International Journal of Computer and Communication Technology

Volume 2 | Issue 2 Article 9

April 2011

# Performance analysis of Dynamic MANET Ondemand (DYMO) Routing protocol

Sukant Kishoro Bisoyi
*C. V. Raman College of Engineering, Bhubaneswar, India*, sukantabisoyi@yahoo.com

Sarita Sahu
*C.V.Raman College of Engineering, Bhubaneswar, India*, sarita.cvraman@gmail.com



## Recommended Citation

Bisoyi, Sukant Kishoro and Sahu, Sarita (2011) "Performance analysis of Dynamic MANET Ondemand (DYMO) Routing protocol," *International Journal of Computer and Communication Technology*: Vol. 2 : Iss. 2 , Article 9.
DOI: 10.47893/IJCCT.2011.1082
Available at: https://www.interscience.in/ijcct/vol2/iss2/9

# Performance analysis of Dynamic MANET On-demand (DYMO) Routing protocol

Sukant Kishoro Bisoyi[1], Sarita Sahu[2]
[1,2] *C.V.Raman College of Engineering, Bhubaneswar, India*
[1] <sukantabisoyi@yahoo.com>, [2] <sarita.cvraman@gmail.com>

**Abstract- Routing in a MANET is challenging because of the dynamic topology and the lack of an existing fixed infrastructure. In such a scenario a mobile host can act as both a host and a router forwarding packets for other mobile nodes in the network. Routing protocols used in mobile ad hoc networks (MANET) must adapt to frequent or continual changes of topology, while simultaneously limiting the impact of tracking these changes on wireless resources. The DYMO protocol intended for the use by mobile nodes in wireless multihop ad hoc networks. It can adapt to the changing network topology and determine unicast routes between nodes within the network. This paper presents a comprehensive summarization and a comparative study of the Dynamic MANET On-demand (DYMO) protocol for MANET and simulation analysis of existing protocols DSR and AODV and comparison among them under varying number of nodes. Comparative study shows that DYMO is only a good choice if the nodes are mobile and wireless multihop. We have compared the performance of DSR and AODV with DYMO protocol by taking some performance metrics. Result shows that DYMO simulation provides better performance than DSR when compared in a given network topology with respect to throughput, packet loss, delay, packet delivery ratio, normalized routing load.**
***Keywords* - MANET, DYMO, DSR,AODV,NS2,**

## I. Introduction

The advancement in wireless communications and lightweight, small-size, portable computing devices have made pervasive and mobile computing possible. One wireless network architecture that has attracted a lot of attention recently is the mobile ad-hoc network (MANET). A Mobile Ad-Hoc Network is a collection of mobile nodes with no pre-established infrastructure, self organizing wireless network which forms a temporary network [1]. Each of the nodes has a wireless interface and communicates each other over either radio or infrared signals. In ad hoc networks [2] all nodes are mobile and can be connected dynamically in an arbitrary manner.


(a) An infrastructured network with two base stations. (b) A mobile ad-hoc network.

Figure 1: Infrastructured and ad-hoc networks.

One area of research, which has been a focal point of research in Ad hoc networks, is Routing. Generally, Ad hoc routing protocols can be classified broadly into two categories, these are *proactive, Reactive*.

## II. ROUTING IN MANET

MANET routing protocols can be divided into two categories. In table driven/ proactive routing protocols [3], nodes periodically exchange routing information and attempt to keep up-to-date routing information [4]. In on-demand/reactive routing protocols [5], nodes only try to find a route to a destination when it is actually needed for communication. A brief classification of Ad-hoc routing protocols is given in figure-2.

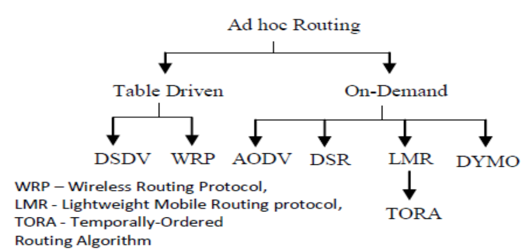


Figure 2: Classification of Routing Protocols in MANET

### A. *Reactive/On-Demand Routing Protocols*

On demand protocols use two different operations to find and maintain routes: the route discovery process operation and the route maintenance operation. In this routing information is acquired on-demand. This is the route discovery operation. Route maintenance is the process of responding to changes in topology that happens after a route has initially been created. Examples of on-demand protocols are DSR, AODV, and DYMO. The main advantage is the control traffic in the network is minimized, but this is the cost of long setup delay therefore this scheme is not suitable for routing real-time traffic. Another drawback of this scheme is that the message size increases because the entire path information is in the message and when the sending node has to discover a route to the destination, the initial delay before data is exchanged between two nodes can be long.

B. *Proactive/Table-Driven Routing Protocols*

Proactive routing protocols maintain routing information continuously. Typically, a node has a table containing information on how to reach every other node and the algorithm tries to keep this table up-to-date. Changes in network topology are propagated throughout the network.

*1. AODV Routing Protocol*

The Ad Hoc On-Demand Distance Vector [6] [7] [8] (AODV) routing protocol is a reactive protocol.

*Route Discovery*

When a node S wishes to communicate with a node T it initiates RREQ message including the last known sequence number for T and a unique RREQ id that each node maintains and increments upon the sending of an RREQ. The message is flooded throughout the network in a controlled manner. Each node forwarding the RREQ creates a reverse route for itself back to S using the address of the previous hop as the next hop entry for the node originating the RREQ. When the RREQ reaches a node with a route to T a RREP, containing the number of hops to T and the sequence number for that route, is sent back along the reverse path. An intermediate node must only reply if it has a fresh route, i.e., the sequence number for T is greater than or equal to the destination sequence number of the RREQ. Since replies are sent on the reverse path. Route discovery is illustrated in figure 3.

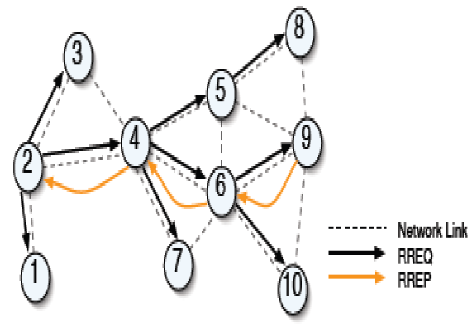


Figure 3: Route discovery in AODV. Node 2 wants to communicate with node 9. Each node forwarding the RREQ creates a reverse route to node 2 used when sending back the RREP.

If an intermediate node has a route to a requested destination and sends back an RREP, it must discard the RREQ. Furthermore, it may send a gratuitous RREP to the destination node containing address and sequence number for the node originating the RREQ. Gratuitous RREPs are sent to alleviate any route discovery initiated by the destination node.

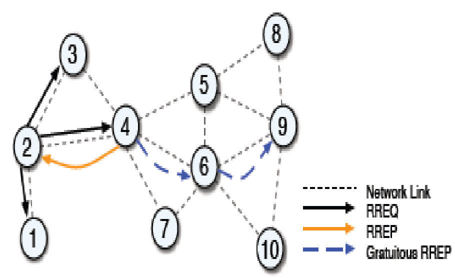


Figure 4: Generation of an RREP by an intermediate node. Node 4 has a route to node 9 and sends an RREP to node 2 and a gratuitous RREP to node 9.

*Route Maintenance*

It is the process of responding to changes in topology. To maintain paths, nodes continuously try to detect link failures. Nodes listen to RREQ and RREP messages to do this. Furthermore, each node promises to send a message every n seconds. If no RREQ or RREP is sent during that period, a Hello message is sent to indicate that the node is still present. Alternately, a link layer mechanism can be used to detect link failures. When a node detects a link break or it receives a data packet it does not have a route for, it creates and sends a Route Error (RERR) packet to inform other nodes about the error. The RERR contains a list of the unreachable destinations. If a link break occurs, the node adds the unreachable neighbour to the list. If a node receives a packet it does not have a route for, the node adds the unreachable destination to the list. In both cases, all entries in the routing table that make use of the route through the unreachable destination, are added to the list. The list is pruned, as destinations with empty precursor lists, i.e., destinations that no neighbours currently make use of, are removed. The RERR message is either unicasted (in case of a single recipient) or broadcasted to all neighbours having a route to the destinations in the generated list. This specific set of neighbours is obtained from the precursor lists of the routing table entries for the included destinations in the RERR list. When a node receives an RERR, it compares the destinations found in the RERR with the local routing table and any entries that have the transmitter of the RERR as the next hop, remains in the list of unreachable nodes. The RERR is then either broadcasted or unicasted as described above. The intention is to inform all nodes using a link when a failure occurs. For example, in figure 5, a link between node 6 and node 9 has broken and node 6 receives a data packet for node 9. Node 6 generates a RERR message, which is propagated backwards toward node 2.

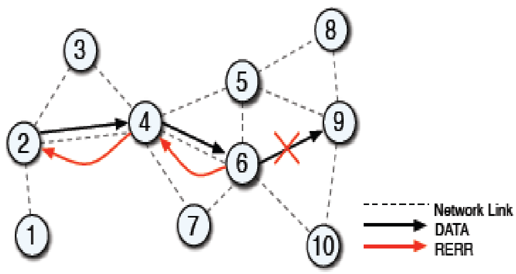


Figure 5: Generation of RERR messages. The link between node 6 and node 9 has broken, and node 6 generates an RERR.

To find a new route, the source node can initiate a route discovery for the unreachable destination, or the node upstream of the break may locally try to repair the route, in either cases by sending an RREQ with the sequence number for the destination increased by one.

*2. Dynamic Source Routing (DSR)*

The Dynamic Source Routing (DSR) [9] protocol is a simple and efficient, highly reactive, routing protocol, which is designed specifically for use in multi-hop wireless ad-hoc networks. The Dynamic Source Routing protocol (DSR) allows any host to dynamically discover a source route to any destination in the network. A packet is moved through a network using a path predetermined by the source node. The path information to use during the routing is placed in the packet.

*Basic Route Discovery*

Route discovery mechanism is illustrated in figure-6. Node 2 has a data packet to send to node 9 and floods a RREQ in the network. The RREQ packet contains a unique request id generated by the source node and a record listing the addresses of all intermediate nodes. Each node receiving the RREQ rebroadcasts the packet, if the node is not the target, it has not forwarded the packet previously, and it does not find its own address already listed in the route record. The request id of the RREQ is used to check for already forwarded packets, i.e., duplicate RREQs. Finally, the node appends its address to the route record of the packet.

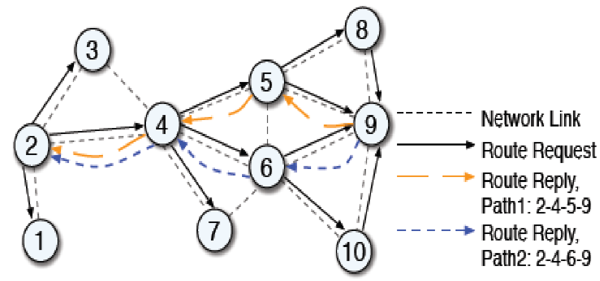


Figure 6: The route discovery process for DSR. Node 2 is the initiator and node 9 is the target.

The RREQ arrives at node 9 via different routes and the node then returns a Route Reply (RREP) to node 2, the initiator of the route discovery, containing the recorded route. When node 2 receives the RREP sent by node 9, it saves the listed route in its route cache for use for subsequent sendings. The RREP can be returned various ways shown in above figure-6.

*Route Maintenance*

Each node transmitting a packet is responsible for ensuring that the next hop neighbor receives the packet. This can be performed in three ways:

It can either per-hop acknowledgements, passive acknowledgements, or finally a flag set in a DSR control packet requesting explicit next hop acknowledgement. Upon detection of a link break when forwarding a packet, a RRER error packet is sent to the node originating the packet, stating the link that is currently broken. For example, in figure, node 9 has moved outside the transmission range of node 6 and it is unable to deliver the data packet to node 9.

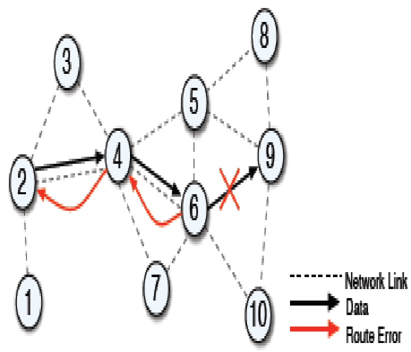


Figure 7: Route maintenance. Node 9 cannot be reached by node 6 anymore and a RERR is returned to node 2.

Node 6 then returns RERR to node 4 that in return propagates it to node 2, the original sender, which removes the route from its route cache. It can then use another cached route (for example, the path 2-4-5-9 learned from the previous route discovery), or perform a new route discovery for node 9.

### 3. *The DYMO Routing Protocol*

The Dynamic MANET On-demand DYMO routing protocol is a newly proposed protocol currently defined in an IETF Internet-Draft [10] in its sixth revision and is still work in progress. DYMO is a successor of the AODV routing protocol [6]. It operates similarly to AODV. DYMO does not add extra features or extend the AODV protocol, but rather simplifies it, while retaining the basic mode of operation. As is the case with all reactive ad hoc routing protocols, DYMO consists of two protocol operations: route discovery and route maintenance. Routes are discovered on-demand when a node needs to send a packet to a destination currently not in its routing table. A route request message is flooded in the network using broadcast and if the packet reaches its destination, a reply message is sent back containing the discovered, accumulated path.

Each entry in the routing table consists of the following fields:

Destination Address , Sequence Number, Hop Count, Next Hop Address, Next Hop Interface, Is Gateway, Prefix, Valid Timeout, Delete Timeout

#### *Route Discovery*

When a node S wishes to communicate with a node T, it initiates a RREQ message. The RREQ message and the RREP message, which is known as Routing Messages (RM). The sequence number maintained by the node is incremented before it is added to the RREQ. We illustrate the route discovery process using figure 8 as an example. In the figure, node 2 wants to communicate with node 9 and thus, node 2 is S, the source, and node 9 is T, the target destination. In the RREQ message, the node 2 includes its own address and its sequence number, which is incremented before it is added to the RREQ. Finally, a hop count for the originator is added with the value 1. Then information about the target destination 9 is added. The most important part is the address of the target. If the originating node knows a sequence number and hop count for the target, these values are also included. The message is flooded using broadcast, in a controlled manner, throughout the network, i.e., a node only forwards an RREQ if it has not done so before. The sequence number is used to detect this. Each node forwarding an RREQ may append its own address, sequence number, prefix, and gateway information to the

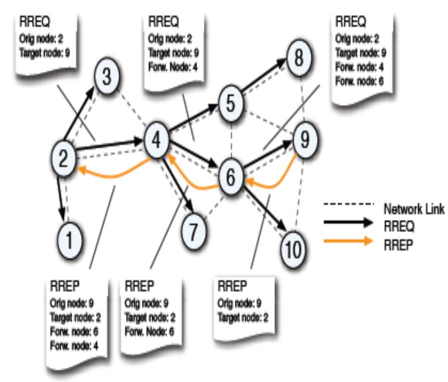


Figure 8: The DYMO route discovery process. Node 2 wants to communicate with node 9. Each node forwarding the RREQ creates a reverse route to 2 used when sending back the RREP. When sending back the RREP, nodes on the reverse route create routes to node 9.

RREQ, similar to the originator node. Upon sending the RREQ, the originating node will await the reception of an RREP message from the target. If no RREP is received within RREQ WAIT TIME, the node may again try to discover a route by issuing another RREQ. RREQ WAIT TIME is a constant defined in the DYMO specification and the default value is 1000 milliseconds. In figure-8, the nodes 4 and 6 append information to the RREQ when they propagate the RREQ from node 2. When a node receives an RREQ, it processes the addresses and associated information found in the message. An RREP message is then created as a response to the RREQ, containing information about node 9, i.e., address, sequence number, prefix, and gateway information, and the RREP message is sent back along the reverse path using unicast. Since replies are sent on the reverse path, DYMO does not support asymmetric links. The packet processing done by nodes forwarding the RREP is identical to the processing that nodes forwarding an RREQ perform, i.e., the information found in the RREP can be used to create forward routes to nodes that have added their address block to the RREP.

We shortly summarize the route discovery process depicted in figure-8. Node 2 wants to communicate with node 9 and floods an RREQ message in the network. As can be seen in the figure, when node 2 begins route discovery, the RREQ initially contains the address of the originator and target destination. When node 4 receives the RREQ, it installs a route to node 2. After node 4 has forwarded the RREQ, it has added its own address to the RREQ, which means it now contains three addresses. Identical processing occurs at node 6 and it installs a route to node 2 with a hop count of 2 and node 4 as the next hop node. When node 9 receives the RREQ, it contains four addresses and has travelled three hops. Node 9 processes the RREQ and install routes using the accumulated information and as it is the target of the RREQ, it furthermore creates an RREP as a response. The RREP is sent back along the reverse route. Similar to the RREQ dissemination, every node forwarding the RREP adds its own address to the RREP and installs routes to node 9.

*Route Maintenance*

Route maintenance is the process of responding to changes in topology that happens after a route has initially been created. To maintain paths, nodes continuously monitor the active links and update the Valid Timeout field of entries in its routing table when receiving and sending data packets. If a node receives a data packet for a destination it does not have a valid route for, it must respond with a Route Error (RERR) message. When creating the RERR message, the node makes a list containing the address and sequence number of the unreachable node. In addition, the node adds all entries in the routing table that is dependent on the unreachable destination as next hop entry. The purpose is to notify about additional routes that are no longer available. The node sends the list in the RERR packet. The RERR message is broadcasted. The dissemination process is illustrated in figure-9. A link between node 6 and node 9 breaks and node 6 receives a data packet for node 9. When we say a link is broken, it could just be that the time stamp in the route table entry for a node timed out and the entry has become invalid. Node 6 generates an RERR message, which is propagated backwards towards node 2.

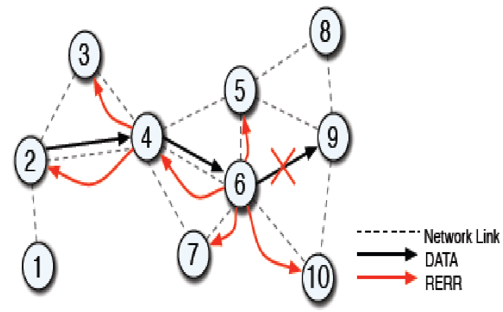


Figure 9: Generation and dissemination of RERR messages. The link between nodes 6 and 9 breaks, and node 6 generates an RERR. Only nodes having a route table entry for node 9 propagate the RERR message further.

When a node receives an RERR, it compares the list of nodes contained in the RERR to the corresponding entries in its routing table. If a route table entry for a node from the RERR exists, it is invalidated if the next hop node is the same as the node the RERR was received from and the sequence number of the entry is greater than or equal to the sequence number found in the RERR. If a route table entry is not invalidated, the corresponding entry in the list of unreachable nodes from the RERR must be removed. If no entries remain, the node does not propagate this RERR further. Otherwise, the RERR is broadcasted further. The sequence number check mentioned is performed to only invalidate fresh routes and to prevent propagating old information. The intention of the RERR distribution is to inform all nodes that may be using a link, when a failure occurs. RERR propagation is guaranteed to terminate as a node only forwards an RERR message once. In figure-9, when the RERR is broadcasted, additional nodes beside node 4 and 2 will receive the message, for example, the nodes 5, 7, and 10. As none of these use nodes 6 as a next hop towards node 9, they all drop the RERR after processing the message. In addition to acting upon receiving a packet to a destination without a valid route table entry, nodes must continuously try to detect link failures to maintain active links.

## III. PERFORMANCE METRICS

The following performance metrics are used to compare the performance of the routing protocols in the simulation:

**Throughput**: It is the amount of data per time unit that is delivered from one node to another via a communication link [11]. The throughput is measured in bits per second (bit/s or bps).

**Packet Loss:** It occurs when one or more packets traveling across a network fail to reach their destination.

**Latency:** In a network, latency, which is a synonym for delay, is an expression of how much time it takes for a data packet to get from one node to another.

**Packet delivery ratio(PDF):** it is ratio between number of packets received by destination and number of packet originated by application (TCP and CBR) [12].

PDF = (data_agt_rec / data_agt_sent)*100;

**Normalized Routing Load(NRL):** The number of routing packets transmitted per data packet delivered at the destination. This metric gives an idea of the extra bandwidth consumed by overhead to deliver data packet.

NRL = ((cp_sent + cp_forw) / data_agt_rec)*100;

cp_sent = rreq + rrep + rerr;
cp_sent =Controll Packets sent
cp_forw=Control packet forwarded
data_agt_rec=Datapacketsreceived
rreq= route request
rrep=route reply
rerr=routeerror

## IV. SIMULATION AND RESULTS

This section describes the simulation tool and parameters chosen to simulate the routing protocols. In this paper the Ubuntu Operating System was used because it is a user-friendly platform and easy to manage and to setup a simulator. For simulation software, Network Simulation 2(NS2.29) was used as the simulator to evaluate the performance of AODV, DSR and DYMO routing protocols. In this project, the simulation environment consist of two different number of nodes which are 3, 6. We have simulated the entire above mentioned algorithm under different condition. In first simulation environment we have created 6 nodes and for node movement we have used seed. Constant Bit Rate (CBR) traffic generators is used as sources to run the simulation when node 0 is source and 3 is destination and FTP is used when node 0 is source and node 1 is destination. Each CBR packet contained 512 Bytes and packets were transmitted at 20Kb and FTP of 960 Bytes at 0.01Kb. Parameters used in simulations are shown in table-1 and comparisons of AODV, DSR and DYMO are shown in table-2. In second scenario we have created 3 nodes and node movement was done by setdist command. In this CBR traffic was used to establish communication between node 0 and node 1. Parameters used in simulations are shown in table-3 and comparisons of AODV, DSR and DYMO are shown in table-4. Figure-10 and figure-11 shows the graph of number packet received versus simulation time and number of packet dropped versus simulation time.

TABLE-1
PARAMETER USED IN FIRST SIMULATION

| PARAMETER | VALUE |
|---|---|
| Channel type | Wireless channel |
| No. of mobile nodes selected | 6 |
| Data Packet | CBR of 512 bytes and FTP of 960 bytes packet size |
| MAC Protocols | Mac/802_11 |
| Node Placement | Random |
| Size of interface queue | 15 |
| Time of simulation | 120 msec. |
| Area of simulation | 500*500 |
| Seed | 1 |

TABLE-2
COMPARISION OF DYMO, DSR AND AODV

| PARAMETER | DYMO | DSR | AODV |
|---|---|---|---|
| Number of CBR data packets generated | 1146 | 682 | 658 |
| Number of TCP data packets generated | 18024 | 1497 | 5 |
| Number of CBR data packet send | 582 | 582 | 582 |
| Number of TCP data packet send | 9090 | 760 | 05 |
| Number of dropped packets | 328 | 513 | 510 |
| Number of forwarded packets | 0 | 109 | 78 |
| Packet delivery ratio (CBR and TCP ) in % | 0.98 | 0.62 | 0.12 |

| | | | |
|---|---|---|---|
| Data packet lost | 174 | 505 | 512 |
| Average Delay | 0.099 | 2.474 | 1.696 |
| Number of control Packets send | 755 | 47 | 30 |
| Control packet forwarded | 0 | 09 | 01 |
| Normalized routing load | 7.95 | 6.69 | 40.789 |

TABLE-3
PARAMETER USED IN SECOND SIMULATION

| PARAMETER | VALUE |
|---|---|
| Channel type | Wireless channel |
| No. of mobile nodes selected | 3 |
| Data Packet | CBR of 512 bytes packet size |
| MAC Protocols | Mac/802_11 |
| Node Placement | Not Random |
| Size of interface queue | 50 |
| Time of simulation | 120 msec. |
| Area of simulation | 500*400 |

TABLE-4
COMPARISION OF DYMO, DSR AND AODV

| PARAMETER | DYMO | DSR | AODV |
|---|---|---|---|
| Number of CBR data packets generated | 998 | 1031 | 1061 |
| Number of CBR data packet send | 582 | 582 | 582 |
| Number of dropped packets | 165 | 133 | 102 |
| Number of forwarded packets | 417 | 168 | 481 |
| Packet delivery ratio (CBR) in % | 71.477 | 77.147 | 83.302 |
| Data packet lost | 166 | 133 | 102 |
| Average Delay | 0.011 | 1.065 | 0.930 |
| Number of control Packets send | 87 | 31 | 21 |
| Control packet forwarded | 0 | 4 | 1 |
| Normalized routing load | 12.5 | 4.009 | 2.5 |

The packet transmission details of the three protocols in first scenario generally indicate that the protocols DYMO tend to have a higher packet delivery fraction (ratio) (see table-2). The losses suffered by AODV and DSR may have happened in response to a dynamic changing topology. So each routing protocol requires a robust Route Discovery and Route Maintenance to cope with the dynamic changing topology. Also DYMO has more throughput as both are reactive (route cache) protocol. The delays experienced by the protocols are a crucial factor which can adversely affect the performance of the protocol. The delay experienced by DYMO (table-2) is the lowest which is much lower than the delay experienced by DSR. The packet lost suffered in DYMO is much lower than AODV and DSR (table-2). AODV exhibited the highest normalized routing overhead compared to DYMO and DSR. This metric gives an idea of the extra bandwidth consumed by overhead to deliver data packet. It is because more routing packets are generated and delivered by AODV but control overhead is more in case of DYMO (see table-2).

In second scenario also these three protocols were compared by creating 3 nodes. In this node 0 is source and node 1 is destination and node 2 is intermediate node. The packet delivery ratio (PDF) of AODV is slightly higher than DSR and DYMO (table-4). For small spaces, for example 500m x 400m, DYMO perform well in terms of stable and low average end to end delay. The delay experienced by DSR and AODV (see table-4) is slightly more than DYMO. But packets dropping and packet loss of AODV slightly less than others which are shown in figure 10 and figure 11.

For the first simulation scenario we have calculated number of packet dropped for different pause time of DYMO, DSR, AODV algorithm and found that more number of packets are dropped in case of DSR algorithm as compared to others two which is shown in figure-12. For the first simulation scenario, table-5 illustrated the packet delivery ratio (PDF) of AODV, DSR AND DYMO versus pause time. DYMO algorithm performs better for all pause time compared to others. In the first simulation the delay experienced by DYMO (table-6)protocol is the lowest which is much lower than the delay experienced by AODV for different speed(5, 10, 15, 20).

TABLE-5
PDF Vs PAUSE TIME(Sec)

| Algorithm | 20 | 40 | 60 | 80 | 100 | 120 |
|---|---|---|---|---|---|---|
| DYMO | 0.983 | 0.981 | 0.983 | 0.991 | 0.984 | 0 |
| DSR | 0.969 | 0.983 | 0.984 | 0.985 | 0 | 0.876 |
| AODV | 0.943 | 0.981 | 0.982 | 0.982 | 0.977 | 0 |

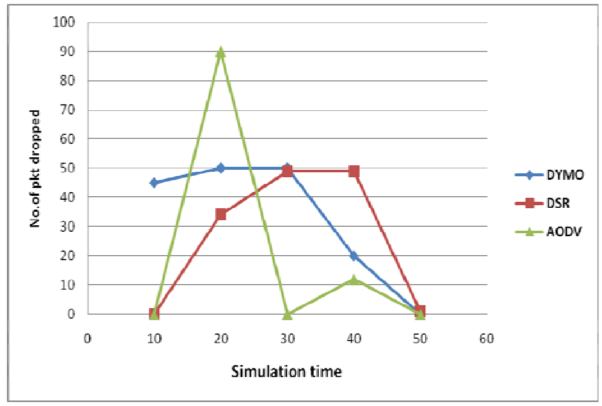


Figure 11: No of packet dropped Vs Simulation time

TABLE-6
AVERAGE DELAY Vs SPEED

| Algorithm/speed | 5 | 10 | 15 | 20 |
|---|---|---|---|---|
| DYMO | 0.0991 | 0.7387 | 0.1207 | 0.1369 |
| AODV | 1.696 | 0.0987 | 0.2858 | 0.501 |

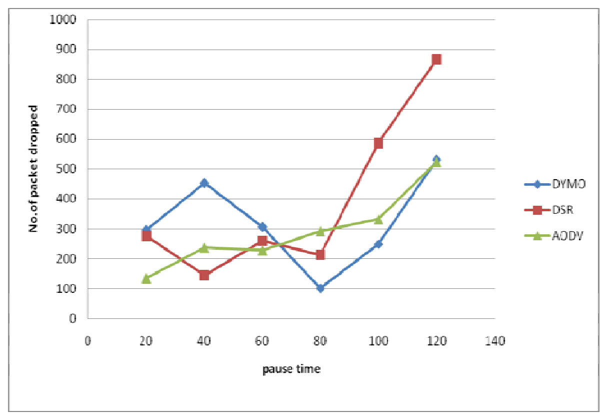


Figure 12: No of packet dropped Vs Pause time

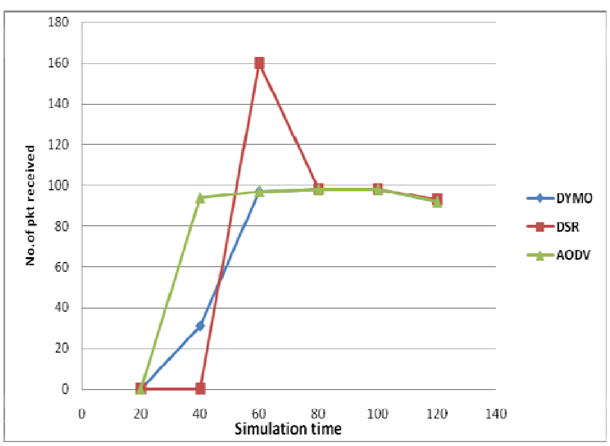


Figure 10: No of packet received Vs Simulation time

## V. CONCLUSION

This study was conducted to evaluate the MANET routing protocols AODV, DSR and DYMO. Performance of each routing protocol has been analyzed and evaluated accordingly based on different number of nodes over different speed and different pause time and we found DYMO is a better routing protocol than DSR and AODV routing protocol for Mobile Ad hoc Network with respect to Quality of Service (QOS) parameters, i.e., throughput, packet delivery ratio, delay, normalized routing load. In terms of routing overhead AODV performs better than others. Most of these results are based on the simulation or small scale experiments in laboratory settings.

The suitability for large scale networks still has to be proven. Hopefully, the result of this study can be used as reference for the future work.

REFERENCES

[1]. Mehdi Alilou, Mehdi Dehghan.t, Upgrading Performance of DSR Routing Protocol in Mobile Ad Hoc Networks, World Academy of Science, Engineering and Technology 5, March 29 2005.

[2]. Magnus Frodigh, Per Johansson and Peter Larsson. Wireless ad hoc Networking-The art of networking without a network.

[3]. Koojana Kuladinithi, Asanga Udugama, Nikolaus A. Fikouras, and Carmelita G¨org. Experimental performance evaluation of AODV implementations in static environments. http://www.comnets.uni-bremen.de/~koo/AODV-Perf-ComNets.pdf

[4]. Jeremie Allard, Paul Gonin, Minoo Singh, and Golden G. Richard III A user level framework for ad hoc routing. In LCN '02: Proceedings of the 27th Annual IEEE Conference on Local Computer Networks, page 13, Washington, DC, USA, November 2002. IEEE.

[5]. Elizabeth Belding-Royer, Ian Chakeres, David Johnson, and Charlie Perkins. DYMO – dynamic MANET on-demand routing protocol. In Rebecca Bunch, editor, Proceedings of the Sixty-First Internet Engineering Task Force, Washington, DC, USA, November 2004. IETF.

[6]. Charles E. Perkins, Elizabeth M. Belding - Royer, and Samir R. Das. Ad hoc on-demand distance vector (AODV) routing. RFC 3561, IETF, July 2003. rfc356.txt.

[7]. C.E. Perkins, E.M. Belding-Royerand, I.D. Chakeres, Ad Hoc On-Demand Distance Vector (AODV) Routing, IETF Internet Draft, draft-perkins-manet-aodvbis-01.txt, January 2004.

[8]. C.E. Perkins and E. Royer, Ad hoc on-demand distance vector (AODV) routing, in: Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA), February 1999, pp. 90–100.

[9]. D.B. Johnson, D.A. Maltz, Y. Hu, The dynamic source routing protocol for mobile ad hoc network, IETF Internet Draft, draft-ietfmanet-dsr-09.txt, April 2003.

[10]. Ian D. Chakeres and Charles E. Perkins. Dynamic MANET ondemand (DYMO) routing protocol. Internet-Draft Version 4, IETF, March 2006. draft-ietf-manet-dymo-04.txt, (Work in Progress).

[11]. Scalable Network Technologies, "QualNet simulator 4.0 Version",tutorial on http://www.cs.binghamton.edu/~vinkolar/QualNet/qualnettut1.pdf.

[12]. JunZhao Sun Media Team, "Mobile Ad Hoc Networking: An Essential Technology for Pervasive Computing", Machine Vision and Media Processing Unit, University of Oulu, Finland.

[13] NS-2, The ns Manual (formally known as NS Documentation) available at http: //www. isi.edu/*nsnam/ns/doc.*