Simulation Based Comparison of Routing Protocols for Mobile Ad hoc Network

S.S. Tyagi
Department of Computer Science and Engineering
Manav Rachna International University
Faridabad

R.K. Chauhan
Department of Computer Science and Applications
Kurukshetra University, Kurukshetra

Abstract: *Mobile Ad hoc networks are the collection of wireless nodes that can exchange information dynamically among them without pre existing fixed infrastructure. Because of highly dynamic in nature, performance of routing protocols is an important issue. In addition to this routing protocols face many challenges like limited battery backup, limited processing capability and limited memory resources. Other than efficient routing, efficient utilization of battery capacity and security are the major concerns for routing protocols. This paper presents simulation based comparison and performance analysis on different parameters like PDF, Average e-e delay, Routing Overheads and Packet Loss. The study is about three main protocols DSR, AODV (Reactive) and DSDV (Proactive).*

Keywords: *Ad hoc Network, Routing Protocols, Performance, Simulation, DSR, AODV, DSDV, Average e-e delay (AED), PDF, Routing Overhead, Packet Loss*

1. INTRODUCTION

Nowadays wireless mobile nodes are becoming more and more capable and have improved a lot over those available in the past. But mobile nodes and their applications will become indispensable at the places where necessary infrastructure is not available. Ad hoc networks are the future of existing networks, where all the wireless mobile devices will be capable to communicate with each other in the absence of infrastructure. Ad hoc network allows all wireless devices within range of each other without involving any central access point and administration. Routing protocols are challenging to design as performance degrades with the growth of number of nodes in the environment and a large ad hoc network is difficult to manage. Proactive protocol DSDV is considered to be a traditional protocol which finds routes between all source – destination pairs regardless of the use or need for such routes. The key motivation behind the development of reactive routing protocols like DSR and AODV is the reduction of routing load. There will be impact on performance for low bandwidth wireless link if high routing load is there. Many simulation studies have been reported for the routing protocols. This paper has been organised as: the three protocols DSDV, DSR and AODV are first briefly reviewed and then description is given of the performance metrices which form the basis for the comparison of the protocols. This is followed by explanation of simulation model, the results are obtained and graphs generated. Next comparison and analysis of results is done with the help of performance metrices. The simulation based comparative performance analysis of routing protocols DSR, AODV (Reactive) and DSDV (Proactive) has been presented and finally conclusion has been made regarding which protocol is better under certain traffic conditions and scenarios.

2. AD HOC ROUTING PROTOCOLS

Ad hoc routing protocols are categorized in to Table driven routing protocols and Reactive routing protocol. DSDV is a Table driven (proactive) protocol while DSR and AODV are typical reactive protocols. Tables are maintained to store information for routing and updated through control packets in proactive algorithms. Updates also respond to the changes in topology of the network.

On demand (reactive) protocols AODV and DSR routes are computed to a specific destination only on demand or when needed. So it is not needed to maintain routing table containing all the nodes as entries do not have to be maintained in each node. Route discovery mechanism is initiated when a source wants to send a packet to the destination. Route remains valid till the destination is reachable or until the route is no longer required. Challenges and issues for base Ad hoc routing protocols are explained below:-

A. Destination Sequence Distance Vector (DSDV)

DSDV[5] is considered to be successor of distance vector in wired routing protocol and guarantees a loop free path to each destination. In this protocol every node maintains a routing table that contains next hop entry and no of hops needed for all reachable destinations. Each route table entry is tagged with a sequence number that is originated by the destination node. Each node transmits updates periodically to maintain the consistency in dynamic environment. Because with the time, entries in the list may change so the advertisement must be made often or periodically to each of its current neighbor nodes. When a mobile node receives new routing information, Either 'Full Dump' or 'incremental' that information is compared to the information already available from previous routing information packets. Any route with the recent sequence no is used and routes with older sequence number is discarded. When such updating takes place, each update is broadcasted in the network, which leads to a heavy network load situation and affects the bandwidth. With more number of nodes network load increases and deteriorates the situation. In response to the topology changes, mobile nodes may cause broken links and these broken links may be detected by layer-2 protocol.

B. *Dynamic Source Routing (DSR)*

DSR[2] is designed specifically for use in multi-hop wireless ad hoc network. This protocol is composed of two mechanisms of route discovery and route maintenance, which work together to allow nodes to discover and maintain source routes to arbitrary destinations in the ad hoc network. Route discovery takes place when source already does not know route to destination. Route cache is also maintained where all learned routes to any given node in the network exist. When a source sends a packet to destination, it obtains a route from route cache of previously learned routes. If no route is found then route REQUEST message is broadcasted to initiate route discovery protocol. When a node receives a route REQUEST message it returns route REPLY message to the initiator, if it is the target of the request. Simply when a node receives a route request it searches the route cache where all routes are stored. If not found then route REQUEST is broadcasted and flooded over the network until the destination node is found. In fact there is a aggressive use of source routing and caching in DSR. No special mechanism is needed to detect the routing loops. Although several optimization techniques have been proposed and have been evaluated very effective by the authors of the protocol[2] like Salvaging, Gratuitous route repai and promiscuous listening. Each route REQUEST message contains a hop limit that may be used to limit the number of intermediate nodes allowed to forward, that copy of the route REQUEST. As the REQUEST is forwarded limit is decremented and packet is discarded if limit reaches to zero. Another mechanism of expanding ring search for the target where a node can initiate another route REQUEST with hop limit of one. For each route REQUEST no route REPLY is received. Node can double the hop limit as previously attempted.

C. *Ad Hoc On Demand Distance Vector (AODV)*

AODV[3] mixes the properties of DSR and DSDV. Routes are discovered as on-demand basis and are maintained as long as they are required. Like DSDV it maintains a sequence number, which it increases each time it finds a change in the topology of its neighborhood. This sequence number ensures that the most recent route is selected for execution of the route discovery. AODV is able to provide unicast, multicast and broadcast communication ability. Combination of the three makes it an advantage protocol. AODV is capable of operating on both wired and wireless media, although it has been designed specifically for wireless domain. Route tables used by AODV store the destination and next hop IP addresses as well as the destination sequence number. AODV also provides quick deletion of invalid routes in response the route ERROR messages generated due to link breakage. If a node fails to receive three consecutive HELLO messages from a neighbor, it is concluded that link is broken for the specific node and a RERR message is broadcasted to any upstream node. In fact a more conservative routing table and sequence number driven approach is utilized in AODV.

3. SIMULATION MODEL

3.1 Experimental setup

We conducted the extensive simulation using ns2 simulator[1] and compared DSR, AODV and DSDV protocols. We have followed the general ideas as of previous study of simulation [RFC 2501]. In simulation, we first generate scenario files considering the area of 800mx1200m and divided them into four different categories as under:

1. Scenario files for varying number of nodes and keeping Pause time (2 sec), Max Speed(10m/s) and Simulation Time (1200 sec) constant. (9 files).
2. Scenario files for Varying Speed and keeping number of nodes (25), Pause Time and Simulation Time Constant. (11 files).
3. Scenario files for varying Pause Time and keeping number of nodes (25), Speed (10 m/s) and simulation Time (1200 sec) constant. (10 files).
4. Scenario files for varying Simulation Time and keeping number of nodes (25), Speed (10m/s) and pause time (100 sec) constant. (10 files).

3.2 Syntax for Generating the Scenario Files

```
./setdest [-n num_of_nodes] [-p pausetime] [-M maxspeed] [-t simtime] [-x maxx] [-y maxy] > [scenario_output_file]
```

Here above syntax can be explained as if we place 50 as num_of_nodes then there are 50 nodes under consideration, Similarly pausetime indicates pause between movement of nodes, maxspeed indicates maximum moving speed of nodes, simtime indicates that simulation will stop in this mentioned simtime, maxx and maxy define the topology boundary in x and y direction.

Senario Files for Varying Number of Nodes

Command Line

```
./setdest -n nn -p 2.0 -M 10.0 t 1200 -x 800 -y 1200 > scene-nn-test
```

Where nn= 10, 25, 50, 75, 100, 125, 150, 175, or 200

Using the above syntax we generated 9 scenario files as output is stored in their respective scenario files. Here in the above command topology boundary is 800mx1200m (800m in x direction and 1200m in y direction), pause between the movement of nodes is 2.0s maximum moving speed of nodes is 10.0m/s, Simulation will stop in 1200s and finally output will be stored in scenario file mentioned as scen-nn-test. Size of the file varies and depends on the number of nodes. Higher the number of nodes, more is the size of the file.

Scenario Files for Varying Speed

Command Line

```
/setdest -n 50 -p 2.0 -M m -t 1200 -x 800 -y 1200 > scene-ms-test
```

Where m = 10, 20, 30, 40, 50, 60, 70, 80, 90 or 100

Using the above syntax, we generated 10 scenario files as output is stored in their respective scenario files. Here again topology boundary is 800mx1200m, number of nodes are considered as 50, pause between the movement of nodes is 2.0s, simulation will stop in 1200s, whereas speed of nodes varies from 10m/s to 100m/s and finally output is stored in scenario file mentioned as scene-ms-test. Size of the file was found to be increased with the incresing speed of the nodes.

Scenario Files for Varying Pause Time

Command Line

```
./setdest -n 50 -p Pt -M 10.0 -t 1200 -x 800 -y 1200 > scene-Ptp-test
```

Where **Pt** = 0,100, 200, 300, 400, 500, 600, 700, 800, 900 or 1000

Using the above syntax we generated 10 scenario files, here again topology boundary is 800mx1200m, number of nodes are constant as 50 and maximum speed of nodes is 10.0m/s, simulation is to be completed in 1200s and pause between the movements of nodes is varying from 0s to 1000s. And finally output is stored in the output file mentioned as scen-**Ptp**-test. Size of the file was found to be increased with the higher pause time between the movement of nodes.

Scenario Files for Varying Simulation Time

Command Line

```
./setdest -n 50 -p 2.0 -M 10.0 -t S -x 800 -y 1200 > scene-St-test
```

Where **S** = 300, 400, 500, 600, 700, 800, 900, 1000, 1100 or 1200

Using the above syntax, we generated 10 scenario files, here again topology boundary is 800mx1200m, number of nodes are constant as 50, pause between movement of nodes is 2.0s, maximum speed of nodes is 10.0m/s, whereas simulation time varies between 300s to 1200s in the interval of 100s. And finally output is stored in the output file mentioned as scen-St-test. Size of the file increases with increase in simulation time.

After generating the scenario files we generated traffic files using *cbrgen* utility of ns2. The number of maximum connections was mentioned as number of nodes for a particular file and data communication rate was defined as 4 packets per second. 9 traffic files were generated for the varying number of nodes ranging from 10 nodes to 200 nodes. The procedure for generating the traffic files is as under.

3.3 Traffic File Generation

To generate the network traffic we made use of cbrgen utility of ns2. This was done as tcl file, which is in

$NS2_HOME/indep-utils/cmu-scen-gen/cbrgen.tcl

For this network traffic generating tool, random traffic connections of TCP and CBR can be setup between nodes. For creating CBR and TCP traffic-connection files, we need to define the type of traffic connections (CBR or TCP), the number of nodes and maximum number of connections to be setup between them, a random seed and in case of CBR connections, a rate whose inverse value is used to compute the interval time between the CBR packets. So the command line is:

Syntax for generating Traffic file

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

In the above syntax, "type cbr/tcp" defines the type of traffic connections either cbr or TCP. "-nn nodes" means the number of nodes could be used, "-seed seed" means a random seed, if it is not equal to 0, the traffic pattern will reappear if all the other parameters are the same. "-mc connections" means maximum number of connections to be setup between those nodes, "-rate rate" means a rate whose inverse value is used to compute the interval time or say packet sending rate. "cbr-nn-nn-rate" is the file used to store the output.

CBR Traffic generation for 10 nodes

Command Line

```
ns cbrgen.tcl -type cbr -nn 10 -seed 1.0 -mc 10 -
rate 4.0 > cbr-10-10-4
```

CBR traffic generation for 25 nodes

Command Line

```
ns cbrgen.tcl -type cbr -nn 25 -seed 1.0 -mc 25 -
rate 4.0 > cbr-25-25-4
```

CBR traffic generation for 50 nodes

Command Line

```
ns cbrgen.tcl -type cbr -nn 50 -seed 1.0 -mc 50 -
rate 4.0 > cbr-50-50-4
```

CBR traffic generation for 75 nodes

Command Line

```
ns cbrgen.tcl -type cbr -nn 75 -seed 1.0 -mc 75 -
rate 4.0 > cbr-75-75-4
```

CBR traffic generation for 100 nodes

Command Line

```
ns cbrgen.tcl -type cbr -nn 100 -seed 1.0 -mc 100
-rate 4.0 > cbr-100-100-4
```

CBR traffic generation for 125 nodes

Command Line

```
ns cbrgen.tcl -type cbr -nn 125 -seed 1.0 -mc 125
-rate 4.0 > cbr-125-125-4
```

CBR traffic generation for 150 nodes

Command Line

```
ns cbrgen.tcl -type cbr -nn 150 -seed 1.0 -mc 150
-rate 4.0 > cbr-150-150-4
```

CBR traffic generation for 175 nodes

Command Line

```
ns cbrgen.tcl -type cbr -nn 175 -seed 1.0 -mc 175
-rate 4.0 > cbr-175-175-4
```

CBR traffic generation for 200 nodes

Command Line

```
ns cbrgen.tcl -type cbr -nn 200 -seed 1.0 -mc 200
-rate 4.0 > cbr-200-200-4
```

Nine (9) files were generated to store the traffic (CBR) output. Each file was utilized to generate the trace file which later on was used to produce the actual output on the basis of which we analyzed the performance of the different protocols.

Before starting the simulation it was ensured that the computer system was having a good processing speed and large storage capacity as 120 trace files were generated and each file was of the capacity in the range of 1gigabyte to 50 gigabytes. Tcl script was run over to generate the trace files for various protocols DSR, AODV and DSDV. Also it was very time consuming as some simulation took approximate 15-20 hours to generate a single trace file especially in case of higher number of nodes. After analyzing these 120 file trace files with awk script we concluded the results for various parameters to be calculated and plotted the appropriate graphs.

Every simulation was done for 1200 seconds (20 minutes).

3.4 Metrics

a. *Packet Delivery Ratio* : The ratio of the data packets delivered to the destinations to those generated by the CBR sources.

b. *Average e-e delay* : Average amount of time taken by a packet to go from source to destination. This includes all possible delays caused by buffering during route discovery latency, queuing at the interface queue, retransmission delays at the MAC and propogation and transfer times.

c. *Packet Loss* : It is the measure of the number of packets dropped by the routers due to various reasons.

d. *Routing Overhead* : The ratio between the total number of routing packets transmitted to data packets.

Simulation Paramaters

Protocols	DSDV, AODV, DSR
Simulation Area	800X1200 m
Simulation Time	1200sec (20 Min)
Varying No of Nodes	10 –200
Transmission Range	250m
Mobility Model	Random Way Point
Varying Speed	(10-100) m/s
Varying Pause Time	(0-1000)sec
Type of Traffic	CBR
Size of Payload	512 bytes
Packet Rate	4 Packets/sec
Maximum Connection	25

III Graphical Results

Case 1 Results for varying number of nodes and constant pause time, Speed and Simulation Time

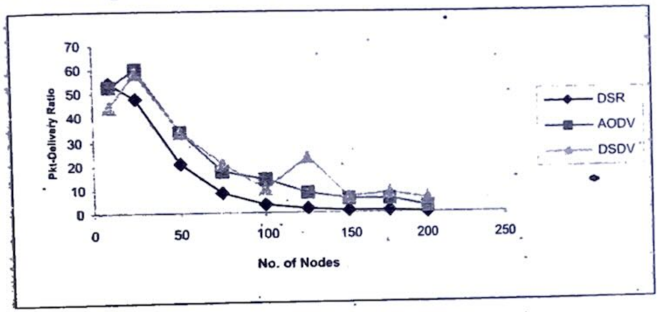

Fig 1. Number of nodes *vs* PDF

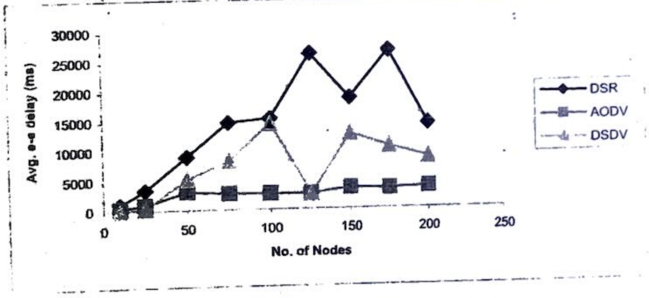

Fig. 2. Number of nodes *vs* AED

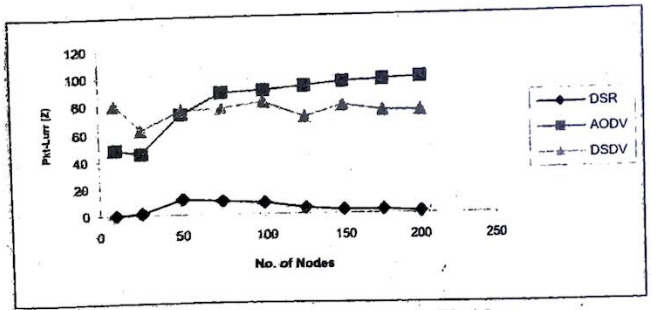

Fig. 3. Number of nodes *vs* Pkt. loss

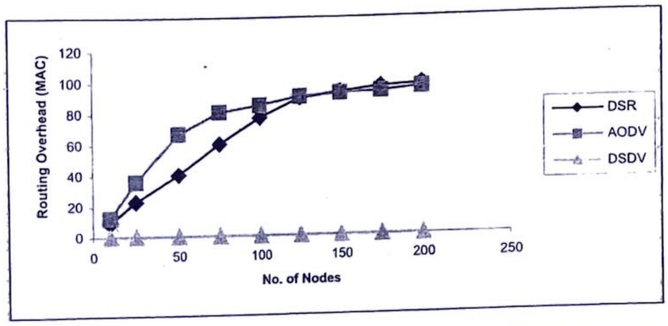

Fig 4. Number of nodes *vs* routing overhead (MAC Layer)

Case 2 Results for varying Pause Time and constant number of nodes, Speed and Simulation Time.

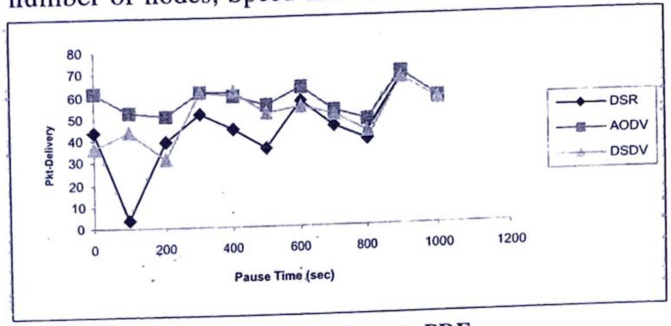

Fig. 5. Pause time *vs* PDF

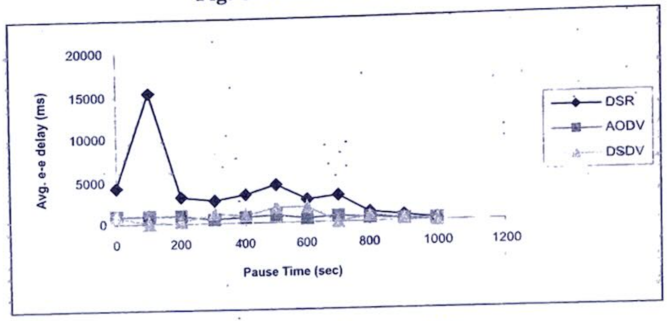

Fig. 6. Pause time *vs* AED

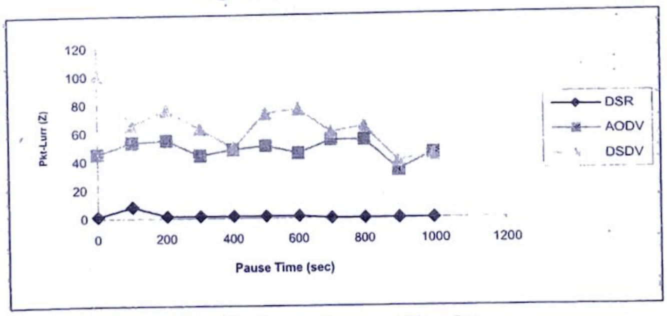

Fig. 7. Pause time *vs* Pkt. loss

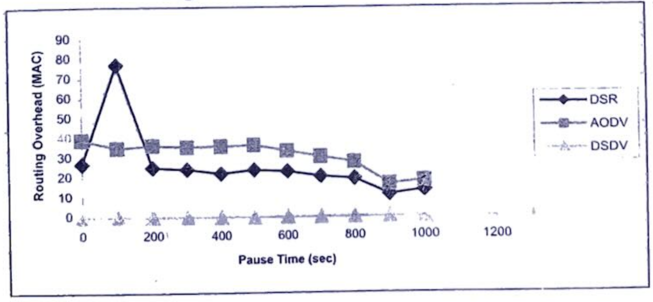

Fig. 8. Pause time *vs* routing overhead (MAC Layer)

Case 3 Results for varying Speed and constant number of nodes, Pause Tme and Simulation Time.

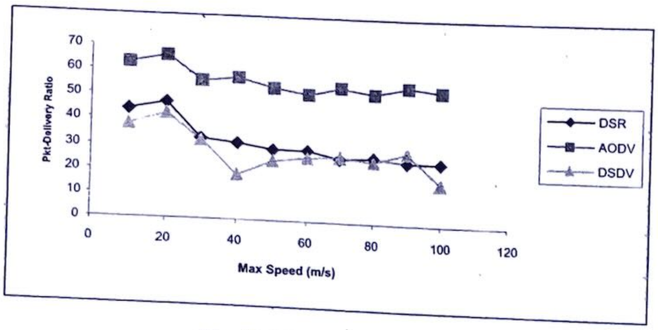

Fig. 9. Max speed *vs* PDR

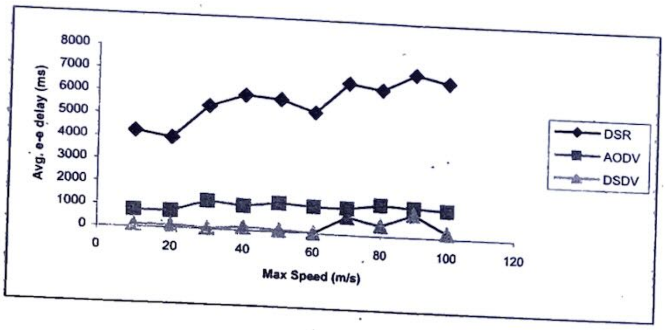

Fig. 10. Max speed *vs* AED

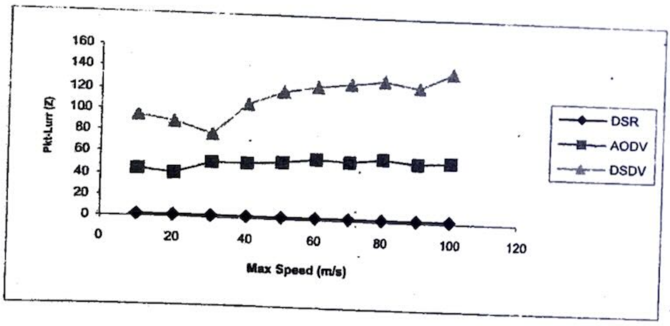

Fig. 11. Max speed *vs* Pkt. loss

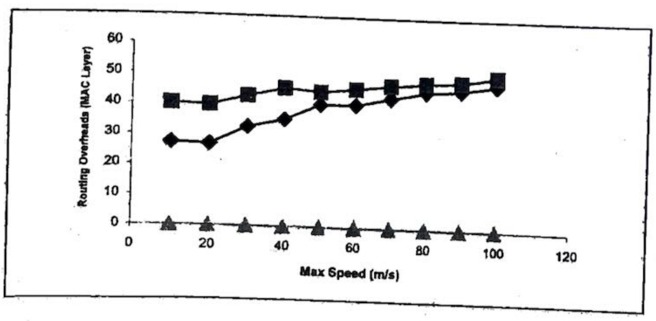

Fig. 12. Max speed *vs* routing overhead (MAC Layer)

Case 4 Results for varying Simulation Time and constant number of nodes, Pause Time and Speed.

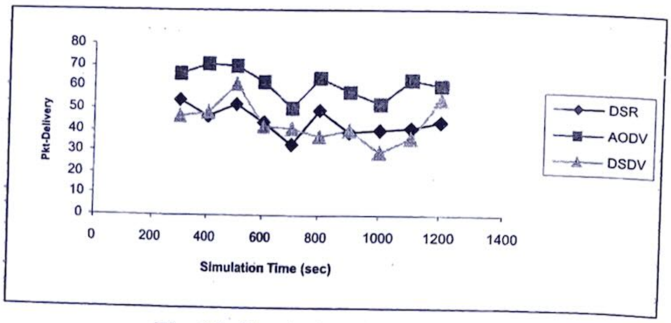

Fig. 13. Simulation time *vs* PDF

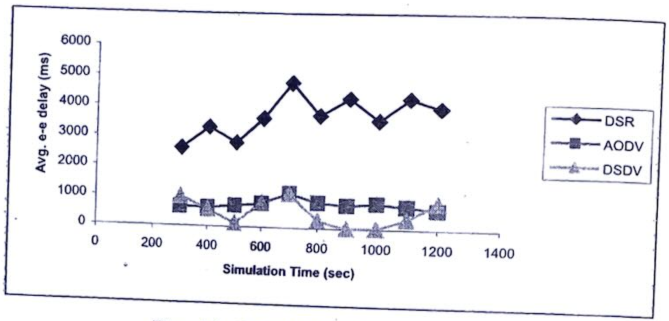

Fig. 14. Simulation time *vs* AED

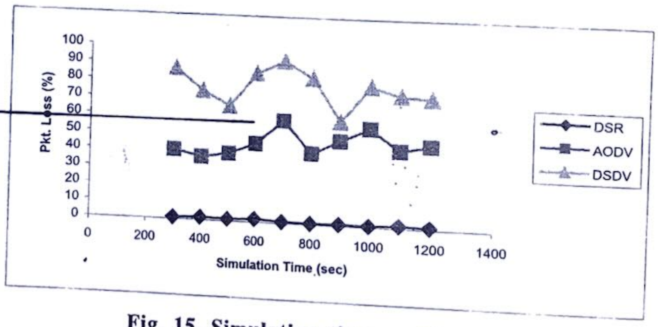

Fig. 15. Simulation time *vs* Pkt. loss

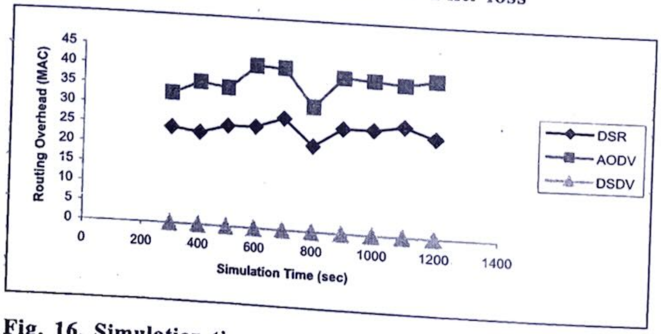

Fig. 16. Simulation time *vs* routing overhead (MAC Layer)

3.5 Analysis of results

a. Packet Delivery Fraction

It is the ratio of data packets delivered to the destinations to those generated by the CBR sources. The PDF tells about the performance of a protocol that how successfully the packets have been delivered. Higher the value gives the better results. In our simulation it has been noticed that AODV outperforms DSR and DSDV in almost all the scenarios we have taken into account. It has been concluded that performance of DSR decreases with the increasing number of nodes as DSR is designed for up to two hundred nodes.

b. Average end to end delay

Average end-to-end delay is an average end-to-end delay of data packets. Buffering during route discovery latency, queuing at interface queue, retransmission delays at the MAC and transfer times, may cause this delay. Once the time difference between every CBR packets sent and received was recorded, dividing the total time difference over the total number of CBR packets received gave the average end-to-end delay for the received packets. Lower the end to end delay, better is the performance of the protocol. From the graphs it is very clear that AODV and DSDV out perform DSR for the scenarios of varying pause time, varying simulation time, varying speed and varying number of nodes. In case of DSR and DSDV delay time increases very sharply with the increasing number of nodes while AODV is consistent with the increasing number of nodes.

c. Packet Loss

It is defined as the difference between the number of packets sent by the source and received by the sink. In our results we have calculated packet loss at network layer as well as at MAC layer. The routing protocol forwards the packet to destination if a valid route is known, otherwise it is buffered until a route is available. There are two cases when a packet is dropped: the buffer is full when the packet needs to be buffered and the time exceeds the limit when packet has been buffered. Lower is the packet loss, better is the performance of the protocol. In case of DSR packet loss is minimum in all the cases as compared to AODV and DSDV. DSDV is having the poorest performance and having the maximum packet losses in case of varying pause time, varying speed and varying Simulation Time. AODV was ahead of DSDV up to around 50 nodes but packet loss was more than DSDV beyond 50 nodes.

d. Routing Overhead

Routing overhead has been calculated at the MAC layer which is defined as the ratio of total number of routing packets to data packets. From the critiques point of view DSR makes use of caching aggressively and replies to all requests reaching the destination from a single request cycle. Thus source learns many alternate routes to destination. Having access to many alternate routes saves flooding of route discovery which is a performance bottleneck. In comparison of AODV, DSR has performed well and supported the previous work. Conceptually routing overheads are negligible in case of DSDV and our results supported the same. DSR performed well in most of the cases when number of nodes were less and around 100 for the particular scenario but AODV outperforms DSR when number of nodes are above 100. Since AODV is having more routing control packets than DSR, routing overhead of AODV is always higher even in stressful environment. It has been concluded from the results that AODV outperforms DSR under heavy load, as routing overheads are more for DSR when number of nodes are more.

4. CONCLUSION

In this paper analysis and investigations are carried out on acquired simulation results of three prominent protocols, AODV, DSR and DSDV using ns2. DSDV is selected as representative of proactive routing protocol while AODV and DSR are the representative of reactive routing protocols. As AODV is designed for up to thousands of nodes while DSR is designed up to two hundred nodes. AODV performed better in dense environment except packet loss. DSR and AODV both performed well. AODV and DSR are proved to be better than DSDV. While it is not very clear that any one protocol is best for all the scenarios, each protocol is having its own advantages and disadvantages and may be well suited for certain scenarios. Although the field of Ad hoc network is rapidly growing and new developments are coming day by day, still there are many challenges to be met.

REFERENCES

[1] Network Simulator- ns-2. http://www.isi.edu/nsnam/ns/.

[2] D.B. Johnson and D.A. Maltz "Dynamic Source Routing in Ad hoc Wireless Network", Mobile Computing, vol. 353, Kluwer Academic 1996.

[3] C.E. Perkins and E.M Royer, "Ad Hoc On-Demand Distance Vector Routing", Proc 2nd IEEE Workshop Mobile Comp. Sys and Apps., New Orlean LA, Feb 1999, pp. 90-100.

[4] C.E. Perkins, E.M. Belding-Royer, and S.R. Das," Ad Hoc On-Demand Distance Vector Routing", IETF RFC 3561, July 2003.

[5] Charles E. Perkins, Praveen Bhagwat. Highly Dynamic Destination- Sequenceed Distance-Vector Routing (DSDV) for Mobile Computers. In Proceedings of the SIGCOMM 94, Aug 1994.

[6] Guntupalli Lakshmikant, A Gaiwak, P.D. Vyavahare, "Simulation Based Performance Analysis of Adhoc Routing Protocols". In proceedings of TENCON 2008.

[7] E.Royer, C.K.Toh "A Reiview of current routing protocols for Adhoc Mobile wireless network" RFC 2409, IEE personal communication, 1999.

[8] Chales E. Perkins "Ad Hoc Networking" published by Dorling Kindersley (India) Pvt Ltd, 2008

[9] Yinfei Pan, "Design Routing Protocol Performance Comparision in NS2: AODV Comparing to DSR as Example, Deptt of CS, SUNY Binghamton, Vestal NY 13850

[10] Yih-chun Hu, David B. Johnson, "Caching strategis in On-Demand Routing Protocols for Wireless Ad Hoc Network", Carnegie Mellon University, Pittsburgh

[11] Elizabeth Royer, C.K. Toh, "A Review of Current Routing Protocols for Ad hoc mobile Wireless Networks", RFC 2409, IEE Personal Communications 1999.

[12] Charles E. Perkins, Praveen Bhagwat, "Highly Dynamic Destination- Sequenceed Distance-Vector Routing (DSDV) for Mobile Computers", In Proceedings of the SIGCOMM 94, Aug 1994.

[13] C. Siva Ram Murthy, B.S. Manoj, "Ad hoc Wireless Networks 'architectures and protocols'", Pearson Publication

[14] E.Royer, C.K.Toh "A Reiview of current routing protocols for Adhoc Mobile wireless network" RFC 2409, IEE personal communication, 1999.

[15] Guntupalli Lakshmikant, A Gaiwak, P.D. Vyavahare, "Simulation Based Performance Analysis of Adhoc Routing Protocols". In proceedings of TENCON 2008.

[16] J. Broch, D. Maltz, D. Johnson, Y. Hu, and J. Jetcheva, "A Performance Comparison of Multihop Wireless Ad Hoc Network Routing Protocols," Proceedings of the Fourth Annual ACM/ IEEE International Conference on Mobile Computing and Networking (Mobicom 98), October 1998.