ABBREVIATIONS

AODV	Ad-hoc On Demand Distance Vector
DSDV	Destination Sequenced Distance Vector
DSR	Dynamic Source Routing
MANET	Mobile Ad-Hoc Network
NS2	Network Simulator 2
NAM	Network Animator
Ns2GTFA	Ns2 GUI Trace File Analyzer
RREQ	Route Request
RREP	Route Reply
RERR	Route Error
TCL	Tool Command Language
IP	Internet Protocol
TCP	Transmission Control Protocol

ABSTRACT

In Ad-hoc Networks a collection of wireless mobile hosts form a temporary network without the use of any stand-alone infrastructure or centralized administration. Due to the mobile nature of nodes in the network, they are self-organizing and self-configuring. The nodes not only act as hosts but also as routers.

The primary goal of routing protocol in such networks is to meet the challenges of the dynamically changing topology. In this project we are going to simulate the AODV and DSR protocols and analyze their performance based on how they scale when the number of nodes are changed with the following metrics…

1. Throughput
2. End to End delay
3. Routing overhead
4. Packet delivery ratio

Keywords Ad-Hoc Networks, Wireless Networks, Routing Protocols, Simulation Performance Analysis

TABLE OF CONTENTS

CHAPTER 1 - INTRODUCTION

Wireless networks are composed of large number of nodes. Depending on the different applications of wireless networks they are either deployed manually or randomly. After being deployed either in a manual or random fashion, the nodes self-organize themselves and start communicating. There is an interesting unlimited potential in this wireless technology with various application areas along with crisis management, military, natural disaster, seismic sensing and many more. In general the two types of wireless sensor networks are: unstructured and structured. The structured wireless sensor networks are those in which the sensor nodes deployment is in a planned manner whereas unstructured wireless sensor networks are the one in which sensor nodes deployment is in an ad-hoc manner. As there is no fixed infrastructure between wireless networks for communication, routing becomes an issue in large number of nodes deployed along with other challenges of manufacturing, design and management of these networks. There are different protocols that have been proposed for these issues. The performance of the protocols is studied in this project by evaluation of two routing protocols with the help of some performance metrics considering applications demand as well.

1.1. MANET

A MANET(Mobile Ad-Hoc NETwork) is a group of wireless devices that are self organizing networks where nodes are connected by wireless links and can move freely and the topology of the network changes constantly. It is a distributed infrastructure-less wireless networks. It uses intermediate relay nodes may involve for routing between two nodes, called multi hop routing. The communicating devices are usually powered by battery and there is a big concern on minimizing power consumption. Due to the distributed nature of both networks self self-management is necessary. The topology of these networks are unpredictable and may change rapidly. They can setup at anytime and anywhere.

1.1.1. Characteristics of MANETs

1. Nodes can perform the roles of both hosts and routers.
2. Capable of Multi-hop routing.
3. Decentralized operation.
4. Dynamic Network Topology.
5. Constrained resources of mobile nodes.
6. Limited Physical Security.
7. Higher mobility and high user density.

1.1.2. Advantages of MANETs

1. They are decentralized and can work without a base station.
2. Cost estimation is very less.
3. MANET can be used as temporary networks and require less time.
4. Provide access to information and services regardless of geographic position.

1.1.3. Disadvantages of MANETs

1. Limited resources and physical security.
2. Intrinsic mutual trust vulnerable to attacks.
3. Inferior reliability, efficiency, stability and capacity of links.
4. Fluctuating link bandwidth of wireless links.
5. Lack of authorization facilities.
6. Volatile network topology makes it hard to detect malicious nodes.
7. Security protocols for wired networks cannot work for ad hoc networks.

1.1.4. Applications of MANETs

1. Personal area networks.
2. Military or police exercises.
3. Disaster relief operations.
4. Communication in mining site operations.

1.2. ROUTING

The main challenge among the mobile nodes is how to deliver the packets to the destination node. Routing is the process by which mobile nodes exchange information from one node to another. Nodes that lie within each other's send range can communicate directly and are responsible for dynamically discovering each other. In order to enable Communication between nodes that are not directly within each other's send range, intermediate nodes act as routers that relay packets generated by other nodes to their destination.

1.2.1. Routing Protocols

There are many routing protocol that are used to communicate among mobile nodes. Figure 1.2.1 shows how the routing protocols are classified based on topology, location and energy awareness. Here we are focusing on topology based classification of protocols. Mesh topology is widely used in wireless networks as each wireless transmitter can directly communicate with the wireless receiver or any other host in the network. Ad-Hoc wireless network is implemented in this way.

The topology based routing protocols are further classified into flat routing and hierarchy based routing protocols which is shown in figure 1.2.2.

Flat routing protocols assign the same role to all the participating nodes whereas in Hierarchical routing different roles are assigned to the network nodes. Flat routing is further classified into proactive and reactive routing protocols.

- Proactive protocol maintains an up-to-date routing table which contains the routing information for every node in the network by periodically sending control messages between the nodes. In Reactive protocols, routes are discovered only when a source wants to send some packets to the destination.

- The hybrid protocol used the advantages of both proactive and reactive routing protocols and was made to overpower their weaknesses. Routing is first done through proactive protocols which then activate few nodes which do reactive flooding. We are further narrowing down into reactive protocols namely AODV and DSR for this project.

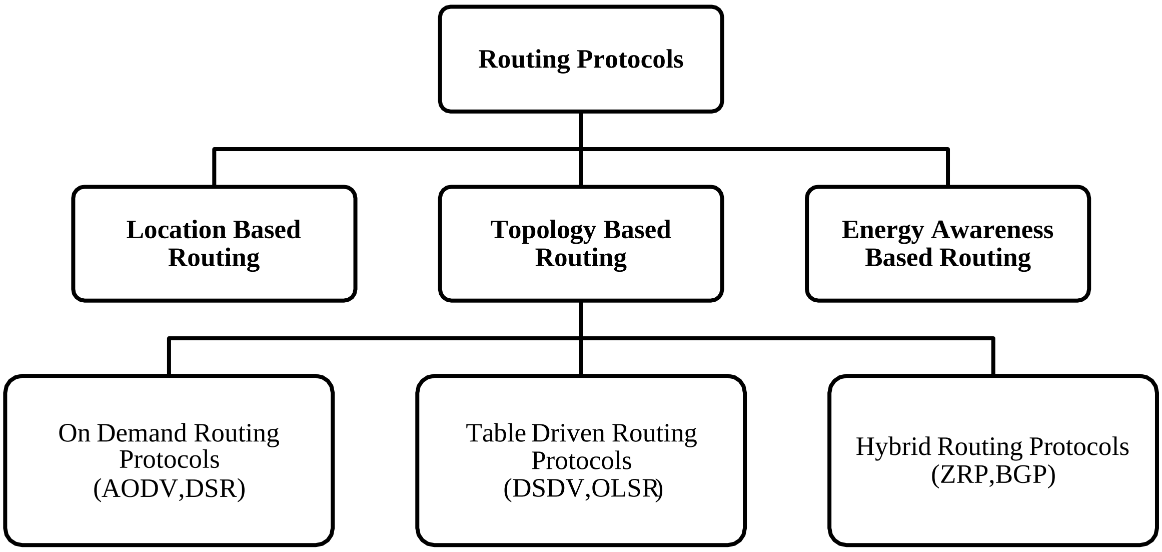


Fig 1.2.1 Classification of Routing protocols in MANETs

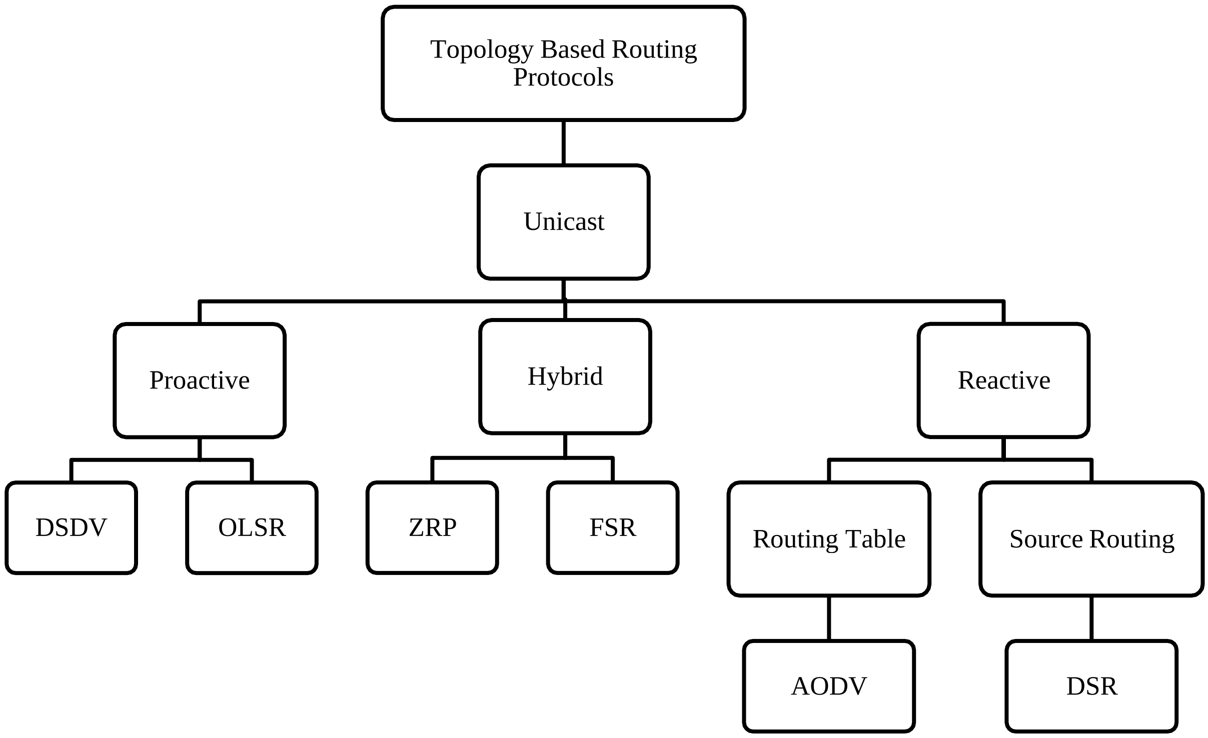


Fig 1.2.2 Further Classification of topology based routing protocols in MANETs

1.2.2. Reactive routing protocol

Reactive routing protocol is also termed as on demand routing protocol. In this protocol, route is found at whatever point it is required. Route discovery fully rely on the nodes and it is used as per requirement. Source node checks its route cache for the possible routes from the source node to the destination node. If no route is determined, it starts route discovery process. Examples are AODV, LMR, DSR and TORA. It has two major components:

1. **Route discovery:** During this phase the source initiates the process of route discovery only when there is a demand. The source node examines its route cache to authenticates which routes are available from source to the destination. If no route is determined, it starts a route discovery process. The packet sent by the source consists of destination node address and the address of the intermediate nodes to the destination.

2. **Route maintenance:** As the network has a dynamic topology, sometimes there is a condition of route failure due to the link breakage. So for that route maintenance needs to be done. These protocols consists of acknowledgement mechanism which helps in route maintenance. But because of this mechanism latency gets added in the network. Every node that is involved in the route maintenance mechanism adds latency in the network. So they are generally preferred in the situations where low there is a requirement of low routing overhead. It helps to achieve stability in the network and to reduce the excessive overhead required in discovering new route.

1.3. DYNAMIC SOURCE ROUTING (DSR) PROTOCOL

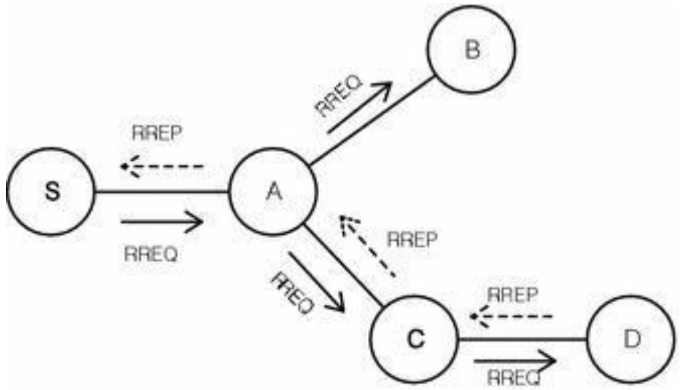


Fig. 1.3.1. DSR Protocol

DSR is a reactive type protocol and working methodology is source route method. It works on the principle of the link state algorithm where the source initiates the route discovery only when there is a demand. DSR does not rely on routing table unlike AODV but instead uses source routing at each intermediate device. The DSR is primarily designed to limit the usage of bandwidth which is consumed by control packets present in Ad hoc networks. During the phase of route construction, it establishes a route by broadcasting route request packet in the network. The destination node upon receiving the respective route request packet sends a response in the form of Route Reply (RREP) packet to the source node carrying the route travelled by Route Request (RREQ) Packet received. The significance of DSR mainly in reactive approach is that it eliminates the necessity of periodical broadcasting in the network which is predominant in table driven approach.

The algorithm for route discovery and route maintenance is given below as follows. First the sender will establish the route from source to destination and it also keeps a note of the address of the intermediate nodes and will save these details to the route record of the packet. This protocol was mainly designed for the multi node network having small diameter.

Considering the route discovery module, the process starts when source node (S) has to send a message to destination node (D) in a specified route. S floods the network with the RREQ, when the adjacent nodes receive the RREQ it checks whether the request is repeated, if repeated then, the adjacent node ignores the request. If the RREQ is not repeated, then it sends an RREP to S (In this case the adjacent node acts as a destination node - D) having the route information. When S receives the RREP it consecutively sends the message in the route specified by RREP. Considering the process of route maintenance, if the link between S and D is disconnected, the respective node sends an error message RERR to S. S receives RRER and consequently deletes all the routes with this invalid link and starts the route discovery process all over again. It is observed that the route discovery process discovers many routes from S to D, but the route chosen would not necessarily be with a minimum hop, hence more energy is consumed during data transmission. Route caching is carried out for two purposes:

1. A cached route is available to the demanding node to reducing the routing latency significantly.
2. Route caching avoids route discovery process for reduces the control.

1.3.1. Advantages of DSR Protocol

- DSR uses a reactive approach. This eliminates the need to periodically flood the network with table update messages which are required in a table-driven approach.
- The intermediate nodes also utilize the route cache information efficiently to reduce the control overhead.

1.3.2. Disadvantages of DSR Protocol

- The route maintenance mechanism does not locally repair a broken down link.
- The connection setup delay is higher than in table-driven protocols.
- The performance degrades rapidly with increasing mobility.
- Considerable routing overhead is involved due to the source-routing mechanism employed in DSR. This routing overhead is directly proportional to the path length.

1.4. AD-HOC ON DEMAND DISTANCE VECTOR (AODV) PROTOCOL

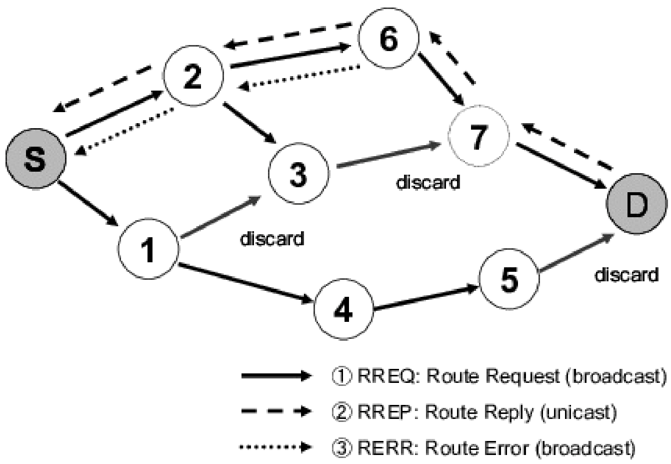


Fig.1.4.1 AODV Protocol

Ad Hoc On-Demand Distance Vector Routing Protocol is working on the principle of fulfilling the demand on order. This will reduce the route broadcast messages as it will provide route on request. When source code requires transferring the packet to the destination it will simply broadcast the route-finding message to its neighbor. Once neighbor receives the broadcasted message it will do the same for transferring the packet to the destination until the packet does not reach the destination. Once the route is defined all nodes saves the route in their route table for acknowledgement to the source node. That node will then reply to the source node. Once source node will get reply from neighboring node, any route-reply after that will be removed by the source node. If network topology changes during the simulation, source node will copy the producer and again broadcast the route request message to carry on the communication in the network. If any link fault is detected, node of that link will simply send an error message to their neighbor node and the process of finding route will start again.

In AODV routing, when a source has data to transmit to a new destination, it broadcast a RREQ for that destination. A neighbor's node receiving the RREQ checks if it has not received the same request before using the ROUTE-ID. It is not the destination and does not have a current route to the destination, it rebroadcasts the RREQ and at same time backward route to the source is created. If the receiving node is the destination or has a current route to the destination, it generates a RREP. The RREP propagates; each intermediate node creates a route to the destination. When the source receives the RREP, it records the forward route to the destination and begins sending data. AODV uses the message types Route Request (RREQ), Route Replies (RREP) and Route Error (RERR) in finding the route from source to destination. AODV performs route discovery, route maintenance and route Caching.

In AODV routing, when a source has data to transmit to a new destination, it broadcast a RREQ for that destination. A neighbor's node receiving the RREQ checks if it has not received the same request before using the ROUTE-ID. It is not the destination and does not have a current route to the destination, it rebroadcasts the RREQ and at same time backward route to the source is created. If the receiving node is the destination or has a current route to the destination, it generates a RREP. The RREP propagates; each intermediate node creates a route to the destination. When the source receives the RREP, it records the forward route to the destination and begins sending data. Once the route is established, a route maintenance protocol provides feedback about the links of the route and to allow the route to be modified maintenance of the discovered / established route is necessary

1.4.1. Advantages of AODV Protocol

- Routes are established on demand and destination sequence numbers are used to find the latest route to the destination.
- Can handle highly dynamic MANETs.
- Less energy consumption since only the information of active nodes are stored in the nodes.
- It supports multicasting.
- Has a small routing table.

1.4.2. Disadvantages of AODV Protocol

- Lacks efficient route maintenance technique.
- Aggregate route is not possible at intermediary nodes.
- Inconsistent routes if the source sequence number is very old.
- Higher route discovery latency.
- The periodic beaconing leads to unnecessary bandwidth consumption.

CHAPTER 2 - SOURCE CODE

2.1. AODV Protocol (aodv.tcl)

```
#AD-HOC ON-DEMAND DISTANCE VECTOR (AODV) ROUTING PROTOCOL
Phy/WirelessPhy set freq_ 2.472e9
Phy/WirelessPhy set RXThresh_ 2.62861e-09; #100m radius
Phy/WirelessPhy set CSThresh_ [expr 0.9*[Phy/WirelessPhy set RXThresh_]]
Phy/WirelessPhy set bandwidth_ 11.0e6
Mac/802_11 set dataRate_ 11Mb
Mac/802_11 set basicRate_ 2Mb

#DEFINING OPTIONS
set val(chan) Channel/WirelessChannel ;         #CHANNEL TYPE
set val(prop) Propagation/TwoRayGround ;    #RADIO-PROPAGATION MODEL
set val(netif) Phy/WirelessPhy ;                #NETWORK INTERFACE TYPE
set val(mac) Mac/802_11 ;                       #MAC TYPE
set val(ifq) Queue/DropTail/PriQueue ;          #INTERFACE QUEUE TYPE
set val(ll) LL ;                                #LINK LAYER TYPE
set val(ant) Antenna/OmniAntenna ;              #ANTENNA MODEL
set val(ifqlen) 50 ;                            #MAXIMUM PACKET IN ifq

#GETTING THE NUMBER OF NODES TO BE GENERATED FROM THE USER
puts "Enter the nodes to generate"
gets stdin MEE
puts "You entered the number of nodes as $MEE"
set val(nn) $MEE ;

#SETTING THE AODV PROTOCOL
set val(rp) AODV;

#BY CHANGING THE VALUE OF X AND Y, WE CAN GENERATE A DIFFERENT
DISTANCE RANGE FOR THE NODE MOVEMENTS.
set val(x) 500 ;          #X-DIMENSION OF TOPOGRAPHY
set val(y) 500 ;          #Y-DIMENSION OF TOPOGRAPHY
set val(stop) 50 ;        #TIME OF SIMULATION

#CREATING AN INSTANCE OF THE SIMULATOR
set ns [new Simulator]
set tracefd [open AODV.tr w]
set winFile [open AODV w]
set namtracefd [open AODV.nam w]
$ns trace-all $tracefd
$ns namtrace-all-wireless $namtracefd $val(x) $val(y)
$ns use-newtrace

#CREATING A TOPOLOGY OBJECT THAT KEEPS TRACK OF NUMBER OF ALL
NODES WITHIN BOUNDARY
set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)
```

```
#CREATING "God (General Operations Director) is the object that is used to store global
information about the state of the environment, network or nodes"
create-god $val(nn)

#CONFIGURING NODES
$ns node-config -adhocRouting $val(rp) \
 -llType $val(ll) \
 -macType $val(mac) \
 -ifqType $val(ifq) \
 -ifqLen $val(ifqlen) \
 -antType $val(ant) \
 -propType $val(prop) \
 -phyType $val(netif) \
 -channelType $val(chan) \
 -topoInstance $topo \
 -agentTrace ON \
 -routerTrace ON \
 -macTrace OFF \
 -movementTrace OFF

#CREATING NODES
for {set i 0} {$i < $val(nn) } {incr i} \
{
set node_($i) [$ns node]
}

#ENABLING RANDOM MOTION OF NODES
for {set i 0} {$i < $val(nn) } { incr i } \
{
$node_($i) random-motion 1
$node_($i) start
}

#LABELLING THE NODES
$node_(0) label "Source"
$node_(2) label "Receiver"

#SETTING A TCP CONNECTION BETWEEN NODE_(0) AND NODE_(2)
set tcp [new Agent/TCP]
set sink [new Agent/TCPSink]
$ns attach-agent $node_(0) $tcp
$ns attach-agent $node_(2) $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 10.0 "$ftp start"

#DEFINING NODE INITIAL POSITION IN NAM
for {set i 0} {$i < $val(nn) } {incr i} \
```

```
{
$ns initial_node_pos $node_($i) 50
}

#TELLING THE NODES WHEN THE SIMULATION ENDS
for {set i 0} {$i < $val(nn) } {incr i} \
{
$ns at $val(stop) "$node_($i) reset"
}

#PRINTING WINDOW SIZE
proc plotWindow {tcpSource file} \
{
global ns
set time 0.01
set now [$ns now]
set cwnd [$tcpSource set cwnd_]
puts $file "$now $cwnd"
$ns at [expr $now+$time] "plotWindow $tcpSource $file"
}
$ns at 10.1 "plotWindow $tcp $winFile"

#ENDING NAM
$ns at $val(stop) "stop"
proc stop {} \
{
global ns tracefd namtracefd
$ns flush-trace
close $tracefd
close $namtracefd
exec nam AODV.nam &
exit 0
}

#STARTING THE SIMULATION
$ns run
```

2.2. DSR PROTOCOL (dsr.tcl)

```
#DYNAMIC SOURCE ROUTING(DSR) PROTOCOL

#DEFINING OPTIONS
set val(chan) Channel/WirelessChannel ;        #CHANNEL TYPE
set val(prop) Propagation/TwoRayGround ;    #RADIO-PROPAGATION MODEL
set val(netif) Phy/WirelessPhy ;                #NETWORK INTERFACE TYPE
set val(mac) Mac/802_11 ;                       #MAC TYPE
set val(ifq) CMUPriQueue ;                      #INTERFACE QUEUE TYPE
set val(ll) LL ;                                #LINK LAYER TYPE
set val(ant) Antenna/OmniAntenna ;              #ANTENNA MODEL
set val(ifqlen) 50 ;                            #MAXIMUM PACKET IN ifq

#GETTING THE NUMBER OF NODES TO BE GENERATED FROM THE USER
puts "Enter the nodes to generate"
gets stdin MEE
puts "You entered the number of nodes as $MEE"
set val(nn) $MEE ;

#SETTING THE DSR PROTOCOL
set val(rp) DSR;

#BY CHANGING THE VALUE OF X AND Y, WE CAN GENERATE A DIFFERENT
DISTANCE RANGE FOR THE NODE MOVEMENTS.
set val(x) 500 ;       #X-DIMENSION OF TOPOGRAPHY
set val(y) 500 ;       #Y-DIMENSION OF TOPOGRAPHY
set val(stop) 50 ;    #TIME OF SIMULATION

#CREATING AN INSTANCE OF THE SIMULATOR
set ns [new Simulator]
set tracefd [open DSR.tr w]
set winFile [open DSR w]
set namtracefd [open DSR.nam w]
$ns trace-all $tracefd
$ns namtrace-all-wireless $namtracefd $val(x) $val(y)
$ns use-newtrace

#CREATING A TOPOLOGY OBJECT THAT KEEPS TRACK OF NUMBER OF ALL
NODES WITHIN BOUNDARY
set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)

#CREATING "God (General Operations Director) is the object that is used to store global
information about the state of the environment, network or nodes"
create-god $val(nn)

#CONFIGURING NODES
$ns node-config -adhocRouting $val(rp) \
 -llType $val(ll) \
```

```
 -macType $val(mac) \
 -ifqType $val(ifq) \
 -ifqLen $val(ifqlen) \
 -antType $val(ant) \
 -propType $val(prop) \
 -phyType $val(netif) \
 -channelType $val(chan) \
 -topoInstance $topo \
 -agentTrace ON \
 -routerTrace ON \
 -macTrace OFF \
 -movementTrace ON

#CREATING NODES
for {set i 0} {$i < $val(nn) } { incr i } \
{
set node_($i) [$ns node]
}

#ENABLING RANDOM MOTION OF NODES
for {set i 0} {$i < $val(nn) } { incr i } \
{
$node_($i) random-motion 1
$node_($i) start
}

#LABELLING THE NODES
$node_(0) label "Source"
$node_(2) label "Receiver"

#SETTING A TCP CONNECTION BETWEEN NODE_(0) AND NODE_(2)
set tcp [new Agent/TCP/Newreno]
$tcp set class_ 2
set sink [new Agent/TCPSink]
$ns attach-agent $node_(0) $tcp
$ns attach-agent $node_(2) $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 10.0 "$ftp start"

# PRINTING WINDOW SIZE
proc plotWindow {tcpSource file} \
{
global ns
set time 0.01
set now [$ns now]
set cwnd [$tcpSource set cwnd_]
puts $file "$now $cwnd"
$ns at [expr $now+$time] "plotWindow $tcpSource $file"
```

```
}
$ns at 10.1 "plotWindow $tcp $winFile"

#DEFINING NODE INITIAL POSITION IN NAM
for {set i 0} {$i < $val(nn)} { incr i } \
{
$ns initial_node_pos $node_($i) 50
}

# TELLING THE NODES WHEN THE SIMULATION ENDS
for {set i 0} {$i < $val(nn) } { incr i } \
{
$ns at $val(stop) "$node_($i) reset";
}

#ENDING NAM
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "stop"
$ns at 150.01 "puts \"end simulation\" ; $ns halt"
proc stop {} \
{
global ns tracefd namtracefd
 $ns flush-trace
 close $tracefd
 close $namtracefd
 exec nam DSR.nam &
 exit 0
}

#STARTING SIMULATION
$ns run
```

CHAPTER 3 – SNAPSHOTS

The snapshots featured were taken during the simulation and analysis of the AODV and DSR protocols. The software used are:

1. Network Simulator (NS2)
2. Network Animator (NAM)
3. NS2 Trace File Analyzer (Ns2GTFA)

3.1. The snapshots below are for the simulation of AODV protocol

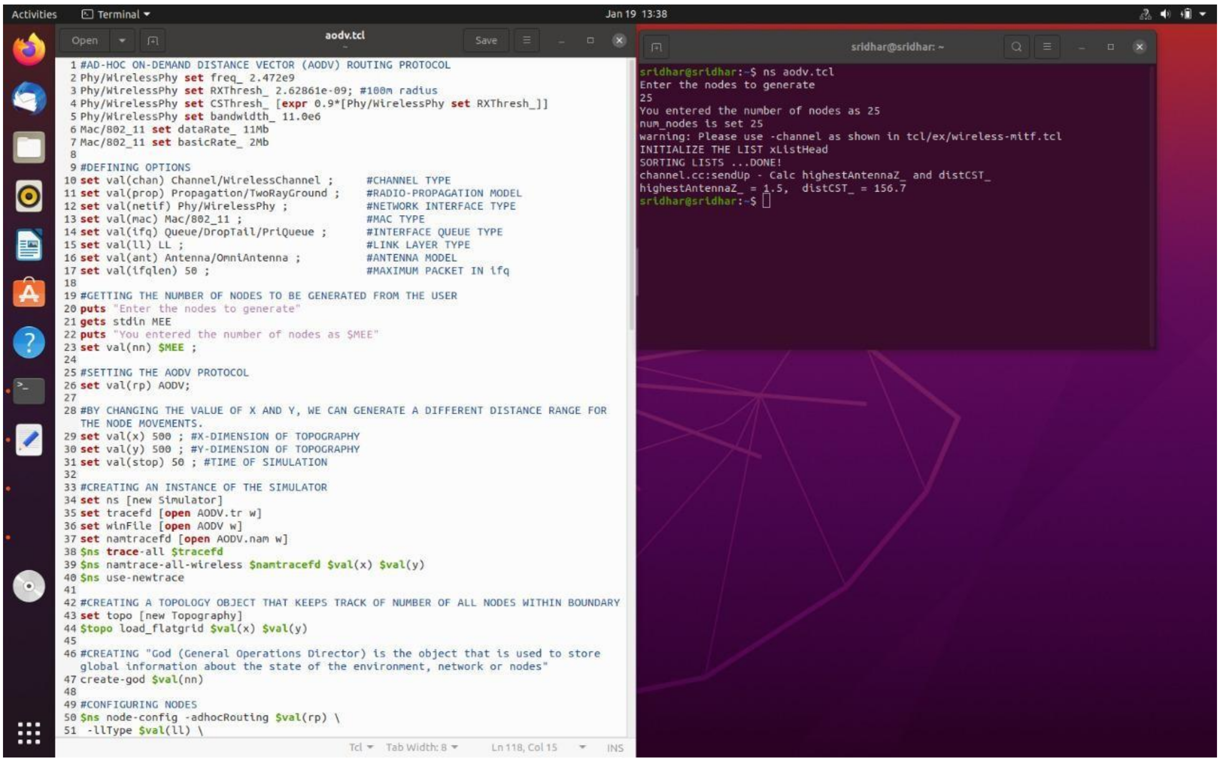


Fig. 3.1.1. ns2 simulation of AODV protocol and its source code

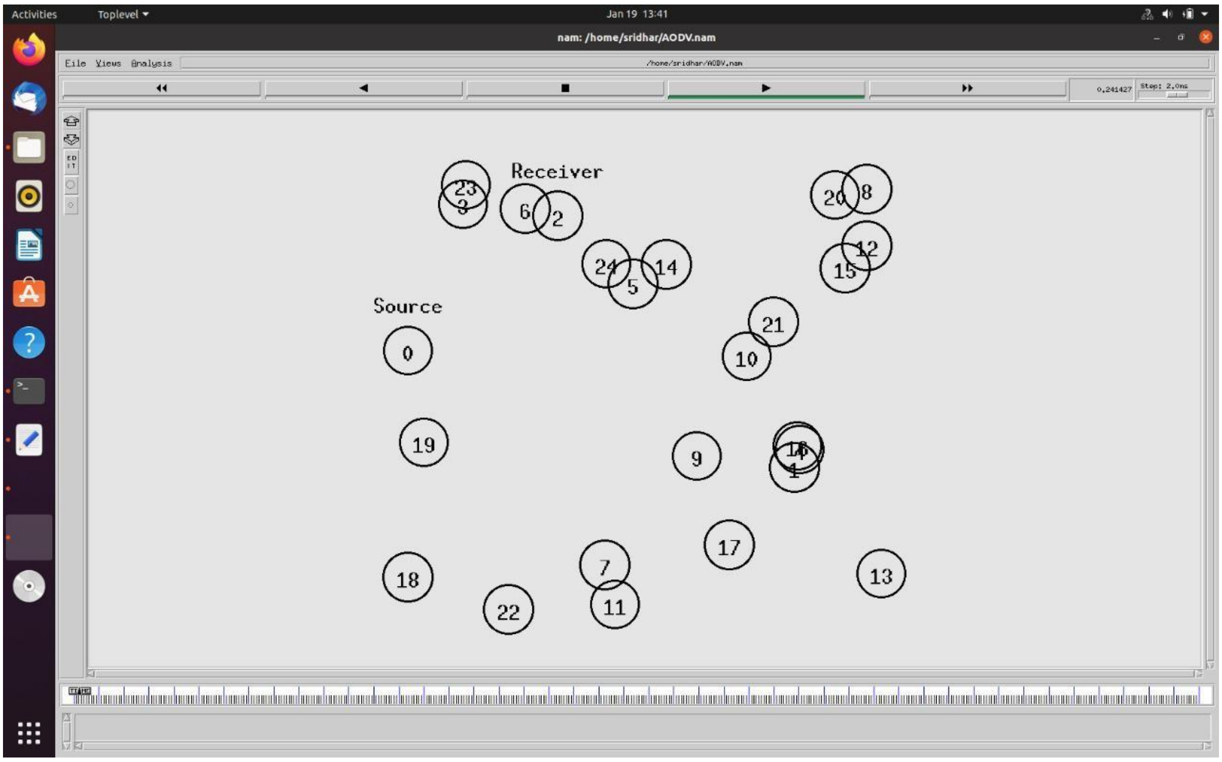


Fig. 3.1.2. NAM output of AODV simulation to view the network traces

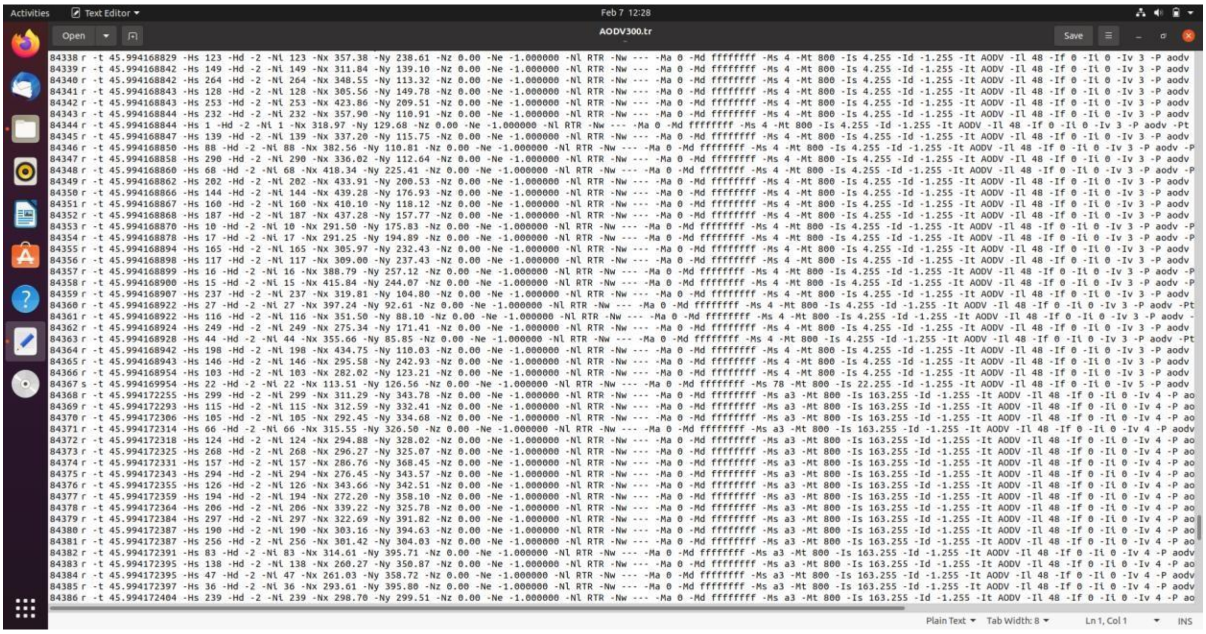


Fig. 3.1.3. Trace file obtained from AODV simulation

3.2. The snapshots below are for the simulation of DSR protocol

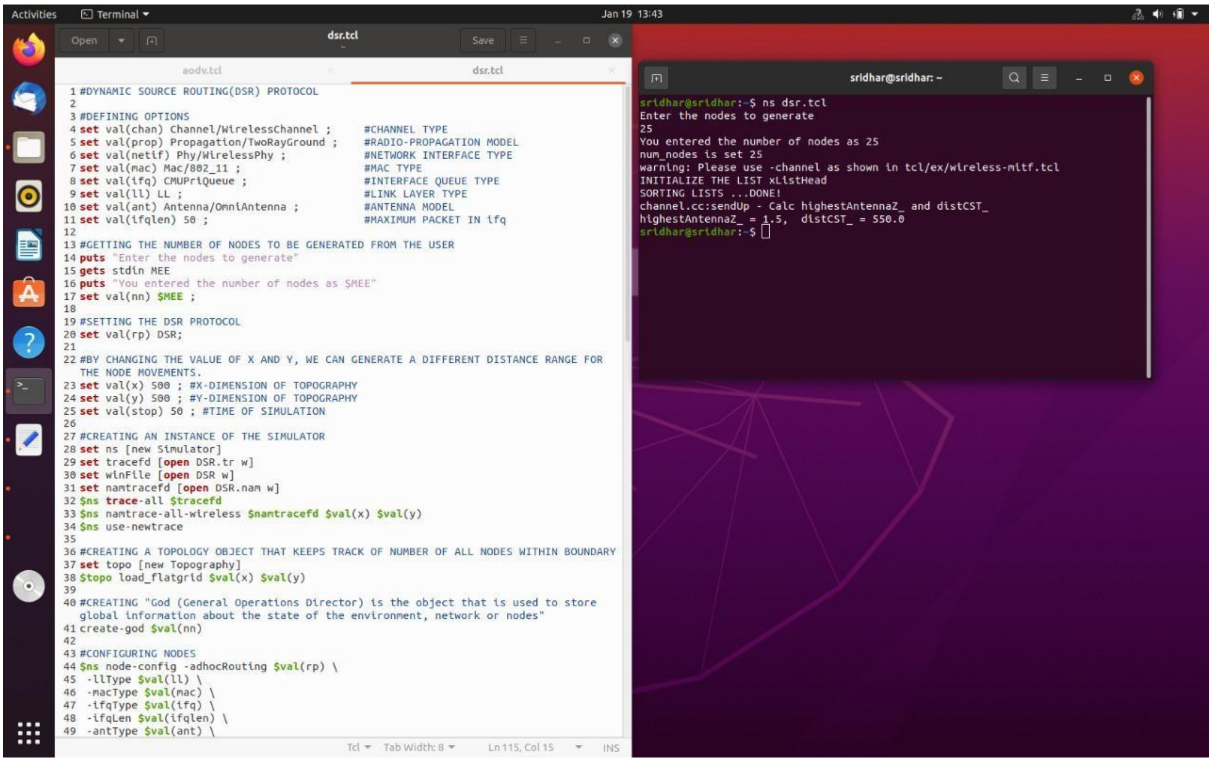


Fig. 3.2.1. ns2 simulation of DSR protocol and its source code

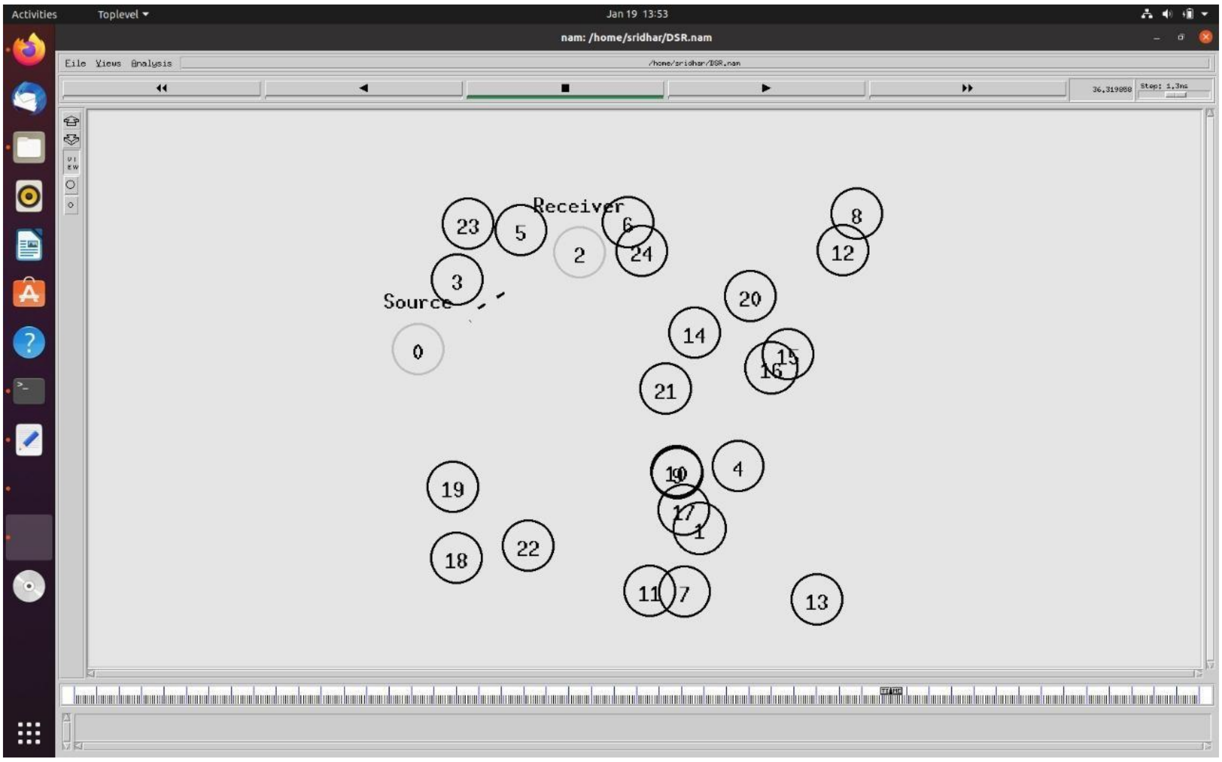


Fig. 3.2.2. NAM output of DSR simulation to view the network traces

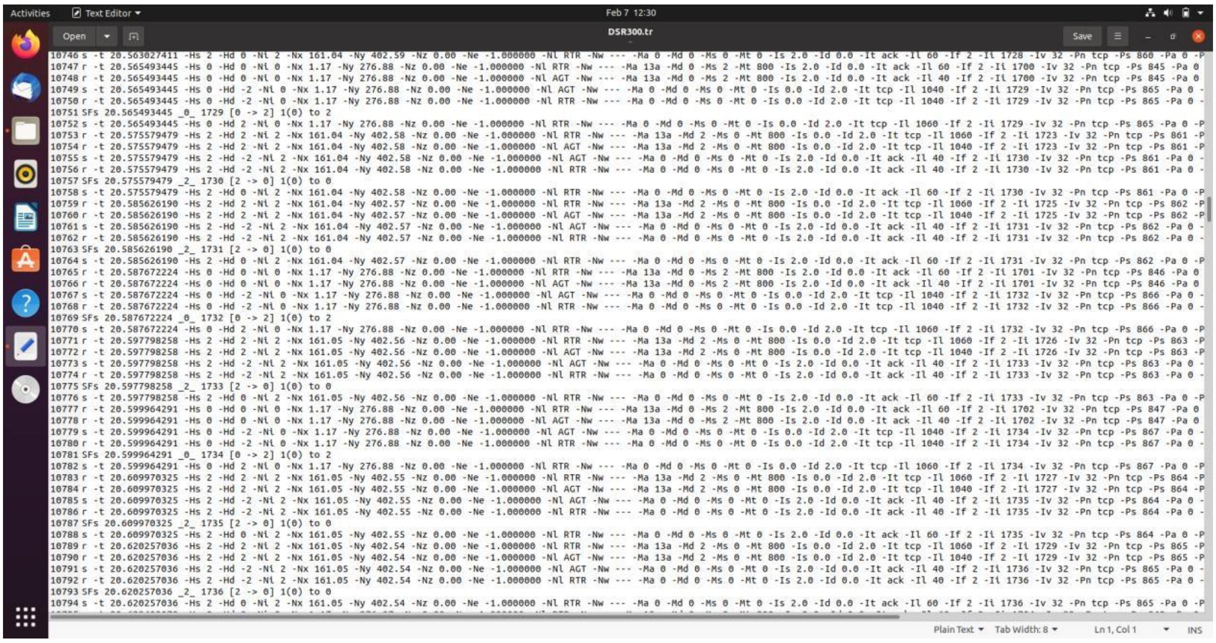


Fig. 3.2.3. Trace file obtained from DSR simulation

3.3. The snapshots below are taken during the analysis of the protocols

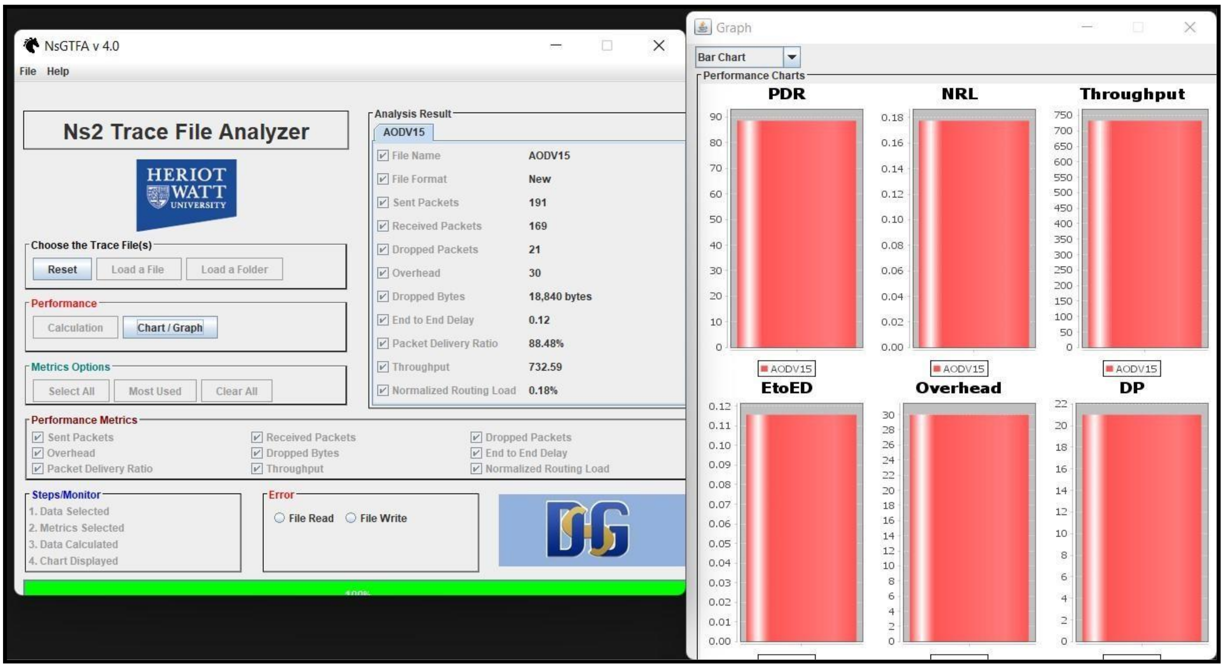


Fig. 3.3.1. Ns2GTFA an open source software used to analyze trace files

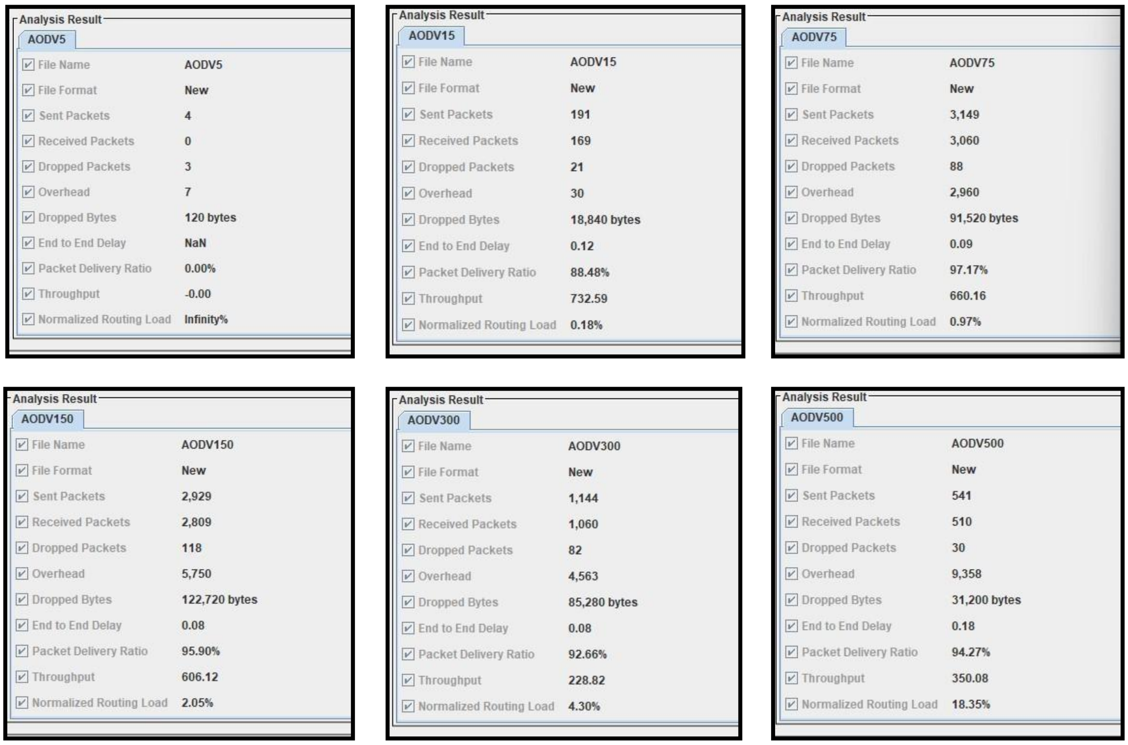


Fig. 3.3.2. Results from analysis of AODV protocol with various number of nodes

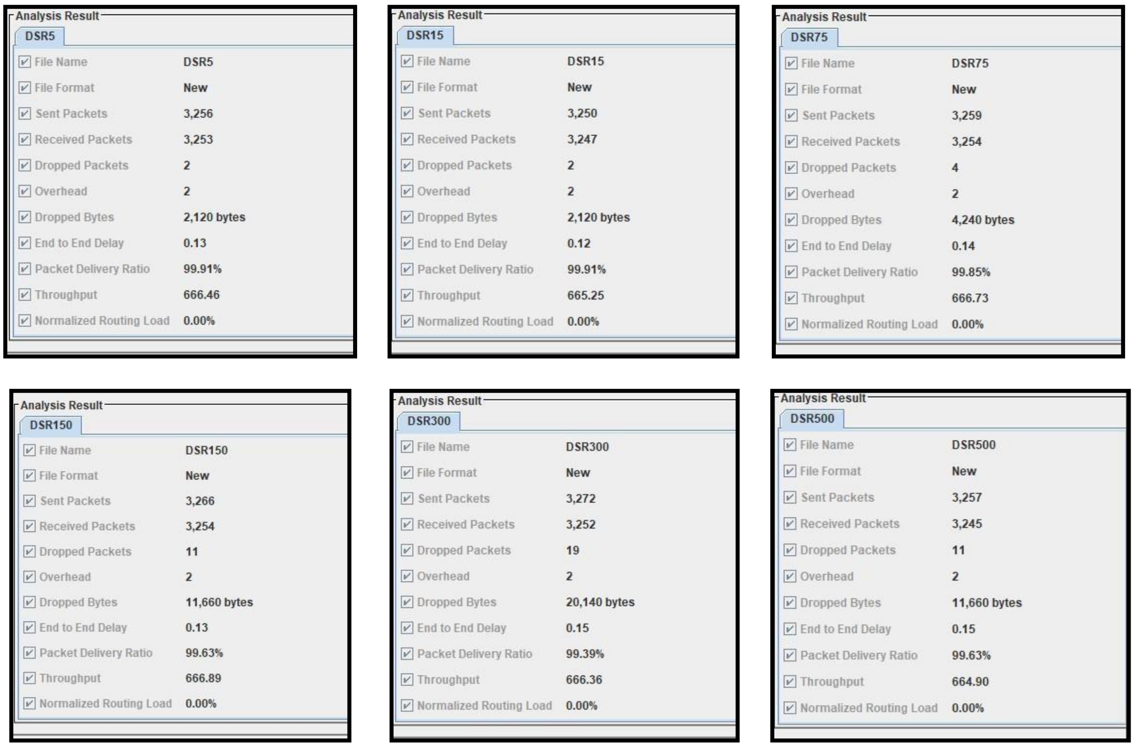


Fig.3.3.3. Results from analysis of DSR protocol with various number of nodes

CHAPTER 4 - SIMULATION AND ANALSYIS

4.1. Simulation

All the work is done has been done under Linux platform, preferably Ubuntu. NS is a discrete event simulator targeted at networking research. It provides substantial support for TCP routing and multicast protocols over wired and wireless networks. It is the open source software for evaluating the performance of the existing network protocols and evaluates new network protocols before use. In NS-2, the frontend of the program is written in TCL (Tool Command Language). The backend of NS-2 simulator is written in C++ and when the TCL program is compiled, a trace file and nam file are created which define the movement pattern of the nodes and keeps track of the number of packets sent, number of hops between 2 nodes, connection type etc. at each instance of time.

DSR and AODV routing protocols can be implemented using Network Simulator 2. Using ns2 simulator to simulate a variety of IP networks, the routing protocols were compared. To run the output of the program in an animator we need a nam file, and to analyze the output we need trace file. So the program must output certain files called nam file and trace file. To analyze the data using NsGTFA, we need the trace files. The format of nam files and trace files have been documented in the ns Manual.

4.2. Simulation Parameters

Platform	Ubuntu 20.04
NS Version	2.35
Number of Nodes	5,15,75,150,300,500
Area Size	500m × 500m
MAC	802.11
Simulation Time	50
Protocol	AODV/DSR
Transmission Range	100m radius
Node Movement Model	Random Motion

Table 4.2.1 Simulation parameters for AODV and DSR Protocols

4.3. Analysis

For MANET simulation, there are many performance metrics which are used to analysis the various proposals. In this project we have used four performance metrics that evaluate routing protocols. The evaluation of performance of routing protocols can be measured under the metrics

Throughput, Packet Ratio Delivery, End-to-End delay and Routing overhead.

1. **Throughput:** The ratio of the total amount of data that reaches a receiver from a sender to the time it takes for the receiver to get the last packet is referred to as throughput. It is measured by bits/sec or packets per second. A high throughput network is desirable.

2. **Packet Delivery Ratio:** Packet delivery ratio is the ratio of number of packets received at the destination nodes to the number of packets sent from the source nodes. The performance is better when packet delivery ratio is high.

3. **End-to-End Delay:** End-to-end delay is the average time delay for data packets from the source node to the destination node. To find out the end-to-end delay the difference of packet sent and received time was stored and then dividing the total time difference over the total number of packet received gave the average end-to-end delay for the received packets. The performance is better when packet end-to-end delay is low.

4. **Routing Overhead:** Routing Overhead is the number of routing packets required for communication. The performance is better when routing overhead is low.

When the dimension of the topography is 500m $\times$ 500m, it is marked as short distance range in the below graphs. From the below figures of graphs, we can see that AODV is better regarding throughput and packet delivery ratio. AODV has high end to end delay because it needs more time for path discovery whereas DSR uses source routing framework. And regarding end to end delay and overhead, DSR is performs better. But as DSR doesn't have frequent route discovery it doesn't perform well in scenarios of high mobility of nodes.

4.4. Graphs of Analysis

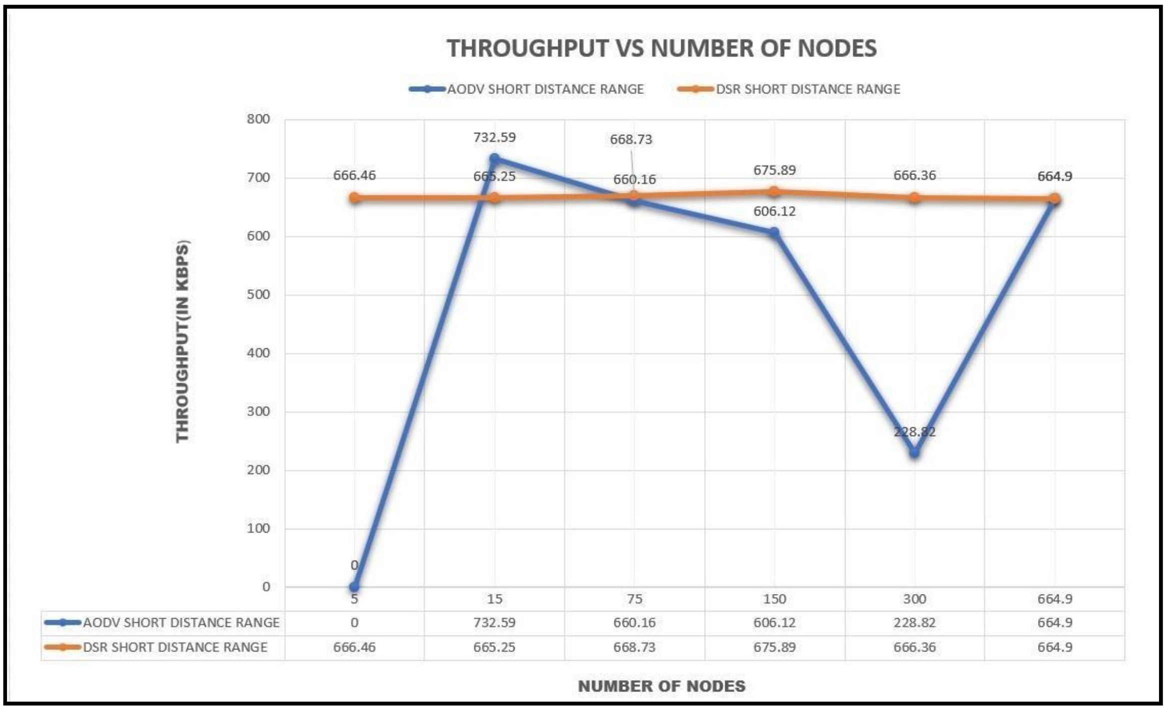


Fig. 4.4.1. Graph of Throughput versus the number of nodes

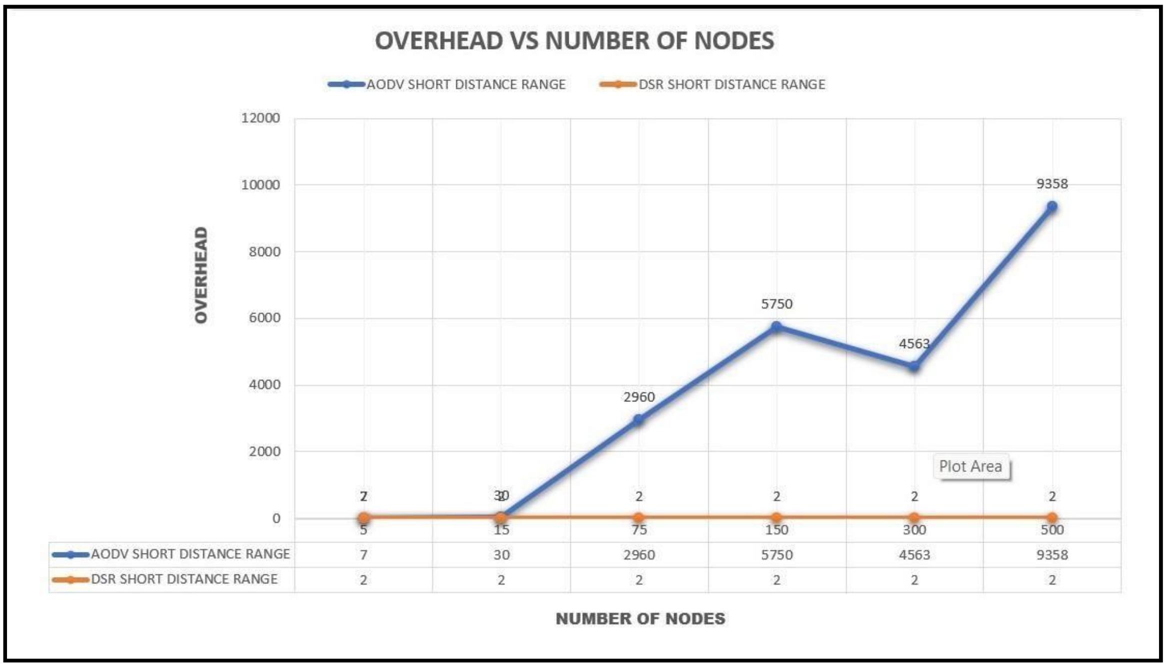
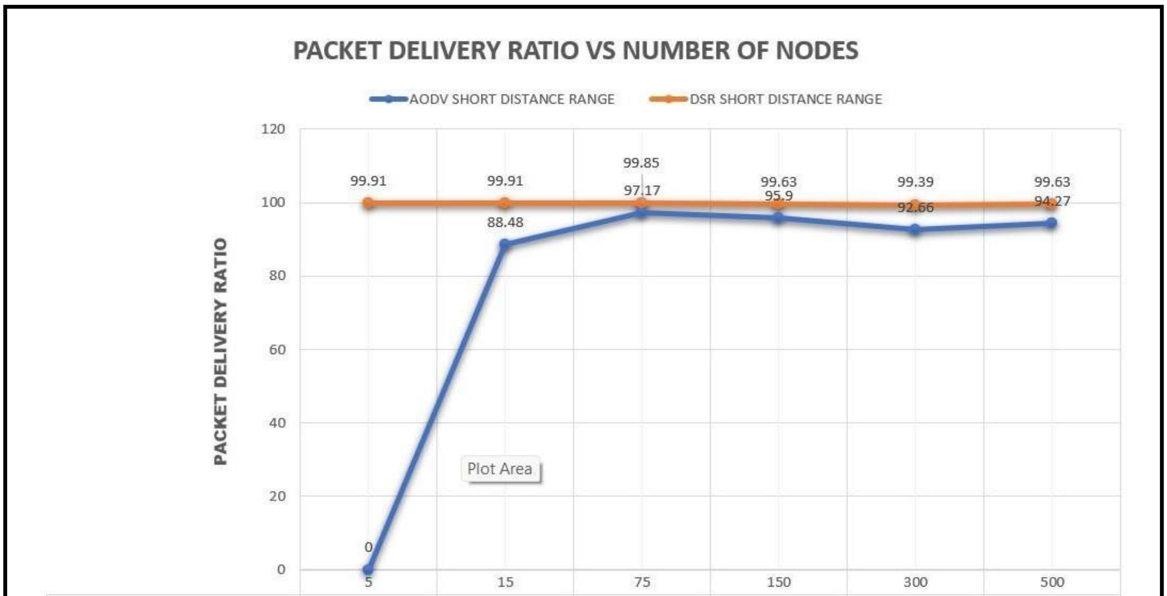
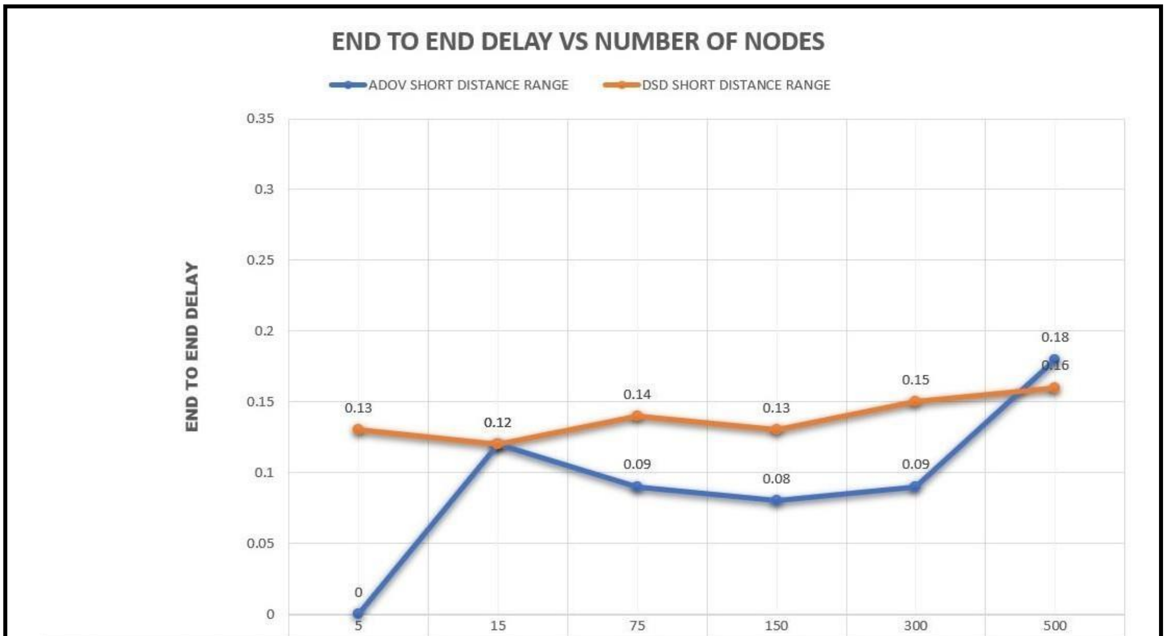


Fig. 4.4.2. Graph of Overhead versus the number of nodes



	5	15	75	150	300	500
AODV SHORT DISTANCE RANGE	0	88.48	97.17	95.9	92.66	94.27
DSR SHORT DISTANCE RANGE	99.91	99.91	99.85	99.63	99.39	99.63

Fig. 4.4.3. Graph of Packet Delivery Ratio versus the number of nodes



	5	15	75	150	300	500
ADOV SHORT DISTANCE RANGE	0	0.12	0.09	0.08	0.09	0.18
DSD SHORT DISTANCE RANGE	0.13	0.12	0.14	0.13	0.15	0.16

Fig. 4.4.4. Graph of End to End Delay versus the number of nodes

CHAPTER 5 - CONCLUSION AND FUTURE PLANS

The performance of the two MANET Routing protocols - AODV and DSR were analyzed in this project using ns2. Comprehensive simulation results of Average End-to-End delay, throughput, overhead and packet delivery ratio over DSR and AODV protocols by varying number of nodes were obtained. Of the two protocols, DSR has better packet delivery ratio when compared to AODV and its throughput is consistent as nodes increases. The Overhead of AODV increases drastically when the number of nodes increases. When the network load is low AODV protocol has lower end to end delay. At moderate traffic DSR is preferrable. Since AODV has higher route discovery, it is preferrable during high mobility situations.

Ad hoc networks provide a challenge in routing as they vary in their characteristics from conventional networks and new techniques are emerging for accommodate the demands. Due to time limitations, this project focuses on two of the protocols. This project can be further extended by analyzing performance metrics against varying simulation time and transmission ranges.

CHAPTER 6 - REFERENCES

1. **Zou, Cliff.** *Introduction to Network Simulator (NS2).* Retrieved from University of Central Florida's website for the Course CDA6530: https://www.cs.ucf.edu/~czou/CDA6530-11/NS2-tutorial.pdf

2. *The Network Simulator - ns-2.* Retrieved from ISI Documentation for NS2 Simulator: https://www.isi.edu/nsnam/ns/

3. **Kanthe, A. M., Simunic, D., & Prasad, R.** *Comparison of AODV and DSR on-demand routing protocols in mobile ad hoc networks.* (2012). International Conference on Emerging Technology Trends in Electronics, Communication and Networking (ET2ECN): https://doi.org/10.1109/ET2ECN.2012.6470118

4. *MANET Routing Protocols*. Retrieved from INET Framework Documentations: https://inet.omnetpp.org/docs/showcases/routing/manet/doc/index.html

5. *Overview (Advantages and Routing Protocols) of MANET.* (2017). Retrieved from Advances in Computational Sciences and Technology.