Analysis of Security and Reliability of Routing Protocols in MANETs

By

Syed Muhammad Raza Gillani

Table of Contents

List of Figures

List of Tables

Abstract

The utmost demand of the future networks is the rapid deployment of independent mobile nodes that can communicate with each other without the need of centralized and organized network infrastructure. This type of network is categorized under the classification of Mobile Ad-Hoc Networks (MANETs). As the nodes in a MANET are mobile, the network topology may change rapidly and unpredictably. While talking about any network type, wireless or wired, the most important issue that need to be resolved is the security and reliability that the network provides. In this thesis, first of all it has been given a detailed introduction about Mobile Ad-hoc networks (MANETs). Then explained the two table-driven and two on-demand routing protocols. Following that it has been analyzed reliability of these four routing protocols by simulating them in ns-2 based on the metrics defined. The On-demand protocols, AODV and DSR perform better than the table-driven DSDV and OLSR protocol. Moreover it has been performed analysis of security issues in Mobile Ad-hoc Networks. The analysis has revealed that DSR is more reliable; therefore, finally a trust-based model for DSR has been proposed to make it more secure and reliable in Mobile-Ad-hoc Networks

1 Introduction to Mobile Ad-Hoc Networks

1.1 Introduction

With recent performance advancements in computer and wireless communications technologies, advanced mobile wireless computing is expected to see increasingly widespread use and application, much of which will involve the use of the Internet Protocol (IP) suite. The vision of mobile ad hoc networking is to support robust and efficient operation in mobile wireless networks by incorporating routing functionality into mobile nodes. Such networks are envisioned to have dynamic, sometimes rapidly-changing, random, multihop topologies which are likely composed of relatively bandwidth-constrained wireless links.

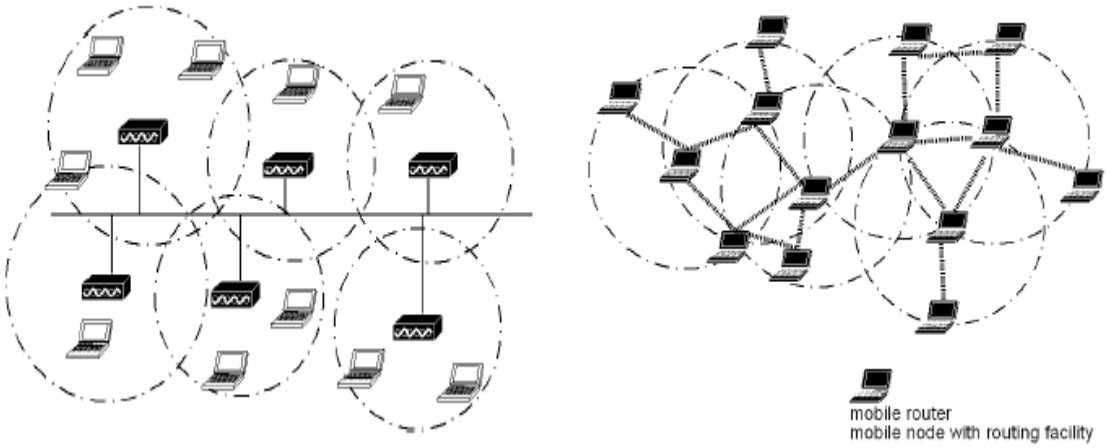
Wireless networking is an emerging technology that allows users to access information and services electronically, regardless of their geographic position. There are currently two variations of mobile wireless networks. The first is known as infrastructured networks, i.e., those networks with fixed and wired gateways. The bridges for these networks are known as base stations. A mobile unit within these networks connects to, and communicates with, the nearest base station which is within its communication radius. As the mobile travels out of range of one base station and into the range of another, a "handoff" occurs from the old base station to the new, and the mobile is able to continue communication seamlessly throughout the network.

The second type of mobile wireless network is the infrastructureless mobile network, commonly known as an ad-hoc network. Infrastructureless networks have no fixed routers; all nodes are capable of movement and can be connected dynamically in an arbitrary manner. Nodes of these networks function as routers which discover and maintain routes to other nodes in the network terrains.

1.2 Mobile Ad-Hoc Networks

A mobile ad-hoc network (MANET) is a collection of nodes, which have the possibility to connect on a wireless medium and form an arbitrary and dynamic network with wireless links. That means that links between the nodes can change during time, new nodes can join the network, and other nodes can leave it. A MANET is expected to be of larger size than the radio range of the wireless antennas, because of this fact it could be

necessary to route the traffic through a multi-hop path to give two nodes the ability to communicate. There are neither fixed routers nor fixed locations for the routers as in cellular networks also known as infrastructure networks (Figure1.1 (a)). Cellular networks consist of a wired backbone which connects the base-stations. The mobile nodes can only communicate over a one-hop wireless link to the base-station; multi-hop wireless links are not possible. By contrast, a MANET has no permanent infrastructure at all. All mobile nodes act as mobile routers. A MANET is depicted in Figure 1.1 (b)



(a) Infrastructure Network (b) Mobile Adhoc Network

Figure 1.1 Wireless Network Structures [1]

A MANET is highly dynamic. Links and participants are often changing and the quality of the links as well. Furthermore, asymmetric links are possible as you can see an example in Figure 1.2. Node A is in transmission range of node B, on the other hand node B is not in range of node A. There exist other reasons for asymmetric links such as a higher signal-to-noise ratio for node A than for node B.

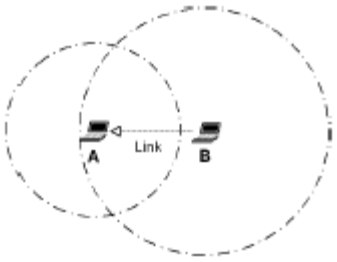


Figure 1.2 Asymmetric Link [1]

1.3 Advantages and Application Areas

Mobile ad-hoc networks have certain advantages over the traditional communication networks. Some of these advantages are:

• Use of ad-hoc networks can increase mobility and flexibility, as ad-hoc networks can be brought up and torn down in a very short time.

• Ad-hoc networks can be more economical in some cases, as they eliminate fixed infrastructure costs and reduce power consumption at mobile nodes.

• Ad-hoc networks can be more robust than conventional wireless networks because of their (non-hierarchical distributed control and management mechanisms).

• Because of multi-hop support in ad-hoc networks, communication beyond the Line of Sight (LOS) is possible (at high frequencies).

• Multi-hop ad-hoc networks can reduce the power consumption of wireless devices. More transmission power is required for sending a signal over any distance in one long hop than in multiple shorter hops.

• Because of short communication links (multi-hop node-to-node communication instead of long-distance node to central base station communication), radio emission levels can be kept low. This reduces interference levels, increases spectrum reuse efficiency, and makes it possible to use unlicensed unregulated frequency bands.

Some applications of MANET technology could include industrial and commercial applications involving cooperative mobile data exchange. In addition, mesh-based mobile networks can be operated as robust, inexpensive alternatives or enhancements to

cell-based mobile network infrastructures. There are also existing and future military networking requirements for robust, IP-compliant data services within mobile wireless communication networks many of these networks consist of highly-dynamic autonomous topology segments. Also, the developing technologies of "wearable" computing and communications may provide applications for MANET technology. When properly combined with satellite-based information delivery, MANET technology can provide an extremely flexible method for establishing communications for fire/safety/rescue operations or other scenarios requiring rapidly-deployable communications with survivable, efficient dynamic networking.

1.4 Challenges of MANETs

Some of the major challenges of the MANET can be summed up as follows:

1. It has a very limited wireless transmission range.
2. Packet loss is frequent due to transmission errors.
3. Route changes frequently due to the mobile nature of the network.
4. Battery constraint is another major concern.
5. The network keeps on getting partitioned frequently.
6. Ease of snooping on wireless transmissions. Problem of security

2 Routing in Mobile Ad-Hoc Network

2.1 Routing Protocols in MANETs

It is the task of routing protocol to create, maintain and recreate routes which preferably should be durable. To run smoothly an Ad Hoc network requires quick and adaptive routing protocol that at the same time does not consume too much of the already scarce wireless bandwidth. Existing routing protocols used in the internet do not function well in such an environment. More specifically, the Routing Information Protocol (RIP) suffers from more convergence and the well known count-to-infinity problem. The Open Shortest Path First (OPSF) convergences only slightly faster than RIP and addition to it has high bandwidth consumption. Over the years several candidates protocols have been proposed both inside and outside the IETF MANET WG and more than a handful of prominent candidates are still active. These protocols must now face the next step towards standardization, namely real-world and interoperability.

2.2 Classification of Routing in MANET

Routing is a difficult problem in a MANET. A lot of solutions have been proposed trying to address a sub-space of the problem domain. Because of complexity and diversity, Internet Engineering Task Force (IETF) has not determined a standard of routing.

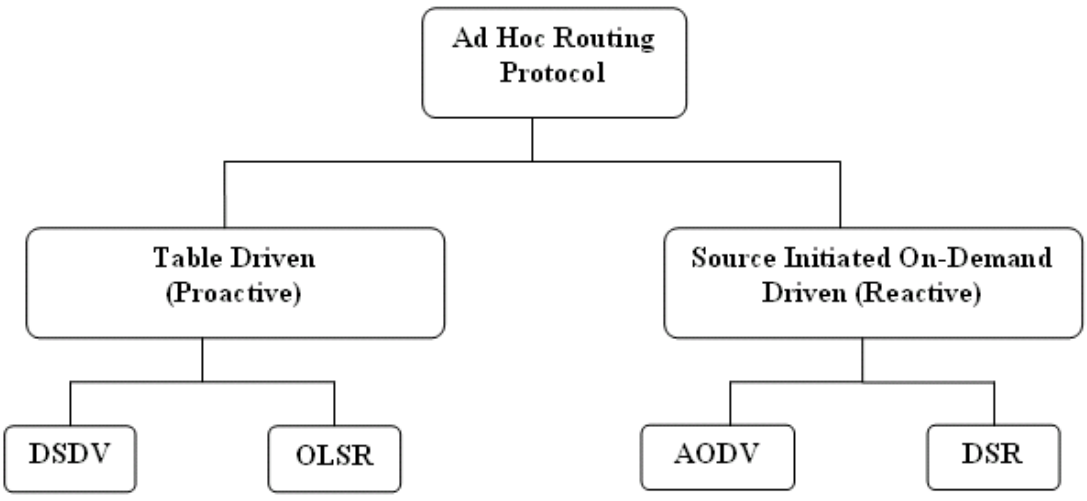


Figure 2.1 Classification of Routing Schemes in MANETs

Figure 2.1 shows the classification of Routing in MANETs. It is clear from the diagram that we can classify the MANET routing protocols into two major categories [1].

1. Proactive routing protocol

2. Reactive routing protocol.

2.2.1 Proactive Routing Protocols

This is also called table-driven routing. It tries to maintain up-to-date information about all the nodes. Periodic route-update messages propagate to all the nodes. The table-driven routing protocols attempt to maintain consistent, up-to-date routing information from each node to every other node in the network. These protocols require each node to maintain one or more tables to store routing information, and they respond to changes in network topology by propagating updates throughout the network in order to maintain a consistent network view. Each node uses the exchanged route information to calculate the costs to reach all possible destinations. Thus if a destination can be reached, a route is always at hand; which avoids the delay associated with finding a route on demand. The areas where they differ are the number of necessary routing related tables and the methods by which changes in network structure are broadcast.

Proactive routing techniques can be divided into either distance vector or link state algorithms. Both techniques require each router to periodically broadcast route information and to calculate the shortest path to others. In distance vector, each router maintains a vector with the distance to all routers and periodically broadcast this vector to each of its neighbor routers. Each router updates its own routing table by calculating the shortest path to each router using the distance vectors received from others.

In link-state, the relationship between whom to collect state about and whom to disseminate this information to be reversed compared to distance vector; each router maintains the state of its neighbors only and broadcast the information to all other routers. Using the link-state information from all others routers, each router computes a complete picture of the network and calculates the shortest path to all nodes. For better performance in Ad-Hoc networks both distance vector and link-state algorithms have been modified. Examples of distance vector protocols are Destination Sequenced

Distance Vector (DSDV), Routing Information Protocol (RIP), and Wireless Routing Protocol (WRP). Examples of link-state protocols are Open Shortest Path First (OPSF), and Optimized Link State Routing (OLSR). But in our project we have studied two proactive routing protocols OLSR and DSDV.

Advantage: route to a destination is always available; there is no initial delay when a route is needed.

Disadvantage: high overhead; slow to converge.

2.2.1.1 Destination Sequenced Distance Vector (DSDV)

The Destination-Sequenced Distance-Vector Routing protocol (DSDV) is a table- driven algorithm based on the classical Bellman-Ford routing mechanism. The improvements made to the Bellman-Ford algorithm include freedom from loops in routing tables. Every mobile node in the network maintains a routing table in which all of the possible destinations within the network and the number of hops to each destination are recorded. Each entry is marked with a sequence number assigned by the destination node. The sequence numbers enable the mobile nodes to distinguish stale routes from new ones, thereby avoiding the formation of routing loops. Routing table updates are periodically transmitted throughout the network in order to maintain table consistency. To help alleviate the potentially large amount of network traffic that such updates can generate, route updates can employ two possible types of packets. The first is known as a "full dump". This type of packet carries all available routing information and can require multiple network protocol data units (NPDUs). During periods of occasional movement, these packets are transmitted infrequently. Smaller "incremental" packets are used to relay only that information which has changed since the last full dump. Each of these broadcasts should fit into a standard size NPDU, thereby decreasing the amount of traffic generated.

The main advantage to DSDV is that it maintains a loop-free fewest-hop path to every destination in the network. However, this protocol also contains both periodic and triggered route updates. While the triggered updates tend to be small (allowing quick discovery of invalid links), the each node's periodic update includes its entire routing table. This means the overhead associated with those updates grows as O(n2), effectively limiting the number of nodes in the network.

Mode of Operation

DSDV operates by having each node maintain a table with information about distances and information about the next node on a route. The protocol can be explained by looking at a small topology, such as the one illustrated in Figure 2-2.

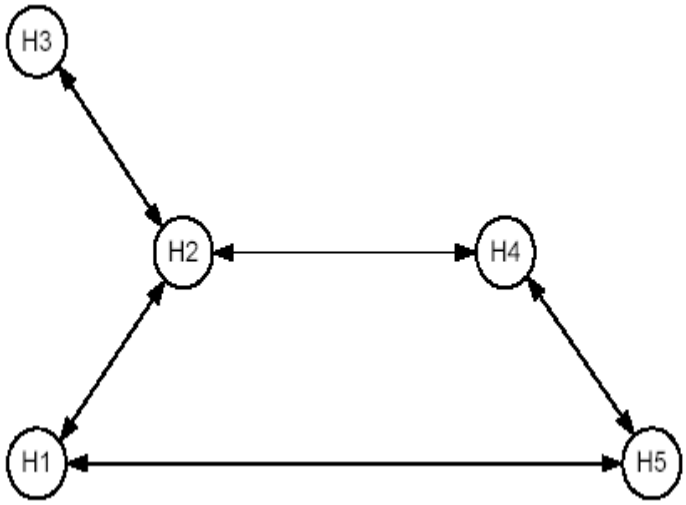


Figure 2.2 A Simple Topology [1]

Table 2-1 illustrates the nodes only stores information about destination and next hop, and not about the entire route. As seen, the route from H4 to H3 goes through H2, which means that the metric is 2 (hops). The next node on the route from H4 to H3 is H2, and H4 will therefore forward packets for H3 to H2. Information concerning the next hop is stored in the Next Hop column.

The sequence numbers in the Seq. No column is used to compare routes. Routes with higher sequence numbers are considered more favorable. If the sequence number is the same the route with the lowest metric is preferred. The value in the Install column is used to help determine when stale routes should be deleted. Each node in the network must periodically transmit its entire routing table to its neighbors. Missing transmissions can be used by neighbor nodes to detect changes (broken links) in the topology. Broken links may also be detected by communication hardware. When a broken link is detected it is assigned a metric value of infinity and the node that detected the broken link broadcasts an update packet, to inform others that the link is broken.

Table 2.1 Routing Table for H4 Node in the DSDV Protocol

Dest	Nexthop	Metric	Seq.No	Install
H1	H2	2	S406 H1	T001H5
H2	H2	1	S128 H2	T001H5
H3	H2	2	S444 H3	T001H5
H4	H4	1	S123 H4	T001H5
H5	H5	1	S489 H5	T001H5

Settling Time

Mobiles also keep track of the settling time of routes, or the weighted average time that routes to a destination will actuate before the route with the best metric is received. By delaying the broadcast of a routing update by the length of the settling time, mobiles can reduce network traffic and optimize routes by eliminating those broadcasts that would occur if a better route was discovered in the very near future. In order to reduce the network overhead generated due to transmission of update messages every time there is a change in the network topology, a settling time is estimated for each route. A settling time is the amount of time it takes a node to get all the update messages for a route. Therefore a node sends out an update message to its neighbors with a new route only if the settling time of the route has expired and the route still remains optimal. Due to the lack of synchronization between nodes in the network, a time delay is imposed to prevent nodes from responding immediately based on a single potentially disruptive update. This settling time allows for the routing table at each node to stabilize before it beings issuing route updates to other nodes.

Resource Usage

Because each node is required to keep track of the full network topology, each node has to maintain a routing table for all routes. This requires additional memory resources compared to some of the other routing protocols we will be discussing (like AODV). Due to the proactive nature of the protocol, and the frequent generation and transmission of topology update messages, the CPU usage increases compared to other routing protocols.

Scalability

Due to the high overhead generated by the transmission of update messages, this protocol is not very scalable. Every time there is a topology change, the nodes detecting this change send update messages to their neighbors, who modify their routing table entries with the new information and then in turn send out update messages to their neighbors informing them of the change. In this manner the topology update information is propagated throughout the network. Therefore, as the number of nodes in the network increases so will the number of messages and time it takes to keep the topology information up-to-date.

2.2.1.2 Optimized Link State Routing (OLSR)

The Optimized Link state Protocol (OLSR) is a proactive link state routing protocol. The OLSR protocol is an improvement over the older and less effective proactive routing protocol, the Destination-Sequenced Distance-Vector (DSDV) protocol. It uses a different routing technique designed to adapt to a network which is dense and where data transmission is assumed to occur frequently between large numbers of nodes. It uses periodic messages for updating the topology information. OLSR is based on the following mechanisms:

- Neighbor sensing based on periodic exchange of HELLO messages
- Efficient flooding of control traffic using the concept of multipoint relays
- Computation of an optimal route using the shortest-path algorithm

Neighbor Sensing

Neighbor sensing is the detection of changes in the neighborhood of the node. Node **A** is called neighbor of node **B** if the two nodes are directly linked, allowing data transmission in both directions of the link. The node **C** is called a two-hop neighbor of **A**, if node **C** is not neighbor of node and there exists a symmetric link between **A** and **B** and a symmetric link between **B** and **C**. For neighbor sensing the node periodically emits HELLO messages. The HELLO message consists of the emitting node's address, the list of his neighbors, including the link status (e.g. asymmetric or symmetric). A node thereby informs its neighbors of which neighbors it has confirmed communication. By receiving a HELLO message, a node generates information describing its two-hop neighborhood

and the quality of the links in its neighborhood. Each node maintains this information set which is valid for a limited time only and has to be refreshed to keep it valid.

Message Flooding and Multipoint Relays

HELLO-messages are exchanged between neighbors only. These messages provide topology information for the nodes. Because the size of a MANET can be considerable, there is a need for efficient distribution of topological information in a network of any size. The task is to provide a mechanism which allows spreading information to each node without unnecessary, duplicate retransmissions. The multipoint relay (MPR) concept decreases the flooding overhead in contrast with full flooding.

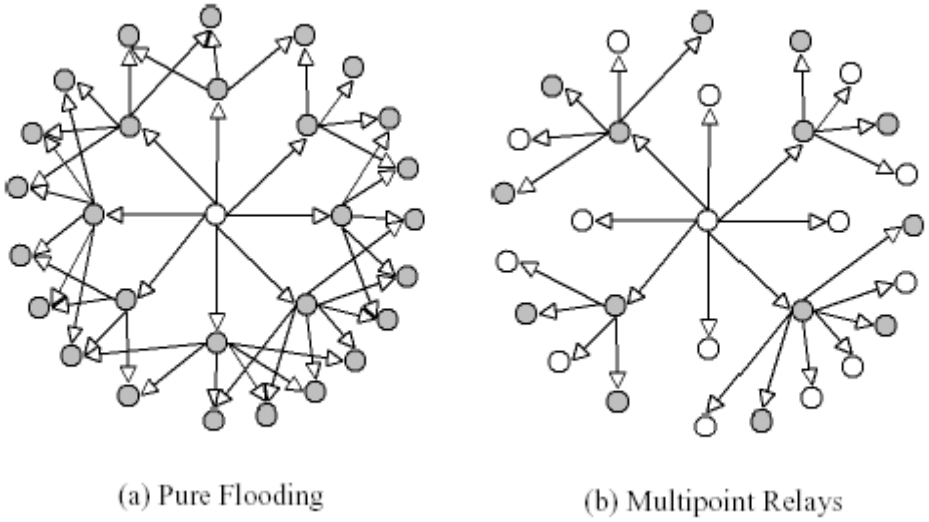


Figure 2.3 Comparisons of Two Flooding Techniques [11]

Full flooding A node retransmits broadcast packet after reception of its first copy, further duplicate receptions are dropped and not forwarded (Figure 2.2(a)).

MPR flooding Each node chooses independently a set of nodes as MPRs (multipoint relays). For this purpose it utilizes the information about its two-hop neighbors to get a minimal MPR set. This set is chosen so that a node reaches all its two-hop neighbors through its MPR relays. Each node maintains a list of nodes which selected it as MPR (MPR selector set). A MPR node only retransmits a broadcast packet if it is received from a node for which it is located in the MPR selector set, further receptions of the same packet are dropped (Figure 2.2(b)).

Spreading Topology Information and Calculating Routes

Finally it is important to spread the topology information to all nodes. All nodes with a non-empty MPR selector set periodically send a topology control message (TC-message). A TC message contains the address of its originator and the MPR set of that node. All MPRs of a node get the reachability information of that node. As a result all nodes will receive a partial topology graph by using that information and the links of their set of links to their MPR selectors. The shortest path algorithm is applied to the partial topology graph for computing the optimal path. Topology information in each node is only valid for a specific period of time and when it is expired it is removed from the graph.

2.2.2 Reactive Routing Protocol

A different approach from table-driven routing is source-initiated on-demand routing. This type of routing creates routes only when desired by the source node. When a node requires a route to a destination, it initiates a route discovery process within the network. This process is completed once a route is found or all possible route permutations have been examined. Once a route has been established, it is maintained by some form of route maintenance procedure until either the destination becomes inaccessible along every path from the source or until the route is no longer desired. There are various types of reactive routing protocols; we have studied DSR and AODV in our project.

Advantage: less overhead due to “route-messages”.

Disadvantage: source must wait until route is discovered

2.2.2.1 Dynamic Source Routing (DSR)

The dynamic Source Routing (DSR) is an on-demand routing protocol that is based on the concept of source routing. Mobile nodes are required to maintain route caches that contain the source routes of which the mobile is aware. Entries in the route cache are continually updated as new routes are learned. The Dynamic Source Routing protocol (DSR) is a simple and efficient routing protocol designed specifically for use in multi-hop wireless ad hoc networks of mobile nodes. Using DSR, the network is completely self-organizing and self-configuring, requiring no existing network infrastructure or administration.

Mode of Operation

DSR operate on demand, which means that no data, such as route advertisement messages, is send periodically and therefore routing traffic caused by DSR can scale down and overhead packages can be avoided. DSR is a source routing protocol, which means the entire route is known before a packet transmission is begun. DSR stores discovered routes in a Route Cache.

The DSR protocol is composed of two main mechanisms that work together to allow the discovery and maintenance of source routes in the ad hoc network:

Route Discovery

When a node S sends a packet to the destination D, it first searches its Route Cache for a suitable route to D. If no route from S to D exists in S's route cache, S initiates Route Discovery and sends out a ROUTE REQUEST message to find a route. The fields of the ROUTE REQUEST message are explained in Table 2.2.

The initiator initializes the Address List to an empty list and set the Initiator ID, the Target Id and the Unique Request Id in the ROUTE REQUEST message and then broadcasts the message. This causes the packet to be received by nodes within the wireless transmission range. The initiator keeps a copy of the packet in a buffer, referred to as the send buffer. It timestamps the message so it can be examined later to determine if it should be send again. If no route is discovered within a specified time frame, the packet is dropped from the send buffer. Packets are also dropped from the send buffer if the buffer overruns.

When a node receives a ROUTE REQUEST message it examines the Target ID to determine if it is the target of the message. If the node is not the target it searches its own route cache for a route to the target. If a route is found it is returned. If not, the nodes own id is appended to the Address List and the ROUTE REQUEST is broadcasted. If a node subsequently receives two ROUTE REQUESTs with the same Request id, it is possible to specify that only the first should be handled and the subsequent discarded. If the node is the target it returns a ROUTE REPLY message to the initiator. This ROUTE REPLY message includes the accumulated route from the ROUTE REQUEST message. The target searches its own Route Cache for a route to the initiator. The reason that the target node doesn't just reverse the found route and use it is that that would require bi-

directional links. If a route is not found in the targets Route Cache, it performs a route discovery of its own and sends out a ROUTE REQUEST where it piggybacks the ROUTE REPLY for the initiator.

Table 2.2 Fields of the ROUTE REQUEST Message. The Italic font is used to indicate fields used for the more advanced features of DSR.

Fields	Explanation
Initiator ID	The address of the initiator.
Target ID	The address of the target
Unique Request ID	A unique ID that can identify the message.
Address List	A list of all addresses of intermediate nodes that the message passes before its destination. This is empty when the message is first send.
Hop Limit	The hop limit can be used to limit the number of nodes that the message is allowed to pass.
Network Interface List	If nodes have several network interfaces this information can be stored in this list.
Acknowledgment bit	There is an option of setting a bit so that the receiver returns an acknowledgment when a packet is received.

Route Maintenance

Since nodes move in and out of transmission range of other nodes and thereby creates and breaks routes, it is necessary to maintain the routes that are stored in the Route Cache. When a node receives a packet it is responsible for confirming that the packet reaches the next node on the route. Figure 2-4 that the mechanism works like a chain where each link has to make sure that the link in front of it is not broken. The figure also illustrates that node C might use another route to communicate to node A.

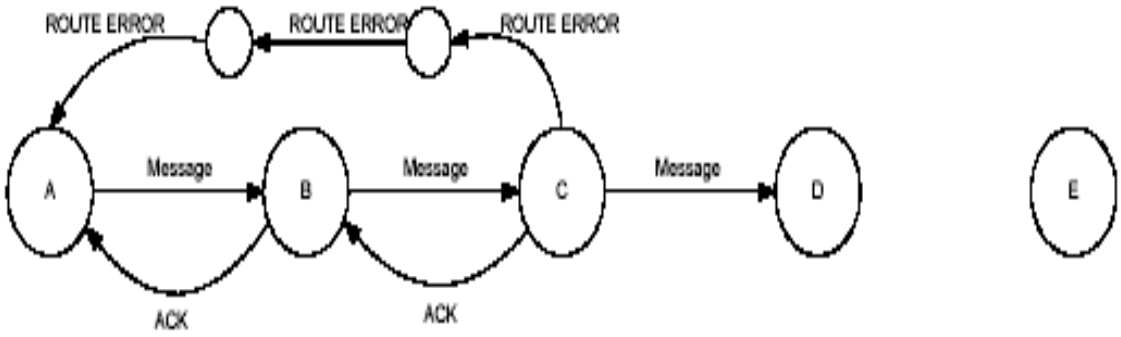


Figure 2.4 The Acknowledgement Mechanism Works Like a Chain [22]

Acknowledgment can be performed either by using mechanisms in the underlying protocol such as link-level acknowledgment or passive acknowledgment. If none of these mechanisms are available, the transmitting node can set a bit in the packets header to request a specific DSR acknowledgment. If a node transmits a packet and does not receive an acknowledgment it tries to retransmit a fixed number of times. If no acknowledgement is received after the retransmissions, it returns a ROUTE ERROR message to the initiator of the packet. In this message the link that was broken is included. The initiator removes the route from its Route Cache and tries to transmit using another route from its Route Cache. If no route is available in the Route Cache a ROUTE REQUEST is transmitted in order to establish a new route.

Advantages

Routes are maintained only between nodes that need to communicate. This reduces the overhead of route maintenance. Route caching can further reduce route discovery overhead. A single route discovery may yield many routes to the destination, due to intermediate nodes replying from their local caches.

Disadvantages

Packet header size keeps on growing with the route length. Flooding problems can take place every now and then. Care needs to be taken to avoid collisions between route requests propagated by neighboring nodes. Insertion of random delays before forwarding RREQ may be a measure to minimize this collision.

Resource usage

Typical routing protocols such as distance-vector store just the next hop for any route, but DSR requires each node to maintain a full topology for all hosts with which it wants to communicate. Hence this adds a load on memory resources. DSR uses more CPU time than other routing protocols like AODV and DSDV. One reason for this could be that DSR requires each host to monitor all of the network traffic going on within its receiving range.

Scalability

DSR uses an optimized form of flooding to reduce network overhead. On route discovery it sends one broadcast packet to all its neighbors. If it does not receive information from them on how to reach the destination node, then it sends a network-wide broadcast. Due

to the use of such optimizing techniques, DSR produces a significantly lower amount of network overhead; However, DSR may still not be a very scalable protocol because each node is required to maintain full knowledge of the paths over which it needs to communicate. The more destinations, the more memory is required, a likely condition as the network gets busier.

2.2.2.2 Ad hoc On-Demand Distance Vector (AODV)

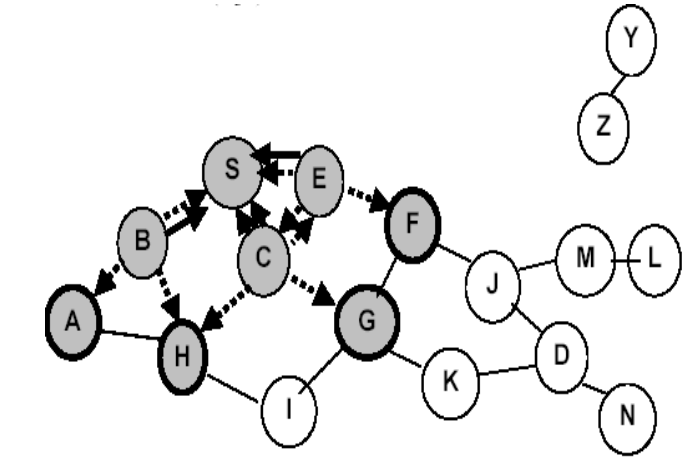
The Ad hoc On-Demand Distance Vector (AODV) algorithm enables dynamic, self-starting, multihop routing between participating mobile nodes wishing to establish and maintain an ad hoc network. AODV allows mobile nodes to obtain routes quickly for new destinations, and does not require nodes to maintain routes to destinations that are not in active communication. AODV allows mobile nodes to respond to link breakages and changes in network topology in a timely manner. The operation of AODV is loop-free, and by avoiding the Bellman-Ford "counting to infinity" problem offers quick convergence when the ad hoc network topology changes (typically, when a node moves in the network). When links break, AODV causes the affected set of nodes to be notified so that they are able to invalidate the routes using the lost link. One distinguishing feature of AODV is its use of a destination sequence number for each route entry. The destination sequence number is created by the destination to be included along with any route information it sends to requesting nodes. Using destination sequence numbers ensures loop freedom and is simple to program. Given the choice between two routes to a destination, a requesting node is required to select the one with the greatest sequence number.

Route Discovery

Whenever there exists a valid route between two communication peers, AODV Route Discovery is not used. As soon as a route is missing between the two communications partners, e.g. when a new route to a destination is needed, a link is broken, or the route has expired, the source node **S** broadcasts a ROUTE REQUEST message in order to find a route to the destination **D**.

• Route Requests (RREQ) are forwarded in a manner similar to DSR described above

• When a node re-broadcasts a Route Request, it sets up a reverse path pointing towards the source (Figure 2.5).



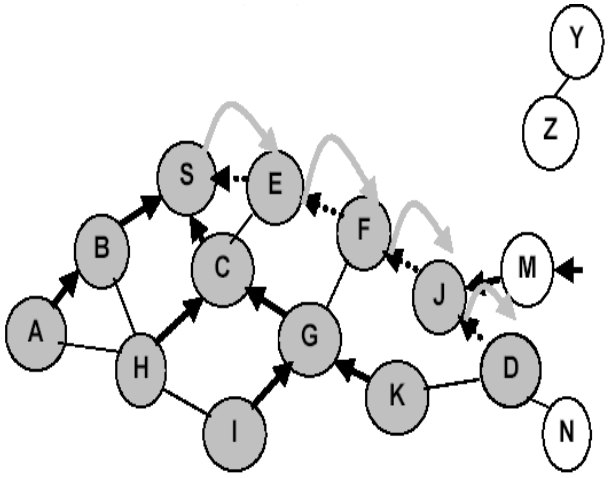
Represents links on the reverse path (in Figure 2.5).

Figure 2.5 Route Discovery [21]

• Here again nodes do not forward RREQ if they have already forwarded it once.

• When the intended destination receives a Route Request, it replies by sending a Route Reply message. The destination does not forward the Route Request message as it is intended for itself.

• The Route Reply message travels along the reverse path set-up when Route Request was forwarded (in Figure 2.6).



Represents links on path taken by RREP (in Figure 2.6).

Represents a link on the forward path (in Figure 2.6).

Figure 2.6 Route Reply [21]

• Forward links are set up when RREP travels along the reverse path. This information is stored in the routing table.

• These routing table entries are used to forward the data packets and is not included in the packet header.

Timeouts

A routing table entry maintaining a reverse path is purged after a timeout interval. In this case timeout should be long enough to allow RREP to come back. A routing table entry maintaining a forward path is purged if not used for a active_route_timeout interval. That if no data is being sent using a particular routing table entry, that entry will get deleted from the routing table (even if the route may actually still be valid).

The Business of Sequence Numbers

An intermediate node (not the destination) may also send a Route Reply (RREP) provided that it knows a more recent path than the one previously known to sender S. To determine whether the path known to an intermediate node is more recent, destination sequence numbers are used.

Route Maintenance

To maintain routes the nodes survey the link status of their next hop neighbors in active routes. The node detecting a link break sends a ROUTE ERROR message to each of its upstream neighbors to invalidate this route and these propagate the ROUTE ERROR to their upstream neighbors. This continues until the source node is reached. Normally the nodes in AODV sends periodic HELLO messages and the failure of reception of three consecutive HELLO messages from a neighbor are handled as link error. Another possibility of link breakage detection uses link layer notification. This alternative results in a pure on-demand nature of the link breakage detection. A broken link cannot be identified until packets should be sent over the link. By contrast the HELLO messages in standard AODV allows the detection of broken links before a packet must be forwarded, but this has the disadvantage of use of bandwidth for the periodic transmission of HELLO messages. The ROUTE ERROR message contains a infinite metric for the destination and causes the receiver to invalidate the route. Now the node must start a new Route Discovery for a connection to this destination.

Use of Sequence numbers

The sequence numbers are mainly used for the following purposes:

a) To avoid using old/broken routes.

b) To prevent formation of loops (counting to infinity problem).

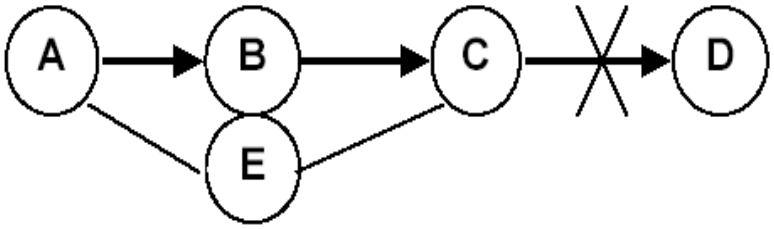


Figure 2.7 Use of Sequence Numbers [21]

• Let us assume that A does not know about failure of link C-D because RERR sent by C is lost.

• Now C performs a route discovery for D. Node A receives the RREQ (say, via path C-E-A).

• Node A will reply since A knows a route to D via node B.

• Results in a loop (for instance, C-E-A-B-C).

Resource Usage

The routing table maintained at each node contains the following information: destination, next hop, sequence number, and status of the link. However, because AODV is an on-demand protocol, the actual size of the route table is much smaller on average compared to the table maintained by DSDV. The size of the routing table at each node is directly proportional to the number of active destination nodes. Thus even though some memory is required to maintain these routing tables, it is less than the amount required to maintain the routing tables for DSDV.

The CPU is used to route packets and discover the routes to the destination, so this approach does not impose any additional load on the CPU compared to DSDV.

Scalability

As the number of nodes in a network increases, the number of routing packets sent is likely to increase as well. Increasing network size most likely translates to an increase in

number of destinations to which each node must maintain working routes. Also the incremental cost for nodes added to the network decreases, because the new nodes use the information learned from one route discovery to fill their tables with information from previous route discoveries already captured at other nodes.

Advantage: Routes need not be included in packet headers.

Disadvantage: Unused routes expire even if topology does not change.

3 Introduction to Network Simulator

3.1 Network Simulator 2 (ns2)

Network Simulation is a technique where a program models the network behavior either by calculating the interaction between the different network entities by actually capturing and playing back observations from a production network.

Network Simulators are relatively fast and inexpensive as they allow the engineers to test scenarios that might be particularly difficult or expensive to emulate using real hardware. For example, simulating the effects of a sudden burst in traffic or a DoS attack on a network service. These allow designers to test new networking protocols or change the existing ones in a controlled environment.

A typical network simulator like NS2 encompasses a wide range of networking technologies and helps the users to build complex networks from basic building blocks like variety of nodes and links.

Most of the commercial simulators are GUI driven, while some network simulators require input scripts or commands (network parameters). The important output of simulations is the trace files. The network parameters describe the state of the network (node placement, existing links) and the events (data transmissions, link failures, etc). Trace files can document every event that occurred in the simulation and are used for analysis.

3.1.1 Version of Ns-2 Used in This Project

By default ns-2 has the support for AODV, DSDV, DSR and TORA. Version 2.30 of ns-2 for the simulation of AODV, DSDV, and DSR has been used. For the simulation of OLSR patch in version 2.28 of ns-2 has been installed.

3.2 Structure of Ns2

Ns2 is built using object oriented methods in C++ and OTcl (object oriented variant of Tcl). The Figure 2.1shows that ns2 interprets the simulation scripts written in OTcl. Some parts of ns2 are written in C++ for efficiency reasons. The C++ objects are controlled by OTcl objects. Results obtained by ns 2 (trace files) have to be processed by other tools, e.g. the Network Animator (*NAM*), a *perl* or *awk* script and *gnuplot*

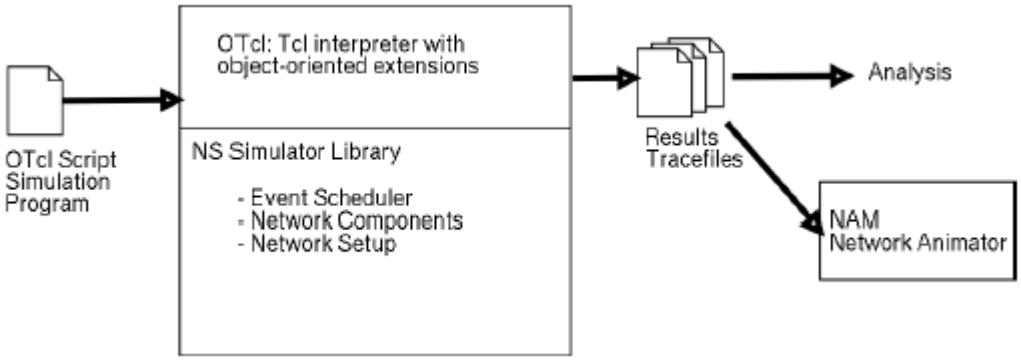


Figure 3.1 Simplified User's View of NS [15]

'Nam' is an animation tool for viewing network simulation results and real world packet traces. 'Xgraph' is basically a plotting program which can be used to create graphic representations of simulation results.

3.2.1 Awk

The awk utility shall execute programs written in the awk programming language, which is specialized for textual data manipulation. An awk program is a sequence of patterns and corresponding actions. When input is read that matches a pattern, the action associated with that pattern is carried out. Input shall be interpreted as a sequence of records. For each pattern matched, the associated action shall be executed. Programs in awk are different from programs in most other languages, because awk programs are *data-driven*.

An Awk program has the general form:

BEGIN {<initializations>}

<Search pattern 1> {<program actions>}

<Search pattern 2> {<program actions>}

END {<final actions>}

When awk is run, an awk program is specified that tells awk what to do. The program consists of a series of *rules*. It may also contain *function definitions*. Each rule specifies one pattern to search for and one action to perform upon finding the pattern.

There are several ways to run an awk program. If the program is short, it is easiest to include it in the command that runs awk, like this:

awk '*program*' *input-file1 input-file2...*

When the program is long, it is usually more convenient to put it in a file and run it with a command like this:

`awk -f` *program-file input-file1 input-file2* `...`

3.3 Generation of Node-Movement and Traffic-Connection for Wireless Scenarios

Normally for large topologies, the node movement and traffic connection patterns are defined in separate files for convenience. These movement and traffic files may be generated using CMU's movement- and connection-generators.

3.3.1 Traffic Models

Random traffic connections of TCP and CBR can be setup between mobile nodes using a traffic-scenario generator script. This traffic generator script is available under ~ns/indep-utils/cmu-scen-gen and is called cbrgen.tcl. It can be used to create CBR and TCP traffics connections between wireless mobile nodes. So the command line looks like the following:

```
ns cbrgen.tcl [-type cbr|tcp] [-nn nodes] [-seed seed] [-mc connections][-rate rate]
```

3.3.2 Mobility Models

The node-movement generator is available under ~ns/indep-utils/cmu-scen-gen/setdest directory and consists of setdest {.cc, .h}. The command would look like

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

4 Analysis of Routing Protocols in MANETs through Simulation

4.1.1 Simulation

The four routing protocols AODV, DSDV, DSR and OLSR have been simulated in ns2.All these protocols are provided with identical traffic load and mobility patterns. UDP has been considered as transport protocol and CBR as traffic generator. protocol evaluations are based on the simulation using ns2 and the graphs are generated using X-graph.

4.2 Environmental Factors Influence Simulation

There are several environmental factors affect the simulation performance, stability and accuracy. These factors can be listed as Degree of Connectivity among Nodes, Degree of Mobility, Number and Duration of Data Flows.

4.2.1 Degree of Connectivity among Nodes

In a highly dense network, almost every node has at least a path to any other node, usually just a few hops away. Meanwhile due to the high volume of routing control messages, congestion happens frequently in such networks. A sparsely connected ad hoc network bears different characteristics. In such a network, paths between two nodes do not always exist, and routing choices are more obviously affected by the mobility of the network. In simulation study, it has been ran simulations in both sparse and dense networks, changing the area between to be 500 x 500 up to 2000 x 2000, and the number of nodes to be 10, 20, and 50.

4.2.2 Degree of Mobility

Varying the degree of mobility, or the moving speed of each node in the network, is a useful way to test how adjustable a routing protocol is to the dynamic environment. There has been several mobility models used in literature. It has been chosen the generated movement scenario because this has been used more widely than other mobility models. In this model, each node begins the simulation by predetermined for

"pause time" seconds. It then selects a random destination in the simulation space and moves to that destination at a speed distributed uniformly between a minimum and a maximum speed. Upon reaching the destination, the node pauses again for "pause time" seconds, selects another destination, and proceeds there as previously described, repeating this behavior for the duration of the simulation. In simulations, it has been changed maximum speed to be between 10m/sec 20m/sec, and varied the "pause time" between 1 and 5 seconds.

4.2.3 Number and Duration of Data Flows

Because on-demand protocols query routes only when data flows exist for them, the number of data flows would influence the number of paths found and the control overhead for on demand protocols, such as AODV and DSR. How well a protocol adjusts to the change of data flows is another important criterion for evaluating a routing protocol. In simulations, it has been varied the number of data flows to be 10, 20, and 40. In simulation studies, each data flow started at an early time of the simulation period, and continued until almost the end of the period.

4.3 Prevent Simulations from the Inaccurate Comparisons

For the health of the simulation, the same traffic and movement scenarios were used in all DSDV, DSR, AODV and OLSR protocols applications. There are several traffic and movement scenario files which were generated with the scenario generators located in the ~ns/indep-utils/cmu-scen-gen. It can be used to create CBR and TCP traffics connections between wireless mobile nodes. In order to create a traffic-connection file, it needs to define the type of traffic connection (CBR or TCP), the number of nodes and maximum number of connections to be setup between them, a random seed and incase of CBR connections, a rate whose inverse value is used to compute the interval time between the CBR packets.

4.4 Performance Metrics

In this study the following metrics are used to evaluate the performance of the routing protocols;

Packet Delivery Ratio: The packet delivery ratio in this simulation is defined as the ratio between the number of packets sent by constant bit rate sources (CBR,"application layer") and the number of received packets by the CBR sink at destination.

$$\text{Packet delivery ratio} = \frac{\sum \text{CBR packets received by CBR sinks}}{\sum \text{CBR packets sent by CBR sources}}$$

It describes percentage of the packets which reach the destination.

Throughput: It is defined as total number of packets received by the destination. It is a measure of effectiveness of a routing protocol. Finally what matters is the number of packets delivered successfully.

$$Throughput = \sum packets\ received\ by\ CBR\ source$$

Packets Lost: It is a measure of the number of packets dropped by the routers due to various reasons. The reasons we have considered for evaluation are Collisions, time outs, looping, errors.

$$Packet\ loss = \sum Data\ packets\ Drop$$

Routing Overhead: The sum of all transmissions of routing packets sent during the simulation. For packets transmitted over multiple hops, each transmission over one hop, counts as one transmission.

$$Routing\ overhead = \sum Transmissions\ of\ routing\ packets$$

Routing overhead is important to compare the scalability of the routing protocols, the adoption to low-bandwidth environments and its efficiency in relation to node battery power (in that sending more routing packets consumes more power). Sending more routing packets also increases the probability of packet collision and can delay data packets in the queues.

Table 4.1 These four characters specify the action that was processed to the packet.

s	send
r	receive
d	drop
f	forward

Normalized Routing Load: Ratio of send network control packets to all received data packets. It is a measure of efficiency of the protocol. Less value of normalized routing load shows more efficient protocol.

$$\textit{Normalized Routing Load} = \frac{\sum \textit{Routing Packets Send by Routing protocol}}{\sum \textit{CBR packets received by CBR sources}}$$

4.5 Scenarios and Results

There are several scenarios that have been created, but it wants to discuss eight of them here. Because of both simulation and tracing the trace files are very time consuming issues when the network topology becomes larger and complex, it has been illustrated simulations up to 50 nodes and the topologies that have up to 2000 x 2000 dimensions.

4.5.1 Scenario 1

In the first simulation scen_10node_2s_10mps_200sim_500x500 and cbr_10node_10con_3rate scenario files have been used as movement scenario and traffic scenario respectively. It can easily be inferred from the name of the scenario files, it have 10 mobile nodes with a 2 seconds of pause time and with a maximum speed of 10m/s in a 500x500 region. After the simulation and analyzing the trace files, it have been obtained the graphs as presented;

For the current situation packets drop is minimum in DSR and AODV and packet delivery fraction is almost same in AODV and DSR. And packets drop, normalized

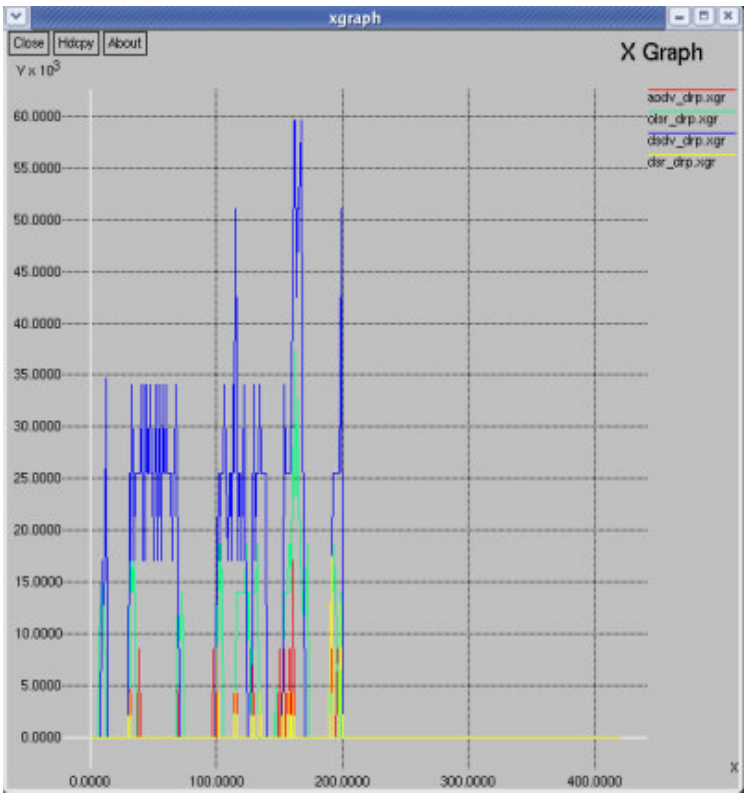


Figure 4.1 No of Packets Dropped

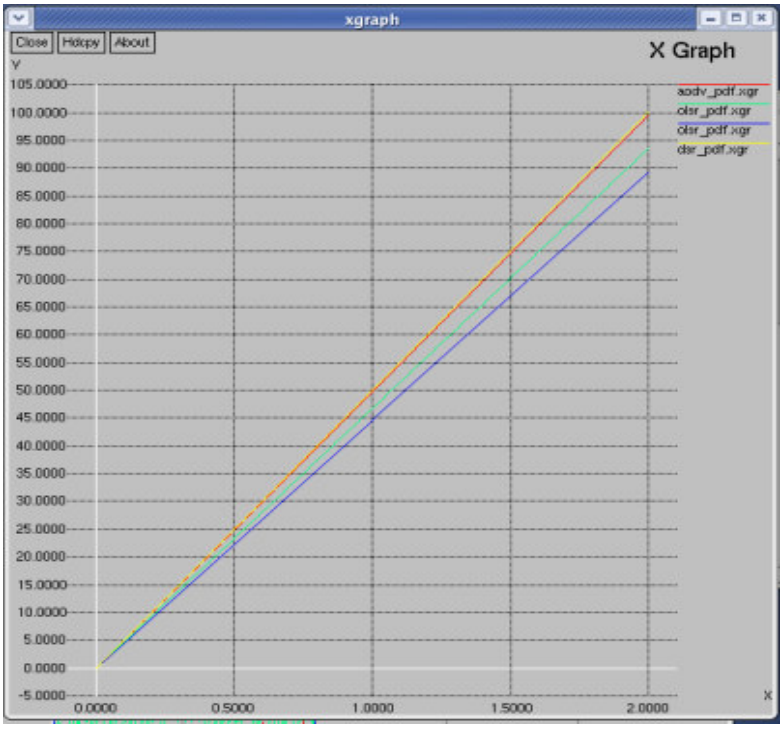


Figure 4.2 Packet Delivery Fraction

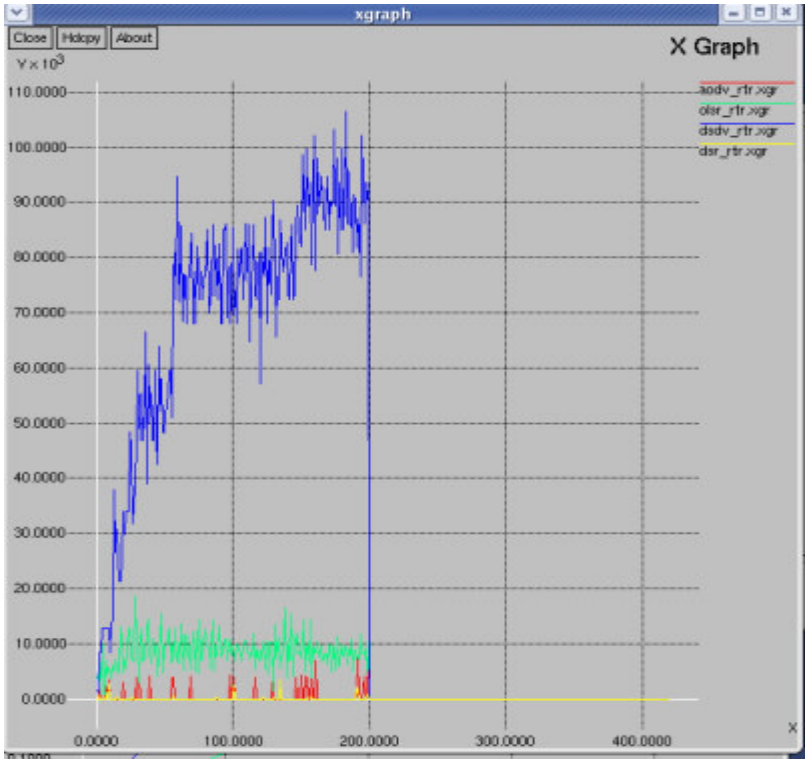


Figure 4.3 Routing Overhead

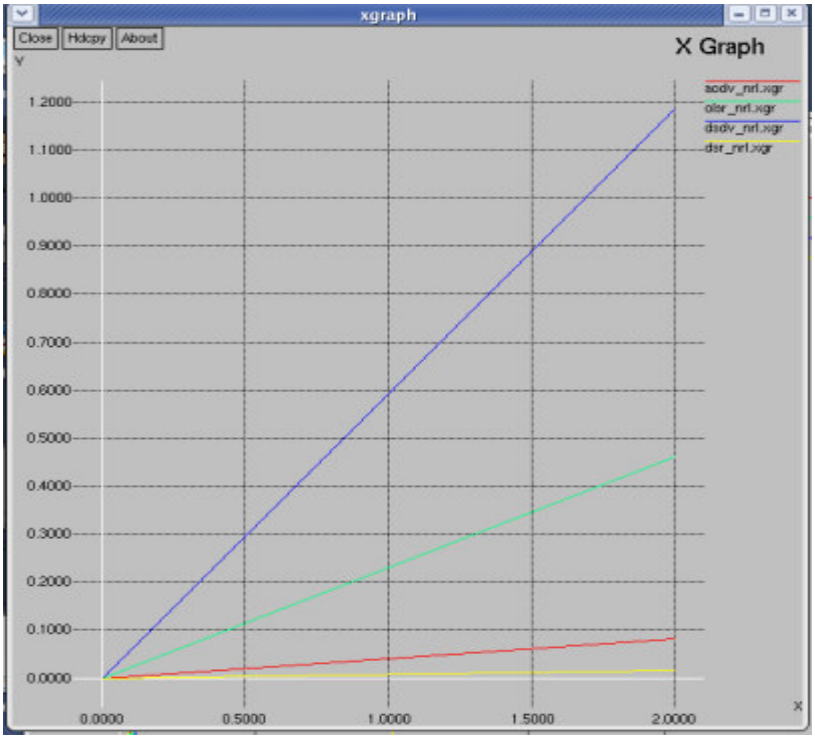


Figure 4.4 Normalized Routing Load

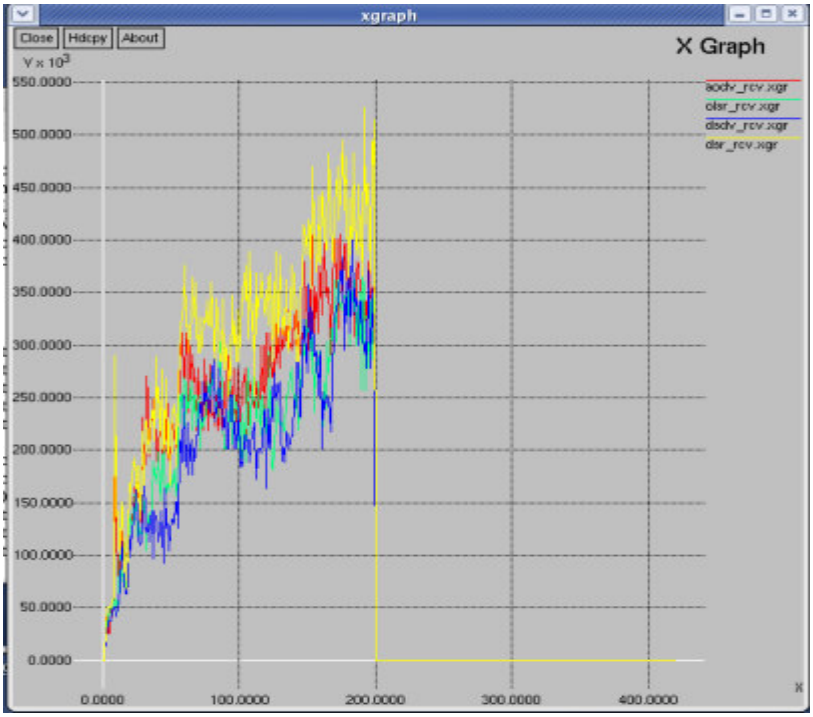


Figure 4.5 No of Packets Received

Routing load and routing overhead is highest in DSDV. OLSR have the good throughput but have more normalized routing load. Also DSR has maximum throughput and minimum normalized routing load and routing overhead. It means DSR is more reliable and useful for the current situation.

4.5.2 Scenario 2

In the second simulation scen_10node_2s_10mps_200sim_1250x1250 and cbr_10node_10con_3rate scenario files have been used as movement scenario and traffic scenario respectively. The region size is expanded by 1250x1250, so it expected to see the performance of the protocols when the density of the nodes is decreased in the region. Only different parameter is the region size than the first simulations since other factors are the same. After the simulation and analyzing the trace files, it have been obtained the table 4.2.

As it can be inferred from the table, OLSR have the good throughput of generated and sent packets. It also has the lower dropped packets. The performance of the AODV and DSR is comparably less than the performance of the first simulation.

Table 4.2 Outputs of the Simulation under Scenario 2

Metrics	DSDV	OLSR	AODV	DSR
No of Packets Received	226304	188160	124480	17864
No of Packets Drop	68096	62256	212800	63840
Normalized Routing Load	1.5291	0.9731	0.2694	0.131
Packet Delivery Fraction	36.548	42.516	49.294	49.6
Routing Overhead	52000	4928	1152	512

Also DSDV has good throughput and has minimum packets drop in this scenario than the first one but have maximum routing overhead. OLSR performed the much better results than the others in a large region and with a less number of nodes.

4.5.3 Scenario 3

In the third simulation scen_50node_2s_10mps_200sim_2000x2000 and cbr_50node_20con_3rate scenario files have been used as movement scenario and traffic scenario respectively. In this simulation, performance of the protocols in more complex scenario can be analyzed, because both the number of mobile nodes increased considerably and the region size is expanded. The connectivity of the nodes is also limited. So the congestion on the intermediate nodes will be increased and the characteristics of the protocols will start to show themselves. After the simulation and analyzing the trace files, we have obtained the table 4.3

The OLSR seems that it is going to be down. Both throughput of generated and throughput of sent packets reduce drastically and it has maximum routing overhead. DSDV has maximum normalized routing load and minimum throughput. DSR have a performance above the average and notice that when the destination can not be reached in single hop, DSR start to reflect flooding behavior.

It can be obtained from the throughput of the forwarded packets. AODV have the higher throughput of generated packets, higher throughput of sent packets and lower throughput of dropped packets. It means AODV is preferable for the current situation.

Table 4.3 Outputs of the Simulation under Scenario 3

Metrics	DSDV	OLSR	AODV	DSR
No of Packets Received	147904	1518144	926720	265171
No of Packets Drop	319616	18406	12768	159744
Normalized Routing Load	8.0191	6	3.1556	0.8841
Packet Delivery Fraction	10.576	24.3356	44.0993	47.963
Routing Overhead	262784	16790	35872	39360

4.5.4 Scenario 4

In the fourth simulation scen_20node_1s_10mps_200sim_500x500 and cbr_20node_20con_3rate scenario files have been used as movement scenario and traffic scenario respectively. This simulation may enable us to see what would be the performances of the protocols when the number of nodes increased.

After the simulation and analyzing the trace files, we have obtained the graphs from which we concluded that; the performances of the protocols are approximately similar with the first simulation performances. Again DSR protocol is extremely reliable when look at throughput and packet delivery fraction.

4.5.5 Scenario 5

In the fifth simulation scen_10node_1s_10mps_200sim_500x500 and cbr_10node_10con_3rate scenario files have been used as movement scenario and traffic scenario respectively. Similarly to first simulation, the aim is to obtain the behavior of the protocols when the mobility of the nodes is changing. Contrast to the previous one, the pause time is decreased down to 1 seconds, it means now nodes are more mobile and topology is changing more often. This simulation may enable us to see what would be the performances of the protocols when the nodes are more willing to move to new destiny.

After the simulation and analyzing the trace files, we have obtained the graphs from which we have concluded that; it is inferred there is no considerable changes in

performance between the fifth simulation and first one. We expected, there should be more considerable changes in performance when the pause time decreases. We guess, this is not the case because the actual region size and the number of nodes are dominant parameters for the current simulation environment. Here, DSR looks like more reliable.

4.5.6 Scenario 6

In the sixth simulation scen_10node_5s_20mps_200sim_500x500 and cbr_10node_10con_3rate scenario files have been used as movement scenario and traffic scenario respectively. The two parameters different from the previous simulation are the pause time of the movement and the maximum speed of the mobile nodes. Pause time is increased up to 5 seconds, it means now nodes are less mobile than the previous one and the maximum speed is increased up to 20m/s. It means mobile nodes reach their destination positions faster; and they stay there much more time. This simulation may enable us to see what would be the performances of the protocols when the nodes are moving faster and when they are less willing to move to new destiny. After the simulation and analyzing the trace files, it has been obtained the graphs;

When the speed of nodes is increased but the mobility of the nodes is decreased, throughput of the generated and throughput of sent packets are increasing in AODV and DSR compared with the previous simulation. However, throughput of the generated and throughput of the sent packets are decreasing in OLSR, but it is negligible. Overall performance of DSR is dominant for this simulation.

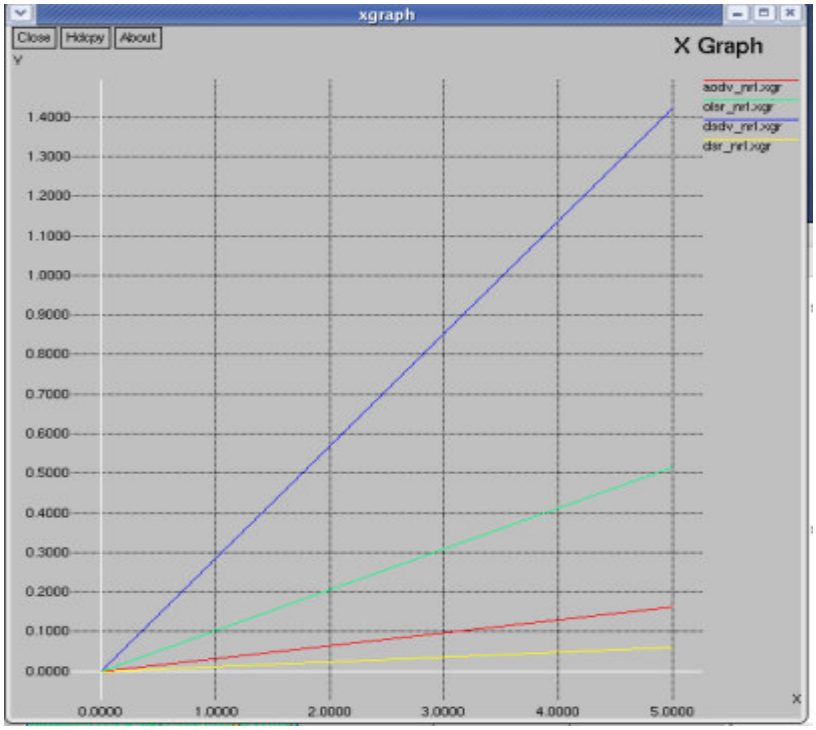


Figure 4.6 Normalized Routing Load

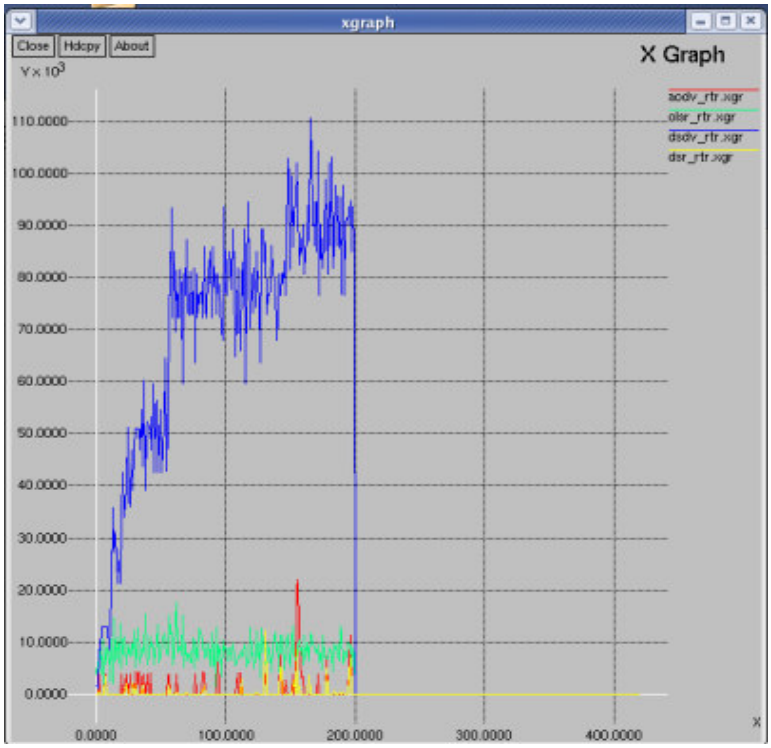


Figure 4.7 Routing Overhead

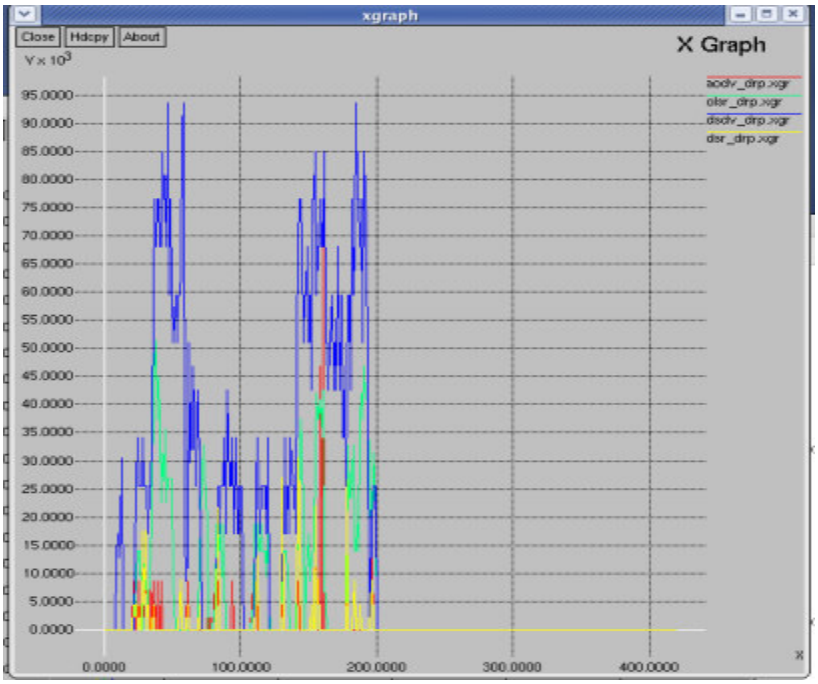


Figure 4.8 No of Packets Drop

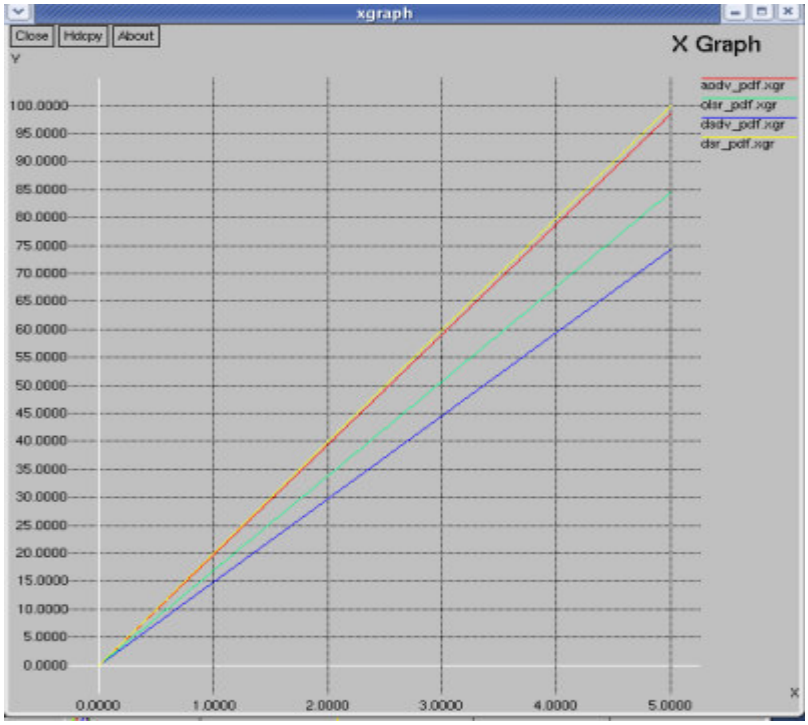


Figure 4.9 Packet Delivery Fraction

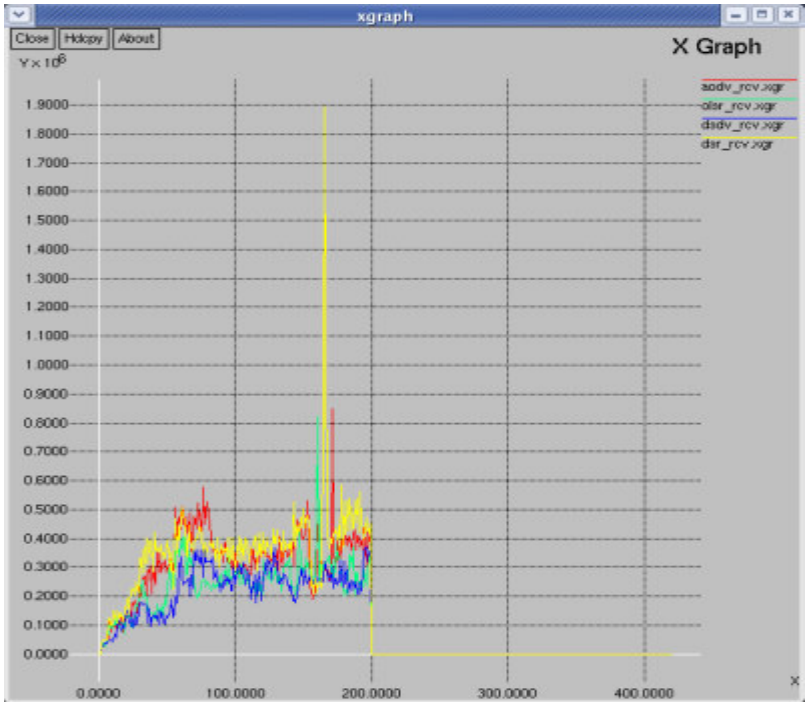


Figure 4.10 No of Packets Received

4.5.7 Scenario 7

In the second simulation scen_10node_1s_10mps_200sim_500x500 and cbr_10node_5con_3rate scenario files have been used as movement scenario and traffic scenario respectively. The only parameter different from the scenario 1 is the connectivity of the nodes. The aim of this simulation is the recognizing what happens when the connectivity of the nodes are decreased since other factors are constant. After the simulation and analyzing the trace files, we have obtained the table 4.4;

It is normal to decrease the throughput of the generated packets when the connectivity between mobile nodes decreases. It means less congestion on intermediate nodes. Packet delivery fraction and normalized routing load is almost same in AODV and DSR. AODV has minimum packets drop and lower routing overhead compared to previous scenarios. DSDV and OLSR has same normalized routing load. So AODV and DSR perform well in this scenario.

Table 4.4 Outputs of the Simulation under Scenario 7

Metrics	DSDV	OLSR	AODV	DSR
No of Packets Received	193536	240192	291264	322880
No of Packets Drop	25536	1929344	8512	17408
Normalized Routing Load	1.18824	0.68847	0.06449	0.02977
Packet Delivery Fraction	91.8271	92.9373	99.5812	99.7643
Routing Overhead	60512	10560	256	4288

4.5.8 Scenario 8

In the seventh simulation scen_10node_1s_10mps_200sim_500x500 and cbr_10node_10con_1rate scenario files have been used as movement scenario and traffic scenario respectively. It expect to see the performance of the protocols when the data sent rate is reduced. After the simulation and analyzing the trace files, it has been obtained the graphs as presented.

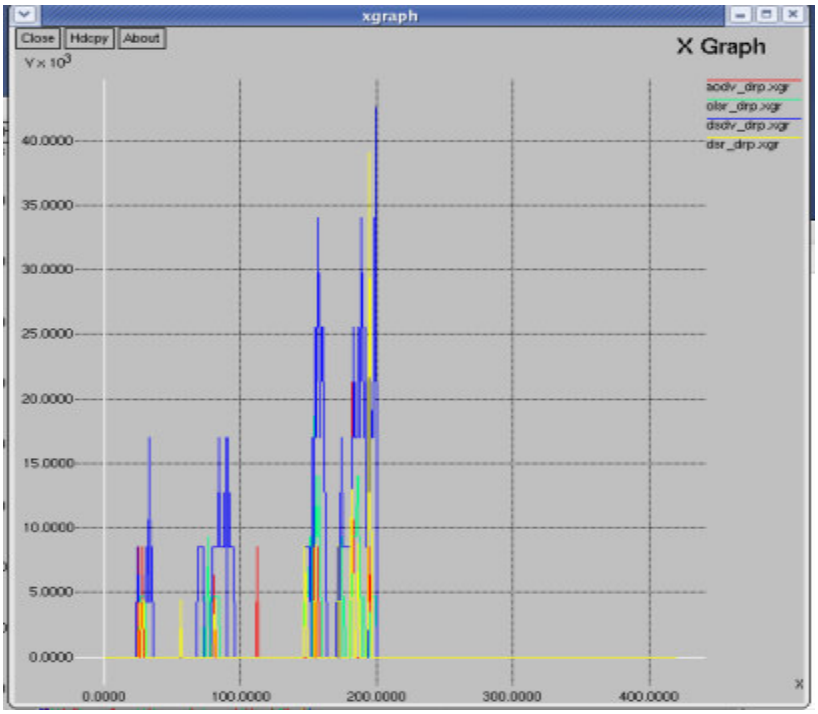


Figure 4.11 No of Packets Dropped

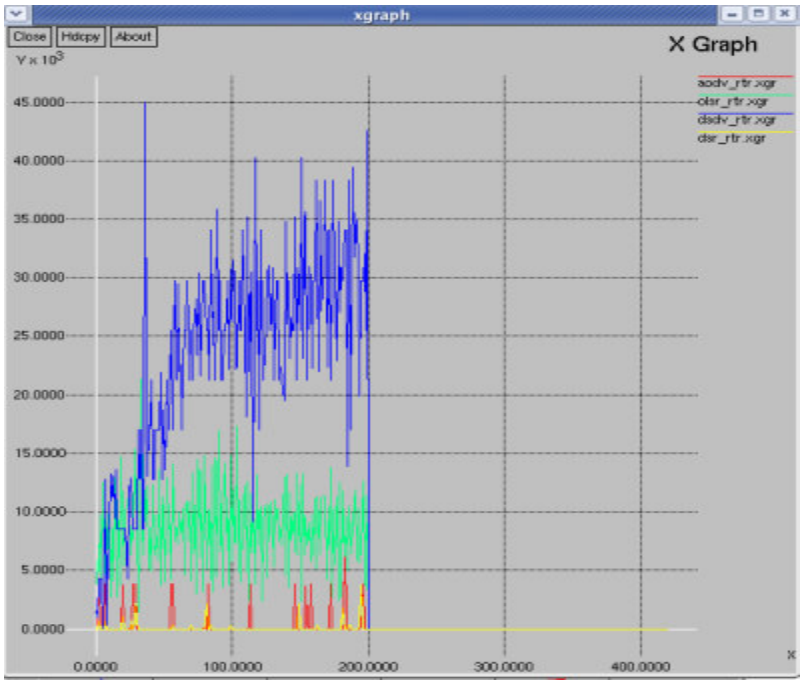


Figure 4.12 Routing Overhead

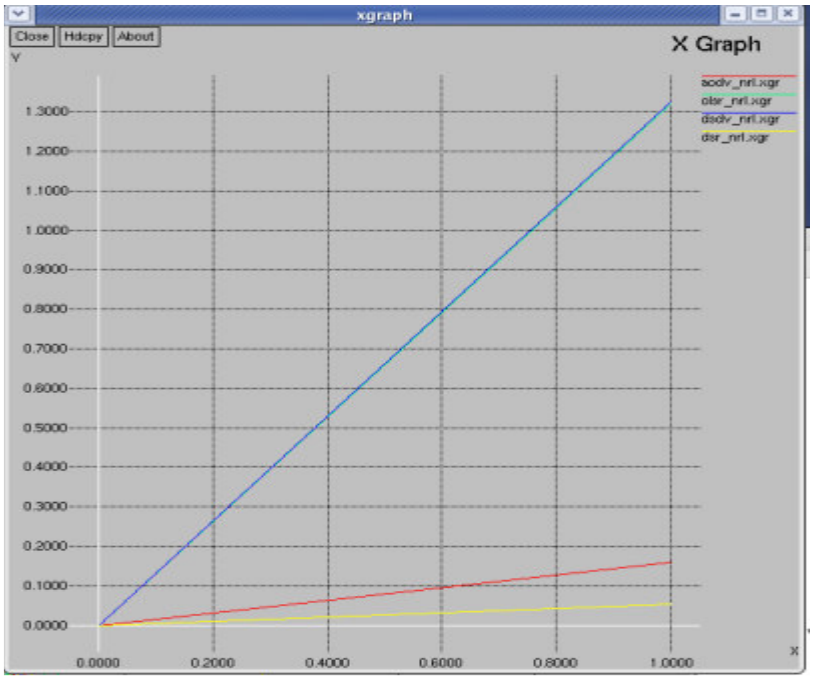


Figure 4.13 Normalized Routing Load

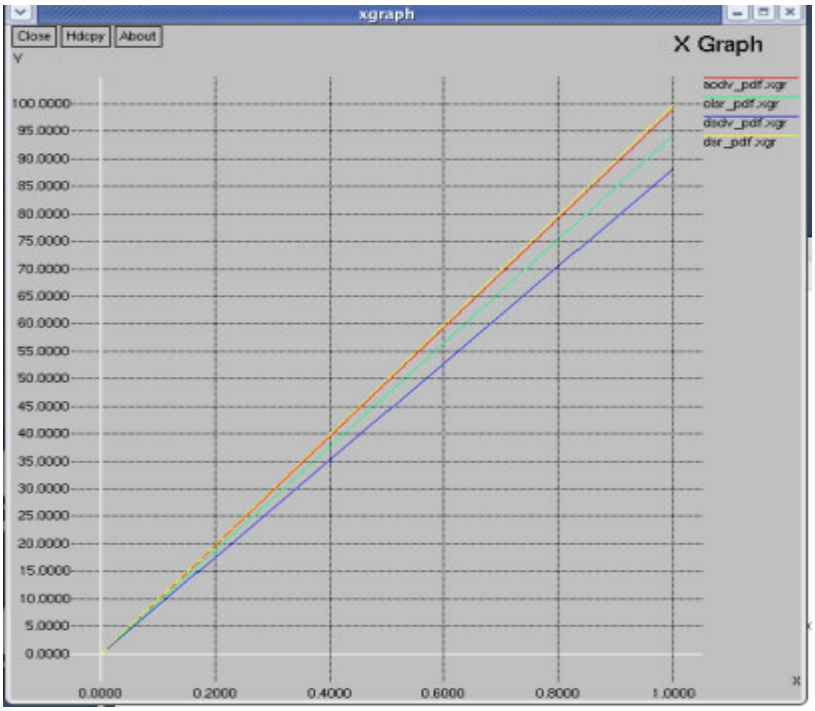


Figure 4.14 Packet Delivery Fraction

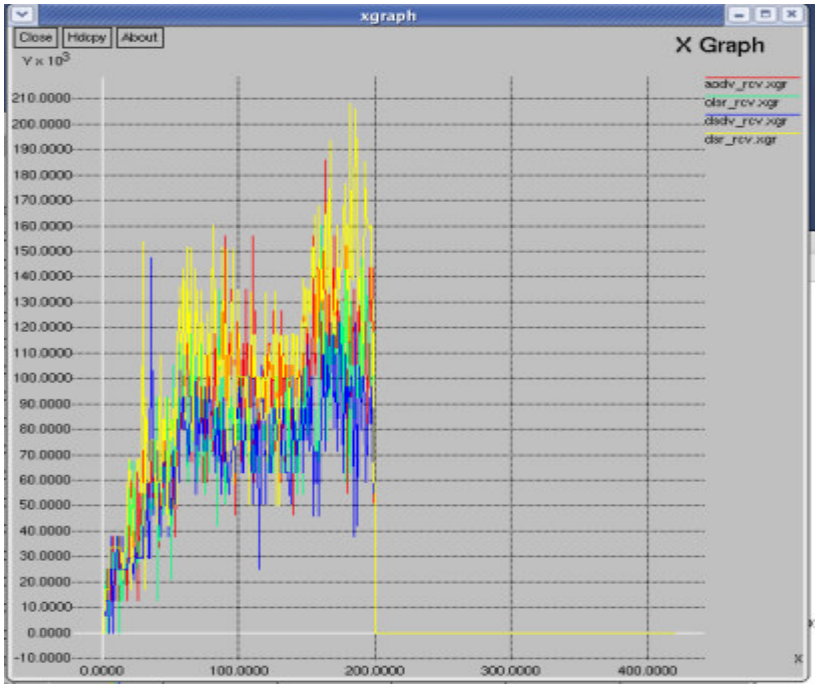


Figure 4.15 No of Packets Received

Most interesting result of this simulation is the throughput of generated packets and throughput of sent packets of the OLSR. It is considerably higher than the previous scenarios. Also throughput of dropped packets of OLSR is lower. However, AODV has minimum packets drop and also it has minimum routing overhead.

4.6 Conclusion

Now is the time to sum up what have been obtained from all the previous simulations and their results. To make explanations clear, it wants to obtain the conclusion step by step.

First, in a small size region with a few numbers of nodes, OLSR and DSDV performs well, because of less congestion on routes and it needs less calculation so it can generate and sent much more packets. When the topology becomes more complex, OLSR generates also higher number of packets but its delivery ratio decreases dramatically. DSDV produces the largest amount of overhead regardless of node mobility. DSDV failed to perform well when the mobility of the nodes increased, and DSDV performed poorly on the packet delivery rate compared to the other protocols presented.

Second, DSR produced the least amount of overhead among all the protocols. It may not be a very scalable protocol, however, because each node is required to maintain full knowledge of the paths over which it needs to communicate instead of just the next-hop entries. This problem gets even worse as the number of nodes with which each node is communicating increases. DSR works well when the size of the region is small and less number of nodes available in the region, so that DSR need less calculation to obtain the routes. Varying the other parameters does not make considerable sense on performance of DSR when the certain population density is obtained in the region. However, the certain equilibrium of the density of nodes in the region is changed, DSR become a flooding machine. In such a situation, it produces considerable overhead on the intermediate nodes, to obtain the available routes and to forward the appropriate packets to their destinations.

Third, AODV shows its power in such situations where the topology is more complex because of number of the nodes and the size of the region, so that the movements of the mobile nodes become more sophisticated. AODV, on the other hand, is well suited for scalability, as it successfully reduces the amount of network overhead, which goes down

further as the mobility within the network decreases. DSR always demonstrates a lower routing load than AODV. The major contribution to AODV's routing over-head is from route requests, while route replies constitute a large fraction of DSR's routing overhead. Furthermore, AODV has more route requests than DSR, and the converse is true for route replies.

As the final words, it is not an easy and also is not possible to say "X" protocol is the best one. In the various simulations that we have explained, there were many parameters changing usually, and depending on the topology and its dynamics, there will be different proper solutions. Clearly, the aim of this study is not ranking the protocols, but is to understand operating principles, algorithms that stay on behind of them and be able to analyze different situations and actions in the operating environment.

5 Security for Mobile Ad-Hoc Networks

5.1 Security Issues

The contemporary routing protocols for ad-hoc networks cope well with dynamically changing topology but are not designed to accommodate defense against malicious attackers. Today's routing algorithms are not able to thwart common security threats. Most of the existing ad hoc routing protocols do not accommodate any security and are highly vulnerable to attacks. Routers exchange network topology informally in order to establish routes between nodes - another potential target for malicious attackers who intend to bring down the network. External attackers inject erroneous routing information, replaying old routing information or distort routing information in order to partition a network or overload a network with retransmissions, thereby causing congestion, and hence a denial of service. Internally compromised nodes are harder to detect and correct. Routing information signed by each node will not work since compromised nodes can generate valid signatures using their private keys. Detection of compromised nodes through routing information is also difficult due to the dynamic topology of ad-hoc networks.

In mobile ad-hoc networks, nodes do not rely on any routing infrastructure but relay packets for each other. Thus communication in mobile ad-hoc networks functions properly only if the participating nodes cooperate in routing and forwarding. However, it may be advantageous for individual nodes not to cooperate, for example to save power or to launch security attacks such as denial-of-service. In this paper, we give an overview of potential vulnerabilities and security requirements of mobile ad-hoc networks, and proposed prevention, detection and reaction mechanisms to thwart attacks.

5.2 Classification of Techniques Used to Secure Ad-Hoc Networks

In order to provide solutions to the security issues involved in ad-hoc networks, we must elaborate on the two of the most commonly used approaches in use today

- Prevention
- Detection and Reaction

Prevention dictates solutions that are designed such that malicious nodes are thwarted from actively initiating attacks. Prevention mechanisms require encryption techniques to provide authentication, confidentiality, integrity and non-repudiation of routing information. Among the existing preventive approaches, some proposals use symmetric algorithms, some use asymmetric algorithms, while the others use one-way hashing, each having different trade-offs and goals. Prevention mechanisms, by themselves cannot ensure complete cooperation among nodes in the network. Detection on the other hand specifics solutions that attempt to identify clues of any malicious activity in the network and take punitive actions against such nodes.

The following broad classifications are:

1. Prevention using asymmetric cryptography
   -Using symmetric cryptography
   -Using one-way hash chains
2. Detection and Reaction

5.2.1 Prevention Using Asymmetric Cryptography

Asymmetric cryptographic techniques specify the underlined basic methodology of operation for protocols under this category. A secure wired networks or a similar network is required to distribute public keys or digital certificates in the ad-hoc network. Mathematically speaking a network with n nodes would require n public keys stored in the network. SAODV (an extension to AODV routing protocol) and ARAN are two of the protocols defined in this category.

Secure Ad-hoc On-demand Distance Vector Routing Protocol (SAODV)

SAODV adds security to the famous AODV protocol. Its basic functionality lies in securing the ADOV protocol by authenticating the non-mutable fields of the routing message using digital signatures.

It also provides an end-to-end authentication and node-to-node verification of these messages. The underlined process is relatively simple. The source node digitally signs the route request packet (RREQ) and broadcasts it to its neighbors. When an intermediate node receives a RREQ message, it first verifies the signature before creating or updating a reverse route to its predecessor. It then stores or updates the route only if the signature

is verified. A similar procedure is followed for the route reply packet (RREP). As an optimization, intermediate nodes can reply with RREP messages, if they have a “fresh enough” route to the destination. Since the intermediate node will have to digitally sign the RREP message as if it came from the destination, it uses the double signature extension described in this protocol.

The only mutable field in SAODV messages is the hop-count value. In order to prevent wormhole attacks this protocol computes a hash of the hop count field.

5.2.2 Prevention Using Symmetric Cryptography

Symmetric cryptographic techniques are used to avoid attacks on routing protocols. We assume that symmetric keys are pre-negotiated via a secured wired connection. Taking a mathematical approach we see that a network with ‘n’ nodes would require n * (n + 1) / 2 pair wise keys stored in the network. SAR and SRP are the two protocols that belong to this category.

Security-Aware Ad hoc Routing (SAR)

SAR is an attempt to use traditional shared symmetric key encryption in order to provide a higher level of security in ad-hoc networks. SAR can basically extend any of the current ad-hoc routing protocols without any major issues.

The SAR protocol makes use of trust levels (security attributes assigned to nodes) to make informed, secure routing decision. Although current routing protocols discover the shortest path between two nodes, SAR can discover a path with desired security attributes (E.g. a path through nodes with a particular shared key). The different trust levels are implemented using shared symmetric keys. In order for a node to forward or receive a packet it first has to decrypt it and therefore it needs the required key. Any nodes not on the requested trust level will not have the key and cannot forward or read the packets every node sending a packet decides what trust level to use for the transfer and thereby decides the trust level required by every node that will forward the packet to its final destination.

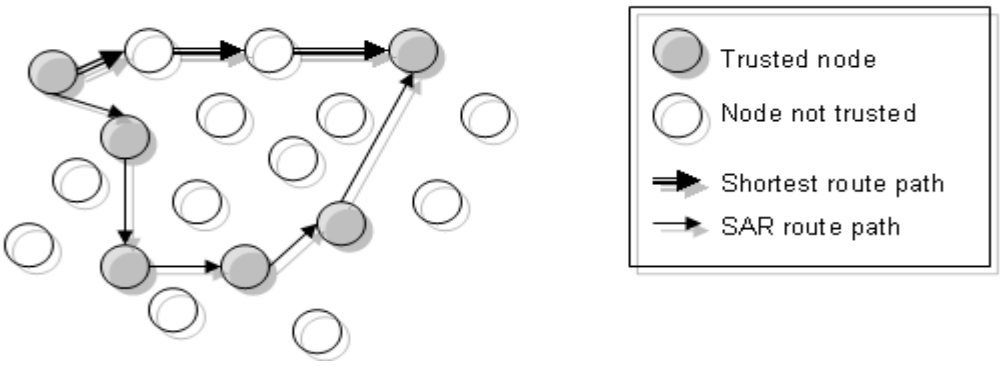


Figure 5.1 Variation of shortest path route selection between SAR and other routing algorithms [23]

SAR is indeed secure in the way that it does ensure that only nodes having the required trust level will read and reroute the packets being sent. Unfortunately, SAR still leaves a lot of security issues uncovered and still open for attacks such as:

- Nothing is done to prevent intervention of a possibly malicious node from being used for routing, as long as they have the required key
- If a malicious node somehow retrieves the required key the protocol has no further security measure to prevent against the attacker from bringing the entire network to a standstill.
- There is excessive encryption and decryption required at each hop. Since we are dealing with mobile environments the extra processing leading to increased power consumption can be a problem.

SAR is intended for the managed-open environment as it requires some sort of key distribution system in order to distribute the trust level keys to the correct devices.

5.2.3 Prevention Using One-way Hash Chains

This category defines a one-way hash chain to prevent attacks on routing protocols. They protect modification of routing information such as metric, sequence number and source route. SEAD and Ariadne fall into this category.

Ariadne

The ARIADNE protocol relies only on highly efficient symmetric cryptography. The protocol primarily discusses the use of a broadcast authentication protocol namely TESLA, because of its efficiency and requires low synchronization time rather than the

high key setup overhead of using pair-wise shared keys. Other authentication protocols such as BiBa are / can also be used for this purpose.

This proposal is an on-demand routing protocol. The design of Ariadne can be viewed as a 3 step process:

- **Authentication of RREQ by target:** To convince the target of the legitimacy of each field in a RREQ, the initiator includes a MAC computed with a shared key over a timestamp.
- **Mechanisms for authenticating data in RREQ and RREP**: The scheme allows the initiator to authenticate each individual node in the node list of the RREP. The target can authenticate each node in the node list of the RREQ, so that it will return RREP only along paths that contain legitimate nodes. 3 alternative techniques are available to achieve the node list authentication. These are the TESLA protocol, Digital Signatures and standard MAC. Out of these TESLA is the most widely used due to its inexpensive requirements.

  The working of TESLA is very straightforward. Whenever an intermediate node receives a RREQ message, it appends a MAC into the message, the key for which is released in a future time set by the source. The target buffers the RREP until intermediates nodes can release the corresponding TESLA keys. The TESLA security condition is verified at the target, and the target includes a MAC in the reply to certify that the security condition was met.

- **Per-hop hashing technique:** A one-way hash function is used to avoid a node from being removed from the node list in the RREQ message. The source initializes the hash chain to a MAC with a key shared between the source and target. When an intermediate node receives the request, it appends its identifier to the hash chain and rehashes it. The target verifies each hop of the path by comparing the received hash and the computed hash of the MAC. To change or remove a previous hop, the attacker must be able to invert the one-way hash function, which has been proved computationally infeasible

The failing of this protocol, similar to that seen in the SAODV, is that although hashing the hop-count value prevents malicious nodes in advertising shorter routes, it does not

prevent nodes from advertising longer routes. Also it can be seen that since this idea is based on a routing protocol with periodic updates, it has a high overhead. Thus it is not suitable to be deployed in resource-constrained mobile ad hoc networks.

Since Ariadne assumes clock synchronization between participating nodes, thus there exists a high complexity in obtaining such precise clock synchronization.

5.2.4 Detection and Reaction

Detection on the other hand specifics solutions that attempt to identify clues of any malicious activity in the network and take punitive actions against such nodes. All protocols in this category are designed such that they are able to detect malicious activates and react to the threat as needed. Byzantine, CONFIDANT, DSR, CORE and a protocol that uses Watchdog and Pathrater are the few protocols specified in this category.

Confidant

Confidant attempts to detect and isolate misbehaving nodes (or nodes with grudges) in an ad-hoc network, thus making it unattractive to deny cooperation and participation. Trust relationships and routing decisions are made based on experienced, observed, or reported routing and forwarding behavior of other nodes. The protocol has been described using Dynamic Source Routing (DSR) in the network layer.

Each node consists of 4 basic components:

1. The Monitor: watches its neighbors for any malicious behavior. If such behavior is detected, the reputation system is invoked.
2. The Reputation System: manages a table consisting of entries for each node and its ratings. Ratings are changed according to a rate function that assigns different weights to the type of behavior detected.
3. The Trust Manager: responsible for calculating trust levels of nodes and dealing with all incoming and outgoing alarm messages.
4. The Path Manager: manages all path information, i.e. adds, deletes or updates paths according to the feedback it receives from the reputation system

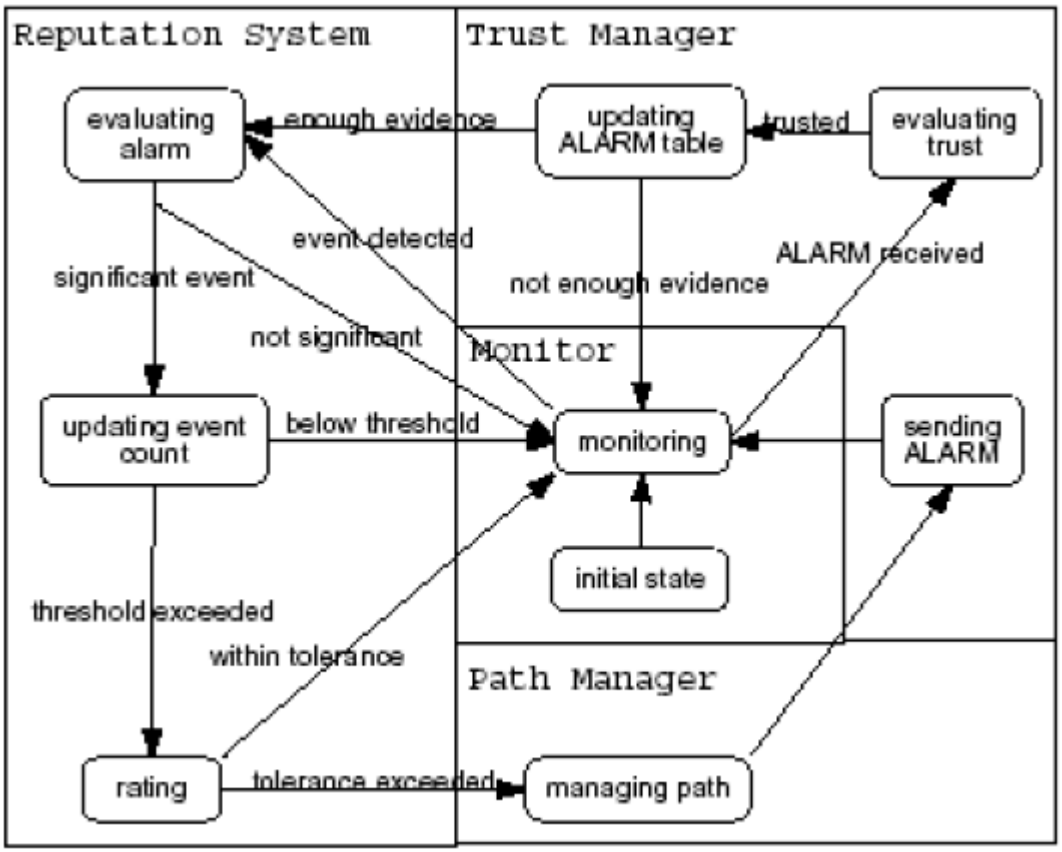


Figure 5.5 Trust Architecture and FMS within each Node of a Confidant

6 DSR Trust Model

In chapter 4 the four protocols DSDV, OLSR, DSR and AODV were simulated. None of the routing protocols described in chapter 4 handle security issues. All assumes that no nodes are malicious and are willing to participate in the routing protocol.

The results of the evaluations indicated that DSR and AODV both delivered a high percentage of the send packets. One of the main differences between DSR and AODV is that DSR uses source routing and stores the entire routes, where AODV only stores information about the next hop on the route and used hop-by-hop routing. It would be quite a challenge to apply trust based routing to AODV since the only information that is available to build some sort of trust based decision on, is the next node on the route.

Since DSR is a source route protocol and therefore stores the entire routes it is possible to make a much better trust based evaluation of the route. So it have been decided to develop security trust model for DSR to make it more secure.

6.1 Trust

Even though trust is widely used in our daily life, and by many people it is an extremely complex subject to work with. Many trust-based decisions are made on a subconscious level, and it is often difficult for people to determine why and if they trust one person and not another. Furthermore, one person's reasons for trusting somebody might differ from other persons. One of the reasons for its complexity is that it is difficult to define exactly what trust is. This is reflected by the many different definitions that exist in literature. Trust is also difficult to measure, which adds to the complexity of the subject. Further it seems that trust in some way is related to the risk that is associated with a given situation or action.

Definition of trust comes from Diego Gambetta who has gathered thoughts from diverse areas such as economics and biology. In his work from 1990 he gives the following definition of trust:

> *Trust (or, symmetrical distrust) is a particular level of subjective probability with which an agent assesses that another agent or group of agents will perform a particular action, both before he can monitor it (or independently of his capacity ever to be able to monitor it) and in a context in which it affects his own action.*

Gambetta expresses trust as a probability, which means that it can be given a value in the range from 0 to 1. The definition indicates that the ability to monitor whether the trusted action is performed is of importance.

6.2 The Proposed Trust Model

It have been analyzed the DSR protocol in chapter 4 through simulation where DSR is more reliable among all protocols. There are several vulnerabilities in DSR that can be exploited by malicious nodes. In order to fortify the DSR protocol it is necessary to find ways to estimate whether a node is malicious or not.

To handle these vulnerabilities we have been proposed a trust model. It has been decided to design a solution that will implement the following:

- Nodes will store trust values of each encountered node that express the nodes trust. These values will be adjusted based on the experiences that a node has with other nodes.
- In order to require acknowledgements for received packages an extension to the existing DSR header will be implemented.
- When nodes receive acknowledgements or data packets they will update trust values for the nodes on the route, based on some trust updating policy. Nodes that are encountered for the first time will have an initial trust value assigned based on some trust formation strategy.
- If a requested acknowledgement is not received within some timeframe the nodes on the used route should have their trust values decreased.
- Route selection will be based on some strategy that uses the trust in the nodes on the route to conduct an evaluation of the entire routes trust value.
- The extension will only seek to deal with malicious behavior that express it self as nodes dropping packets that they were supposed to forward.

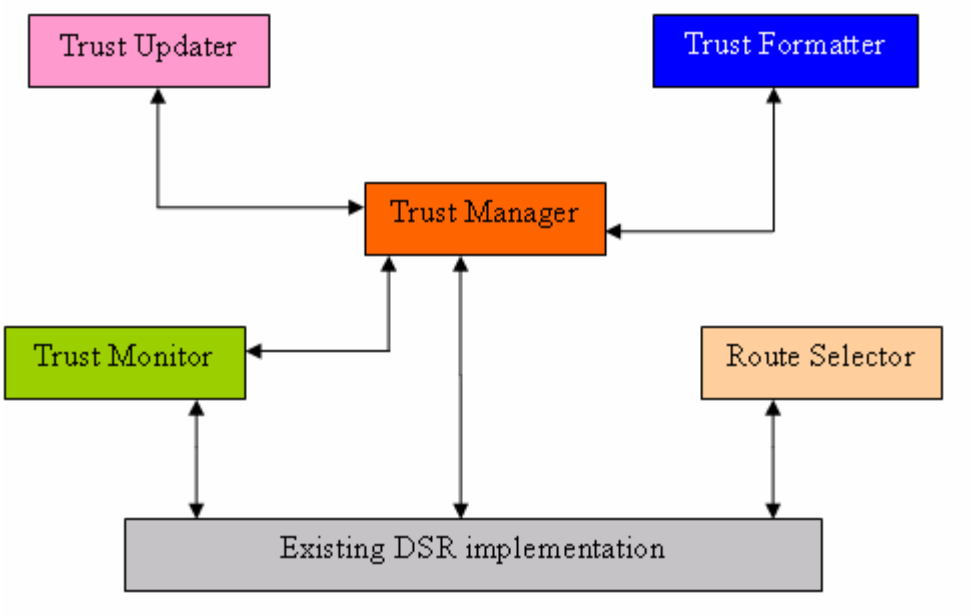


Figure 6.1 Design for Modified DSR Implementation

The above diagram is the trust model that has been proposed.

6.2.1 Trust Formatter

When new mobile nodes encountered in the network, the trust formatter component implements methods to assign trust values to these nodes. An initial trust value will be assigned to the new nodes when the first route is discovered because all nodes on the route will be unknown. The value of this parameter is quite important because it determines how close the node is to achieve maximal trust. It would be best to assign a low valued trust value in an environment with many malicious nodes. If a route contains known nodes, the trust value of these nodes is used to base the assignment of the initial trust value.

6.2.2 The Trust Updater

The trust updater module implements the functions for updating trust. The trust value depends on a given nodes experience in a given situation. This means that it is not reasonable to construct a general method for updating trust values that will be applicable to all applications in all domains. The function designed here is aims to function in domains with several malicious nodes.

A function for updating trust can depend on several parameters.

- Previous trust values.
- Lowest and Highest trust value ever assigned.
- Nr of positive and negative experiences in the past.
- The situation and value of an experience.

We have used the following trust updating function in the model implementation:

Fd (Ev, TV) = d*TV + (1-d) *Ev

TV: The existing trust value

Ev: The experience

d: A constant used to express the inflation of trust

Based on the observation there proposes the intervals [-1, 1], that interval is used for both experiences and trust values. The experience set will consist of three experiences that correspond to: Acknowledgement received ok, Acknowledgement timed out and Data packet received. An acknowledgement indicates that all nodes on the route could be trusted and therefore a value of 1 (maximum trust) will be assigned to this experience. The opposite counts when an acknowledgement is not received within the timeframe. This means that it must be assumed that the packet never reached its destination. Since the cause cannot be determined a value of –1 is assigned to this experience. The final experience is receiving a data packet, which means that nodes on the route have forwarded the package. This is not considered as powerful as an acknowledgement, because an acknowledgement is a response and confirmation of the trust that a node put in other nodes, where the receipt of data packet can be seen as a recommendation from the source to the destination. Therefore this experience will be given a value of 0.7.

The graph illustrates that a high d value will lead to a faster trust evolution towards maximum (or minimum) trust value. Only full positive experiences with a value of 1 and pure negative experiences with a value of -1 are used and the initial trust is set to 0.5. With a d value of -0.5 maximum trust will be placed in node after only a few experiences. This corresponds to trust evolving as balanced fast. With a d value of -0.9 trust will evolve as balanced slow. It is also possible to use one value for d when for positive experiences and another for negative experiences and thereby let trust evolve faster in one direction than the other. Based on the fact that a good node can also drop packages unintended d will be given the value -0.9. This means that good nodes that accidentally

drop packages will not loose trust to fast. At the same time it means that a node with a high trust value that causes a negative experience will have its trust value lowered in a perceptible way.

6.2.3 Route Selection

The main task of the route selection component is to evaluate routes based on the trust value of the nodes that constitute the route and selects a route based on this evaluation. The routes are evaluated and the route with the highest rating should be used. This means that the best route will be considered as the one that has the highest trust rating. A good route is considered to be a route that does not contain malicious nodes. To decide whether one route that results in a packet being delivered is better than another that achieves the same is difficult. Here metrics such as latency could be used. It can be concluded that a route that contains a malicious node is not good because it will always result in a packet drop. As the coming discussions of route strategies will illustrate, determining the best route can actually lead to a good route being discarded and a route containing a malicious node being chosen. Defining a route selection strategy is not an unambiguous task. Nodes are grouped as one because they are on the same route, but actually they do not have anything in common that, from a sociological point of view, can substantiate the grouping. This makes it quite difficult to argue for one routing strategy over another. For all routing strategies is that must not take the destination of the packet into account when the rating of the route is calculated, because the destination might be identified as a malicious node and therefore have a low trust value. This is necessary because the traffic is generated randomly for the simulations, and therefore malicious nodes will also be the destination of packets. Whether or not a node would actually send packets to a node that was identified as malicious is not treated here.

Furthermore, all strategies return a maximum rating if the route only consists of two nodes, since that means that the destination is a neighbor. If a maximum rating is returned, the route is used without examining further routes. This is actually a performance improvement compared to the implemented strategy used by standard DSR in Ns-2, where all routes to a destination is examined even though the destination is a neighbor.

The designed routing strategies are basic strategies that can be considered archetypes from which more complex and sophisticated strategies can be derived. It is chosen to design simple strategies, because it will make it easier to determine the effect and difference between the strategies.

Route Selection Strategy 1

The first route selection strategy will be to return the average of all nodes on the route. Using the average presents the issue, illustrated in Table 5-1, that routes containing nodes with very low trust values might still be rated high.

Table 6.1 Route Selection Strategy 1

Route	Trust values				Ratings
	Node 1	Node 2	Node 3	Node 4	
1	0.3	0.8	0.5	0.2	0.45
2	1	1	-1	1	0.75

Table 6.1 illustrates an example where two routes are evaluated by route selection Strategy 1. Route consists of four nodes with values between 0.7 and 0.5. With an initial trust value below 0.5 this would mean that the node evaluating the route has had positive experiences with all nodes. Route 2 consists of four nodes where three have maximum trust value and one has minimum trust value. That the node has minimum value indicates that it is a malicious node. However, as the Rating column shows route 2 is rated higher than route 1. The example illustrates an extreme case and it is difficult to predict how often such a situation occurs.

Table 6.2 Route Selection Strategy 2

Experience	Experience value	Trust value	Avg of the experiences
1	1	0.55	1
2	1	0.60	1
3	1	0.64	1
4	-1	0.47	0.5
5	-1	0.32	0.2

Route Selection Strategy 2

The second scenario evaluates the nodes based on the average value of the past experiences. Only 5 past experiences are remembered by this strategy, to calculate the average value of experiences. Nodes with a high trust value that suddenly starts to drop packets will be identified faster than by using the trust values.

6.2.4 Trust Management

The trust manager module stores trust information about all known nodes during run time, and offers methods to query for information about stored trust values. It also functions as the main interface between the existing implementation of the DSR protocol and the trust updater and trust formatter module. In a real life scenario it is likely that nodes will move about in the same environment for some time. This means that the same nodes can be encountered on a regular basis. For this reason the trust management module implements IO methods for storing trust values in a persistent way so they can be loaded again.

6.2.5 Acknowledgement Monitoring

As described in section 6.2 it is necessary to use an acknowledgement mechanism to base trust updating on. In general acknowledgement leads to a packet overhead that of course should be minimized. One way to minimize the packet overhead is to limit the amount of acknowledgements send, by using a sliding window mechanism. To keep it simple it is decided not to include a sliding window mechanism in the trust model but instead require acknowledgements for all data packets and not for protocol packets.

The purpose of the acknowledgement mechanism is to use received acknowledgements or lack of acknowledgements to adjust trust values and not, as known for many other protocols, to base decisions on retransmission on. Since the trust values are used to base routing decisions on, and because a node can be part of many routes it is important that a missing acknowledgement is detected fast.

A short time frame might cause routes that where simply slow, but forwarded data packets and acknowledgements, to be rated low. On the other hand a large timeframe might result in a bad route being used several times before it has its trust values decreased.

An acknowledgement id is stored when the packet is send, and if an acknowledgement is received within the time frame nodes on the stored route have their trust values updated. If a requested acknowledgement is not received within some time frame, the packet is considered dropped. In this case the nodes trust values should be adjusted in a negative way. The time it takes for a packet to reach its destination will depend on the length of the routes. The number of links a packet will pass is one less than the number of nodes on the route. Since the return route of the acknowledgement will depend on the destinations route selection it is unknown how many nodes it will include and therefore the double of the length of the outgoing route is used as the best estimate.

The following formula for estimating the total time that a node will wait for an acknowledgement will be used.

$$TO = (2L-2)*tc$$

Where

TO = Total time out value,

tc = Time out constant

The time the packet will spend on the actual physical wire is considered small compared to the time it will take for a node to process it and therefore these two times has been combined to one timeout constant. Since the node will not be in a state where it is waiting to receive the acknowledgement before it can continue, it is expected that a relatively high timeout value can be accepted.

APPENDIX

Tcl Code for Adhoc Wireless Protocols

```
#===================================================================
#   Define options
#===================================================================
set val(chan)       Channel/WirelessChannel
set val(prop)       Propagation/TwoRayGround
set val(netif)      Phy/WirelessPhy
set val(mac)        Mac/802_11
set val(ifq)        CMUPriQueue
set val(ll)         LL
set val(ant)        Antenna/OmniAntenna
set val(x)              500          ;# X dimension of the topography
set val(y)              500          ;# Y dimension of the topography
set val(ifqlen)          50           ;# max packet in ifq
set val(seed)            1.0          ;# random seed
set val(adhocRouting) (DSR, DSDV,OLSR, AODV)   ;# routing protocol
set val(nn)              10           ;# how many nodes are simulated
set val(cp)             "cbr10_5"       ;# traffic model file
set val(sc)           "scen10_1p" ;#mobilityfile
set val(stop)            200.0        ;# simulation time

#===================================================================
# Main Program
#===================================================================
# create simulator instance
set ns_              [new Simulator]
# setup topography object
set topo [new Topography]
# create trace object for ns and nam
set tracefd         [open out_$val(adhocRouting).tr w]
$ns_ trace-all $tracefd
set namtrace    [open out.nam w]
$ns_ namtrace-all-wireless $namtrace $val(x) $val(y)
# define topology
$topo load_flatgrid $val(x) $val(y)
# Create God
set god_ [create-god $val(nn)]
# define how node should be created
set chan_1_ [new $val(chan)]
#global node setting
$ns_ node-config -adhocRouting $val(adhocRouting) \
                 -llType $val(ll) \
                 -macType $val(mac) \
                 -ifqType $val(ifq) \
                 -ifqLen $val(ifqlen) \
```

```
                -antType $val(ant) \
                -propType $val(prop) \
                -phyType $val(netif) \
                -channel $chan_1_ \
                        -topoInstance $topo \
                        -agentTrace ON \
                -routerTrace ON \
                -macTrace ON \
                -movementTrace ON
#  Create the specified number of nodes [$val(nn)]
for {set i 0} {$i < $val(nn) } {incr i} {
            set node_($i) [$ns_ node]
            $node_($i) random-motion 0 ;# disable random motion
}
# Define node movement model
puts "Loading connection pattern..."
source $val(cp)
# Define traffic model
puts "Loading scenario file..."
source $val(sc)
# Define node initial position in nam
for {set i 0} {$i < $val(nn)} {incr i} {
  $ns_ initial_node_pos $node_($i) 20
}
# Tell nodes when the simulation ends
for {set i 0} {$i < $val(nn) } {incr i} {
    $ns_ at $val(stop).0 "$node_($i) reset";
}
# stop procedure which generates the nam and graph files
proc stop {} {
  global ns_ tracefd namtrace val
  $ns_ flush-trace
  close $tracefd
  close $namtrace
  if {$val(adhocRouting) == "AODV"} {
    exec awk -f tput.awk out_AODV.tr > aodv_rcv.xgr
    exec awk -f drp.awk out_AODV.tr > aodv_drp.xgr
    exec awk -f rtr_aodv.awk out_AODV.tr > aodv_rtr.xgr
    if {[string match "*_5*" $val(sc)]} {
      exec awk -f pdf_30p.awk out_AODV.tr >> aodv_pdf.xgr
      exec awk -f nrl_30p_aodv.awk out_AODV.tr >> aodv_nrl.xgr
    } elseif {[string match "*_1*" $val(sc)]} {
      exec awk -f pdf_10p.awk out_AODV.tr >> aodv_pdf.xgr
      exec awk -f nrl_10p_aodv.awk out_AODV.tr >> aodv_nrl.xgr
    } elseif {[string match "*_2*" $val(sc)]} {
      exec awk -f pdf_20p.awk out_AODV.tr > aodv_pdf.xgr
      exec awk -f nrl_20p_aodv.awk out_AODV.tr > aodv_nrl.xgr
    }
  } elseif {$val(adhocRouting) == "OLSR"} {
    exec awk -f tput.awk out_OLSR.tr > olsr_rcv.xgr
    exec awk -f drp.awk out_OLSR.tr > olsr_drp.xgr
```

```
    exec awk -f rtr_olsr.awk out_OLSR.tr > olsr_rtr.xgr
    if {[string match "*_5*" $val(sc)]} {
      exec awk -f pdf_30p.awk out_OLSR.tr >> olsr_pdf.xgr
      exec awk -f nrl_30p_olsr.awk out_OLSR.tr >> olsr_nrl.xgr
    } elseif {[string match "*_2*" $val(sc)]} {
      exec awk -f pdf_20p.awk out_OLSR.tr >> olsr_pdf.xgr
      exec awk -f nrl_20p_olsr.awk out_OLSR.tr >> olsr_nrl.xgr
    } elseif {[string match "*_1*" $val(sc)]} {
      exec awk -f pdf_10p.awk out_OLSR.tr >> olsr_pdf.xgr
      exec awk -f nrl_10p_olsr.awk out_OLSR.tr >> olsr_nrl.xgr
    }
  }  elseif {$val(adhocRouting) == "DSDV"} {
    exec awk -f tput.awk out_DSDV.tr > dsdv_rcv.xgr
    exec awk -f drp.awk out_DSDV.tr > dsdv_drp.xgr
    exec awk -f rtr_dsdv.awk out_DSDV.tr > dsdv_rtr.xgr
    if {[string match "*_5*" $val(sc)]} {
      exec awk -f pdf_30p.awk out_DSDV.tr >> dsdv_pdf.xgr
      exec awk -f nrl_30p_dsdv.awk out_DSDV.tr >> dsdv_nrl.xgr
    } elseif {[string match "*_2*" $val(sc)]} {
      exec awk -f pdf_20p.awk out_DSDV.tr >> dsdv_pdf.xgr
      exec awk -f nrl_20p_dsdv.awk out_DSDV.tr >> dsdv_nrl.xgr
    } elseif {[string match "*_1*" $val(sc)]} {
      exec awk -f pdf_10p.awk out_DSDV.tr >> dsdv_pdf.xgr
      exec awk -f nrl_10p_dsdv.awk out_DSDV.tr >> dsdv_nrl.xgr
    }
  } elseif {$val(adhocRouting) == "DSR"} {
    exec awk -f tput.awk out_DSR.tr > dsr_rcv.xgr
    exec awk -f drp.awk out_DSR.tr > dsr_drp.xgr
    exec awk -f rtr_dsr.awk out_DSR.tr > dsr_rtr.xgr
    if {[string match "*_5*" $val(sc)]} {
      exec awk -f pdf_30p.awk out_DSR.tr >> dsr_pdf.xgr
      exec awk -f nrl_30p_dsr.awk out_DSR.tr >> dsr_nrl.xgr
    } elseif {[string match "*_2*" $val(sc)]} {
      exec awk -f pdf_20p.awk out_DSR.tr >> dsr_pdf.xgr
      exec awk -f nrl_20p_dsr.awk out_DSR.tr >> dsr_nrl.xgr
    } elseif {[string match "*_1*" $val(sc)]} {
      exec awk -f pdf_10p.awk out_DSR.tr >> dsr_pdf.xgr
      exec awk -f nrl_10p_dsr.awk out_DSR.tr >> dsr_nrl.xgr
    }
  }
  exec  nam out.nam &
}
$ns_ at  $val(stop) "stop"
$ns_ at  $val(stop).0002 "puts \"NS EXITING...\" ; $ns_ halt"
puts $tracefd "M nn $val(nn) x $val(x) y $val(y) rp $val(adhocRouting)"
puts $tracefd "M sc $val(sc) cp $val(cp) seed $val(seed)"
puts $tracefd "M Prop $val(prop) Ant $val(ant)"
puts "Starting Simulation..."
$ns_ run
```

References

[1] Elizabeth M. Royer and C-K Toh "A *Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks"IEEE* 2003.

[2] Adrian Pullin "*A Realistic Model for the Evaluation of MANET* " IEEE International Conference on Communication, ICC 98, 1998. Volume: 1.

[3] Anne Aaron and Jie Weng "*Performance Comparison of Ad-hoc Routing Protocols for Networks with Node Energy Constraints"* Class Project Spring 2000-2001.

[4] Hong Jiang and J. J. Garcia-Luna-Aceves "*Performance Comparison of Three Routing Protocols for Ad Hoc Networks"* 48th IEEE Vehicular Technology Conference, VTC 98, 1998.

[5] Sampo Naski Helsinki "*Performance of Ad Hoc Routing Protocols: Characteristics and Comparison "* IEEE Journal on Selected Areas in Communication, Volume: 17, Issue: 8, Aug. 1999.

[6] M. Gerla, K. Xu, X. Hong, "*Exploiting Mobility in Large Scale Ad Hoc Wireless Networks*", 2002.

[7] J. Wu, "*Extended Dominating-Set-Based Routing in Ad Hoc Wireless Networks with Unidirectional Links*", IEEE 2002.

[8] F. De Rango, A. Iera , A. Molinaro, S. Marano, "*A Modified Loation-aid3ed routing protocol for the Reduction of Control Overhead in Ad Hoc Networks*", 10th International Conference on Telecommunications, ICT 2003, 2003. Volume 2.

[9] S. Murthy and J. J. Garcia-Luna-Aceves, " *Loop-Free Internet Routing Using Hierarchical Routing Trees*," Proceedings of INFOCOM '97, April 7-11, 1997.

[10] Dr. Wenye WangFinal Amit Singh, Parth Pathak, Rikin Gandhi, Sidhharth Bhai,Srinivas and Annambhotla. "*Comparison of Routing Protocols in Mobile Ad-hoc Networks"* ECE 575 Introduction to Wireless Networking Project Report May 3, 2006

[11] Thomas Staub , Ad-hoc and Hybrid Networks Performance Comparison of MANET Routing Protocols in Ad-hoc and Hybrid Networks .

[12] S. Ahmed and M. S. Alam *, "Performance Evaluation of Important Ad Hoc Network Protocols"* USA Received 15 July 2005; Accepted 12 December 2005.

[13] T. Camp, J. Boleng, B. Williams, L. Wilcox, W. Navidi, "*Performance Comparision of Two Location Based Routing Protocols for Ad Hoc Networks*", Proc. of INFOCOM, 2002.

[14] Megha Sharma , "*Comparison of Routing Protocols for Ad-hoc networks"* 2005.

[15] Jianping Wang, "ns-2 Tutorial (2) by Multimedia Networking Group", 2002.

[16] L. Ji and M. S. Corson, "*A Lightweight Adaptive Multicast Algorithm,"* Proceedings of GLOBECOM '98, pp. 1036{1042, November 1998.

[17] T. Clausen and P. Jacquet, "Optimized Link State Routing Protocol (OLSR)", IETF: The Internet Engineering Taskforce RFC 3626, Oct 2003, last accessed on Jan 2006.

[18] J. Liebeherr and N. Christin. Rate allocation and buffer management for differentiated services. *Computer Networks*, 40(1):89–110, September 2002.

.

[19] Lennart Conrad, M.Sc. "*Thesis Secure Routing in Mobile Ad Hoc Networks"* 2003.

[20] Per Johansson, Tony Larsson, Nicklas Hedman, and Bartosz Mielczarek. "*Routing protocols for mobile ad-hoc networks – a comparative performance analysis*". In *Proceedings of the 5th International Conference on Mobile Computing and Networking (ACM MOBICOM'99)*, August 1999. to appear.

[21] C. E. Perkins, E. M. Belding-Royer, and S. R. Das. "*Ad hoc on-demand distance vector (AODV) routing*". Rfc 3561, IETF, July 2003.

[22] E. M. Belding-Royer and C. E. Perkins. "*Evolution and future directions of the ad hoc on-demand distance vector routing protocol*". *Ad hoc Networks Journal*, 1(1):125–150, July 2003.

[23] J.-P. Hubaux, L. Buttyan, and S. Capkun, "*The quest for security in mobile ad hoc networks,*" in *MobiHoc '01: Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*, New York, NY, USA, 2001, pp. 146–155.

[24] D. Johnson and D. Maltz." *Dynamic source routing in ad hoc wirelessnetworks* ". In T. Imielinski and H. Korth, editors, *Mobile computing*, chapter 5. Kluwer Academic, 1996.

[25] www.isi.edu/nsnam/ns/tutorial/

[26] www.cis.ohio-state.edu/~jain/cis788-99/adhoc_routing