

## Chapter 1

# Introduction

### 1.1 General Description

Wireless mobile communications for personal use is and will continue to be a part of our everyday life. Standards for wireless communications have boosted the market; manufacturers equip everything, from powerful laptops to small devices, with hardware support for different radio technologies. With the development of *IEEE 802.11* wireless standards [1], the popularity of wireless networks has increased dramatically over the last few years, both in industry and in home networking. *Bluetooth* [2] is another technology, which aims at being small, cheap and built-in in almost everything. Both IEEE 802.11 and Bluetooth utilize the license-free 2.4 GHz band [3]. The work presented in this thesis uses IEEE 802.11 based wireless cards for deploying a *Mobile Ad hoc Network (MANET)* [4].

The term “ad hoc” often means improvised or for the needs of the moment for a specific purpose. A MANET is a collection of self-organized wireless mobile hosts forming a network without the aid of any established infrastructure or centralized administration stations unlike cellular wireless networks [5]. Since the mobile hosts move around causing the network topology to change frequently, the hosts must rely on each other to relay the data packets in such a multi-hop situation. The type of devices in MANET can range from embedded systems like sensors to powerful computers inside vehicles.

A typical usage scenario for MANET is a disaster area where pre-existing communications infrastructure has been eliminated due to e.g., an earthquake or a terror attack. In an emergency situation, the rescue teams need to quickly establish connections to the other rescue teams for saving lives. Given the maturity of the technology, a MANET would be a good choice for deployment. All nodes – carried by people and/ or vehicles – would continuously organize themselves and cooperate in order to forward each other’s traffic when needed. Another usage would be a university campus area or a *Small Office HOme (SOHO)* network with no, or limited, wireless LAN coverage, where groups might wish to establish temporary wireless network in order to work on collaborative projects. A MANET can also be connected to other networks such as an Internet or a *General Packet Radio Service (GPRS)* [6] network, via a gateway with minimal changes. Possible applications in a MANET do, of course, depend on the scenario. In a disaster area or military scenario it could be voice communication or transmission of camera snap shots. In a SOHO or university scenario, it could be spontaneous meeting; the applications could be file sharing, or in case where the Internet

connectivity is present. The surrounding physical environment significantly attenuates and distorts the radio transmissions, and the signal quality is inversely proportional to distance. So the effective transmission area of any node is limited and thus the effective throughput may be less than the radio's maximum transmission capacity. The research in MANET is primarily simulation studies, which may not incorporate all the factors of nature, so it is necessary to evaluate MANET in real world scenarios for commercial availability.

## **1.2 Research Problems Addressed in this Thesis**

The following are the issues that were investigated in this thesis.

- (i) Comparing two MANET implementations - The two implementations by Uppsala University (UU) and University of California, Santa Barbara (UCSB) were tested with node mobility.
- (ii) Comparing the performance of MANET in an indoor/outdoor environment - The effect of the real world environment on the routing protocol was assessed by comparing the experimental results obtained in an indoor environment (building) with an outdoor environment (parking lot).
- (iii) Key parameters in determining MANET performance - A network designer needs to select appropriate metrics and a methodology to evaluate any networks. The metrics used in this thesis are throughput and latency. One of the key parameters that affect these metrics is block size in addition to well-known system parameters such as link bandwidth, link error rate, traffic intensity, and node mobility. This

thesis presents a unique and a more visual means to analyze MANETs and find the optimal block size.

- (iv) Differences between simulations and the real world - Although simulations can reduce the time of development of any research, sometimes it can mislead the research path, so it is important to conduct real world experimental results and compare it with simulation results to assess the credibility of the simulator.
- (v) Analyzing the effects of interference in a small MANET using simulation - Interference is a common problem particularly in MANET scenarios, so this thesis presents scenarios that affect the routing protocol in MANETs and investigates the results.

The rest of this paper is organized as follows. Chapter 2 describes an overview of MANET routing protocols. Chapter 3 describes experimental test bed and presents the results, addressing the issues (i), (ii) and (iii) as listed above. Chapter 4 describes simulation methodology and presents the results on the issues (iv) and (v).

## Chapter 2

# Routing in Mobile Ad-Hoc Networks with AODV

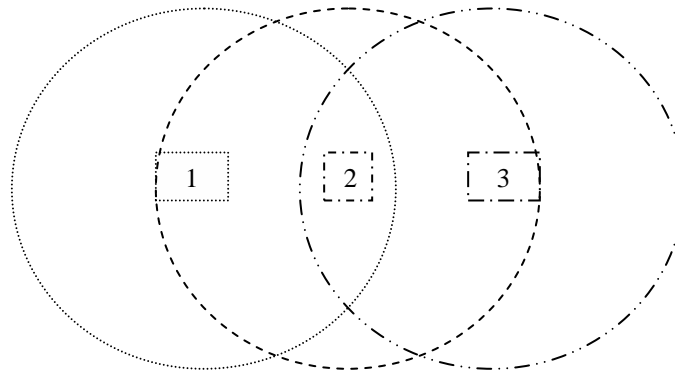
This chapter describes the routing protocol *Ad-hoc On Demand Distance Vector (AODV)* [7] used in MANETs. Section 2.1 gives a brief introduction about the routing protocols in MANETs. Sections 2.2 and 2.3 describe key features of *Destination Sequence Distance Vector (DSDV)* [8] and *Dynamic Source Routing (DSR)* [9], which offer the basic building blocks of AODV. Section 2.4 describes AODV algorithm.

### **2.1 General Description**

Conventional routing protocols are based on either *distance vector* or *link state* algorithms [8]. Distance vector protocol makes shortest path decisions based on a hop count metric, while link state makes decisions based on cost of each link. These

algorithms assume that links between routers occasionally go down or come up, and sometimes the cost of a link may change due to congestion, but routers do not generally move around dynamically, shifting major portions of the network topology back and forth.

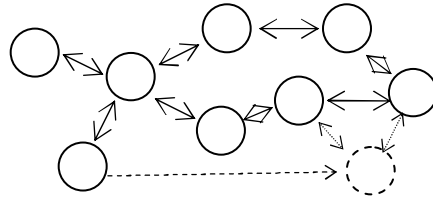
However, they are not designed for the type of dynamic topology changes that may be present in MANET. A natural method to provide routing in an ad hoc network is to simply treat each one of mobile hosts as a router and to run a conventional routing protocol among them [10]. For example, mobile host 2 in Figure 2.1 acts as a router between the “network” directly reachable by mobile host 1 and the “network” directly reachable by mobile host 3, i.e. mobile host 1 transmits its packets for mobile host 3 through mobile host 2.



**Figure 2.1** A typical ad hoc network.

Since topology changes at any time in MANET scenarios, convergence to stable routes may be quite slow, particularly with distance vector algorithms. Link state protocols will take a path which has more hops, but that uses a faster medium over a path using a slower medium with fewer hops. But distance vector algorithms require less

processor overhead, compared to link state. The speed of convergence, in distance vector algorithms, may be improved by sending routing updates more frequently, but such a shift only wastes more bandwidth and battery power when topology does not change much.



**Figure 2.2** A mobile ad hoc network (MANET).

The primary function of a MANET routing protocol is to create, maintain and re-create routes. To run a MANET (as in Figure 2.2) smoothly, it requires a quick and adaptive routing protocol, at the same time it should not consume too much of bandwidth. Some of the desired properties for such a protocol are distributed operation, loop free, demand based operation, unidirectional link support, security, power conservation, multiple routes and quality of service support [11, 12]. It is difficult to incorporate all the properties mentioned above into one protocol.

The initial approaches for routing in MANETs were *proactive* i.e. the routes are established and updated even if they are never needed and this requires the protocol to exchange control messages periodically. However, in MANETs, channel bandwidth and node energy are two important limiting factors and hence it is a good idea to use *reactive* routing, where routes are established as and when needed. The *Ad-hoc On-demand Distance Vector (AODV)* [7] is one such algorithm, which is a combination of *proactive* and *reactive* routing.

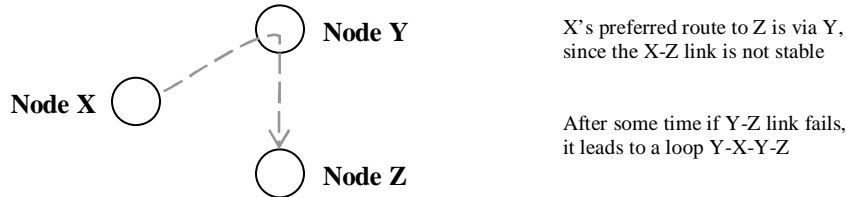
The initial design of AODV was undertaken to improve upon the deficiencies in DSDV, a proactive protocol. The main goal of AODV was to reduce the need for system-wide broadcasts to counter the problems due to the characteristics of wireless medium. Since it is necessary to conserve the bandwidth as much as possible, the architects of AODV found it necessary to incorporate the on-demand route discovery and route maintenance mechanisms from DSR, a reactive protocol. The following sections illustrate the development of AODV, based on DSDV and DSR.

## **2.2 DSDV (Destination Sequence Distance Vector)**

DSDV [8] is a hop-by-hop distance vector routing protocol requiring each node to periodically broadcast routing updates. DSDV is an entirely proactive protocol. All nodes keep a routing table that holds the routes for all reachable nodes. The advantage of this approach is that a packet can be forwarded immediately if there is an entry for its destination in the routing table.

DSDV routing protocol is derived from a classical distance vector algorithm, *Distributed Bellman-Ford* (DBF) algorithm, where in each node maintains the shortest distance to all destinations through all of its neighbors. Periodically each node creates a vector containing shortest distance to each destination, and sends this vector to its neighbors. Upon receiving a vector from a neighbor, a node updates its minimum distances to all the destinations via this neighbor. DBF was designed for fixed networks, where the links are stable. But in wireless networks the links are not stable so routing loops are formed. Imagine X, Y, Z are connected in a triangle as shown in Figure 2.3. X's preferred route to Z is via Y while Y's is direct. Now the Y-Z link fails; Y will see that X

is advertising a route to Z, and so Y's new preferred route to Z will be via X. So we get the routing loop Y-X-Y-Z, which is incorrect.



**Figure 2.3** Basic functioning of DSDV.

Enhancements are made in DSDV in order to avoid this looping problem. DSDV includes a sequence number with each route. Each node periodically broadcasts its own routing table. The transmitted routing table includes the sequence number created by the source. A route decision is based on the sequence number (larger sequence number means a more recent route which is favorable). If two routes have the same sequence number then the decision is based on the metric (lower metric is favorable).

While loops are usually formed due to stale routes, DSDV uses the sequence numbers to remove stale routes from the routing table of each node. Consider the network shown in Figure 2.3; let  $S(X)$  and  $S(Y)$  be the destination sequence numbers for node Z as stored at node X and node Y respectively. As node X receives routing information from node Y about a route to node Z, node X verifies the following:

- If  $S(X) > S(Y)$ , then node X ignores the routing information received from node Y.
- If  $S(X) = S(Y)$ , and cost of going through node Y is smaller than the route known to node X, then node X sets node Y as the next hop to node Z.

- If  $S(X) < S(Y)$ , then node X sets node Y as the next hop node Z, and  $S(X)$  is updated to  $S(Y)$ .

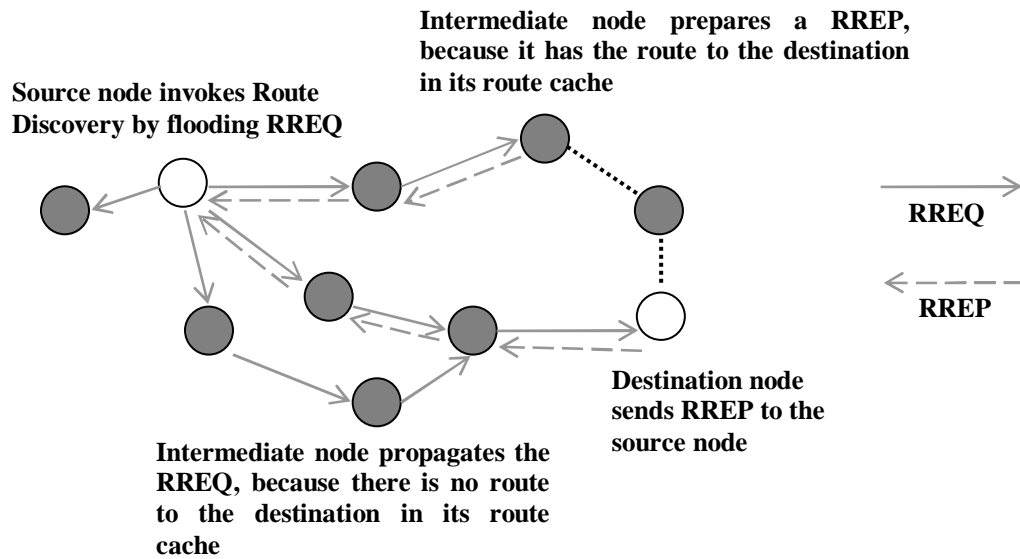
### 2.3 DSR (Dynamic Source Routing)

DSR [9] uses source routing rather than hop-by-hop routing. When using source routing, each packet to be routed carries in its header the complete, ordered list of nodes through which the packet must pass. A key advantage of source routing is that intermediate nodes do not need to maintain up-to-date routing information in order to route the packets they receive, since the packets themselves already contain all the necessary routing information. This fact, coupled with the dynamic, on-demand nature of the DSR's route discovery, completely eliminates the need for the periodic route advertisement and neighbor detection packets, present in other protocols.

The DSR protocol consists of two mechanisms: *route discovery* and *route maintenance*. Route discovery is the mechanism whereby a source node wishing to send a packet to a destination node obtains a source route. Route maintenance is the mechanism, where the source node is able to detect if the network topology has changed, such that it can no longer use the route to the destination node. When route maintenance indicates a source route is broken, the source node attempts to use any other route it happens to know, or invokes route discovery again to find a new route.

To perform route discovery, the source node broadcasts a *Route Request* (RREQ) packet as shown in Figure 2.4. Each node that hears the RREQ packet forwards a copy of the request, if appropriate, by adding its own address to a source route being recorded in the request packet and then rebroadcasts the RREQ packet. The forwarding of RREQs

is constructed so that copies of the RREQ propagate hop-by-hop outward from the source node, until it reaches the destination node or it reaches another node that can supply a route to the destination. The RREQs are forwarded (1) if the node is not the target of the request, (2) if the node is not already listed in the recorded source route in this copy of the request, and (3) if the node has not recently seen another RREQ packet belonging to this same route discovery. When the destination node receives the RREQ, the recorded source route in the request identifies the sequence of hops over which this copy of the request reached the destination. Destination node copies this recorded source route into a *Route Reply* (RREP) packet and sends this RREP packet back to the source node.



**Figure 2.4** DSR's route discovery mechanism

All source routes learned by a node are kept in a *route cache*, which is used to reduce the cost of route discovery. Further, when an intermediate node receives a RREQ packet it searches its own route cache for a route to the destination node. If the

intermediate node finds a route, it will not propagate the RREQ packet, but instead it sends a RREP to the source node (as shown in Figure 2.4), by concatenating the recorded source route contained in the RREQ packet to the cached route to the destination node present in its route cache.

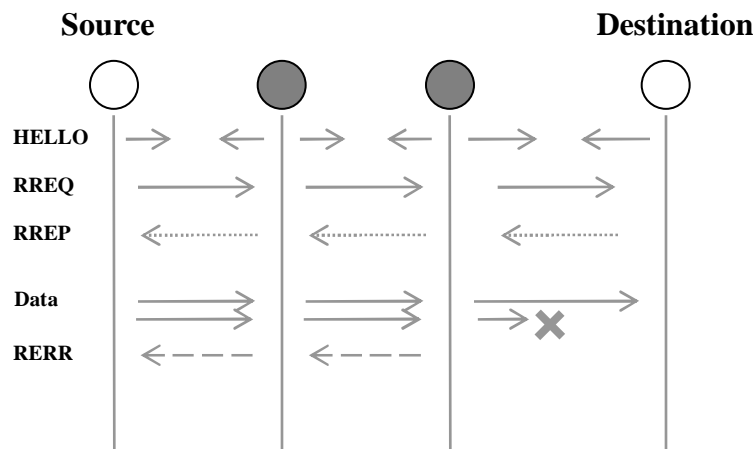
## 2.4 AODV (Ad-hoc On-Demand Distance Vector)

AODV [7] is essentially a combination of both DSR and DSDV. It borrows the basic on-demand mechanism of route discovery and route maintenance mechanisms from DSR, plus hop-by-hop routing, sequence numbers, and periodic beacons from DSDV.

When a source node needs a route to a destination node, it broadcasts a RREQ message to its neighbors, including the last known sequence number for that destination. The RREQ is flooded in a controlled manner through the network until it reaches a node that has a route to the destination. Each node that forwards the RREQ creates a *reverse route* for itself back to source. When the RREQ reaches the destination node, this node generates a RREP that contains the number of hops necessary to reach the destination node and the sequence number most recently seen by the node generating the RREP. Each node forwarding the RREP packet to the originator of the RREQ (source node), creates a *forward route* to the destination node. The state created in each node along the path from source to destination is hop-by-hop state; i.e., each node remembers only the next hop and not the entire route, as would be done in source routing.

In order to maintain routes, AODV requires that each node periodically transmit a HELLO message, with a default rate of once per second. Failure to receive three consecutive HELLO messages from a neighbor is taken as an indication that the link to

the neighbor in question is down. Alternatively, the AODV specification briefly suggests that a node may use physical layer or link layer methods to detect link breakages. When a link goes down, any upstream node that has recently forwarded packets to a destination using that link is notified via a Route Error (RERR) containing an infinite metric for that destination. Upon receipt of such a RERR, the source node must acquire a new route to the destination node using the route discovery mechanism.



**Figure 2.5** Basic functioning of AODV.

## Chapter 3

# Evaluation and Test Bed

Routing in MANET is a difficult problem and has received a lot of attention. Many routing protocols such as DSR [9], AODV [7], DSDV [8], and TORA [13], have been proposed for MANETs. Most of these protocols are validated based on simulation study. However it is necessary to validate MANET routing protocols in real world. This chapter discusses the implementation and performance evaluation of a MANET in real world scenarios with AODV routing protocol. Section 3.1 describes the survey of existing AODV implementations. Section 3.2 describes different real world scenarios tested, and their significance. Section 3.3 describes the evaluation tool used to assess the MANET performance. Section 3.4 presents experimental results and discussion.

### 3.1 Survey of Existing Implementations

Few implementations of AODV are publicly available: *mad-hoc* [14], *AODV-UIUC* [15], *AODV-UU* [16] and *AODV-UCSB* [17], which are user space implementation and *kernel AODV* [18], which is a kernel space implementation. The most publicly recommended implementations of AODV are kernel AODV, AODV-UU and AODV-UCSB [19]. Kernel AODV, a kernel space implementation, has few advantages; it operates faster than user space implementation and it does not require any mechanism for transferring packets from kernel to user space and vice versa. However, AODV-UU and AODV-UCSB were chosen in this thesis because it kernel AODV is less portable and difficult to maintain, and reduces the protocol functionality and weakens memory management. Also kernel AODV has some known bugs like memory leaks, mishandling `malloc ()` function, assertion failures, routing loops and etc [21].

#### Implementation of AODV-UU and AODV-UCSB

AODV-UU and AODV-UCSB are implemented in the Linux and written in C. The basic logical structure of AODV-UU and AODV-UCSB is the same. AODV-UU is based on packet handling support provided by *netfilter* [20] and two kernel modules *k\_route* and *libipq*. Netfilter's ability to process packets in user-space is what makes it effective in developing AODV. Netfilter queues packets in kernel space and sends the information about the packet to user-space over a netlink socket. The packets are queued so as to allow the AODV-UU routing daemon to process the packets in user-space. This approach is independent of kernel modifications. The *k\_route* module constantly updates the kernel routing table depending on the information sent by the AODV-UU routing daemon after

the packet processing. Routes may be added, changed or deleted. These are all the functionalities used at the kernel level. The libipq is used for use-space queuing of IP packets.

AODV-UCSB at the kernel level implements similar logic, but it does not use netfilter, instead they developed their own schemes to complement the functionality of netfilter. For instance when a packet arrives at the IP layer, AODV-UCSB queues the packet, and then sends the information about the packet to the AODV-UCSB routing daemon in the user-space for packet processing, and in the mean time it creates a dummy route table for the packet with a short expiry time, and when the AODV-UCSB routing daemon sends a desired action the kernel routing table is updated accordingly.

The packet processing in both implementations is quite similar, and is performed in user space. All the incoming and outgoing packets are processed; route discovery and packet buffering are performed as needed. If a packet is destined for another node, and an active node is available, the packet is forwarded using the next hop information from the AODV routing table.

### **Differences between AODV-UU and AODV-UCSB**

AODV-UCSB and AODV-UU are basically the same [21], except that AODV-UU supports *unidirectional link* detection and avoidance. In MANET scenarios, unidirectional links originate either due to difference in radio transmission power level (or receiver sensitivity) of the nodes or due to the difference in interference (or noise). Both these cases frequently occur in MANET scenarios. It is important to note that these unidirectional links formed can be transient in nature, so the effective throughput is

reduced. This thesis provides an insight into the effect of this feature in AODV-UU by comparing with AODV-UCSB that does not support unidirectional links.

### **3.2 Test Bed**

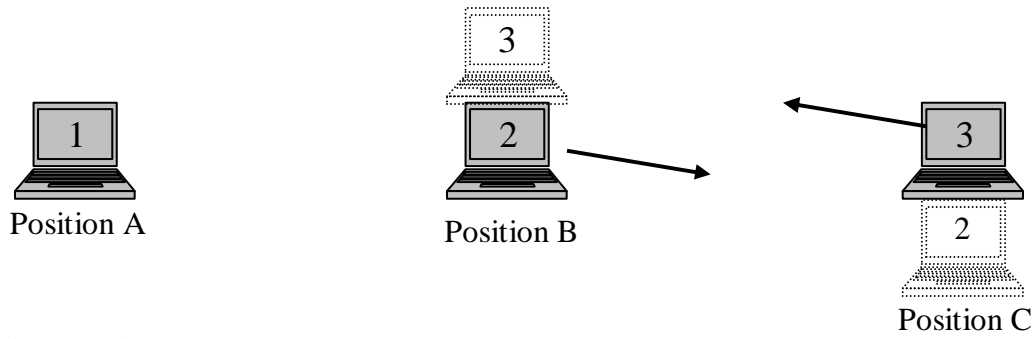
Three Toshiba laptops (with Intel Pentium III processors) running the Red Hat Linux operating systems (version 7.1) were used for these experiments. To evaluate the performance of a MANET, it is important to design a topology that characterizes the properties of a MANET. The basic idea was to study the performance of a MANET protocol according to the changes in the physical environment. For this purpose, the experiments are conducted in an outdoor environment and an indoor environment using AODV-UU. Also, we compare AODV-UU and AODV-UCSB to observe the effect of unidirectional links on the MANET protocol when all the nodes in the network are static. The effect of the physical structure on the MANET's route discovery time is also analyzed by implementing the protocol for a particular mobility pattern. The topologies implemented are explained in detail in the below.

#### **Experiment Environment**

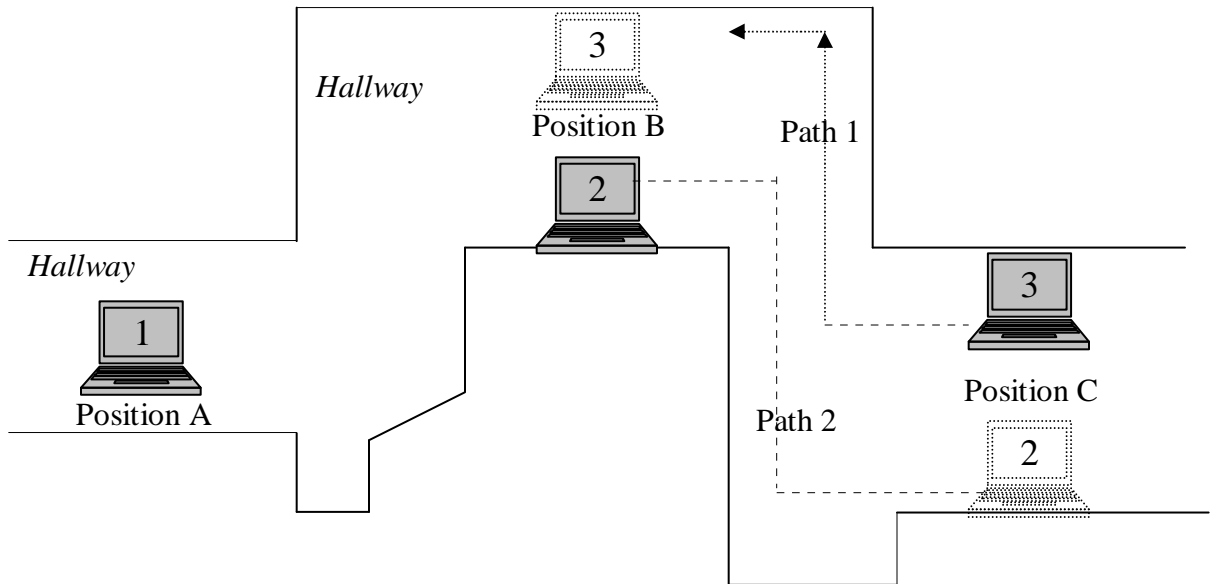
The experiments are conducted for both the static and mobile scenarios; and repeated for indoor and outdoor environments. The experiments for the outdoor scenario are implemented in a parking lot and those for the indoor case are implemented in a college building.

- Static scenario: All the laptops in this scenario were static and the experiment was conducted for varying packet sizes. The arrangements of the laptops for outdoor

and indoor scenarios are shown in Figures 3.1 and 3.2. Laptop 1 was placed at position A while laptop 2 and laptop 3 were placed at position B and position C respectively such that laptop 1 and laptop 3 are out of range.



**Figure 3.1** Experimental scenario: outdoor environment (static and mobile scenarios).



**Figure 3.2** Experimental scenario: indoor environment (static and mobile scenarios).

- Mobile scenario: Initially, the laptops were kept in the same positions as in previous case. Then, Laptop 3 was made to traverse towards position B and simultaneously. Laptop 2 was made to traverse towards position C as shown in

Figures 3.2 and 3.3. After Laptop 3 and Laptop 2 reach position B and position C respectively, they stay there for some amount of time, after which they are taken back to their original position. When Laptop 3 is at position B, it is in direct contact with Laptop 1. Laptops 2 and 3 were moving at normal walking speed and the experiment was conducted for a fixed packet size.

### 3.3 Evaluation Tool

The protocol was evaluated using software called NetPIPE [22]. NetPIPE helps answer the following questions, which surround network communications inherent to communication bound applications,

- a) How soon will a given data block of size  $k$  arrive at the destination?
- b) Which network and protocol will transmit size  $k$  blocks the fastest?
- c) What is a given network's effective maximum throughput and saturation level?
- d) What is block size  $k$  for which the throughput is maximized?

All these questions are answered by analyzing *throughput graph*, *network signature graph*, *saturation graph* and *mobility graph*. The NetPIPE software sends packets with increasing size and provides us with the following information: *time taken to transfer the block*, *throughput in bits/sec*, *number of bits in the block transferred*, and *number of bytes in the block transferred*. The idea behind increasing the packet size is to measure and compare the throughput for various packet sizes.

NetPIPE performs simple ping-pong tests, bouncing messages of increasing size between two nodes, across a network. Each data point involves many ping-pong tests to

provide an accurate timing. Latencies are calculated by dividing the round trip time in half. The following four graphs were used to analyze MANETs with NetPIPE.

- *Throughput graph* plots throughput versus block size; it is used to observe the throughput for a network.
  
- *Network signature graph* plots throughput versus response time. This graph is a new and unique way of viewing network performance data; the key is to use a logarithmic time scale horizontally instead of the transfer block size. Although unconventional, this graph represents perhaps a better approach to visualize network performance. All the necessary data are clearly visible and easy to extrapolate. The network latency coincides with the time of the first data point on the graph. The maximum attainable throughput is clearly shown as the maximum point on the graph.
  
- *Saturation graph* plots block size versus response time on a logarithmic scale. The key feature of this graph is to identify the saturation point. This is the point after which an increase in block size results in a near-linear increase in transfer time, effectively the knee of the curve. The time interval between the saturation point and the end of the recorded data is referred to as the saturation interval. In this interval, the graph monotonically increases at a constant rate i.e., the network throughput cannot be improved upon by increasing the block size. Thus the saturation point is nothing but the best suitable block size for the network.

- *Mobility graph* plots response time for each packet. This graph is unique to MANETs, because it analyzes the route discovery time in MANETs, and this graph provides the necessary information to monitor the route discovery time.

Throughout the experiments, we also concerned about the appropriate block size that optimizes MANET performance. Block size is the size of each data packet transmitted. The block size actually affects the performance of the protocol quite a bit, and it is essential for the network designer to choose the correct packet size to be transmitted to achieve optimal performance. At higher loads, the latency can be quite large with high degree of variability, which affects the performance of higher layer protocols such as UDP or TCP due to long packet delays. At lower loads, the number of packets required to transmit a file is high, so the average throughput achieved would be quite low when compared to the high load. So it is very important to choose appropriate block size to achieve the best results. This is probably the most neglected metric, since most of the researchers fix a block size usually 512 bytes and perform experiments or simulations. This thesis aims to at finding the appropriate block size for MANET scenarios.

### **3.4 Results and Discussion**

This section presents the results obtained from the experiments conducted. Section 3.4.1 discusses the test to verify the basic network connectivity. Section 3.4.2 compares

AODV-UCSB and AODV-UU. Section 3.4.3 deals with AODV-UU but for different environments i.e. outdoor and indoor environment as explained in section 3.2.

### **3.4.1 Network Connectivity Test**

We used *ping* command in our tests to see if all the nodes are active, and to monitor the node's ideal performance especially during mobile scenarios.

#### **Hello**

Each node periodically broadcasts HELLO messages. The connected node receives the HELLO messages, and installs a route to the other node. The following actions are verified:

- Correct reception of neighboring HELLO messages.
- Correct installation of route to neighboring node.
- Deletion of route when nodes are disconnected.

#### **2-hop RREQ/RREP**

Consider the network as shown in Figure 3.1, where in node 1, node 2 and node 3 form an ad hoc network. It is necessary to ensure if the RREQ/RREP procedures are performed correctly. So, the following actions are verified:

- Node 1 issues a RREQ for node 3.
- Node 2 receives the RREQ, and replies with a RREP.
- Node 1 receives the RREP, install the route, and pings are correctly received.

## **RERR**

After creating a route between node 1 and node 3 as in the previous test, the link between node 2 and node 3 was disconnected. The correct receipt of a Route Error (RERR) at node 1 demonstrates the correct operation of precursor nodes. The following actions are verified:

- Node 2 issues a RERR for node 3 and removes its route to node 3.
- Node 1 receives this RERR, and removes its route to node 2 also.

Successfully running all the above tests ensured that the AODV software was working properly. Later, the NetPIPE software was used to analyze the network.

### **3.4.2 Comparison of AODV-UCSB and AODV-UU**

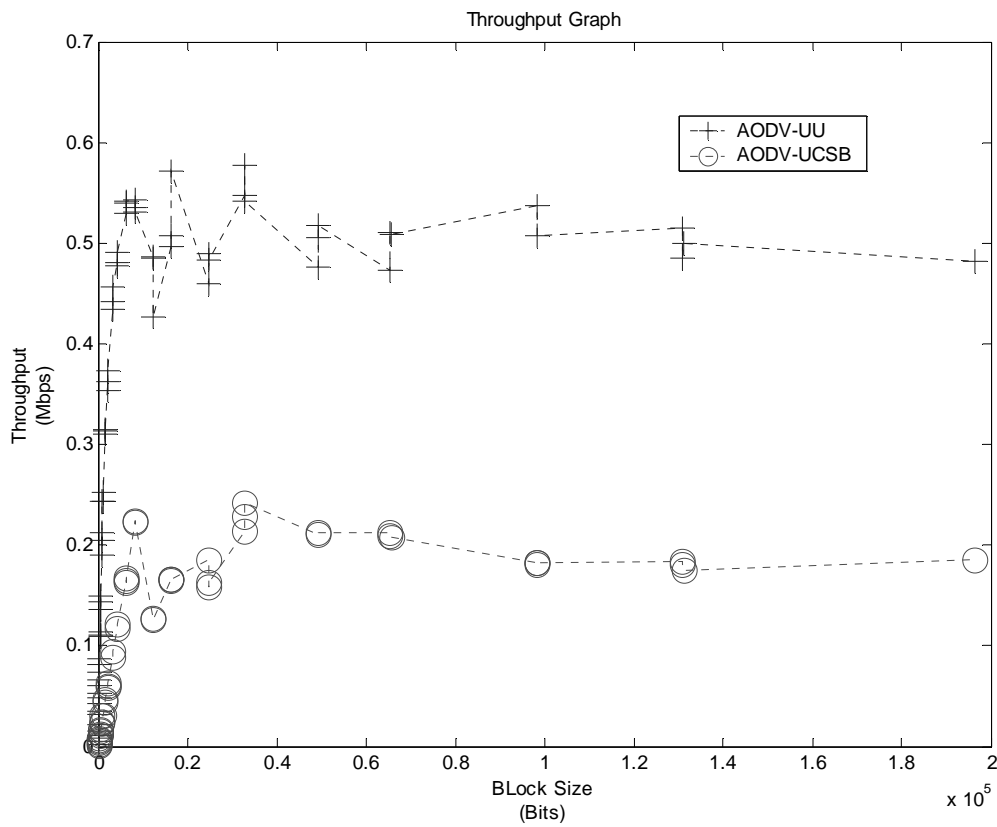
This thesis aims at observing the effect of unidirectional links on the performance of AODV, by comparing AODV-UCSB and AODV-UU software. It is necessary to extensively test the MANET routing protocols for them to be used in reality. The testing of AODV was limited to 3 nodes due to various administrative constraints.

#### **Results with Static scenario**

The Laptop 1, 2 and 3 were static and placed at position A, B and C respectively as shown in Figure 3.1. Laptop 3 transmits packets to Laptop 1 via Laptop 2.

### Throughput graph

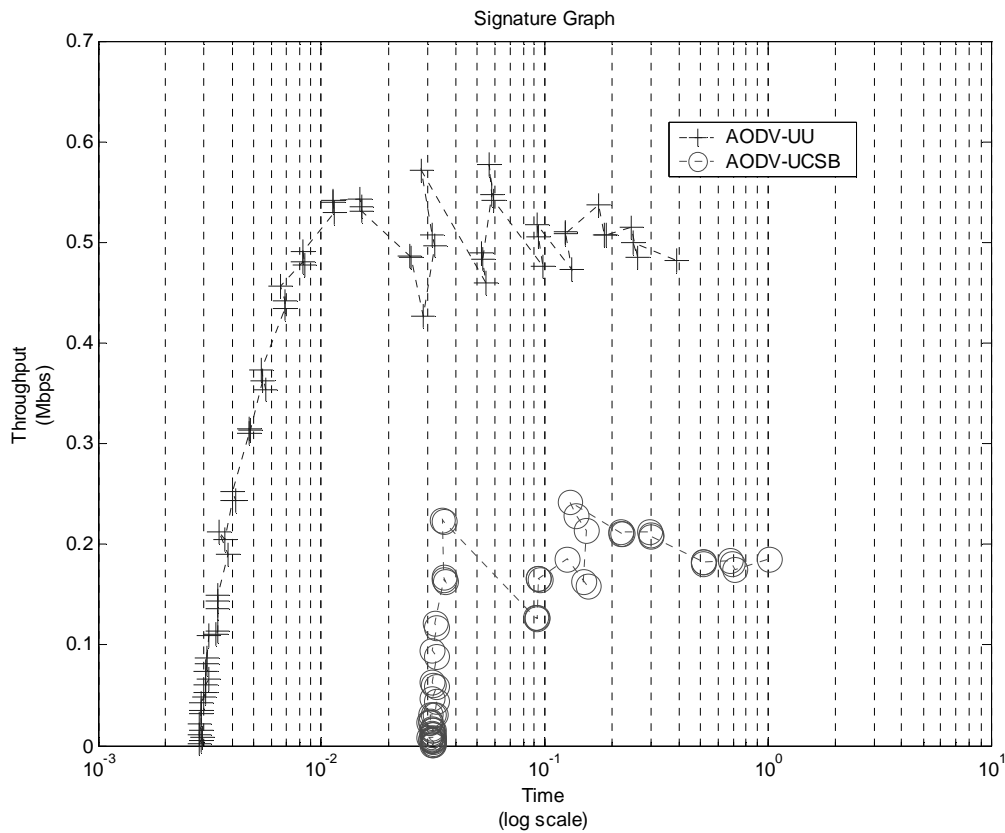
The throughput graph is a plot of throughput vs block size, used to observe the maximum obtainable throughput of the protocol. As seen from Figure 3.3, the throughput for AODV-UCSB and AODV-UU gradually increases and stabilizes after a certain value of the block size. AODV-UCSB stabilizes around 0.2 Mbps while AODV-UU stabilizes around 0.5 Mbps.



**Figure 3.3** Throughput graph comparing AODV-UU and AODV-UCSB.

### Network Signature Graph

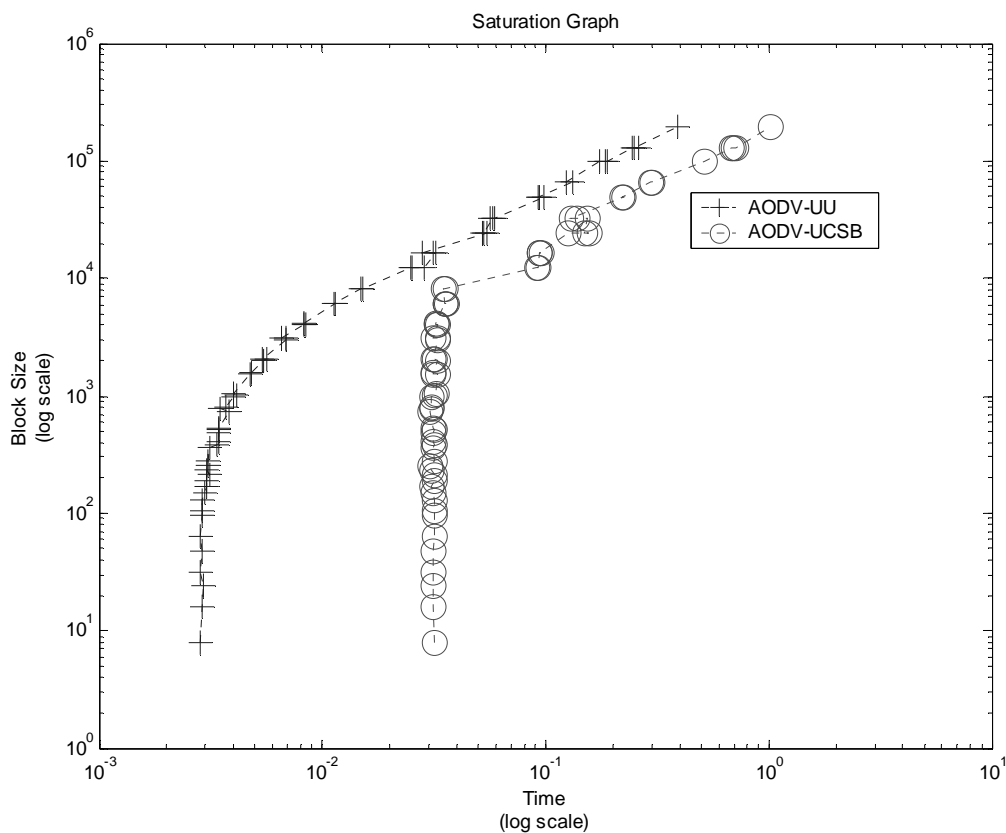
The network signature graph is a plot of throughput vs response time, used to observe the latency. The scale on the x-axis of the graph is logarithmic while the one on the y-axis is a normal scale. The scale on x-axis was changed to logarithmic so that the latency between AODV-UCSB and AODV-UU can be seen distinctly. As seen in the Figure 3.4 the latency associated with AODV-UCSB is greater than that of AODV-UU. The latency associated with AODV-UU was found to be 0.002938 seconds while that in AODV-UCSB was 0.03200 seconds.



**Figure 3.4** Network signature graph comparing AODV-UU and AODV-UCSB.

### Saturation Graph

The saturation graph is a plot of block size vs response time to transfer a block, used to find the suitable block size for the network. As seen in Figure 3.5; as the block size is increased, the response time is constant till certain value of block size, after which the response time increases linearly with the block size. This point is called saturation point. The interval between the saturation point and the end of the recorded data is referred to as the saturation interval. Essentially the saturation point indicates the suitable block size that should be used. In case of AODV-UCSB the suitable block size is around 2000 bytes, and for AODV-UU it is around 800 bytes.



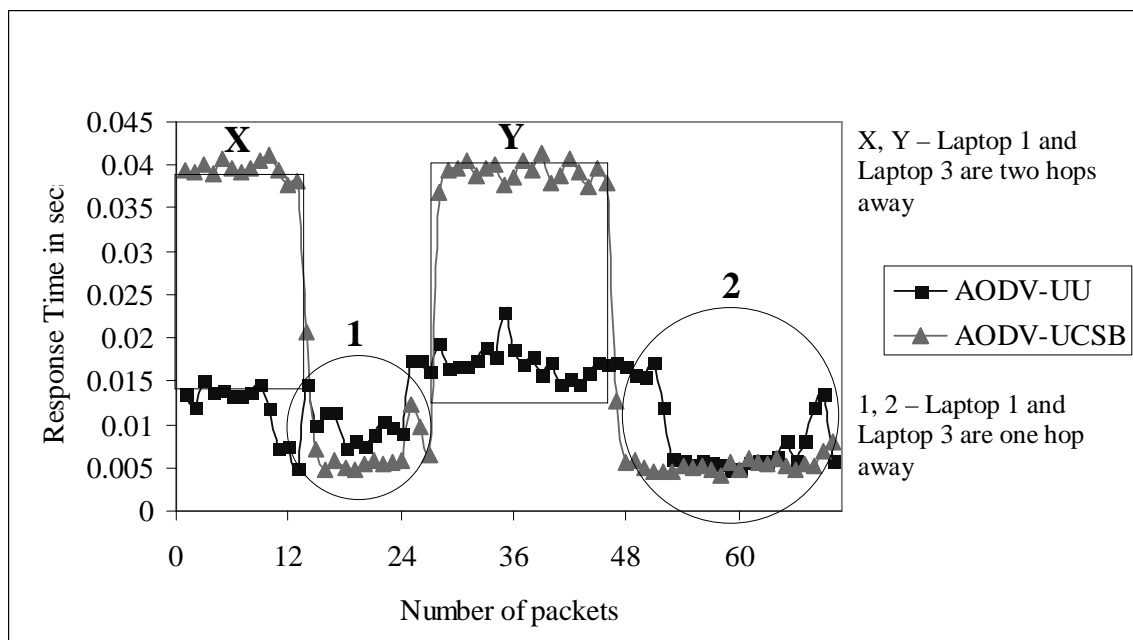
**Figure 3.5** Saturation graph comparing AODV-UU and AODV-UCSB.

## Results with Mobile scenario

Initially laptops were positioned as in the static scenario. After, Laptop 3 was made to traverse path 1 and brought to position B as shown in the Figure 3.2, and then Laptop 2 was made to traverse the path 2 to position C. Thus the link between Laptop 1 and Laptop 3 was broken since Laptop 2 was not in range of Laptop 1. Laptops 2 and 3 were moved back and forth between position B and position C, so as to invoke route discovery. Thus the route discovery was analyzed for both AODV-UU and AODV-UCSB.

### *Mobility Graph*

As seen in Figure 3.6, time to send the packet decreases drastically when the two Laptops are in direct transmission range of each other. Circle 1 and Circle 2 represent the period when the two laptops are in direct transmission range of each other.

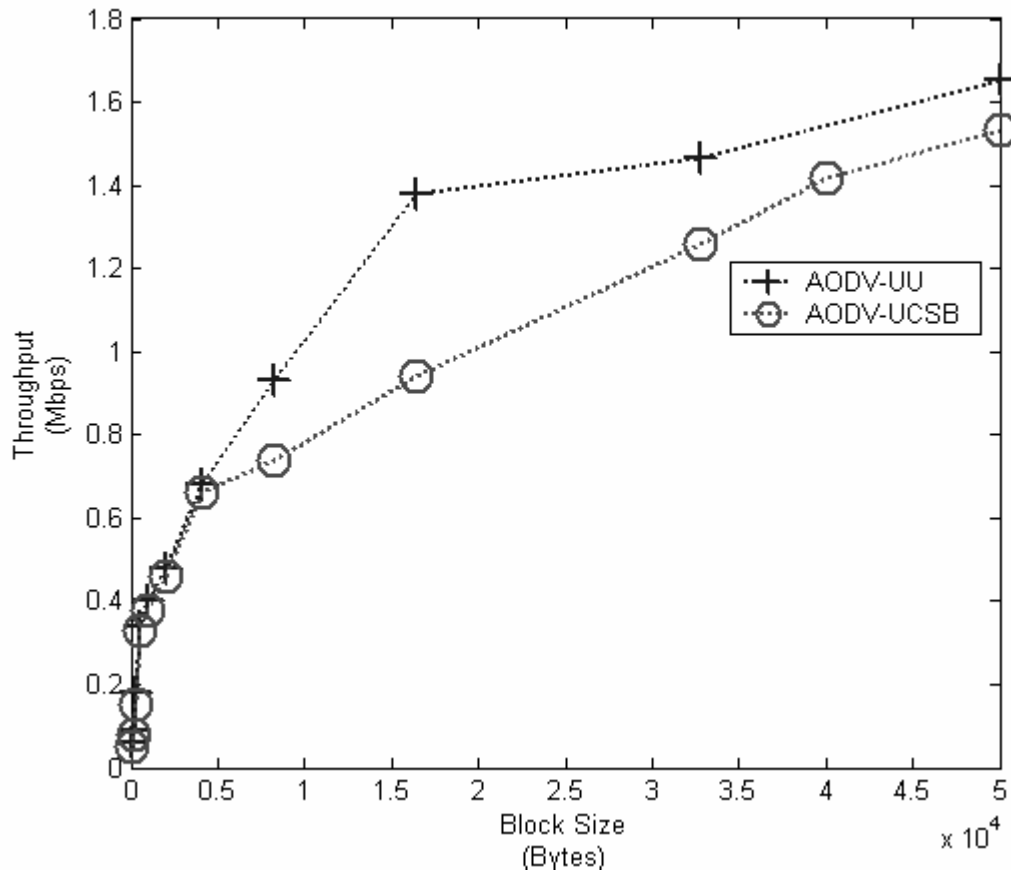


**Figure 3.6** Mobility graph comparing AODV-UU and AODV-UCSB.

The Dashed rectangles X and Y indicate the period for which the two laptops are one hop away from each other. After approximately 25 packets had been transmitted Laptop 3 and Laptop 2 were brought back to their original positions. Laptop 1 and Laptop 3 are one hop away. After approximately 50 packets had been transmitted Laptop 3 and Laptop 2 were made to traverse path 1 and path 2 respectively as shown in Figure 3.6 which shows that both AODV-UU and AODV-UCSB respond to link breakage quite promptly since there is a steep fall and rise when the laptops switch positions.

### **Discussion**

Figure 3.7 compares the throughput obtained for AODV-UU and AODV-UCSB for outside environment, i.e. in this scenario the unidirectional links are not formed as discussed in section 3.2. This shows that AODV-UU and AODV-UCSB both perform similarly when unidirectional links are not formed. But Figure 3.3 shows that AODV-UU achieves throughput almost double to that of AODV-UCSB, this illustrates that unidirectional links are formed in the indoor environment. Along with throughput the latency associated with AODV-UU is much better than that of AODV-UCSB as shown in Figure 3.4. But the route discovery time is not much affected, since both AODV-UU and AODV-UCSB respond to link breakage similarly as shown in Figure 3.7. So it can be concluded that unidirectional links formed due to the obstacles around the mobile nodes cannot be neglected and hence unidirectional link detection and avoidance improves the performance of the protocol.



**Figure 3.7** Throughput graph comparing AODV-UU and AODV-UCSB for outdoor environment.

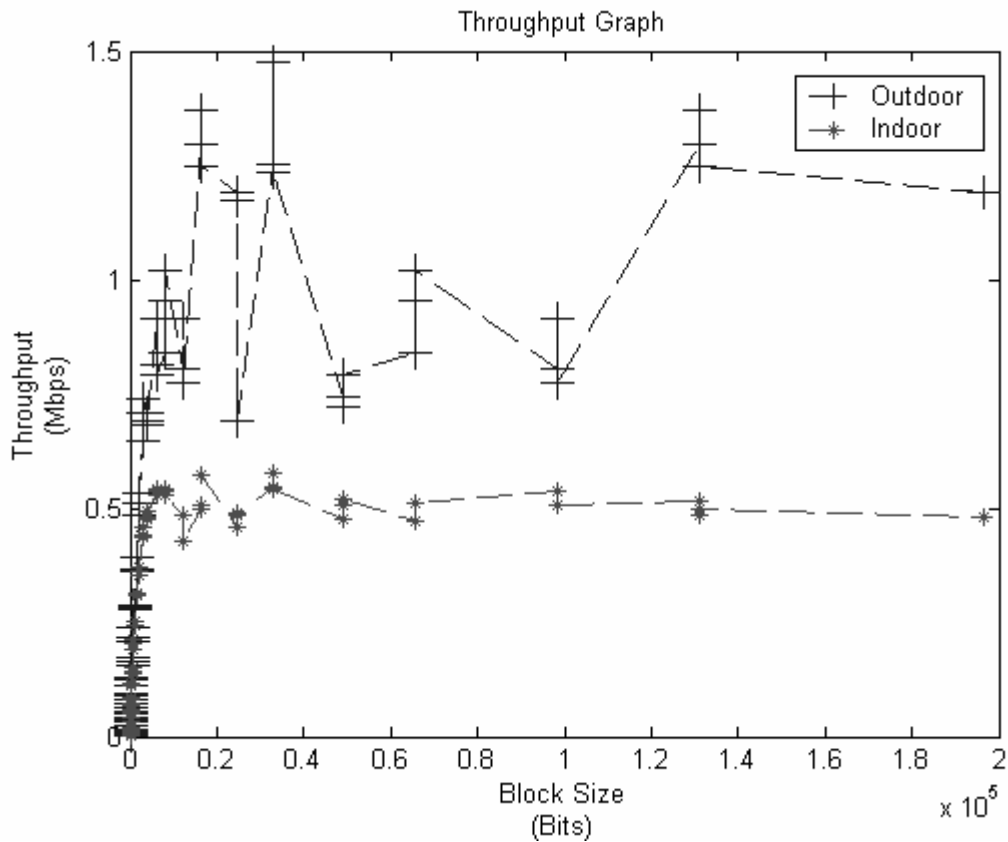
### 3.4.3 Comparison of Outdoor and Indoor Environment

Section 3.4.2 clearly presents that AODV-UU is better software than AODV-UCSB. So AODV-UU was used for other tests. This thesis aims at investigating the effects of the external environment on the performance of AODV protocol. So comparing the results obtained for both the environments, provides an opportunity to assess the performance degradation of AODV in indoor environment. The rest of the section presents the experimental results obtained using AODV-UU for indoor and outdoor environment.

Laptop 1, 2 and 3 were static and positioned at position A, B and C respectively as seen from Figure 3.2. The scenario setup is explained in detail in section 3.2.

### Throughput graph

As seen from Figure 3.8, the throughput for indoor as well as outdoor environment gradually increases and stabilizes after a certain value of the block size. The throughput for outdoor environment stabilizes at 1 Mbps while the indoor environment stabilizes around 0.5 Mbps. The indoor environment plot follows ideally, but the outdoor is not as ideal as it should be. The throughput in indoor environment is considerably degraded

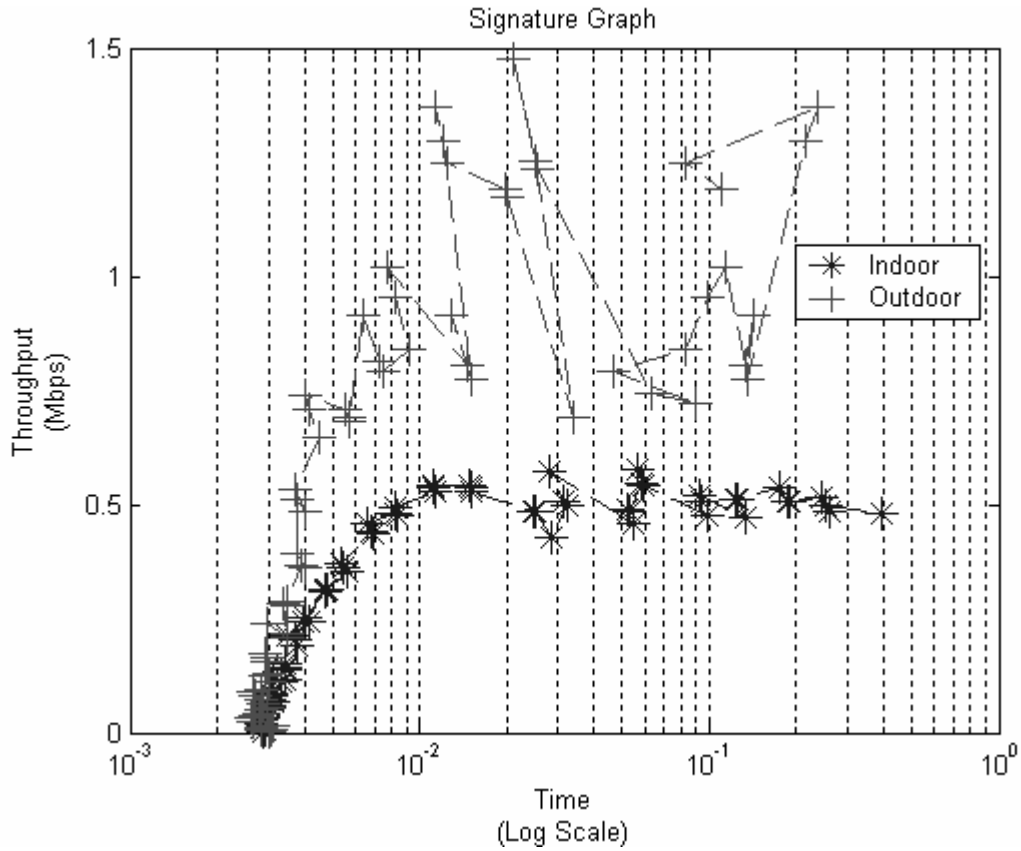


**Figure 3.8** Throughput graph of AODV-UU for outdoor and indoor environment.

when compared to the outdoor environment.

### Network Signature Graph

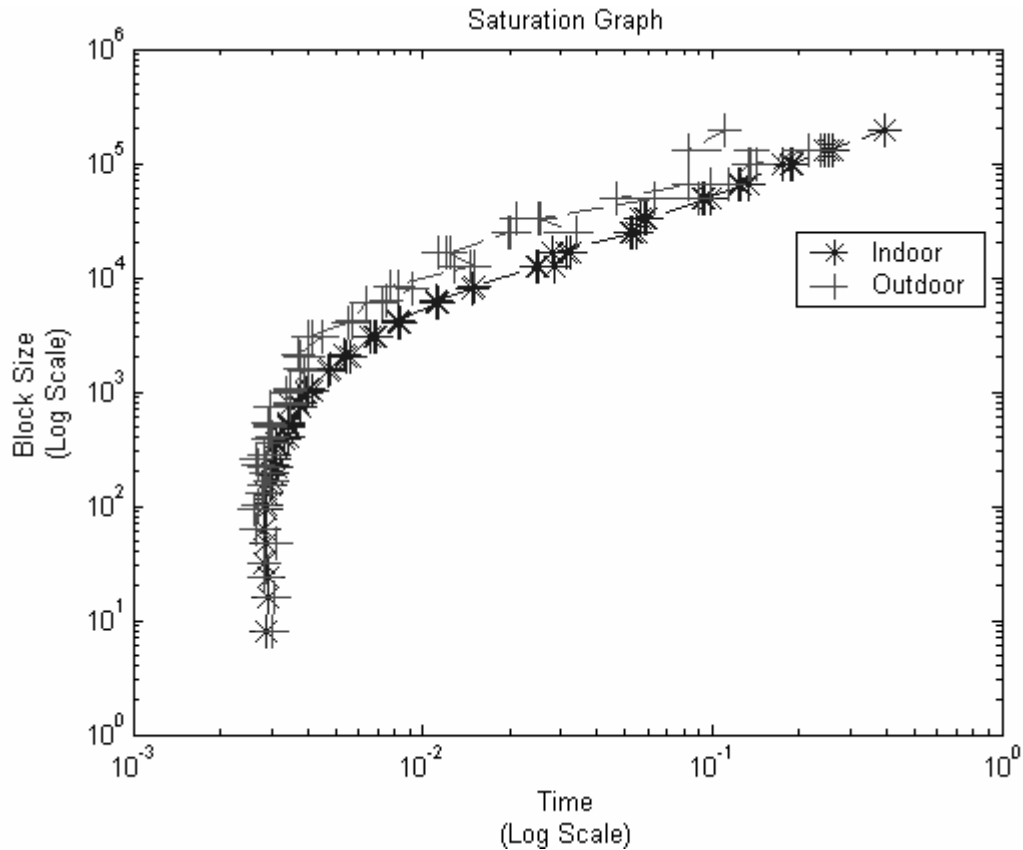
As seen in Figure 3.9, the latency associated with indoor is greater than that of outdoor environment, which is quite obvious. The mean latency associated with indoor environment and outdoor environment are 0.01252 and 0.006679 seconds respectively.



**Figure 3.9** Network signature graph of AODV-UU for outdoor and indoor environment.

### Saturation Graph

As seen in Figure 3.10 the saturation point for both the indoor environment and outdoor environment is almost the same around 800 bytes (block size). So for AODV-UU the ideal block size is between 750-800 bytes.



**Figure 3.10** Saturation graph of AODV-UU for outdoor and indoor environment.

### Discussion

The results clearly indicate that MANET performs poorly in indoor environment. As seen from Figure 3.8, the throughput for the outdoor environment is almost doubled for all the block sizes, but the throughput does not saturate ideally, this may be due to different factors such as birds, wind, etc. The mean latency as observed from Figure 3.9 for the indoor environment is almost 1.87 times that of the outdoor environment. The ideal block

size seems to be independent of the environment and for AODV-UU it is around 750-800 bytes.

## Chapter 4

# Simulation Methodology

This chapter discusses the performance evaluation of MANET using simulation. Section 4.1 describes briefly about the simulator with the mobility patterns and traffic patterns implemented, which affect the MANET. Sections 4.2 and 4.3 present and analyze the simulation results. More specifically, section 4.2 compares the experimental results and simulation results based on scenario without interference. Section 4.3 discusses the results obtained for the scenarios wherein interference is involved.

### **4.1 Simulation Environment**

To simulate the ad-hoc routing protocol AODV, *ns-2* [23] was used. The network simulator, *ns-2*, is a discrete event simulator developed by the University of California at Berkeley and the VINT project [24]. It provides substantial support for simulating variety

of protocols over conventional networks, and it also supports for simulating the physical aspects of multi-hop wireless communication and the wireless MAC protocol mostly based on IEEE 802.11 standard. The two-ray ground reflection model is used as a radio propagation model not only to consider a single line of-sight path between nodes but also the ground reflection. This model gives more accurate predictions of the received power at long distances than the free space model.

Simulations in ns-2 can be logged to *trace files*, which include detailed information about packets in the simulation and allow for post-run processing with some analysis tool. It is also possible to let ns-2 generate a special trace file that can be used by *NAM (Network Animator)*, a visualization tool that is part of the ns-2 distribution. NAM allows simulations to be replayed on screen. A large amount of documentation exists for ns-2, including a reference manual, several tutorials [25], and *ns-users* mailing list [26].

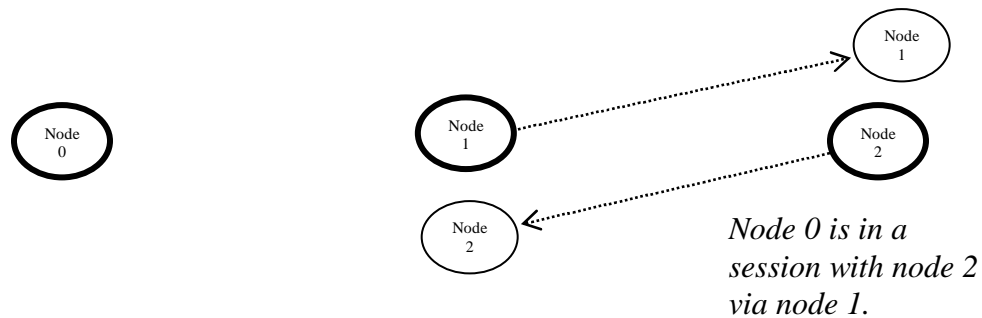
The first step in the simulation study is to assess the credibility of the simulator. The scenarios implemented in real world experiments were simulated using ns-2, and the results were compared. Later under various traffic constraints, the behavior of 5-node MANET for interference effects was observed.

The rest of this section explains about the scenario, communication model, movement model and the radio propagation model used in the simulations. All these models were chosen to make a reasonable comparison with the real world experiments.

## **Mobility Pattern**

The *Mobility Patterns* play a major role in analysis of a MANET. The mobility model usually used is the *Random Waypoint Model*. As per this model, a mobile node remains

stationary for a specified pause time, after which it begins to move with a randomly chosen speed towards a randomly chosen destination within the defined topology. But in the scenarios implemented in this thesis, the mobility patterns were defined such that the nodes move in a specific desired direction but not randomly. In order to assess the credibility of the simulator, a test bed as described in section 3.2 was simulated in ns-2 as shown in Figure 4.1.

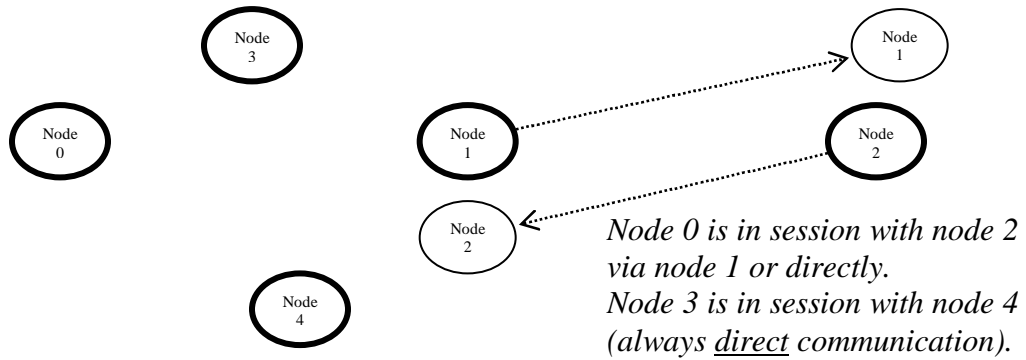


**Figure 4.1** Simple mobility scenario without interference.

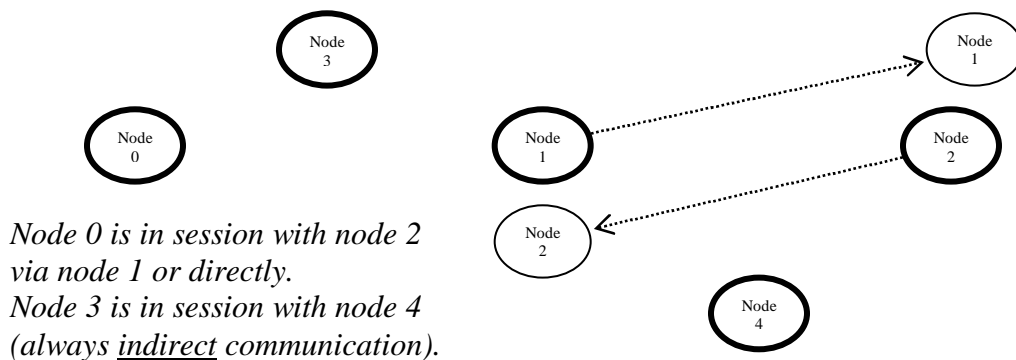
Each node begins the simulation by staying at the initial position for a predefined pause time. Node 2 then selects a target position around node 1's initial position in the simulated area and moves towards it with a speed of 2.5 meters/sec, after a certain time when node 2 is about to reach the vicinity of node 1, node 1 starts to move in the direction of node 2's initial position with a speed of 1.4 meters/sec. Both nodes repeat this mobility behavior during the simulated period of 300 seconds.

To analyze the effects of interference on AODV two scenarios as shown in Figure 4.2 and Figure 4.3 were developed. In Figure 4.2, there are two CBR traffic sessions, one uses AODV (node 0 and node 2) and another is a normal one-to-one session (node 3 and node 4). This scenario provides the basic performance of AODV under interference.

Another scenario, as shown in Figure 4.3 has two CBR sessions one between node 3 and node 4 and another between node 0 and node 2. Both the scenarios use AODV. In this scenario, the effects of interference with respect to AODV can be analyzed. The simulations for all the above scenarios were implemented where in both the sessions try to access the medium at same time; i.e when node 0 tries to access the medium at the same instance node 3 also tries to access the medium, causing a collision.



**Figure 4.2** Mobility scenario with interference (Traffic Pattern 1).



**Figure 4.3** Mobility scenario with interference (Traffic Pattern 2).

The communication model is determined by four factors: number of sources, packet size, packet rate and the communication type. This study uses the CBR communication type, which uses UDP as its transport protocol. Throughout the experiment the packet size is varied. Other simulation parameters are shown in Table 4.1.

Table 4.1. General simulation parameters.

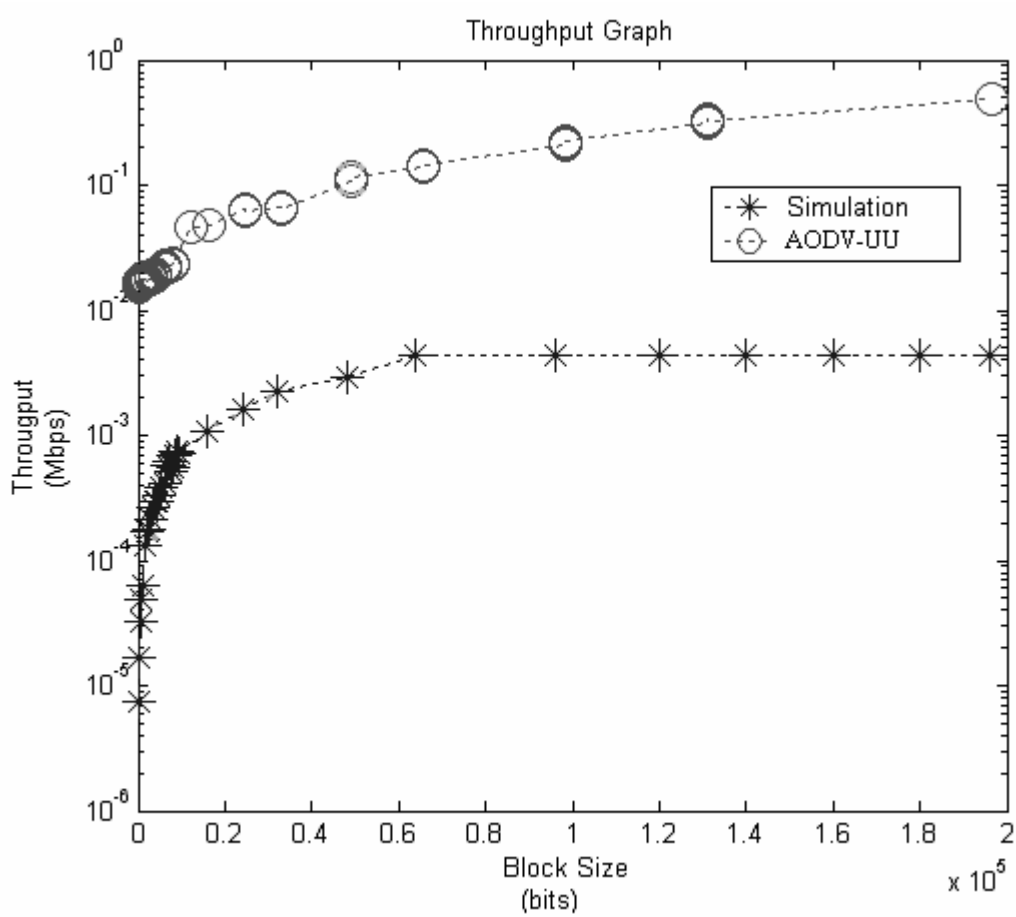
	<b>Parameters</b>	<b>Values</b>
<b>Radio Characteristics</b>	Transmission range	250 meters
	Wireless bandwidth	1 Mbits/sec
<b>Communication Model</b>	Traffic type	Constant bit rate
	Packet size	Varying
<b>Mobility Pattern</b>	Node speed	0 m/s – 5 m/s (walking speed)
<b>Simulation Parameters</b>	Simulation time	300 seconds
	Number of nodes and network area	5 nodes in an area 1000 x 1000m
<b>Routing and MAC Protocols</b>	Routing protocol	AODV
	MAC protocol	IEEE 802.11

The results obtained from all the above scenarios were used to plot the *throughput graph*, *network signature graph* and the *saturation graph*. A detailed analysis of these results is discussed in the following sections.

## 4.2 Simulation Results without Interference

The main aim of this thesis is to extensively test small ad-hoc networks in real world environment. But it is very difficult to conduct the experiments for all possible cases, so simulation was chosen to analyze the performance of MANET without interference in

this section based on the simple mobility scenario in Figure 4.1 and with the effects of



**Figure 4.4** Throughput graph to assess the credibility of the simulator.

interference in the next section with the scenarios wherein interference is involved as shown in Figure 4.2 and 4.3.

Care was taken to re-simulate the real world experimental scenarios, so as to make a credible comparison. This comparison aims at assessing if the simulator follows the same pattern of results as obtained from experiments. Throughput graph is used to plot throughput versus block size. In Figure 4.4 the throughput was taken in log scale. For simulation and real world experiments the throughput increases steeply with the block

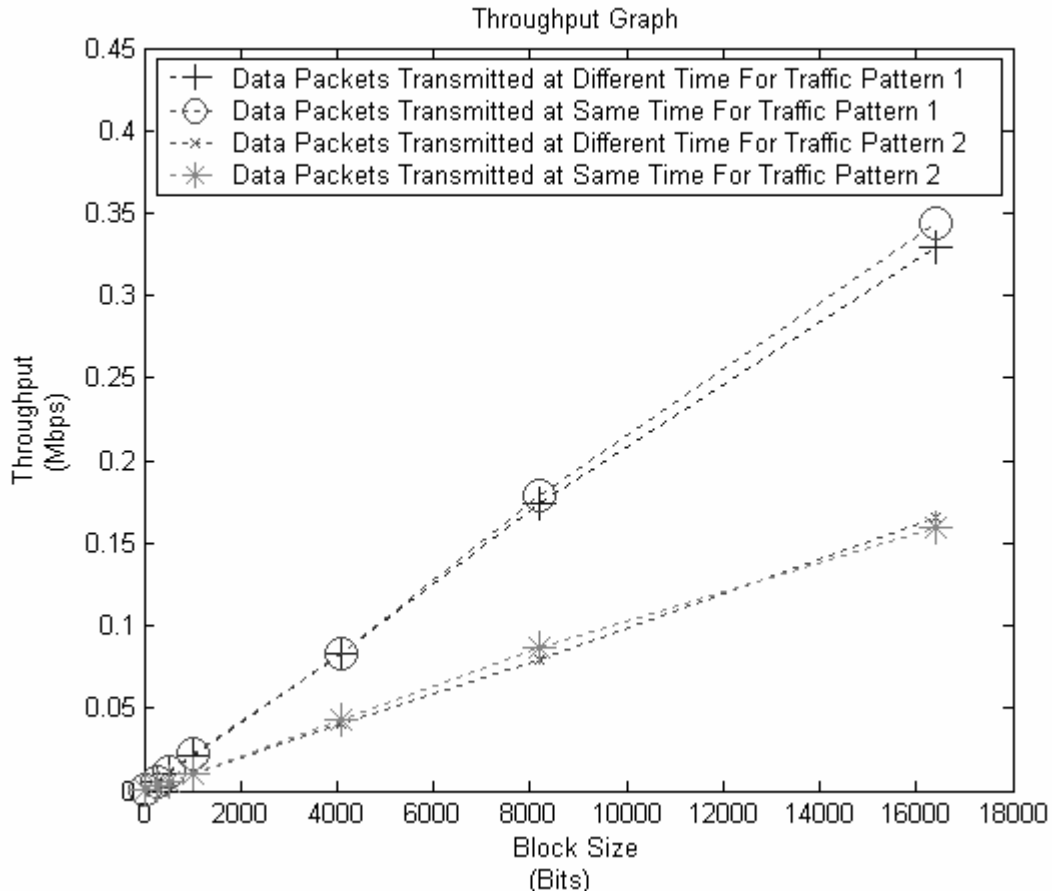
size and saturates after certain block size. But the minimum and maximum throughput levels are different for simulation and experiments. Both the simulation and experiment saturate for almost the same block size.

For MANETs to be commercially successful there is a heavy need to conduct real world tests more frequently. But a simulator can be used to make educated guesses about the performance of the MANETs in many different scenarios. This would save the researchers time and money. As observed from the results obtained, there is a strong correlation between the parametric values obtained the simulations and the real world.

### **4.3 Simulation Results with Interference**

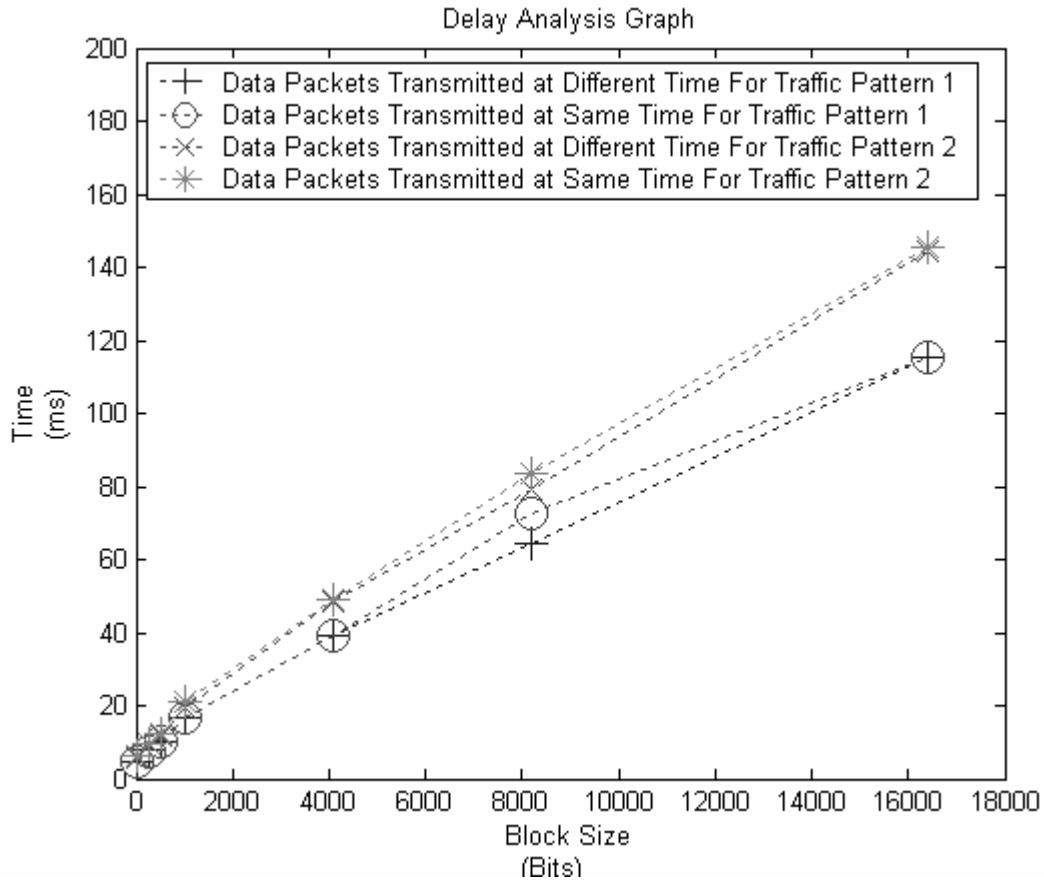
The scenarios depicted in Figure 4.2 and Figure 4.3, are used to analyze the effects of interference. Figure 4.2 is depicted as Traffic Pattern 1 and Figure 4.3 is depicted as Traffic Pattern 2, for further analysis. Traffic Pattern 1 uses two CBR traffic sources, one uses AODV for communication while the other one uses direct link. Traffic Pattern 2 uses two CBR traffic sources, both of which use AODV for communication. In both the Traffic Patterns each CBR packet in different session are sent at the same time, i.e. in Figure 4.2 and Figure 4.3 node 0 sends packet at the same time as node 3. Another simulation was also implemented where in node 0 sends packet at different time when compared to node 3. This is implemented, to investigate if the performance of AODV degrades, when two nodes try to access the medium at the same time, because such a scenario induces collisions.

As seen from Figure 4.5 the throughput does not change much for the interference in both the Traffic Patterns. But the throughput of Traffic Pattern 1 is almost double that of Traffic Pattern 2.



**Figure 4.5** Throughput graph to analyze effect of interference and collisions.

The response time is almost the same for both cases, i.e. for the CBR traffic sent at the same time and different time. But the response time recorded for the Traffic Pattern 1 is about 0 to 30 ms less than that for the Traffic Pattern 2 as shown Figure 4.6. For small packet sizes the response time is almost same for both the Traffic Patterns, but as the packet size increases the response time for Traffic Pattern 2 is more.



**Figure 4.6** Delay Analysis graph to analyze effect of interference and collisions.

The results presented show that as the links using the AODV increase the throughput and response time decreases proportionally. The results also show that even if the packets are sent at the same time or at a different time by two different nodes, it does not really degrade the performance of AODV.

## Chapter 5

# Conclusions

This thesis aims at evaluating MANET in real world environment. It is important to test MANET in real world environment for its commercial use. The two implementations of a MANET routing algorithm by Uppsala University (UU) and University of California, Santa Barbara (UCSB) were compared. As seen from Chapter 3, this comparison tells that different implementation gives a substantially different performance even though the trend is the same. In these particular implementations, we consider it is due to the affect of unidirectional links in a MANET. The next step in this thesis was to compare the performance of MANET in an indoor (building) /outdoor (parking lot) environment; so as to assess effect of the real world obstacles on the MANET protocol.

This thesis presents a unique approach to analyze the performance of a MANET. As observed from the results, the throughput graph, network signature graph and saturation graph assist in choosing a suitable block size for MANETs. The route

discovery time for MANETs is also analyzed. We also conducted a simulation study to assess the credibility of the simulator. The experimental scenarios were simulated also in ns-2 simulation environment. Later the effect of interference and collisions on MANET was investigated. For this purpose, two scenarios with interfering communication traffic were developed and tested.

According to the experimental results, as shown in Chapter 3, the MANET performance degrades considerably for indoor environments. The throughput for indoor as well as outdoor environment gradually increases and stabilizes after a certain value of the block size. The throughput for the outdoor environment is doubled almost for all the block sizes, but the throughput does not saturate ideally, this may be due to different factors such as trees, birds, winds, etc. The mean latency for the indoor environment is almost 1.87 times that of the outdoor environment. The ideal block size seems to be independent of the external factor and is entirely dependent on the implementation. So it can be concluded that the performance of MANET degrades considerably in indoor environment. Probably one solution to improve the performance is to install a static node running a MANET routing protocol in critical areas where the performance degrades.

According to the results, as shown in Chapter 4, there is a strong correlation between the parametric values obtained for simulation and experiments. For simulation and real world experiments the throughput increases with the block size and saturates after certain block size. Both the simulation and experiment saturate for almost the same block size. But the minimum and maximum throughput levels are different for simulation and experiments. Since the main bases of the simulator's development are mathematical equations, the simulation results may not be the same as of the experimental results. But

we can definitely use the simulator to make educated guesses about the performance of the MANETs in many different scenarios, which would save the researchers time and money. So it can be concluded that ns-2 can be used intuitively to analyze MANETs performance.

## **Future work**

MANETs prove ideal in military, emergency and rescue operation where there is lack of readily available communication infrastructure. Development of such applications demands rigorous performance evaluation procedures including extensive change in environments and population of mobile nodes.

The current work is based on experimental procedures involving only 3 mobile nodes. Future studies could involve more number of nodes (30-50) and various scenarios to test the performance of MANETs, thus taking it closer to the real world.

# References

1. IEEE Computer Society LAN MAN Standards Committee. “Wireless LAN Medium Access Control (MAC) and Physical (PHY) Specifications”, 1999.
2. J. Haartsen. “Bluetooth, the universal radio interface for Ad-hoc wireless connectivity”, *Ericsson Review*, 1998.
3. Kamerman. “Coexistence between Bluetooth and IEEE 802.11 CCK Solutions to avoid Mutual Interference”, in IEEE P802.11 Working Group Contribution, *IEEE 802.11-00/162*, July 2000.
4. MANET, <http://www.ietf.org/html.charters/manet-charter.html>
5. C. K. Toh. “Ad Hoc Mobile Wireless Networks: Protocols and Systems”, *Prentice Hall PTR*, 2002.
6. General Packet Radio Service, <http://www.mobilegprs.com>
7. C. E. Perkins, E. M. Belding Royer, and S. Das. “Ad Hoc On Demand Distance Vector (AODV) Routing,” *Internet Engineering Task Force draft*, March 2002.
8. C. E. Perkins and P. Bhagwat. “Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers”, *SIGCOMM '94 Conference on Communications Architectures, Protocols and Applications*, August 1994.
9. D. B. Johnson, D. A. Maltz and Y. C. Hu, “The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks”, *Internet Engineering Task Force draft*, April 2003.
10. J. Jubin and J. D. Tornow. “The DARPA packet radio network protocols”, *IEEE (Special Issue on Packet Radio Networks)* about???, January 1987.

11. L. Buttyan and J. Hubaux. "Report on a Working Session on Security in Wireless Ad Hoc Networks", *ACM Mobile Computing and Communications Review*, October 2002.
12. "Routing protocol performance issues and evaluation considerations", *Internet Engineering Task Force Network Working Group*, January 1999,  
<http://www.ietf.org/rfc/rfc2501.txt>
13. V. Park and S. Corson. "Temporally-Ordered Routing Algorithm (TORA) Version 1 Functional Specification", *Internet Engineering Task Force draft*, July 2001, <http://www.ietf.org/internet-drafts/draft-ietf-manet-tora-spec-04.txt>
14. Mad-hoc: Implementation of AODV, <http://mad-hoc.flyinglinux.net/>
15. Implementation of AODV, University of Illinois at Urbana Champaign,  
<http://aslib.sourceforge.net>
16. Implementation of AODV, Uppsala University, <http://user.it.uu.se/~henrikl/aodv/>
17. Implementation of AODV, University of California at Santa Barbara,  
<http://moment.cs.ucsb.edu/AODV/aodv.html>
18. Implementation of AODV, kernel-AODV,  
[http://w3.antd.nist.gov/wctg/aodv\\_kernel/](http://w3.antd.nist.gov/wctg/aodv_kernel/)
19. Implementations of AODV  
<http://moment.cs.ucsb.edu/AODV/aodv.html#Implementations>
20. Netfilter, <http://www.netfilter.org>
21. V. Kawadia, Y. Zhang and B. Gupta. "System Services for Implementing Ad-hoc Routing Protocols", *International Conference on Parallel Processing Workshops*, April 2002.

22. NetPIPE, <http://www.scl.ameslab.gov/Projects/NetPIPE/>
23. Network Simulator - ns-2, <http://www.isi.edu/nsnam/ns/>
24. The VINT (Virtual InterNetwork Testbed),  
<http://www.isi.edu/nsnam/vint/index.html>
25. Marc Greis and The VINT group. "Tutorial for the Network Simulator ns,"  
December 2000, <http://www.isi.edu/nsnam/ns/tutorial/index.html>
26. Ns-users mailing list, <http://mailman.isi.edu/mailman/listinfo/ns-users>