CHAPTER 6

PERFORMANCE ANALYSIS

6.1. Introduction

This chapter briefly describes the simulation environment and various parameters chosen to simulate the routing protocols. This section presents the performance evaluation of MMDV and EMMD protocol using NS2.3.5. Evaluation is accomplished for all protocols under the same environment of the simulation mentioned earlier in this chapter. The same performance metrics and input parameters are used here again to evaluate the protocols in MANETs.

Simulations are carried out using NS2 version 2.35 on Linux platform - Fedora 5 with the same mobility models. A new routing protocol EMMDV is compared with existing protocol MMDV .The simulated traffic is Constant Bit Rate (CBR). The performance of both protocol are evaluated based on Packet Delivery Ratio, End-to-End Delay, Throughput, Packet loss and Energy.

6.2. Network Simulator

Network Simulator (Version 2), widely known as NS2, is simply an event driven simulation tool that has proved useful in studying the dynamic nature of communication networks. Simulation of wired as well as wireless network functions and protocols (e.g., routing algorithms, TCP, UDP) can be done using NS2 It consists of two simulation tools.

The network simulator (ns2) contains all commonly used IP protocols. The network animator (NAM) is used to visualize the simulations. Ns-2 [85] fully simulates

a layered network from the physical radio transmission channel to high-level applications The simulator was originally developed by the University of California at Berkeley and VINT project the simulator was recently extended to provide simulation support for ad hoc network by Carnegie Mellon University (CMU Monarch Project homepage, 1999 .

NS2 consists of two key languages: C++ and Object-oriented Tool Command Language (OTcl) while the C++ defines the internal mechanism (i.e., a backend) of the simulation objects, the OTcl sets up simulation by assembling and configuring the objects as well as scheduling discrete events (i.e., a frontend). After simulation, NS2 outputs either text-based or animation-based simulation results. To interpret these results graphically and interactively, tools such as NAM (Network AniMator) and XGraph are used.

The result of the simulations is an output trace file that can be used to do data processing (calculate delay, throughput etc) and to visualize the simulation with a program called Network Animator (NAM). NAM is a very good visualization tool that visualizes the packets as they propagate through the network. An overview of how a simulation is done in ns2 is shown in Fig.6.1.

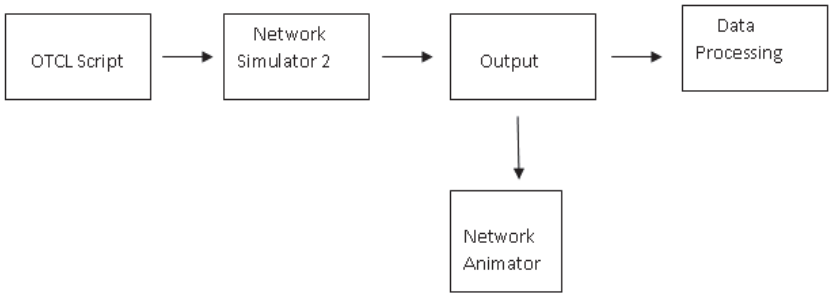

Fig.6.1 Network Simulator 2

6.2.1. Network Animator (NAM)

NAM is a Tcl/TK based animation tool for viewing network simulation traces and real world packet trace data. The first step to use NAM is to produce the trace file. The trace file should contain topology information, e.g., nodes, links, as well as packet traces. Usually, the trace file is generated by ns2. During an ns2 emulation, user can produce topology configurations, layout information, and packet traces using tracing events in ns2 [86].

When the trace file is generated, it is ready to be animated by NAM. Upon startup, NAM will read the trace file, create topology, pop up a window, do layout if necessary and then pause at the time of the first packet in the trace file. Through its user interface, NAM provides control over many aspects of animation.

The main window of NAM is shown below:

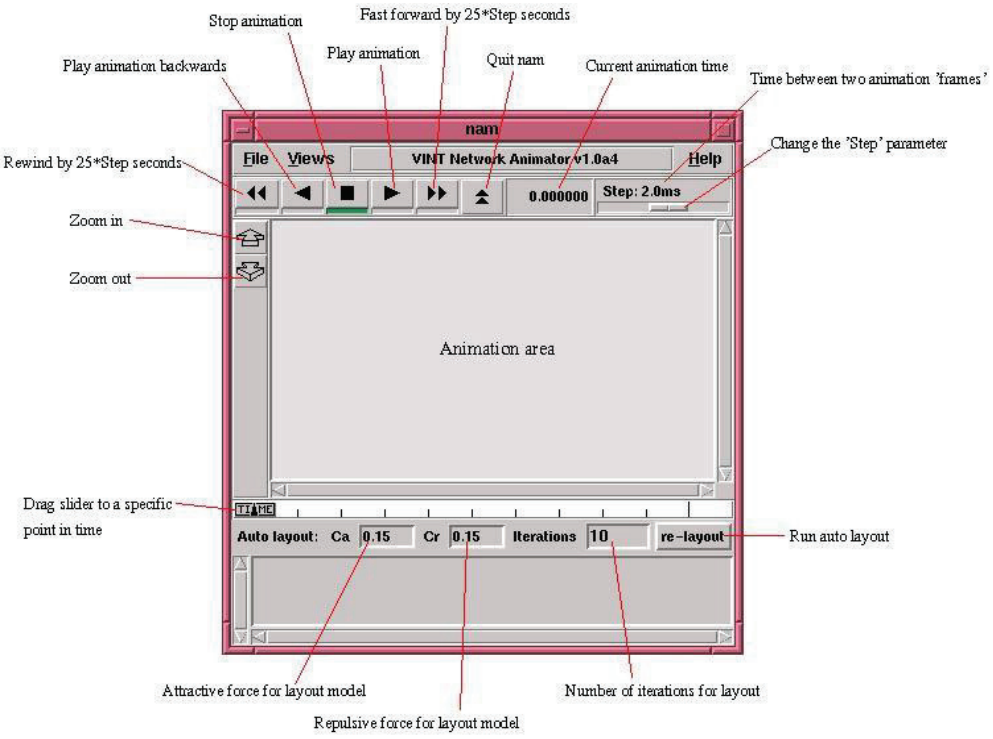

Fig.6.2 NAM Window

6.2.2. Trace file in NS2

The trace data is in ASCII code and are organized in to 12 fields. Each trace line starts with an event descriptor followed by the simulation time (in seconds) of that event, and from and to node, which identifies the link on which the event occurred. The next information in the line is for flags. Since no flags are set here we have “------". Then we have the packet type and size (in Bytes). The next field is flow id (fid) of IP address that a user can set for each flow. Even though fid field may not be used in a simulation, users can use this field for analysis purposes.

The next two fields are source and destination address in forms of "node port". The last field shows the network layer protocol's packet sequence number. Note that even though UDP implementations do not use sequence number, NS-2 keeps track of UDP packet sequence number for analysis purposes. The last field shows the unique id of the packet .To create a trace file, you need to do the following two steps:

1. Create a file to record tracing information.
2. Record the tracing information to the created file.

Step 1: Create a File for Writing

Tcl uses a command “open" to open a file. The syntax of the command "open" is as follows:

open <filename> <purpose>

where <filename> is the name of the file to be opened, and <purpose> can be

- “w" for writing,
- “r" for reading, or
- “a" for appending

This statement returns a file handle, which can be used to refer to the opened file. The following Tcl command is used to open the trace File.

open tracefile.tr <purpose>

Step 2: Record tracing information in the opened trace file.

The next step is to record trace information in the opened file. This can be achieved using the following Tcl statement:

$ns trace-all

Trace file Format

event	time	from node	to node	pkt type	pkt size	flags	fid	scr addr	det addr	seq num	pkt id

r : receive (at to_node)
+ : enqueue (at queue)
- : dequeue (at queue)
d : drop (at queue)

src_addr : node.port
dst_addr : node.port

```
r 1.3556 3 2 ack 40 ------- 1 3.0 0.0 15 201
+ 1.3556 2 0 ack 40 ------- 1 3.0 0.0 15 201
- 1.3556 2 0 ack 40 ------- 1 3.0 0.0 15 201
r 1.35576 0 2 tcp 1000 ------- 1 0.0 3.0 29 199
+ 1.35576 2 3 tcp 1000 ------- 1 0.0 3.0 29 199
d 1.35576 2 3 tcp 1000 ------- 1 0.0 3.0 29 199
+ 1.356 1 2 cbr 1000 ------- 2 1.0 3.1 157 207
- 1.356 1 2 cbr 1000 ------- 2 1.0 3.1 157 207
```

Fig.6.3 Trace File Format

6.2.3. Xgraph in NS2

One part of the ns-allinone package is 'xgraph', a plotting program which can be used to create graphic representations of simulation results. Xgraph is an X-Windows application that includes:

- interactive plotting and graphing
- animation and deritives
- portability and bug fixes

So to plot the characteristics of NS2 parameters like throughput, end to end delay, packets inform the successful installation of NS2 will install xgraph also along with it. To run xgraph from a shell prompt using the following command,

xgraph filename.xg

The xgraph program draws a graph on an X display given data read from either data files or from standard input if no files are specified. It can display up to 64 independent data sets using different colors and/or line styles for each set. It annotates the graph with a title, axis labels, grid lines or tick marks, grid labels, and a legend. There are options to control the appearance of most components of the graph.

The input format is similar to graph but differs slightly. The data consists of a number of data sets. Data sets are separated by a blank line. A new data set is also assumed at the start of each input file. A data set consists of an ordered list of points of the form "X Y". Each point must appear on a separate line. The name of a data set can be specified by a line which begins with a double quote followed by the set name.

6.3. Experimental Results and Discussion

The simulation experiment is carried out in LINUX (FEDORA 8). The detailed simulation model based on network simulator-2 (ver-2.35), is used in the evaluation. The NS2 instructions are used to define the topology structure of the network, the motion mode of the nodes, to configure the service source and the receiver, to create the statistical data track file and so on.

The studied scenario consists of 50 mobile nodes. The topology is a rectangular area with 800m length and 500m width. A rectangular area was chosen in order to force the use of longer routes between nodes than would occur in a square area with equal node density. All simulations are run for 25 seconds of simulated time. All mobile nodes are constant bit rate traffic sources. They are distributed randomly within the mobile ad hoc network. The sources continue sending data until one second before the end of the simulation.

The EMMDV protocol performance was evaluated with the "Network Simulator 2" (ns2). In the case of study, two simulation scenarios were considered.

1) In the first one the simulation was carried out in normal conditions, in other words, all the nodes participated correctly in the routing functions.
2) In the second scenario, Routing is established using EMMDV protocol.

To implement EMMDV protocol, a wireless scenario was configured with a similar set of 50 nodes. The nodes move about within an area whose boundary is defined in this example as 800mX500m. At the beginning of a wireless simulation, the types for each of these network components have to be defined. The type of antenna, the radio-propagation model, sand the type of routing protocol used by mobile nodes is some of the other parameters that are defined.

Random traffic connections of UDP and CBR are setup between mobile nodes using a traffic-scenario generator script. The CBR and UDP traffics connections are created between wireless mobile nodes.

To create a traffic-connection file, the type of traffic connection, 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 are defined.

General parameters used in the experiments are shown in Table 6.1.

Table 6.1 Parameter value

Transmission range	250 m
Simulation time	25s
Topology size	800m x 500m
Number of nodes	25,50,100
Number of source	1
Traffic type	CBR(Constant Bit Rate)
Packet rate	5 packets/s
Packet size	512 bytes
Maximum speed	20 m/s

The node-configuration for EMMDV is as shown below,

```
# Define options
 set val(chan)         Channel/WirelessChannel  ;
```

```
set val(prop)         Propagation/TwoRayGround ;
set val(ant)          Antenna/OmniAntenna;
set val(ll)               LL;
set val(ifq)              Queue/DropTail/PriQueue  ;
set val(ifqlen)           50 ;
set val(netif)            Phy/WirelessPhy;
set val(mac)          Mac/802_11 ;
set val(nn)           51 ;
set val(rp)             AODV ;
set val(x)            900
set val(y)            900

#flush stdout
#set dest_node [gets stdin]
set sour_node 11
set dest_node 29
set ns [new Simulator]
#ns-random 0
set f [open 50.tr w]
$ns trace-all $f
set namtrace [open 50.nam w]
$ns namtrace-all-wireless $namtrace $val(x) $val(y)
set f0 [open rou_packets_received.tr w]
set f1 [open rou_packets_lost.tr w]
```

```
set f2 [open proj_out2.tr w]
set f3 [open proj_out3.tr w]
set topo [new Topography]
$topo load_flatgrid 900 900
create-god $val(nn)
set chan_1 [new $val(chan)]

# CONFIGURE AND CREATE 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 OFF \
                -routerTrace ON \
                   -macTrace OFF \
                -movementTrace ON \
                -channel $chan_1
               # -movementTrace ON \
```

-$ns at 0.001 "movementTrace OFF" \

#-channel $chan_1 # \

Nam Window for Mode creation

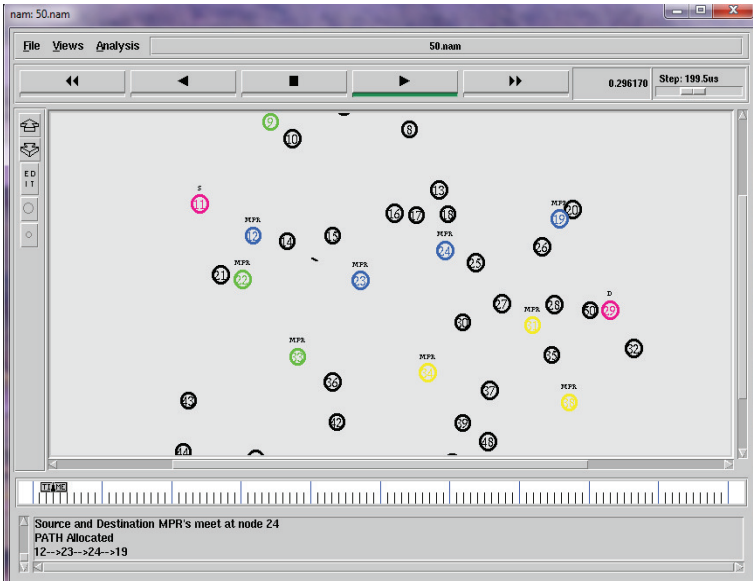

Fig.6.4 Screenshot for Node creation

Nam Window for Routing

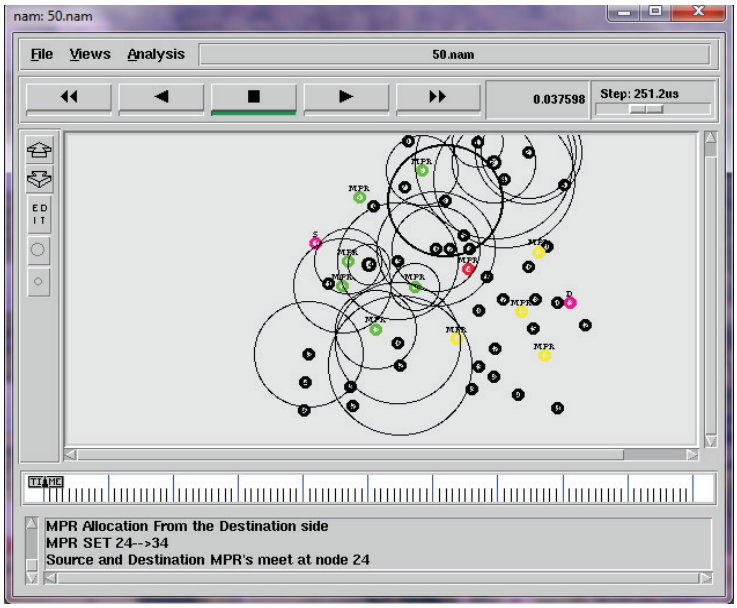

Fig.6.5 Screenshot for routing using EMMDV Protocol

Performance Metrics

The performance of EMMDV and MMDV protocol are evaluated based on the following performance metrics.

Performance Results

Table 6.2 provides the simulated data which is obtained from the experiments. The performance of existing MMDV schemes and proposed EMMDV schemes are investigated under different network sizes such as 25, 50 and 100 nodes.

Table 6.2 Performance analysis Results

	MMDV (EXISTING)			EMMDV (PROPOSED)		
No. of. Nodes	25	50	100	25	50	100
Allocated MPR set	9	17	48	7	11	18
Generated Packets	101	143	167	244	244	144
Received Packets	96	121	110	231	242	142
Packet Loss	5	22	57	13	2	2
Packet Delivery Ratio (PDR)	95.05	84.62	80.87	94.67	96.59	97.02
End-to-End Delay (E2E)	56.5	69.44	143.76	20.72	23.68	61.16

From the above table it is identified that EMMDV scheme generates less number of MPR nodes when compared to that of MMDV scheme under all scenario. As number of MPR nodes is less, flooding of RREQ also gets reduced and end to end path sets quickly. So totally the EMMDV protocol reduces the delay time and also increase the PDR. MMDV scheme provides 80% PDR but EMMDV scheme provides 97% PDR in network size with 100 nodes.

In 25 nodes environment the MMDV scheme, it could reach the destination only after 56.5 ms but in case of EMMDV scheme, it could reach the target within 20.72ms. In 100^{th} nodes MMDV could reach the destination only after 143.76ms, but EMMDV could reach the target within 61.16ms. It is comparatively less than existing scheme. Reduction in end-to-end delay leads to increase in PDR (Packet Delivery Ratio).

6.4. Simulation Results and Analysis

The simulation results are obtained under several experiments .The results for proposed work has been compared with the results of MMDV protocol using Xgraph. XGRAPH has been used to conduct qualitative analysis. The parameters under consideration are Packet delivery ratio, Average Delay or End-to-End Delay, Throughput, Packet Loss and Energy. The following graphical analysis shows the performance results of MMDV and EMMDV protocols.

1. Packet Delivery ratio

Fig.6.6. shows the Xgraph for MMDV and EMMDV with a pause time set to 25ms. The X-axis of the graph indicates the No.of Nodes and the Y-axis shows the

Packet Delivery Ratio. This graph shows at node 50 the PDR ratio of EMMDV is 96.59 % and PDR Ratio of MMDV is 84.62 %. In this figure when the transmission goes from 25th node to 100th node. Here the PDR of EMMDV will increases but the PDR of MMDV will decreases .In fig PDR of MMDV decreases when network size increases where as PDR of EMMDV increases when network size increases because it has less packet loss. MMDV scheme provides 80.87% PDR but EMMDV scheme provides 97.02% PDR in network size with 100 nodes.

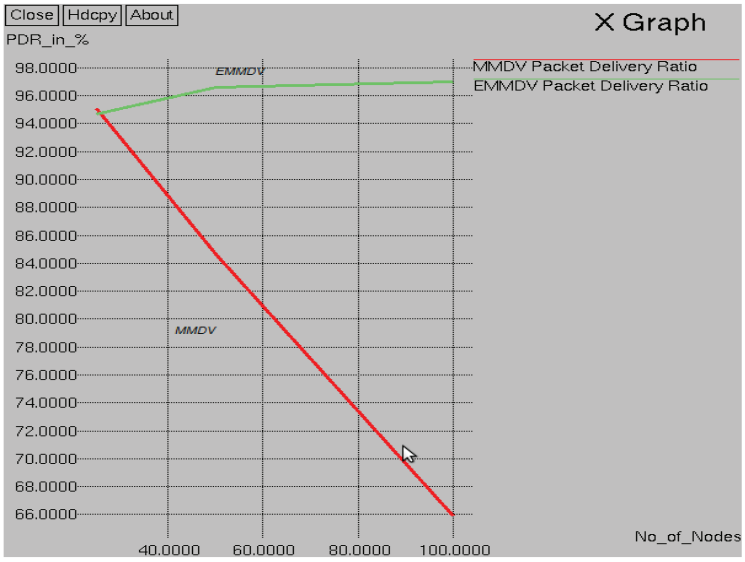

Fig.6.6 Packet Delivery ratio

2. End-to-End Delay

Fig.6.7. shows that End-End delay for MMDV and EMMDV .The X-axis of the graph represents the No. of Nodes and the Y-axis represents End-To-End Delay.While comparing these two protocol, EMMDV raises from 20.72% and it reaches a maximum level of 61.16%.But MMDV raises from 56.5% and it reaches 143.76%.In network size with 50 nodes the delay caused by MMDV (69.44%) increases than that of delay

caused by EMMDV (23.68%). The figure also shows that End-End delay for EMMDV scheme decreases when network size increases. The result depicts that, delay for data transfer gets reduced by using EMMDV scheme instead of using MMDV, when the number of nodes are increased.

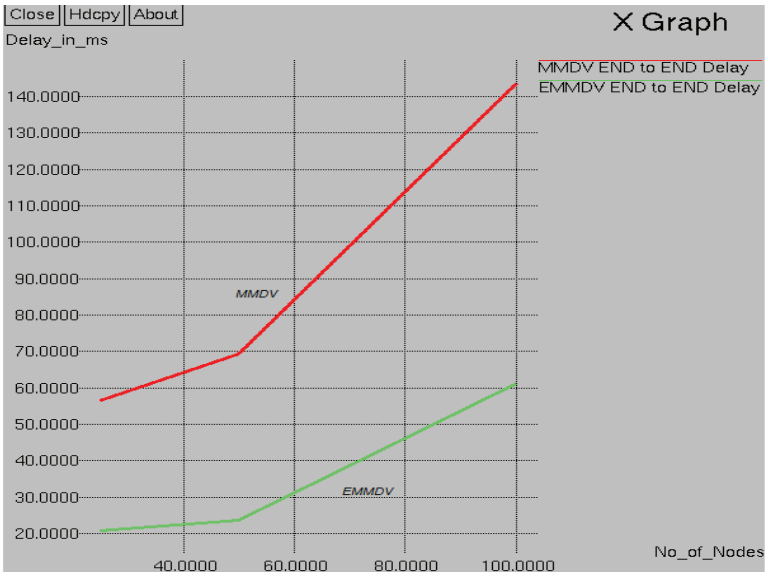

Fig.6.7 End-To-End Delay

3. Throughput

Another important quality of communication networks is the throughput. It is defined as the total useful data received per unit of time. Fig.6.8. illustrate the comparison of throughput for EMMDV and MMDV. In this metric, the throughput of the protocol in terms of number of messages delivered per one second (Mbps) is analyzed.

This fig shows that throughput of MMDV start from 25th node and it reaches 80% at 100th node. But throughputs of EMMDV will increase when network size

increases. It reaches maximum throughput 95% at 100th node. So EMMDV provides highest throughput than MMDV. More routing packets are generated and delivered by EMMDV than MMDV.

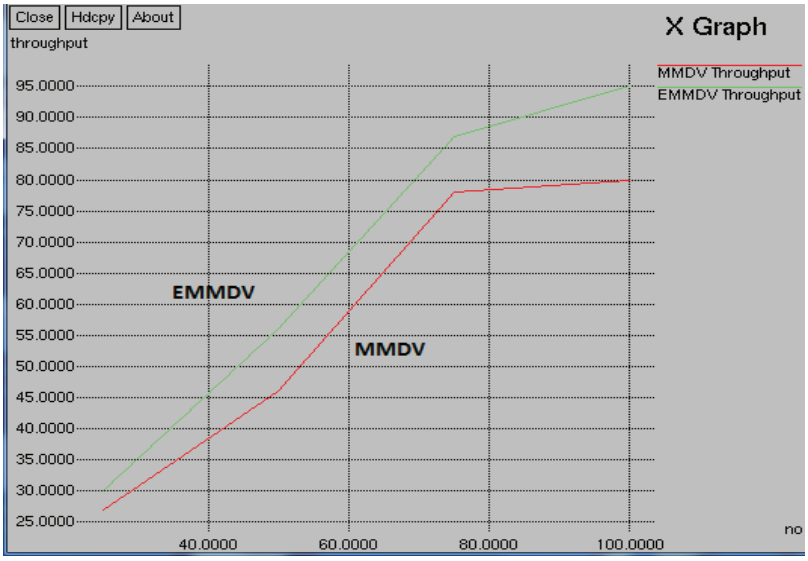

Fig.6.8 Throughput

4. Packet Loss

It is the number of data packets that are not successfully sent to the destination. The figurative transmission is shown in fig.6.9.

In this figure, the MMDV protocol losses 14% of packets at node 25. At the same time EMMDV protocol losses 15% at node 25. But the loss of packet is reduced to 2% while it reaches 100th node. In terms of dropped packets, EMMDV performance is better than MMDV.

In MMDV when the number of nodes increases, the number of packets dropped also increases which means that number of packets not successfully reaching the destination is high. The EMMDV performs consistently well with increase in the

number of nodes. The number of packets dropped is negligible which means that almost all packets reach the destination successfully. The packets dropped are much less compared to performance of MMDV.

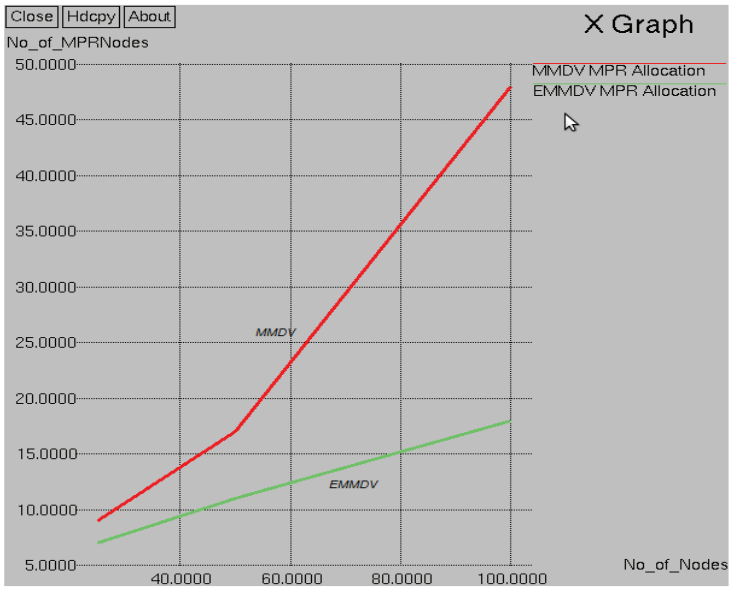

Fig.6.9 Packet Loss

5. Energy

Fig.6.10. shows the total transmission and receiving energy. When numbers of nodes are high, the transmitting energy is more. In this proposal the energy is calculated based on searching time. In MMDV, the source node takes more time for searching the destination.

In EMMDV, source node first searches whether the destination is present in the current network. If available it sends the message to destination. Similarly destination also searches for source node. Parallel execution of this architecture helps to reduce the discovery time. So EMMDV consumes less energy when compare to MMDV.

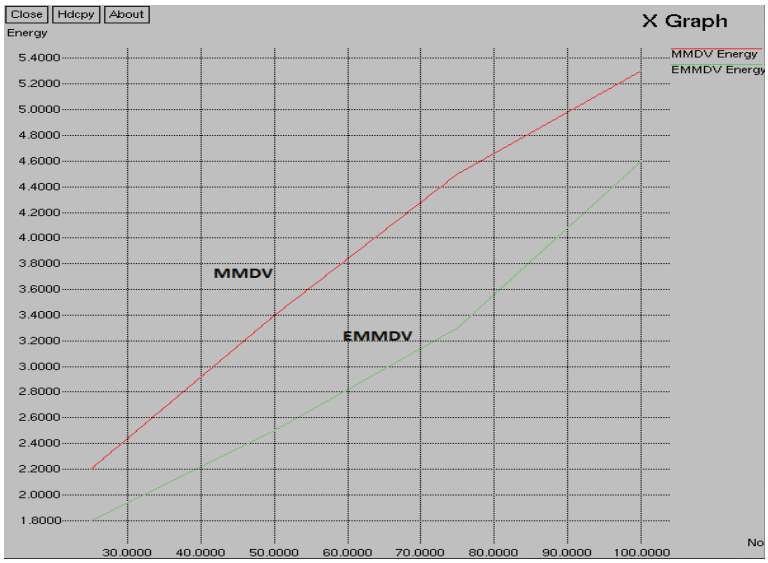

Fig.6.10 Energy Level

6.5. Summary

In this chapter, the performance of the two MANET Routing protocols such as MMDV and EMMDV was analyzed using NS-2.35 Simulator. These routing protocols were compared in terms of Packet delivery ratio, Average end-to-end delay, Throughput, Packet Loss and Energy when subjected to varying number of nodes. The motive was to check the performance of these two routing protocols in MANET in the above mentioned parameters. The conclusions of the experimental results are as follows:

1. The EMMDV protocol provides 97.02% PDR and MMDV protocol provides 80.87% PDR.
2. The EMMDV protocol is raises from 20.72% and it reaches a maximum level of 61.16%.But MMDV raises from 56.5% and it reaches 143.76%.

3. MMDV reaches Maximum throughput 80% at 100^{th} node. But EMMDV reaches maximum throughput 95% at 100^{th} node.
4. MMDV losses 57% of packets but EMMDV losses 2% of packet.
5. The EMMDV saves 80- 90% of energy when compared to MMDV.

The EMMDV protocol show better performance than the MMDV protocol in terms of certain performance metrics. The EMMDV protocol minimizes the duration of route discovery process which enhances the packet delivery ratio and overall performance when compared to that of conventional routing protocols. The experimental result shows that EMMDV scheme provides better performance than MMDV in most of the cases.