International Journal of Scientific & Engineering Research, Volume 4, Issue 5, May-2013 2257
ISSN 2229-5518

Performance Evaluation of Ad-hoc Routing Protocols

M. M. Chandane, S. G. Bhirud, S.V. Bonde

Abstract— A wireless ad hoc network is a decentralized wireless network. The network is ad hoc because it does not rely on a preexisting infrastructure, such as routers in wired networks or access points in managed (infrastructure) wireless networks. Instead, each node participates in routing by forwarding data for other nodes, and so the determination of which nodes forward data is made dynamically based on the network connectivity. Routing protocols plays the important role for the resource requirement for routing. Mobility and scalability has become the crucial parameters for today's ad-hoc network.Objective of this paper is to study various ad-hoc routing protocols and evaluate their performance in terms of scalability and mobility using various performance metrics such as packet loss, throughput, and jitter so as to decide the usability of these protocols. The protocols under comparison are Ad-hoc On-demand Distance Vector Protocol (AODV), Destination Sequence Distance Vector Protocol (DSDV) and Dynamic Source Routing Protocol (DSR). Result shows that AODV performas better than the other two protocols with varying number of nodes and mobility.

Index Terms—: *M*obile Ad-hoc network, Routing, Simulation, Throughput, Jitter, Packet Delivery Fraction (PDF), Normalized Routing Overhead.

1 Introduction

NETWORKING is the place where users can communicate with each other by sending data packets over the network. These data packets are sent to the receiver through routing. Routing is usually performed by a dedicated device called a Router. Routing is a key feature of Internet because it enables messages to pass from one computer to another and eventually reach the target machine. Each intermediary computer performs routing by passing along message to the next computer. Part of this process involves analyzing the routing protocol to determine the best path. Numerous routing protocols have problems while establishing and maintaining the routes in dynamic topology. However some may perform well while finding the easiest and the best path to destination thus giving performance benefit.

A "mobile ad-hoc network" (MANET) is a system of mobile nodes connected through wireless links wherein random movement, arbitrary organization of nodes takes place. Thus network topology is continuously changing. Because of large scalability and mobility, adeqaute routing protocols need to be designed. MANET has derived two major categories of protocols [1]: on-demand such as (AODV) and DSR, and proactive such as DSDV. These two major categories can also be confirmed as the dynamic and static protocols. Conventional protocols are mostly inflexible to be used in ad-hoc mobile networks, as they perform in very strict manner. They are not able to adapt themselves to changing conditions. Contrary the dynamic protocols are well defined and adaptable to observe the changes and respond to them in particular situation.

MANET is characterized by no fixed infrastructure. The network is formed without any preplanning. It should be able to operate in isolation without any reliance on infrastructure based services. Fig. 1 shows the example of simple MANET wherein node A wants to communicate with node C, which is outside of its range, therefore node A has to use the services of node B since node B's range overlaps with that of node A and node C. Routing problem in real ad-hoc network may be more complicated than this simple scenario because of the node mobility and the scalability.

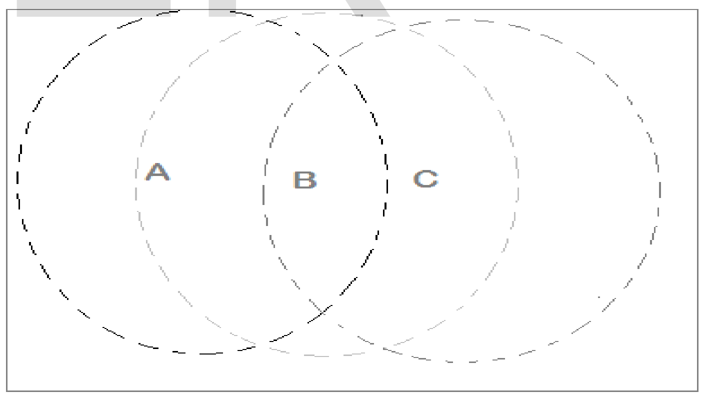


Fig. 1 Simple Ad-hoc Network with Three Nodes

This paper is organized as follows: Section 2 describes the working of AODV, DSDV and DSR protocols. Section 3 presents the performance measurement methodology and available tools. Section 4 gives the simulation setup and performance parameters whereas section 5 describes the simulation results and its analysis. Finally section 6 concludes the paper

2 Working Principle of DSDV, AODV and DSR

Many routing protocols have been developed which support establishing and maintaining multi-hop routes between nodes in MANETs. Fig. 2 show the brief classification of available

- *M M Chandane is currently Assistant Professor in Computer Technology Dept. of VJTI, Mumbai University, India, PH-022 24198164. E-mail: mmchandane@vjti.org.in*
- *S.G. Bhirud is currently Associate Professor in Computer Technology Dept. of VJTI. Mumbai University, India, PH-022 24198164. E-mail:sgbhirud@vjti.org.in*
- *S.V. Bonde is currentlyProfessor in E&TC Dept. of SGGSIE&T,SRTM University,India, E-mail:svbonde11@yahoo.com*

ISSN 2229-5518

routing protocol [2]. As said earlier, protocols are been divided into categories like: proactive and reactive. In the on-demand protocols (reactive protocols), routes are discovered only when the need arises. This provides a reduced overhead of communication and supports for scalability. In the table-driven protocols, routing table contains routing information which is generated and maintained continuously regardless of the need of any given node to communicate at that time. With this approach, the latency for route acquisition is relatively small, which might be necessary for certain applications, but the cost of communications overhead incurred in the continued update of information for routes which might not be used for long time if at all is too high. Furthermore, this approach requires more memory due to significant increase in the size of the routing table. These requirements put limits on the size and density of the network.

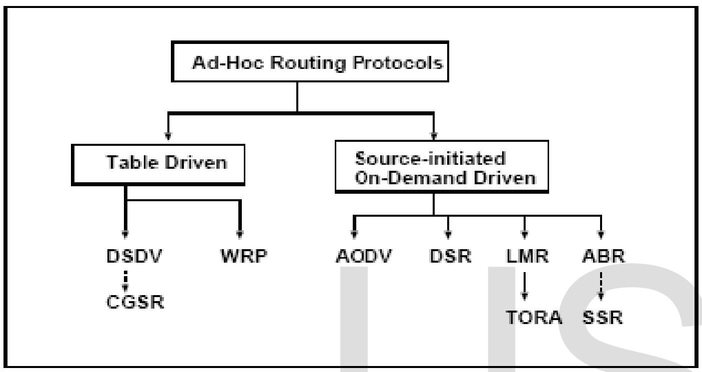


Fig. 2 Classification of Routing Protocols [2]

2.1 Destination-Sequenced Distance-Vector Routing

Destination-Sequenced Distance-Vector Routing (DSDV) is a table-driven routing scheme for ad hoc mobile networks based on the Bellman-Ford algorithm. It was developed by C. Perkins and P.Bhagwat in 1994 [3]. It is a distance vector protocol in which every node i maintains for each destination x a set of distances {dij(x)} for each node j that is a neighbor of i. Node i treats neighbor k as a next hop for a packet destined to x if dik(x) equals minj{dij(x)} [4]. The succession of next hops chosen in this manner leads to x along the shortest path. In order to keep the distance estimates up to date, each node monitors the cost of its outgoing links and periodically broadcasts to all of its neighbors its current estimate of the shortest distance to every other node in the network. The distance vector which is periodically broadcasted contains one entry for each node in the network which includes the distance from the advertising node to the destination. The distance vector algorithm described above is a classical Distributed Bellman-Ford (DBF) algorithm [2].

The main contribution of the algorithm was to solve the Routing Loop problem. Each entry in the routing table contains the sequence number. The number is generated by the destination, and the emitter needs to send out the next update with this number. Routing information is distributed between nodes by sending full dumps infrequently and smaller incremental updates more frequently. Earlier research's have found that, it is quite suitable for creating ad hoc networks with small number of nodes as DSDV requires a regular update of its routing tables, which uses up battery power and a small amount of bandwidth even when the network is idle. Whenever the topology of the network changes, a new sequence number is necessary before the network re-converges.

2.2 Ad-hoc on-Demand Distance Vector Routing

The Ad hoc On Demand Distance Vector (AODV) routing algorithm is a routing protocol designed for ad-hoc mobile networks [5][6] . AODV is capable of both unicast and multicast routing [6]. It is an on demand algorithm, meaning that it builds routes between nodes only when it is required. It maintains these routes as long as they are needed by the sources. Additionally, AODV forms trees which connect multicast group members. The trees are composed of the group members and the nodes needed to connect the members. AODV uses sequence numbers to ensure the freshness of routes. It is loop-free, self-starting, and scales to large numbers of mobile nodes [5].

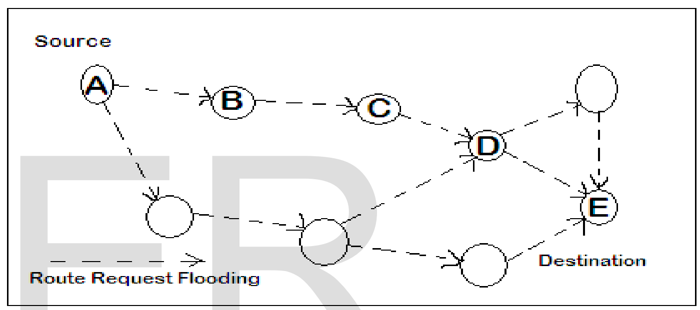


Fig. 3 Route Request Flooding [7]

AODV builds routes using a Route request / reply query cycle. When a source node desires a route to a destination, it broadcast a route request packet (RREQ) [6] across the network. Fig 3 shows the process of route request flooding. Nodes receiving this packet update their information and set up backward pointers to the source node. RREQ also contains the most recent sequence number for the destination of which the source node is aware. A node receiving RREQ may send a route reply (RREP) [6] if it is either the destination or if it has a route to the destination with corresponding sequence numbers greater than or equal to that contained in the RREQ. If this is the case, it uni-cast RREP back to the source. Otherwise, it re-broadcasts RREQ. If they receive RREQ which they have already processed they discard the RREQ. As the RREP propagate back to the source, nodes set up forward pointers to the destination. Fig 4 shows the process of route reply. If the source receives the RREP containing the greater sequence number or contains the same sequence number with a smaller hop count, it may update its routing information for that destination. As long as the route remains active, it will continue to be maintained. Once the source stops sending data packets, the link will time out and eventually will be deleted from the intermediate node's routing tables. If a link break occurs while the route is active, the node up-stream of the break propagates a route error (RERR) message to source node to inform it of the now unreachable destination(s) [6]. After receiving the RERR, if the source node still desires the route, it can re-initiate the route discovery.

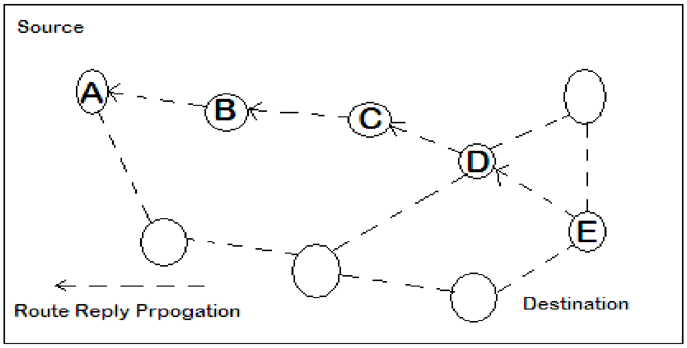


Fig. 4 Route Reply Propagation [7]

2.3 Dynamic Source Routing Protocol

Dynamic Source Routing Protocol (DSR) [8] is one of the well known routing algorithms for ad hoc wireless networks [7]. It was originally developed by Johnson, Maltz, and Broch. DSR uses source routing. Source routing is a routing technique in which the sender of a packet determines the complete sequence of nodes through which it has to forward the packet. The sender node explicitly lists this route in the packet's header, identifying each forwarding "hop" by the address of the next node to which to transmit the packet on its way to the destination host. DSR is used especially in situations where the mobility is low.

The basic algorithm can be seen in the following manner. A source node first searches its route cache to see if it already has a route to the destination. If it does not, it then initiates a route discovery mechanism. This is done by sending a Route Request message. When some intermediate node gets RREQ message, it searches its own cache to see if it has a route to the destination. If it does not, it then appends its id to the packet and forwards the packet to the next node; this continues until either a node with a route to the destination is encountered (i.e. has a route in its own cache) or the destination receives the packet. In that case, the node sends a RREP packet which has a list of all of the nodes that forwarded the packet to reach the destination. This constitutes the routing information needed by the source, which can then send its data packets to the destination using this newly discovered route [1][2]. If fatal transmission occurs, route maintenance is initiated to send route error packets. If some nodes in the route cache are erroneous then all routes containing that node are truncated at that point. In this case, route discovery is initiated again to determine most suitable route to destination.

3 PERFORMANCE MEASUREMENT METHODOLOGY

This section gives the brief information about different techniques available for evaluation of network performance. Performance is a key criterion in the design, procurement and use of computer systems. Many computers, professionals, scientists, analysts and users need knowledge of performance evaluation technique so that highest performance could be obtained for a given cost. There are three techniques such as analytical modeling; simulation and test bed implementation are available for performance measurement. This work uses the simulation method therefore the same is explained in detail in following section.

3.1 Simulation Tools

Simulation is the most suitable technique to get more details about the system. The simulation model needs less assumption, accuracy of the result is good, time required for evaluation and the cost incurred is low as compared to analytical modeling. Following simulators are available as Network Simulators: Network Simulator-2 (NS-2), OPNET Modeler, GloMoSim, OMNeT++ etc. All these simulators except NS-2 are proprietary. Buying and installation cost required for these propriatory items is too high.

We have used NS2.29.3 to perform the experiments. NS-2 is an object-oriented, discrete event driven network simulator developed at UC Berkely and written in C++, OTcl. NS-2 is primarily useful for simulating local and wide area networks. NS-2 supports for networking research and educationist and is also suitable for designing new protocols, comparing different protocols and traffic evaluations. It is open source and is freely distributed. Fig. 5 shows the overview of simulation model of NS-2.

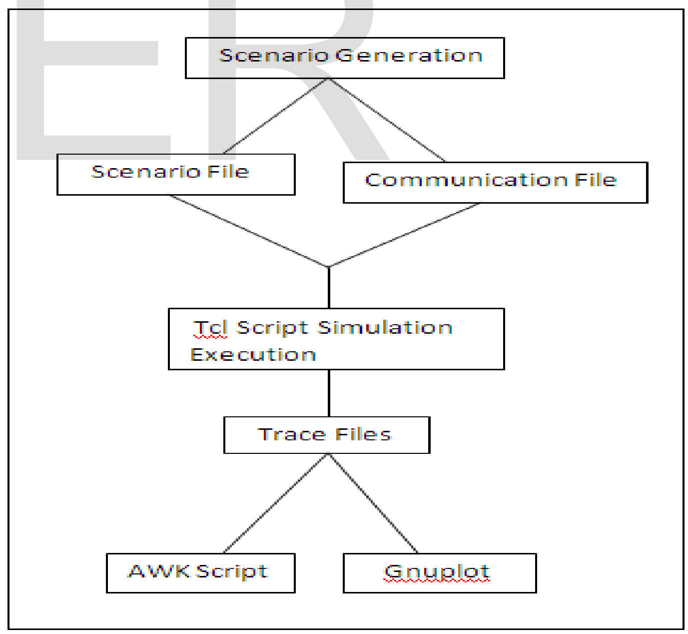


Fig. 5 Overview of Simulation Model

NS-2 uses two languages because any network simulator, in general, has two different kinds of things it needs to do. Detailed simulations of protocols require a systems programming language which can efficiently manipulate bytes, packet headers, and implement algorithms that run over large data sets. For these tasks run-time speed is important and turn-around time (run simulation, find bug, fix bug, recompile, re-run) is less important. On the other hand, a large part of network research involves slightly varying parameters or configurations, or quickly exploring a number of scenarios. In these cases, iteration time (change the model and re-run) is more important. Since configuration runs once (at the beginning of the simulation), run-time of this part of the task is less important [4]. We first create a scenario generation file [9][10][11] where in we describe the location of each node along with mobility and traffic generators well as receivers. After execution of scenario, output is placed in trace files i.e. *.tr [12]. Contents of * .tr files are analyzed using awk utilities and finally outcome of awk could be used to plot graphs [12].

4 SIMULATION ENVIRONMENT

These simulation experiments are carried out using NS-2. We have used a grid of 500*500 meters for the placements of nodes. Packet size of 512 bytes is used for these experiments and the simulation time varied from 6 to 30 seconds. Following parameters are used to analyze simulation environment and to compare the result

Throughput: It is directly proportional to number of packets received by the receiver. It represents the number of packets received within the given time interval.

Jitter: Jitter is measured on the basis of end to end delay. It basically depends on the send-time and received time of the packet. However difference between current and previous delay gives the jitter.

Packet Delivery Fraction: The ratio of the data packets delivered to the destinations to those generated by the CBR sources is known as packet delivery fraction.

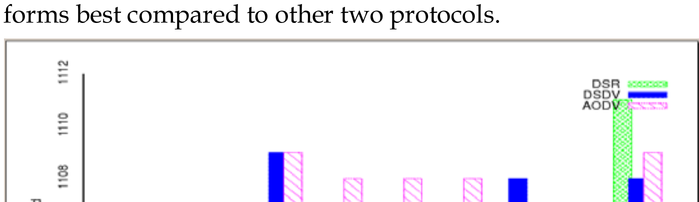
End-to-End Delay: End to end delay includes all possible delays caused by buffering during route discovery latency, retransmission delays at the MAC, queuing at the interface queue, and propagation and transfer times of data packets.

Normalized Routing Overhead: The number of routing packets transmitted per data packet delivered at the destination. Each hop -wise transmission of a routing packet is counted as one transmission.

5 RESULTS AND DISCUSSION

This section presents the results obtained for the experiments. Results are obtained using NS-2 with simulation parameters as discussed in previous section.

Fig 6 and fig 7shows number of packets received and packet loss respectively with constant simulation time of 6 seconds and number of nodes varying from 2 to 10 at different locations and with UDP environment. From fig 6 it can be seen that the more number of packets are being received by the receiver in case of AODV. Thus even in increasing order of nodes, AODV out performs DSDV and DSR and it can adapt to changes quickly as it only maintain one route that is actively used. DSDV delivers lesser data packet compared to AODV because in rapidy changing topology, it is not as adaptive as AODV to route changes and in updating its table. DSR does not have mechanism to know which route in cache is stale, thus maximum packet loss occur in DSR. Thus AODV performs best compared to other two protocols.



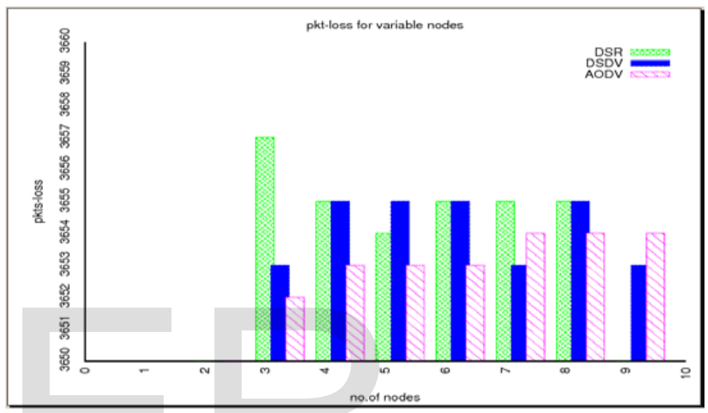


Fig. 7 Packet Lost Vs Number of Nodes

Fig 8, in case of DSR, it is found that throughput increases over a period of time but it generates spikes at certain time which shows a lack of consistency at certain time period. For DSDV protocol, first the throughput decreases and remains constant for some time. However it is not consistant over the period of time. The AODV shows the consistant behavior over a period of time for different number of nodes. Thus AODV shows a better performance as compared to other two protocols. Fig 9 shows the analysis of Jitter. It represents any unwanted variations in one or more signals generated during the packets transfer. For AODV protocol it is found that average initial jitter varies from 0 to 0.5. For DSR it can be observed that the jitter varies from 1 to 3 and continuously varying. Even this shows many spikes, thus shows its inconsistency. For DSDV protocol, the average jitter is less than 0.3, which is the lowest of the three. Although the jitter for three protocols does not show much difference, however in network consideration even small amount of jitter can cause significant delay. Thus it can be seen that DSDV protocol performs better than other two in case of jitter. The highest jitter could be seen in case of DSR protocol. However the modrate behavior is shown by AODV protocol.

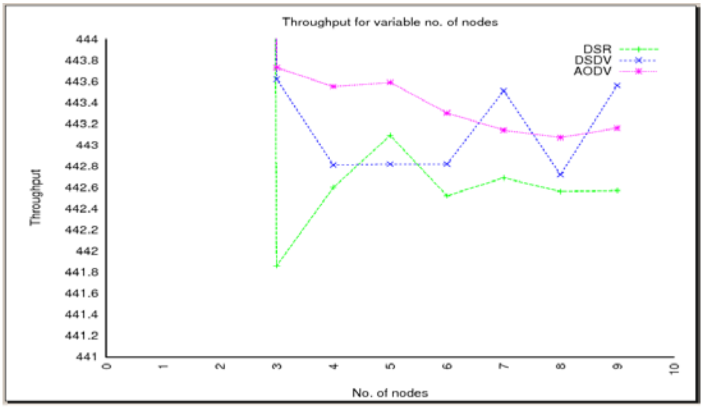


Fig. 8 Throughput Vs Number of Nodes

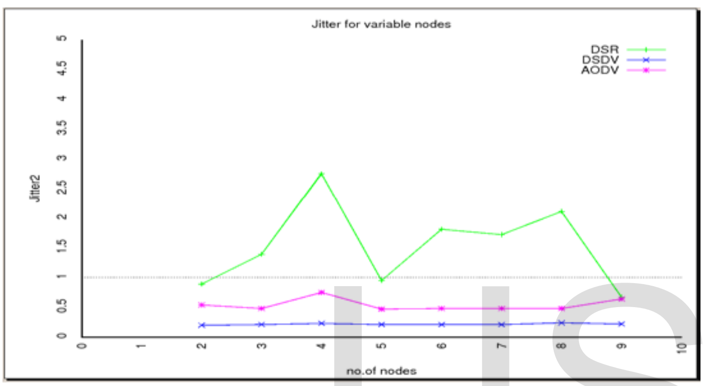


Fig. 9 Jitter Vs Number of Nodes

Normalized routing overhead analysis is shown in fig. 10. It shows that as the number of nodes increases, the static networks shows the sudden growth in routing overhead whereas the dynamic netwok shows the consistent behavior.
The packet drop rate with mobility is shown in fig 11. Figure indicate that the increase in pause time or making node static for a specific time interval, the packet drop rate decreases. So we can say that the packet drop rate increases with the increased mobility speed.

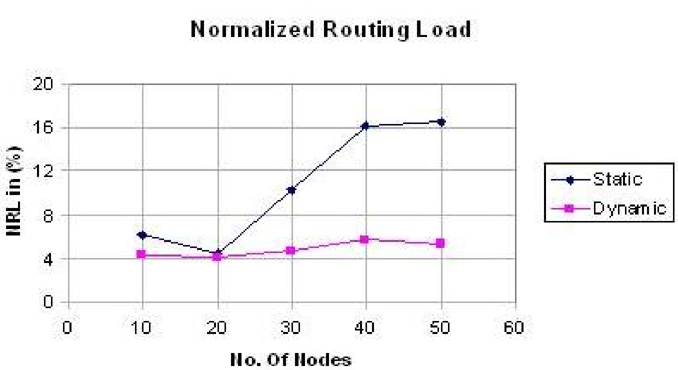


Fig. 10 Normalized Routing Load with Varying Network Size

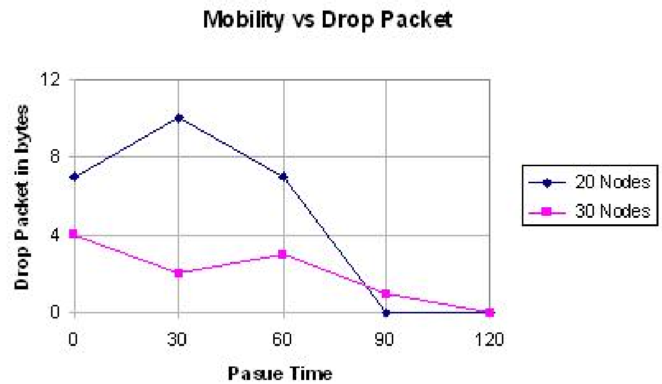


Fig. 11 Drop Packets with Mobility for Varying Network Size

6 Conclusion

This paper does the simulation based comparison study of three routing protocols DSDV, AODV and DSR. The significant observation is that the simulation results agree with the expected results based on theoretical study. Considering the static and dynamic nature of the protocols it is true that the reactive routing protocol for example AODV performas better in UDP environment than the other two protocols. Also for mobility and varying number of nodes, AODV has showed better performance
The DSDV requires a regular update of its routing tables which uses up battery power and a small amount of bandwidth even when the network is idle. Whenever the topology of the network changes, a new sequence number is necessary before the network re-converges; therefore, DSDV shows poor performance for dynamic networks. DSDV performs almost at par with DSR, but it requires transmission of many routing overhead packet. However DSDV is more expensive than DSR at higher rates of mobility.
Within a trial for mobility rates it was found that DSR showed good performance at all mobility rates and movement speed. The DSR protocol is also not far lbehind than AODV however comparatively DSR lags behind than AODV in terms of packet received, throughput and end to end delays.
In the case of Jitter test, although there wasn't much difference in the output of these three protocols, however, DSDV gave slightly better performance than the other two protocols.

Acknowledgment

We are greatful to our B.Tech (IT) students Ashvini Mandpe, Dipika Sose, and Ashwini Warthy for their valuable contribution for the outcome of this work.

References

[1] T. Larsson and N. Hedman,"Routing protocols in wireless ad-hoc networks- a Simulation study", Master thesis in Computer Science and Engineering, 1998.

[2] Elizabeth M. Royer and C-K Toh. "A Review of current Routing Protocols for Ad-hoc Mobile Wireless Networks", IEEE Personal Communications, Vol. 6, No.2, pp. 46-55, April 1999.

[3] Charles E Perkins and Pravin Bhagwat,"Highly Dynamic Destination Sequenced Distance Vector (DSDV) Routing Protocol for Mobile Computers, 1994.

[4] Darshan Singh Kali Rai,"Performance based comparative analysis of aodv, dsr and dsdv protocol w.r.t udp and tcp", Master thesis in Computer Science and Engineering, July 2009

[5] Z.Alexander, "Performance Evaluation of AODV Routing Protocol: Real-Life Measurements", SCC, June 2003.

[6] Charles E. Perkins and Elizabeth M. Royer, "Ad Hoc On-Demand Distance Vector Routing," in Proceedings of the 2nd IEEE Workshop on Mobile Computing Syslems and Applications, pp. 90-100, Feb. 1999.

[7] Salima Hamma & Eddy Cizeron & Haz Issaka & Jean-Pierre "Performance evaluation of reactive and proactive routing protocol in IEEE 802.11 ad hoc network",

[8] David B. Johnson and David A. Maltz, ''Dynamic source routing in adhoc wireless networks," in Mobile computing, Kluwer Academic, 1996, ch.5

[9] Kevin Fall and Kannan Varadhan. "The ns Manual (formerly ns notes and documentation)"

[10] Giovanni Perbellini. "An introduction to NS-2", Università degli Studi di Verona, Facoltà di Scienze MM. FF. NN., Verona, 12/09/2005

[11] J. Williams, "Narrow-Band Analyzer," PhD dissertation, Dept. of Electrical Eng., Harvard Univ., Cambridge, Mass., 1993. (Thesis or dissertation)

[12] Jaroslaw Malek. "Trace graph - Network Simulator NS-2 trace files analyzer" http://www.tracegraph.com