leader}(t) + random() * SDR * max\_speed$

2. $\Theta_{member}(t) = \Theta_{leader}(t) + random() * ADR * max\_angle$

where,
SDR is speed deviation ratio and is greater than equal to zero. ADR is angle deviation ratio and is less than equal to one.

SDR and ADR are used to control the deviation of the velocity of group members from that of the group leader. Two different scenarios were considered to evaluate the performance of ad hoc routing protocols. First, a checkpoint file was generated using mobility generator for ns-2 to get the positions of five nodes(including group leader) over a simulation time period of 300 seconds. We emulated two mobility patterns using

RPGM model. The parameters for first checkpoint file were SDR=60 and ADR=0.9 and for second checkpoint file SDR=10 and ADR=0.4. With the help of checkpoint files, relative distances between the nodes were calculated. These distances were then mapped to Packet Error Rate (PER) of the wireless links between nodes of the group. The matrix for distance mapped to PER is described in section 4.1.3.1.

We evaluated the performance of two routing protocols, namely AODV and OLSR, under the above mentioned scenarios. The performance was evaluated based on two metrics (1) end-to-end throughput and (2) control overhead. With mobility, the valid routes may become unavailable, thus causing the packets to be dropped and resulting in throughput degradation and increased control overhead.

4.2.1 Experimental parameters and Setup

The main objective of this thesis was to study the behavior of AODV and OLSR in mobile environment while keeping other conditions identical for both the protocols. The experiments were conducted on ORBIT testbed as mentioned in section 3. The experimental configurations that were kept same throughout the experiments are:

- Wireless card in 802.11a mode using channel 48.
- Implementation for AODV used was AODV-UU 0.9.5 and for OLSR was OLSRd 0.4.1.
- Topology creation using MACKill.
- The physical layer rate for all the experiments was 36Mbps.
- There were two flows in the network with a offered load of 1Mbps each for all the experiments.

4.2.2 Mobility Experiment Results

4.2.2.1 Straight Line Mobility Pattern

We conducted experiment with five nodes moving together in straight lines at an angle of 45 degrees. The relative distances between the nodes as they move towards the target was mapped to PER values using the matrix constructed in section 4.1.3.1. Note that the experiments are discrete in time and not continuous, i.e., if a node has (x1,y1) coordinates in time t1 then at time t2 it will have (x2,y2) coordinates. Ten distinct time positions were considered for this experiment. The time taken by a node to move from one time position to another was 30 seconds and 60 seconds respectively for the two experiments conducted. The two experiments were performed keeping all the other experimental parameters the same. Two data flows were configured for each scenario with an offered load of 1Mbps each, thus, a total network load of 2Mbps as shown in figure 4.3. In addition to measuring the aggregate throughput, protocol overhead was also calculated. For sniffing the packets from the network to measure the overhead, a node was designated as sniffer node. At the end of the experiment, the packets captured using tcpdump utility were analyzed using Wireshark, a publicly available network protocol analyzer. Figures 4.5(b) and 4.5(a) show the aggregate throughput obtained from the two experiments conducted. The observations of the experiments are as follows:

- After first 40 seconds, presumably once the routes are established, the average throughput for AODV is 1.92 Mbps and the same for OLSR is 1.89 Mbps for 60 seconds scenario. In a static environment with a linear two-hop topology, the throughput achieved was 99% of the offered load whereas in our mobile scenario, it is 96% for AODV and 94% for OLSR. While there is a cost of mobility for both the protocols, AODV is only marginally better than OLSR.

- The standard deviation of throughput for both the protocols is comparable in 60 seconds scenario. However, after first 40 seconds, the standard deviation of OLSR is more than 2.5 times that of AODV. OLSR throughput dropped below 1 Mbps for 3% of the duration indicating that one of the flow was not able to send the data packets. Incidentally, these interruptions happened in approximately 60 seconds intervals matching with the frequency of discrete movements.

- Similar results were obtained for 30 seconds scenario where standard deviation of throughput for OLSR is about 2.74 times that of AODV . The average throughput for AODV was 1.91Mbps and the same for OLSR was 1.84Mbps, which is comparable.

- The protocol overhead for AODV is 42% higher than OLSR for 60 seconds scenario as shown in figure 4.4. To compare it with the two-hop topology in static environment, the overhead for AODV is 2.6 times that of OLSR. It is interesting to note that the increase in overhead due to mobility is much lower for AODV compared to OLSR. Similar results were obtained for 30 seconds scenario.
- In addition to above observation, it was observed that as the time to move from one time position to other is decreased, or in other words mobility speed is increased, overhead for AODV for 30 seconds case jumped to 2.27 times to that of obtained in 60 seconds scenario. Similar increase was observed for OLSR where it jumped 2.17 times.

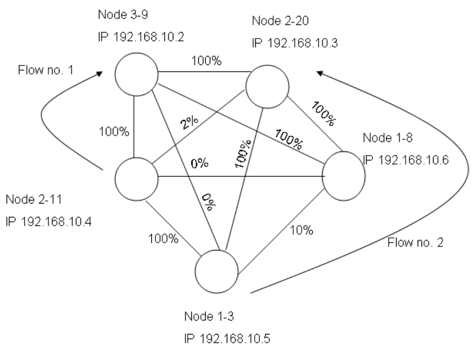


Figure 4.3: Topology - Straight line mobility pattern

4.2.2.2 Reference Point Group Mobility Pattern

We studied two different scenarios for RPGM model. As mentioned in section 4.2, there are two parameters that can be changed to emulate different mobility patterns, namely Angle Deviation Ratio(ADR) and Speed Deviation Ratio(SDR). For first experiment, we kept ADR=0.9 and SDR=60 where as for the second experiment, we kept ADR=0.4 and SDR=10. Also, for the first case, the time taken by a node to move from one coordinate to other coordinate was configured to be 30 seconds and for the other scenario

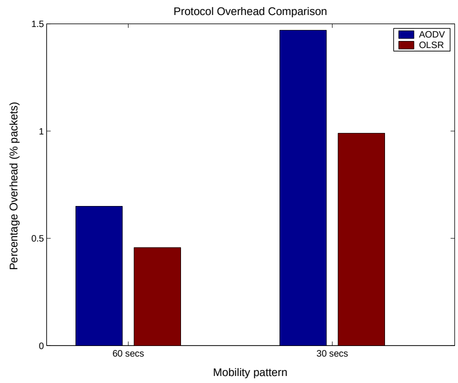


Figure 4.4: Protocol overhead comparison for straight line mobility

it was 5 seconds. First scenario is discrete in time where nodes move at faster speed and snapshots are taken at every 30 seconds. In the second scenario, nodes move slowly and snapshots are taken every 5 seconds. Figures 4.6 and 4.7 show how PER changes with time. It is reasonable to characterize the second scenario as somewhat continuous in time. Rest of the thesis refers the first scenario as "discrete" and the second scenario as "pseudo-continuous".

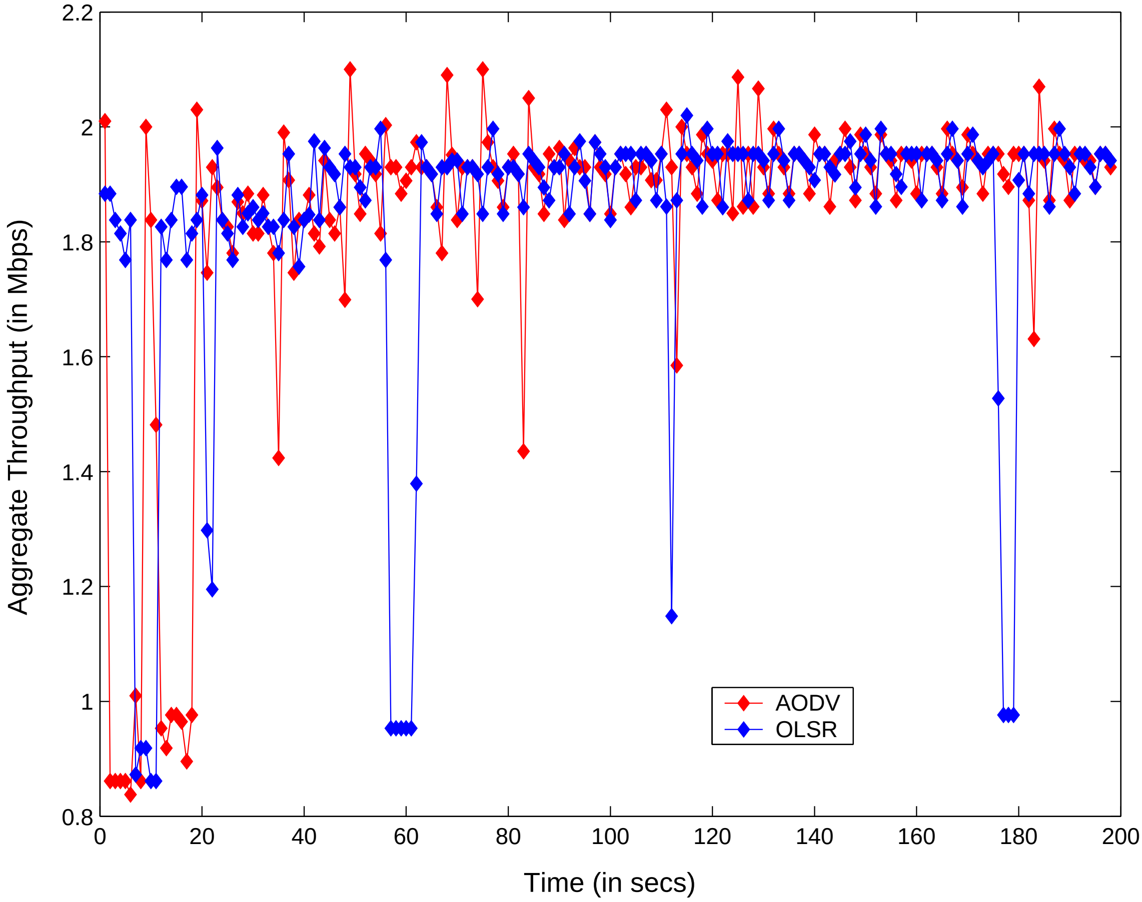
The topology used was same as in case of straight line mobility pattern. The throughput performance is shown in figure 4.8. Below are the main observations:

- As before, first 40 seconds are removed from following calculations. AODV has 8.6% higher throughput compared to that of OLSR for the discrete scenario. On the other hand, AODV has 11% lower throughput compared to that of OLSR for the pseudo-continuous scenario.

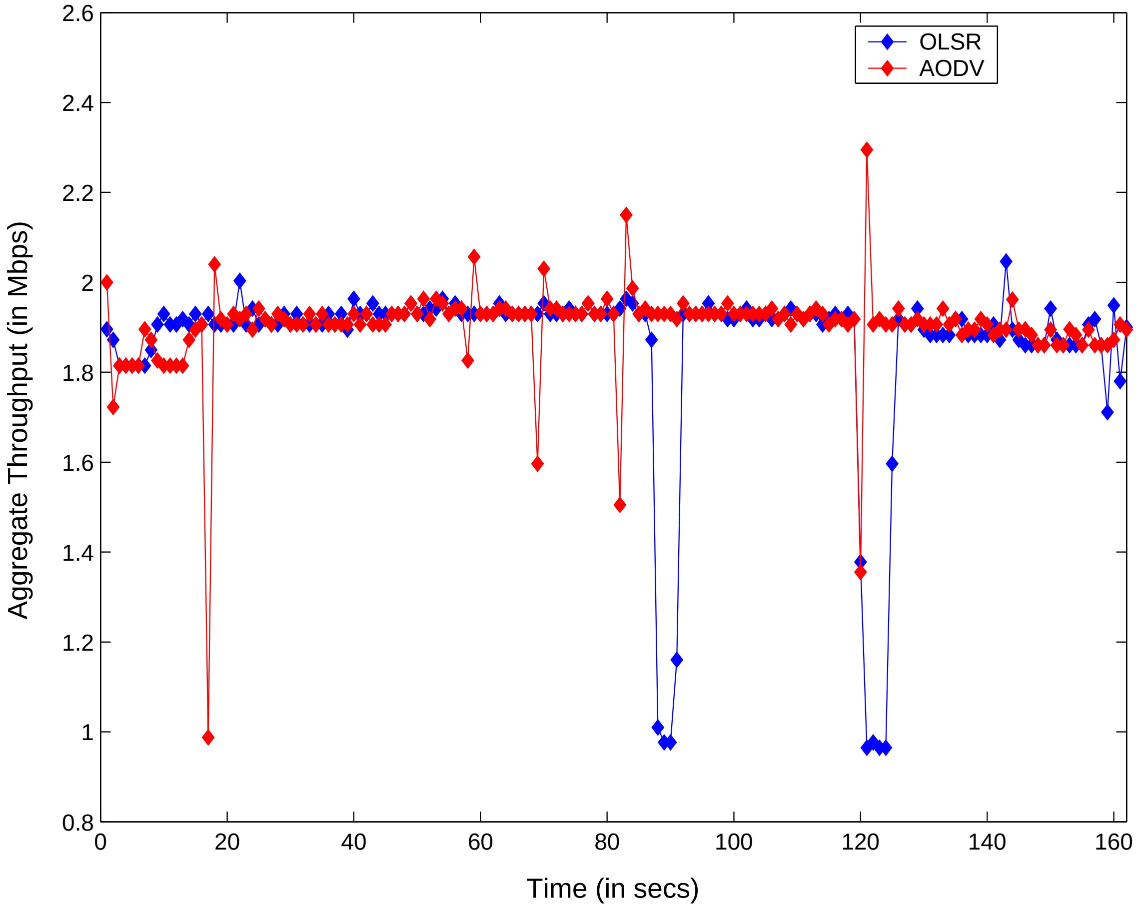
- As compared to straight line mobility, throughput achieved is much lower in both the experiments for both the protocols. For instance, average throughput for AODV under RPGM model is 1.41 Mbps where as the same for straight line model is 1.9 Mbps.

- The standard deviation for AODV is 26% lower than that of OLSR for the discrete scenario. However, the standard deviation for AODV is 12% higher than that of OLSR for the pseudo-continuous scenario.

- In the discrete time experiment, atleast one flow was down for 5% of the time for



(a) Performance - Straight line mobility pattern with 60 secs



(b) Performance - Straight line mobility pattern with 30 secs

Figure 4.5: Performance - Straight line mobility pattern

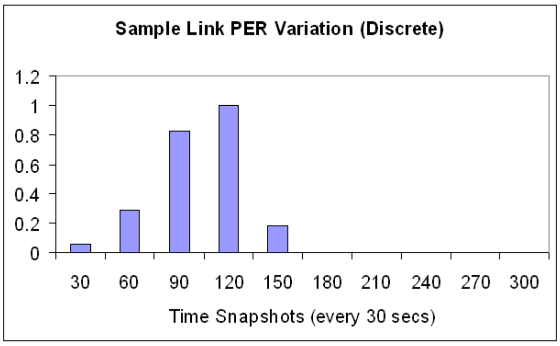


Figure 4.6: Sample PER snapshot for discrete scenario

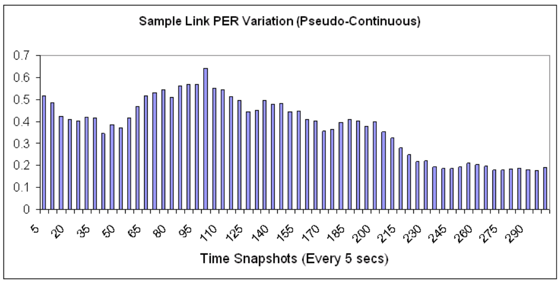


Figure 4.7: Sample PER snapshot for continuous scenario

AODV and 12% of the time for OLSR. The numbers for the pseudo-continuous experiment are 34% and 23% respectively. Again, it is not clear if one protocol performs better than the other. However, as expected, both the protocols show higher fluctuation in throughput in the pseudo-continuous scenario where time positions were changed more frequently.

- AODV suffered from deadlinks for 3% of time where as OLSR had deadlinks for 8% of the time for the discrete time experiment. In the pseudo-continuous scenario, there were less deadlinks for OLSR as compared to AODV. The links were down for 2.7% of the time for AODV and approximately 2% for OLSR.

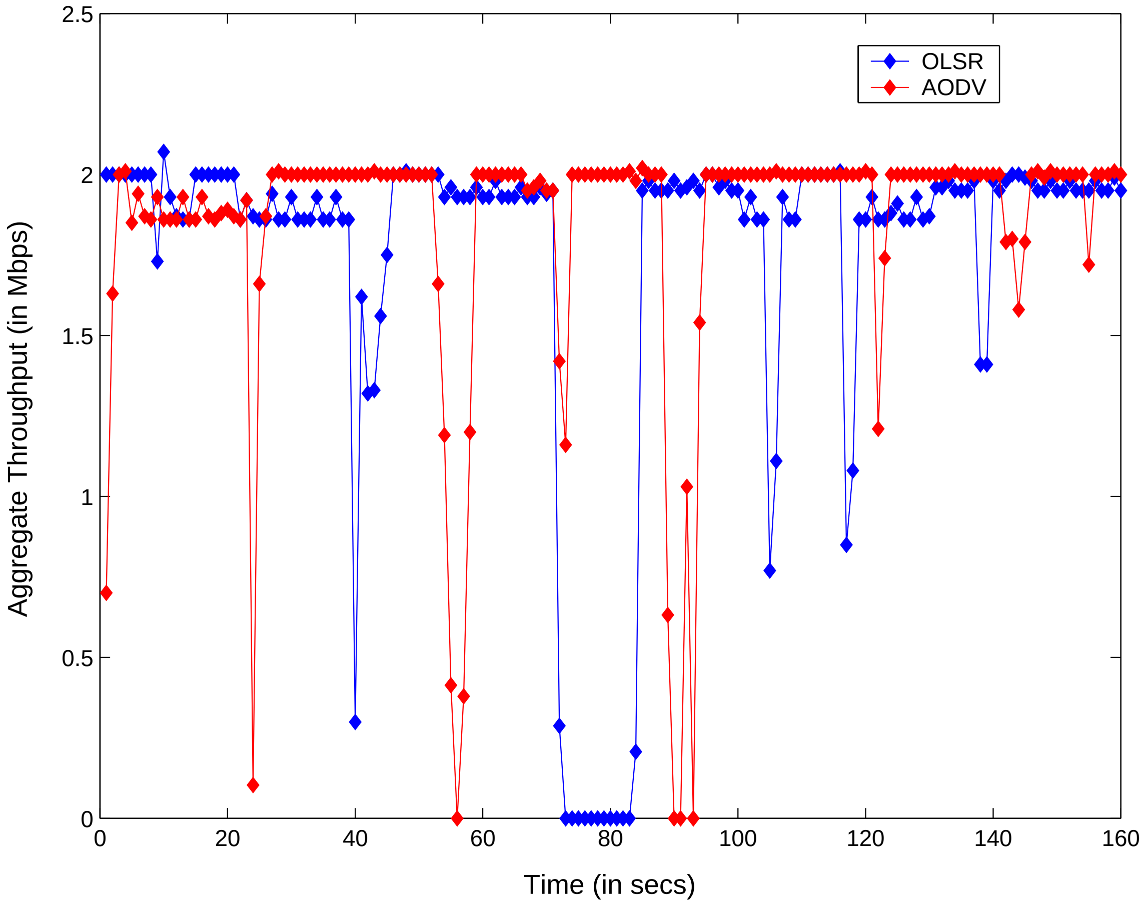
- Protocol overhead, as expected, is high for AODV as compared to OLSR, as shown in table 4.3.

- The protocol overhead for AODV was 1.72 times higher than that of OLSR for the discrete time experiment whereas the same increased to 2.26 times in the pseudo-continuous case. But if we compare the overhead for AODV (in the discrete case) to static two-hop linear topology, it is 6.39 times that in mobile environment. The same comparison for OLSR reveals 9.8 times higher overhead in mobile environment.

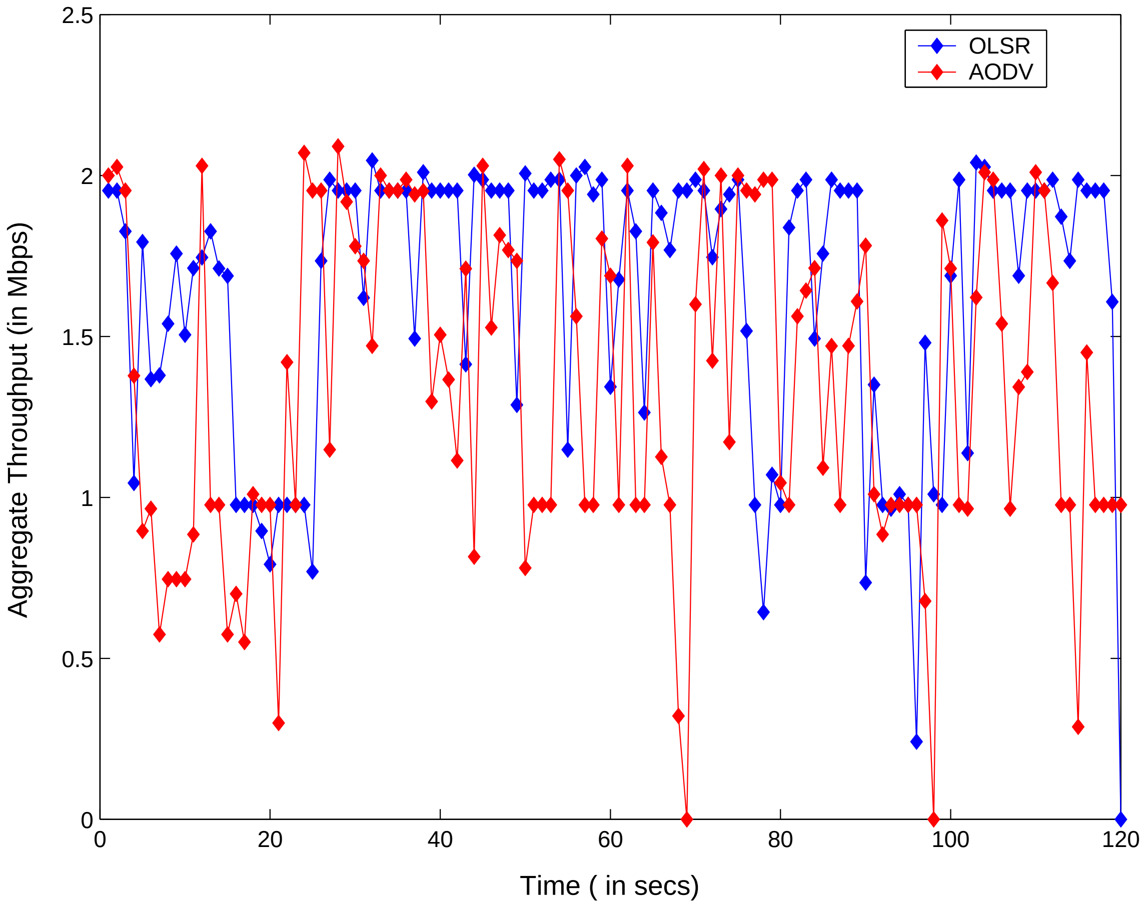
To summarize, in discrete scenario, AODV had marginally higher throughput and lower fluctuations compared to those of OLSR. However, OLSR performed slightly better in terms of mean throughput and standard deviation in pseudo-continuous scenario.

Table 4.3: Protocol overhead (in percent packets)

Protocol	Discrete	Pseudo-continuous
AODV	0.946	1.084
OLSR	0.547	0.479



(a) Performance - RPGM mobility model with SDR=60



(b) Performance - RPGM mobility model with SDR=10

Figure 4.8: Performance - RPGM mobility model

Chapter 5

Conclusion and future work

AODV and OLSR are among the most popular protocols for mobile ad hoc networks. In this thesis, we studied the impact of mobility on the performance of these two routing protocols. It was seen if the nodes are static and are in linear topology, OLSR performs better than AODV in terms of throughput. Another interesting observation was the protocol overhead involved in maintaining links. It is intuitive for on-demand routing protocol to have lower protocol overhead as compared to table-driven because control packets are exchanged only when sender has data packets to transmit. But our experimental results showed higher protocol overhead for AODV. There was no significant relationship found between packet size and throughput achieved for both the protocols. Also, when the number of hops were increased in a route, observed mean throughput for OLSR was higher than AODV, indicating that throughput scales down as the number of nodes go up. Even in static complex topology (a group of 10 nodes) with three consecutive flows, OLSR outperformed AODV. Based on the static experiment results, it can be concluded that OLSR performs better than AODV.

Another observation was that if a node comes into and goes out of transmission range very rapidly (like every 1 second or 5 seconds), both the protocols do not modify their routing protocols to change the route and hence do not respond to the change in topology resulting in a stable throughput throughout the experiment. However, when a node comes into and goes out of transmission range at slower pace (like every 30 second or 60 seconds), both the protocols change their routes. As expected, OLSR, due to its proactive nature, switches to a shorter path much more often than AODV. Due to excessive switching, OLSR provides lower throughput in this scenario.

To study the impact of mobility on performance of routing protocol, we used straight line mobility and group mobility models. For straight line mobility, we emulated the mobility in such a way that a group of nodes moved in straight line. Evaluation results show that AODV had higher mean throughput and lower standard deviation. Note that the time snapshots for this study were discrete, taken every 30 seconds and 60 seconds for a 300 seconds experiment duration. Also, the protocol overhead for AODV was higher than that of OLSR. However, the interesting point to note that the increase in overhead due to mobility for AODV was much lower compared to OLSR.

The second mobility study was done using RPGM mobility model. The model represents the motion for a group of nodes. Each group has a group leader and group of nodes move to reach a pre-defined destination. Each node moves randomly about their own pre-defined reference point that follows group movement. The path that nodes follow depends on the path of group leader. We used two different scenarios for this model. The first scenario took time snapshots at every 60 seconds for a 300 seconds experiment duration. We called this scenario as discrete time scenario. The parameters configuration of the model were ADR=0.9 and SDR=60. For the second scenario, the time snapshots were taken at every 5 seconds and we called it pseudo-continuous scenario. The parameters configuration of the model for this scenario were ADR=0.4 and SDR=10. It was observed that in discrete time scenario, AODV had marginally higher throughput and lower fluctuations compared to those of OLSR. However, OLSR performed slightly better in terms of mean throughput and standard deviation in pseudo-continuous scenario. It seems if time snapshots are taken more frequently, OLSR performs better because it is able to find new routes to the destination quickly. Although it is counter intuitive, AODV had higher overhead in all the mobility experiments. It can be concluded that mobility does affect the performance of the routing protocol and the performance greatly depends on the parameters of mobility model used for their evaluation.

It is clear that there is considerable cost associated with mobility. Both the protocols show decrease in throughput, higher standard deviation, more dead links and higher overhead when compared to their respective performance in static environment.

Since we studied only intragroup interaction, it can be extended to study intergroup interaction. Further study can be done to explore more realistic mobility models like

Freeway and Manhattan. Also, hybrid routing protocol like ZRP can also be studied to get a comparison between proactive, reactive and hybrid. To get a more comprehensive understanding, other researched protocols like DSR, TORA, DSDV etc can also be studied.

References

[1] Haidong Wang, Brian Crilly, Wei Zhao, Chris Autry, and Sean Swank. Implementing mobile ad hoc networking (manet) over legacy tactical radio links. In *Proc. of IEEE Military Communications Conference(MILCOM)*, pages 1–7, Orlando, FL, October 2007.

[2] S.Corson and J.Macker. Mobile ad hoc networking(manet): Routing protocol performance issues and evaluation considerations. `http://www.ietf.org/rfc/rfc2501.txt`, January 1999.

[3] Josh Broch, David A. Maltz, David B. Johnson, Yih chun Hu, and Jorjeta Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Proc. of 4th annual ACM/IEEE Internaltional Conference on Mobile Computing and Networking*, pages 85–97, Dallas, TX, October 1998.

[4] Charles E. Perkins, Elizabeth M. Royer, Samir R Das, and Mahesh K Marina. Performance comparison of two on-demand routing protocols for ad hoc networks. In *Proc. of 19th annual joint Conference of IEEE Computer and Communication Societies(INFOCOM)*, volume 1, pages 3–12, Dallas, TX, October 2000.

[5] M.Corson, S.Batsell, and J.Macker. Architectural considerations for mobile mesh networking. `http://tonnant.itd.ml.nay.mil/mmnet/mmnetRFC.txt`, May 1996.

[6] C.Perkins, E.Royer, and S.Das. Ad hoc on-demand distance vector routing, July 2003. IETF RFC 3561.

[7] T.Clausen and P.Jacquet. Optimized link state routing protocol, October 2003. IETF RFC 3626.

[8] Andrea Tonnesen. Mobile ad-hoc networks. `www.olsr.org/docs/wos3-olsr.pdf`.

[9] D.Raychaudhuri, I.Seskar, M.Ott, S.Ganu, K.Ramachandran, H.Kremo, R.Siracusa, H.Liu, and M.Singh. Overview of the orbit radio grid testbed for evaluation of next-generation wireless network protocols. In *Proc. of IEEE Wireless Communications and Networking Conference(WCNC)*, volume 3, pages 1664–1669, New Orleans, LA, March 2005.

[10] Mark Gates, Ajay Tirumala, Jon Dugan, and Kevin Gibbs. Iperf: The tcp/udp bandwidth measurement tool. `http://dast.nlanr.net/Projects/Iperf/iperfdocs_1.7.0.php`, March 2003.

[11] E.Norstrom. Aodv implementation on linux. aodv-uu. `http://core.it.uu.se/core/index.php/AODV-UU`, 2007.

[12] Olsrd: An ad hoc wireless mesh routing daemon. `http://www.olsr.org/`.

[13] Wireshark. `http://www.wireshark.org/`, 1998.

[14] Xiaoyan Hong, Mario Gerla, Guangyu Pei, and Ching-Chuan Chiang. A group mobility model for ad hoc wireless networks. In *Proc. of ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems(MSWiM)*, pages 53–60, Seattle, WA, August 1999.

[15] Theodore S. Rappaport. *Wireless Communications: Principles and Practice.* Prentice-Hall of India, 2005.

[16] Praveen Gopalakrishnan, Predrag Spasojevic, Larry Greenstein, and Ivan Seskar. A method for predicting the throughput characteristics of rate-adaptive wireless lans. In *Proc. of IEEE Vehicular Technology Conference*, Los Angeles, CA, September 2004.

[17] Fan Bai, Narayan Sadagopan, and Ahmed Helmy. User manual for important mobility tool generators in ns-2 simulator. urlhttp://nile.usc.edu/important/mobility-user-manual.pdf, Feb 2004.