ISSN 2229-5518

Performance Comparison of Dynamic Mobile ad-hoc network on-demand multipath routing protocol with AODV

Archie Budhiraja ,Roopali Garg

Abstract- Communication devices have become one of the most important instruments to stay in touch with each other. Over the years, engineers have been working to enhance the network protocols used by these devices for better communication. Mobile Ad hoc Networks (MANETs) are networks with dynamic topology and limited resources, where routing is highly correlated to their performance. Most routing algorithms focus on the establishment of a single path between the source and the destination. However, multiple routes have been proven beneficial, when used either for load balancing or as backup routing paths. In this paper we propose a novel routing algorithm which is based on the Dynamic MANET On-demand (DYMO) routing protocol and evaluate its performance in detail. Simulation results show that multiple paths can reduce the jitter and increase the throughput.

Keywords- Dynamic MANET on Demand Routing Protocol (DYMO), DSR, DSDV, AODV.

I. INTRODUCTION

Nowadays, there is an increasing need for interconnection of several devices, in order to satisfy particular needs. Mobile Ad hoc Networks (MANETs) are networks formed in cases were networking infrastructure is either unavailable or totally absent. Using wireless interfaces, hosts may communicate with each other directly if each one falls within the communication radius of the other. Fig.1 shows an example of Mobile ad-hoc network. This ad hoc topology may change with time as the nodes move or adjust their transmission and reception parameters.

However, two distant hosts, that cannot communicate directly, may use nodes located between them as relays that forward data for them, thus forming multihop networks. At the same time, users may move around, so existing links may break while new ones may be created. Usually based on limited resources, mainly in terms of bandwidth and energy, MANETs characteristics introduce many critical parameters for network designers, but they also make MANETs the appropriate solution for many networking problems, such as natural disasters, environmental networking or vehicle-to-vehicle communication (VANETs).In this context, maintaining routing paths in such a dynamic network is not a trivial task.

The increasing usage of demanding applications calls for sophisticated routing protocols that permit these applications to be executed smoothly over the network. This means that routing algorithms should provide the data sources with valid paths towards their destinations with the minimum possible cost in bandwidth and energy terms.

Maintaining multiple routes between source-destination pairs has been proposed as a performance enhancement mechanism.

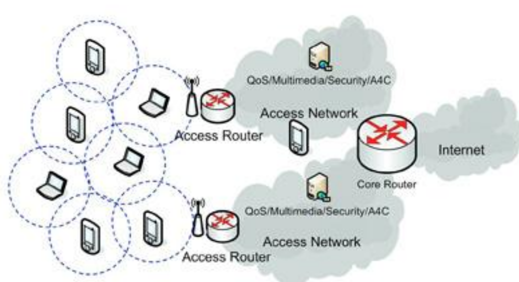

Fig 1: Example of Mobile-ad-hoc-networks

Multipath may be used in several ways. Different routes could be used for delivering packets belonging to different applications, in order to avoid congestion (load balancing). Furthermore, packets belonging to the same service may be sent through different routes to increase reliability. Finally, the additional paths could be used as backup routes, so that to decrease the number of route requests sent out from source. Motivated by the aforementioned characteristics of multipath routing, we modified the Dynamic MANET On-demand (DYMO) routing protocol in order to enable the source and the destination to establish more than one routes between them. The new concept is added between the existing protocol i.e. Dynamic MANET On-demand Multipath routing protocol is discussed in this paper.

ISSN 2229-5518

The remaining of the paper is organized as follows: in the next Section, we describe the routing protocol of MANETs. In Section III we describe the multipath concept in detail. Section IV describes the DYMO protocol. In Section V we describe simulation tools and section VI describes the simulation results and analysis for the following parameters. Finally in last section the conclusion.

II. ROUTING PROTOCOLS OF MANETs

Mobile Ad-hoc Network (MANET) is a self-configuring network of mobile nodes connected by wireless links where each device in a MANET is free to move independently in any direction with capability of changing its links to other devices frequently [1]. A brief classification of Ad-hoc routing protocols is given in figure 2[2].

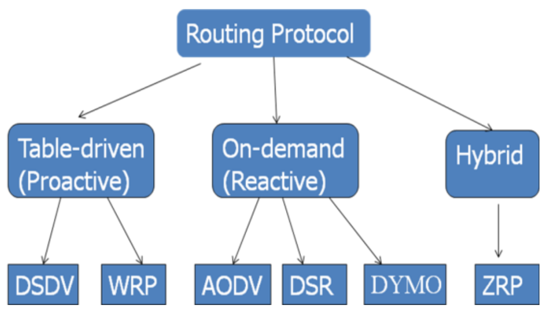

Fig 2: Types of Routing Protocol.

DYMO (Dynamic MANET On-demand) routing protocol is newly intended for use by mobile nodes in wireless multihop networks. MANET working group uses it as the current research in on-demand routing issue. When I. Chakeres and E. Belding-Royer submitted the draft about DYMO routing protocol in Internet-Drafts in 2005, this protocol caused widespread attention [3]. The draft about DYMO has been updated many times, and has grown to the version 21 [4].

DYMO is a descendant of the design of MANET reactive protocols, such as AODV and DSR. Changes to previous MANET reactive protocols stem from research and implementation experiences. The basic operations of the DYMO protocol are route discovery and route management. Therefore, the protocol is not only simple and easy to implement, but also has significant scalability, to allow further enhancements in capability and expansion of the function. The protocol may be extended in the different aspects so that people can easily conduct study on the reactive routing protocols deeply. It is unique to DYMO, and it is the main reason why the DYMO is widely studied now. In addition, DYMO protocol can be used both in IPv4 and IPv6 network and people may use it to connect with the Internet.

III. MULTIPATH CONCEPT

Multipath Routing has been initially proposed and studied for wired networks [5]. In MANETs, reactive protocols are based on route request procedures. When a node wishes to send data towards a destination, it initiates a route request procedure by flooding a Route Request (RREQ) packet in the entire network. When the destination receives this packet, a Route Reply (RREP) packet is sent back to the source. When a link breaks, a Route Error (RERR) packet is sent back to the source, so that a new route request procedure to take place. In this case, establishing more than one path is not a straight forward procedure, since these paths should be created during the same route request procedure, in order to avoid overhead.

These multiple routes may differ is several ways:

->All links of one route may be different from all links of another route (link disjoint routes)

-> All intermediate nodes of one route may be different from all intermediate nodes of another route (node disjoint routes)

-> All intermediate nodes of one route may be further than the interference distance from any intermediate node of another route (interference disjoint routes) [6]

Algorithms for finding multiple routes have been proposed in the framework of graph theory [7]. Since they assume knowledge of the entire network topology, these algorithms can only be applied to proactive routing, since in reactive routing nodes do not maintain information about the entire network, but only for the desired destinations. In the literature, several reactive multipath routing algorithms have been proposed.

A multipath routing based on the well known DSR protocol has been proposed in [8]. In this work, the authors develop an analytical tool for calculating the interval between successive route request discoveries, proving that multipath increases the expected value of this interval. They conclude that routes much longer than the shortest one, as well as maintaining more than two or three alternative routes are not beneficial.

Focusing on the source's ability to find multiple routes, the authors in [9] propose "loser" rules, compared to those of [8], for load balancing. A new path selection criterion is proposed, namely the correlation factor between paths and it is evaluated via simulations. Multipath algorithms based on the AODV protocol have also been proposed.

IV. Dynamic MANET On-Demand Routing Protocol (DYMO)

DYMO [4] is a routing protocol that was created for situations where clients are mobile and communications will be transported through several different clients over a wireless medium, Mobile ad-hoc Network (MANET).DYMO was created to dynamically handle changes in the network. Using Ad hoc on Demand Distance Vector (AODV) as the basis, DYMO borrows "Path Accumulation" from Dynamic Source Routing [10][11] and removes unnecessary Route Reply (RREP), precursor lists and Hello messages (Route exploration messages), thus simplifying AODV [10].It retains sequence numbers, hop count and Route Error (RERR) messages from AODV.It is a simple and fast routing protocol for multihop network. It is a reactive routing protocol mainly based on ideas from AODV. It determines multi-hop unicast routes on-demand in a dynamic network Topology.It uses a reactive route discovery.

Dynamic Manet On-demand (DYMO) routing protocol [4] is a reactive algorithm developed for MANETs. It resembles AODV. However, it is equipped with source routing characteristics, namely the path accumulation technique, which permits
nodes listening to routing messages to acquire knowledge about routes to other nodes without initiating route request discoveries themselves. This method increases the routing packet size, but decreases the required transmissions. In this work, we propose some simple modifications to the DYMO
protocol that permit it to establish multiple routes, whenever this is possible, so that DYMO will also be benefited by the improved performance offered by multipath routing. We called this new protocol DYMOM (DYMO Multipath). Below we describe the DYMO protocol, focusing on the points was
modifications are required for the aforementioned purpose. Let us mention here that the new internet-draft that describes DYMO does not propose any changes that influence the procedures that interest us in this work.

When a node wishes to send data to another node to whom no valid route is available, it initiates a route request procedure. A RREQ packet is flooded in the network. Nodes receiving RREQ packets apply the following rules, in order to decide if they will update their routing table and if they will further forward the packet:

->The information in the received packet is stale, when the source's sequence number reported by the packet is lower than the sequence number stored in the node's routing table.

-> The information in the received packet is loop-prone, when the aforementioned sequence numbers are the same and the reported number of hops (hop count) plus one is lower than the hop count stored in the node's routing table.

->The information in the received packet is inferior, when the sequence numbers are equal, a valid route exists and the reported hop count is greater than the stored hop count.

->The reported and the stored hop counts are equal and the routing packet is a RREQ.

Otherwise, the information is fresh. Only packets containing fresh information are further propagated. In case the path accumulation is enabled, the above rules are applied to the information regarding the intermediate nodes too, without influencing the decision about the retransmission of the packet.

When the destination receives a RREQ, it applies the same rules mentioned above. If the information is fresh, a RREP packet is created and sent to the source node, via the reverse path. Multiple RREPs are possible, since a new RREQ packet may indicate a shorter path than the one currently used. Nodes that receive a RREP update their routing tables following the aforementioned rules. In this way, the source receives the RREPs, updates its routing table accordingly and begins sending the data packets to the destination using the newly established path.

When a node fails to propagate a data packet due to link breakage, it sends back to the source node a RERR packet, indicating the broken link. All nodes receiving the RERR invalidate those paths that used the broken link and so does the source. Hence, when a new packet is received by the source's routing agent, a new route request procedure is initiated.

In order to enable the creation of multiple paths, some modifications are needed to the above procedures. In order to keep the protocol as simple as possible, we decided to minimize the required modifications. In this framework, we propose the alteration of the routing table update rules only for the source and destination nodes. In particular, these nodes should omit the rule referring to the inferior information. The consequence of this modification is that when the destination (source) receives a RREQ (RREP) via a longer path, this information is not discarded but it is taken into account or route establishment.

4.1. DYMO Overview

Reactive and multihop routing can be achieved between the participating nodes that wish to communicate with help of a protocol called Dynamic MANET On-demand (DYMO) routing. As a reactive protocol, DYMO does not explicitly store the network topology. Instead nodes compute a unicast route towards the desired destination only when needed. As a result, little routing information is exchanged, which reduces network traffic overhead and thus saves bandwidth and power.

ISSN 2229-5518

When a node needs a route, it disseminates a Route Request (RREQ), which is a packet asking for a route between an originator and a target node. The packet is flooded to the entire network or within a number of hops from the originator (see figure 3).

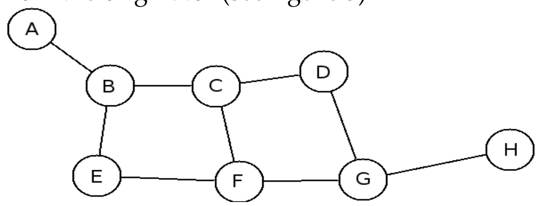

Fig3: Node A wants to communicate to node H

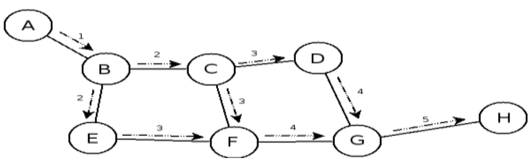

Fig 4: Node A wants a route to node H, it broadcasts a RREQ.

When the packet reaches its target (or a node that has a fresh route towards the target), the node replies with a Route Reply (RREP). A route reply packet is very similar to a route request, but it follows a unicast route and no reply is triggered when the target is reached (see figure 5).

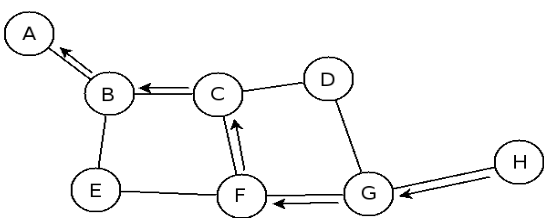

Fig 5:Node H replying to node A,so broadcasting a RREP

When nodes receive a RREQ or a RREP, they cache information about the sender and the originator, so that they know a route to the originator that can be used later (if it is fresh enough) without sending a RREQ. The nodes have the possibility to accumulate the path followed by the packet in the packet itself. So, when nodes disseminate a RREQ or RREP, a lot of information can actually be Obtained from the packet, much more than a route between two nodes.

When routes have not been used for a long time, they are deleted. If a node is requested to forward a packet through a deleted route, it generates a Route Error (RERR) message to warn the originating node (and other nodes) that this route is no longer available (see figure 6).

As another route maintenance mechanism, DYMO uses sequence numbers and hop counts to determine the usefulness and quality of a route.

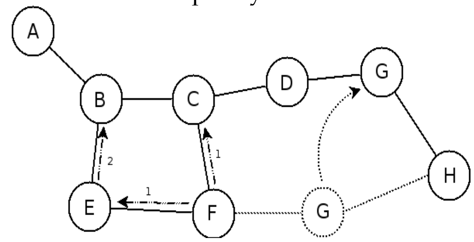

Fig 6:Node F replying to node E and node C,so broadcasting a RERR

We explain the operation of DYMO in more detail in the following sections.

4.2 Route Discovery

4.2.1 Routing Messages: The messages exchanged during the process of route discovery, that is RREQs and RREPs, are called *routing messages*. They always include the target and originator addresses, as well as a hop limit and a hop count which prevent a routing message from being forwarded several times. DYMO uses sequence numbers, as introduced in the distance-vector protocols. So, all nodes have a sequence number and include it in the DYMO packets they send nodes increment their sequence number whenever they estimate that the routing information of the other nodes is too old. This applies to both RREQs and RREPs, they are therefore very similar. The differences are only algorithmic: the way nodes decide if they should increment their sequence number before sending the packet, as well as how the packets are forwarded (since RREQs are broadcast and RREPs are unicast).

4.2.2 Packet Processing and Forwarding: When nodes receive a routing message, they look at all the included routing information. Based on sequence numbers and distances counted in hops, they decide whether the routing information is better than what they already know, and update their routing tables consequently. Therefore, links are assumed to be bidirectional. Also, during this process, the content of the message is updated. Hop counts are incremented and routing information that was not considered useful is removed from the message so that it is not propagated further. After having judiciously updated their routing tables, nodes can append additional routing information. This may decrease the number of RREQs and enable quicker RREPs. Indeed, when a node receives a RREQ with a target for which it knows a fresh route, it can send an intermediate RREP instead of forwarding the RREQ. As a result, the originator of the RREQ receives the RREP sooner, and the RREQ is not propagated further, which reduces the traffic overhead. Appending routing information to routing messages increases the chances that other nodes will send intermediate RREPs at the expense of bigger packets.

4.3 Route Maintenance

Since DYMO applies to a context with a highly dynamic network topology, routes need to be actively monitored after having been established. The protocol does not impose a monitoring mechanism, but specifies how this can be done with route timers.

4.3.1 Length and Freshness of Routes

A part of the route maintenance is keeping routes fresh and as short as possible. Not only fresh information is better, it also ensures loop freedom. When the information is fresh and loop-free, only the shortest path available is kept. This is determined by comparing hop counts, which is the distance from the considered node counted in hops. For instance, on figure 4, nodes F and G forward the RREQ only once, and they do not update their routing table with the second RREQ, because it does not have a better hop count.

4.3.2 Link Monitoring

Each time a node creates or updates a route in its routing table, it can monitor the route with associated timers. To ensure that nodes can rely on the information they receive in RREPs, nodes are expected to keep their routes for a minimum amount of time. Routes also have a maximum age, because keeping a route for a long time in a dynamic context is not safe and can lead to forwarding loops, and also because we do not want to spend memory for a route that is not actively used. Each time a packet is forwarded through a route, the timer for this route is updated. When the timer expires, the route can be deleted. Nodes may also use other methods to monitor links and routes, for instance a neighbour discovery protocol or a link-layer feedback.

4.3.3 Route Errors

When the route monitoring process detects a broken route, a broken flag is set for the corresponding route entry. If a node tries to use this route, a route error (RERR) message is flooded in the network. The RERR contains information about the unreachable node (node G on figure 6 for instance), and may also contain information about nodes (such as node H) previously reachable through this node. A RERR warns other nodes that some nodes are no longer available through the sender of the RERR. Upon receiving a RERR, nodes that do not have superior information about unreachable nodes set a broken flag for the relevant route entries. Unless the routing information included in the RERR is considered too old, the RERR is forwarded to all neighbours.

4.4. DYMO Applicability

DYMO protocols are designed for mobile ad hoc networks since DYMO is capable of handling dynamically altering mobile network patters. The routes between the source and destination are hence determined only when a route was required to be established. Being capable of handling on-demand routes discovery and maintenance, DYMO can also adapt to wide ranging traffic patterns. DYMO can be typically utilized in a large mobile network consisting of large number of nodes where only a part of the nodes communicate with each other. DYMO is also memory efficient since it maintains very little routing information. In DYMO, only routing information that are pertinent to all active sources and destinations is maintained where as other protocols require entire routing information of all nodes with in a network.

4.5 DYMO Messages

DYMO Control Packets include RREQ (Route Request), RREP (Route Reply) and RERR (Route Error). RREQ are used to disseminate routing information on how to reach the originator of the RREQ. RREP are used to disseminate routing information, on how to reach the target, to nodes between the target and the RREQ originator. RERR are used to disseminate that a valid route is not available for a particular destination, or set of destinations. Routing Messages (RM) are used to disseminate routing information. There are two DYMO message types that are RM, RREQ and RREP. They contain the same information, but have slightly different processing rules [4]. A RM requires the following information: IP.DestinationAddress, MsgHdr.HopLimit,
AddBlk.Target.Address, AddBlkOrig.Address,
AddTL V.Orig.SeqNum,etc. A RM may optionally include the following information: AddTL V.Target.SeqNum, ect. A RERR requires the following information: IP.DestinationAddress, MsgHdr.HopLimit, AddBlkUnreachable.Address. A Route Error may optionally include the following information: ddTL V.Unreachable.SeqNum, AddTLV.Node.

4.6 DYMO MULTIPATH

Based on DYMO, a multipath protocol is developed, which supports the usage of multiple node disjoint paths towards the destination. In order to minimize the additional complexity needed so as the source can learn and maintain more than one path towards the destination, the efforts are concentrated on the minimization of the necessary changes.

In this framework, no change is made in the Route Request procedure, regarding the intermediate nodes. An intermediate node applies to routing messages the same rules as DYMO. However, these rules need to be modified in the case of the source and destination nodes. Thus, when a RREQ is received by its target node (i.e. the destination), the latter applies only the rules referring to stale routes and possible routing loops and omits the rule referring to inferior routes. The idea behind this is that we are interested not only in the best route (usually the shortest one) but in longer routes as well. This is necessary in order for the destination to be able to store routes longer than the shortest one.

However, due to the restriction for loop freedom, the alternate routes cannot be more than one hop longer than the shortest one. Applying the above rules we result in the creation of multiple routes between the source and the destination that are at most one hop longer than the shortest one. This restriction is not a drawback, since various previous works indicated that multiple routes are advantageous if they are not much longer than the shortest one. Finally, there is the possibility of comparing all the paths travelled by the RREQs if we exploit the path accumulation mechanism of the DYMO protocol.

In this case, some additional information (the intermediate nodes) is needed to be maintained by the destination node for every path it stores towards the source. This is needed so that they are able to compare the paths and store only the node-disjoint ones. The same requirement is applied to the source node for a similar reason. RREPs are used in the same way as in DYMO for the creation of the reverse paths. The source receives all RREPs and using the same rules as the destination did for the RREQs, it stores multiple routes towards the destination. Then it starts using the shortest one. When this one breaks, a RERR is created that travels towards the source and all nodes receiving it invalidate all the paths through the broken link, and so does the source. Then, the source searches for an alternative path in its routing table. If such a path exists, it uses it to forward the packets, thus avoiding flooding another RREQ in the network. If not, a new Route Request procedure will take place, when the next data packet will be created and ask for a valid route. There is also an alternative for the RERRs: setting the nodes to initiate RERRs for all their stored paths, not just those nodes only but those that are used for data forwarding. In this way, a broken alternative path will be invalidated before it is selected for packet forwarding, so future packet drops are avoided. However, this technique requires the usage of appropriate neighbour discovery mechanisms, such as the well known method of Hello packet transmissions.

In order for this method to be beneficial, the Hello packets should be transmitted frequently. This results enough in bandwidth consumption. In fact, a reactive multipath protocol would require both link-layer and network-layer neighbour discovery mechanisms, link layer is needed for fast identification of broken links and the network layer for discovering link breakages on alternative links.

It was decided to keep the Route Error procedure the same as in the original DYMO protocol, considering that this results in a less complex protocol and realizing that it can have a negative impact on the performance of DYMO multipath in terms of packet delivery ratio and delay. Another issue is the timeout value for the invalidation of paths. DYMO uses timers to invalidate routes, if they are not being used. In order to avoid deleting a valid alternative route when the primary one is used, DYMO multipath uses large timeout values for path invalidation, like other multipath protocols

Summarizing the issue it is stated that by using the path accumulation technique and slightly modifying the DYMO protocol regarding the treatment of RREQs and RREPs by the source and the destination, it resulted in a new protocol that is benefited by the usage of multiple paths. In this research work the numbers of alternative paths were restricted to one alternative node disjoint path. This is because as the addition of more than one alternative routes does not provide large benefits to the overall performance of a multipath routing protocol. In fact, DYMO multipath relies on the reception of RREQs via several paths and doesn't specify any mechanism that intermediate nodes should follow in order to ensure the creation of multiple paths. This means that alternative routes may not be always available, as a result of the particular paths followed by RREQs towards the destination node.

V. SIMULATION TOOLS

The ns-2 [12] was used to evaluate the performance of DYMO multipath, compared to AODV. Nodes were set from 10 to 100.they move inside a 1500m x 400m rectangle area according to the Random Waypoint Model.Sources started sending CBR/UDP traffic for 1500 msec of simulation time each packet being 512 bytes in length.

The main evaluation metrics were:

Average Throughput: It is defined as the total amount of data per time unit that is delivered from one node to another via a communication link.

Jitter: It is the variation in time between arrivals of packets. It is the deviation from the ideal delay or latency. It is caused by network congestion, a sudden network topology change or route changes.

Packet Delivery ratio: This ratio reflects the network throughput, routing protocols that adapt to the effectiveness of changes in network topology and the performance.

VI. SIMLULATION RESULTS AND NALYSIS

6.1 Average Throughput

Figure 7 shows the X-graph between the AODV and DYMO-MULTIPATH. In this xgraph X-Axis denotes the number of nodes from 10 to 100 and Y-Axis denotes the throughput and values ranges from 35 to 145.Green line indicates DYMO-MULTIPATH and red lines indicates AODV protocol.

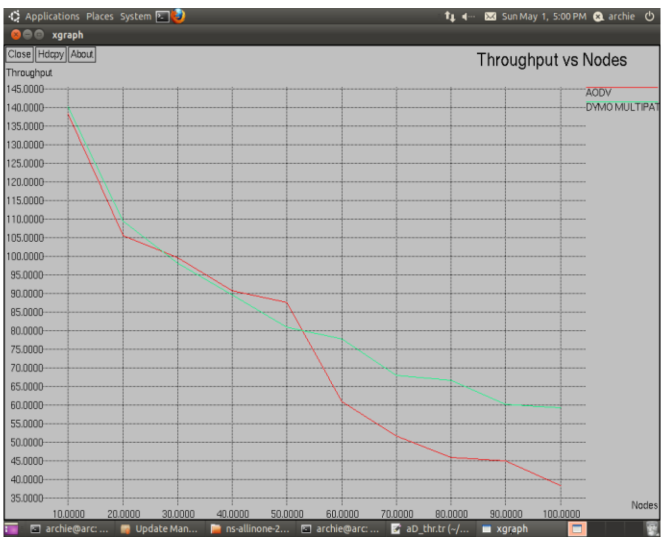

Figure 7: Xgraph for Average Throughput vs nodes

Figure 8 shows the excel graph between average throughput vs nodes. This graph contains the nodes ranging from 10 to 100.

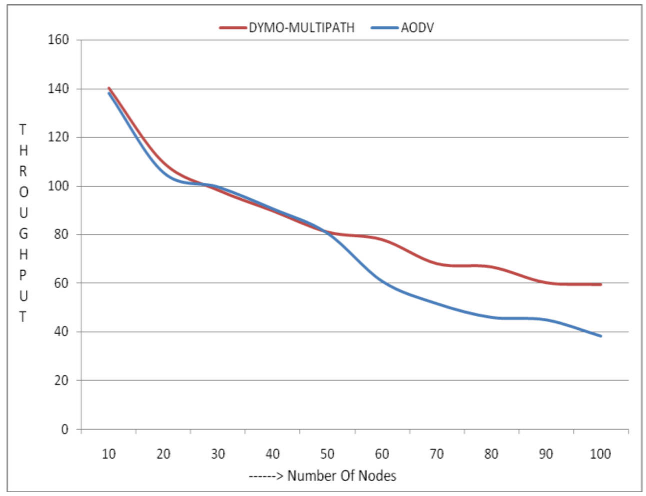

Figure 8 Excel Graph for Average Throughput vs nodes

ANALYSIS: In graph of fig7 x-axis denotes Number of nodes and y axis denotes throughput. Here blue line denotes AODV protocol and Red line denotes DYMO – MULTIPATH. This graph indicates that throughput is nearly same for both protocols for nodes in the range from 10 to 50.From 50 to 100 nodes our protocol i.e. DYMO performs better in comparison to AODV.

6.2 Jitter

Figure 9 shows the Xgraph for Jitter and number of nodes. In this Xgraph X-axis contains nodes from 20 to 60 and Y-axis contains the jitter values from 150 to 700. Green line indicates the DYMO-MULTIPATH and red line indicates AODV.

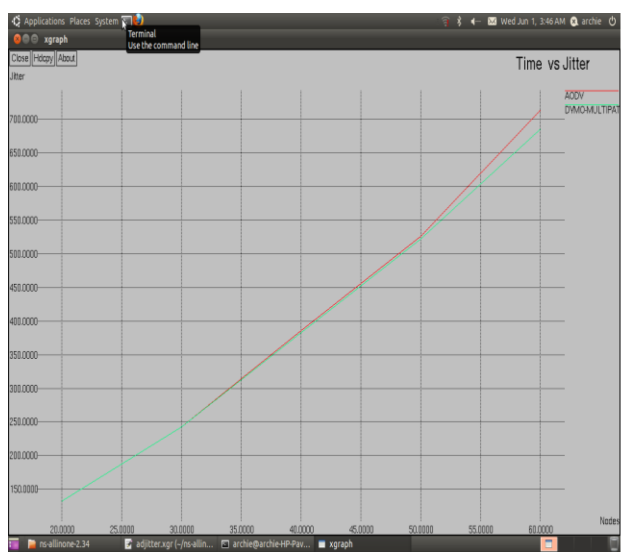

Figure 9 Xgraph for Jitter vs nodes

Figure 10 shows the excel graph between jitter and number of nodes. In this Excel Graph X-axis contains the number of nodes ranging from 20 to 60 nodes and Y-axis contains the jitter values from 100 to 800.

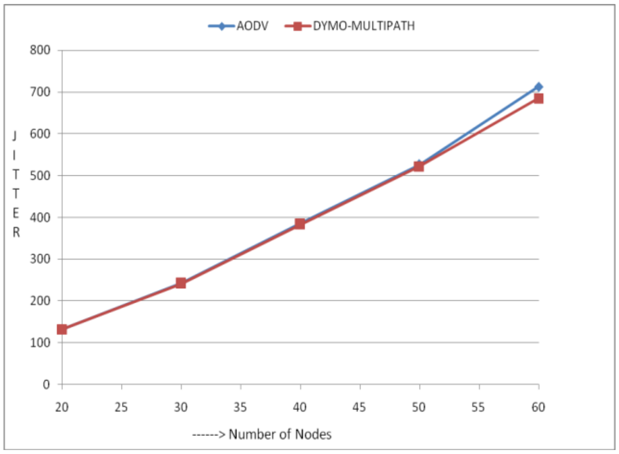

Figure 10 Excel graph for Jitter vs nodes

ANALYSIS: This graph contains the values for two protocols: Red line is for DYMO-MULTIPATH and blue line is for AODV. This Graph shows that with the increase in number of nodes jitter also increases. But our protocol gives less jitter in comparison to AODV.

6.3 Packet Delivery Ratio

Figure 11 shows the Xgraph between Packet Delivery ratio versus number of nodes. In This Xgraph X-axis contains the number of nodes from 10 to 100 and y-axis contains the value of packet delivery ratio ranging from .993 to .997.

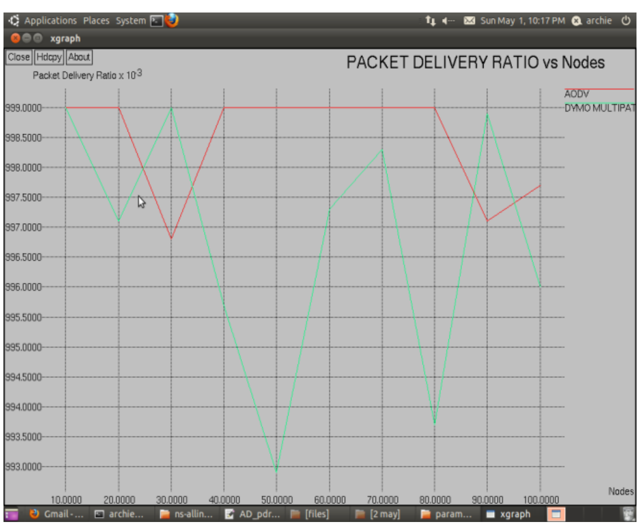

Figure 11 Xgraph for Packet Delivery Ratio vs nodes

Figure 12 shows the excel graph for Packet delivery ratio versus number of nodes. It contains the number of nodes on X-axis and packet delivery ratio on Y-axis.

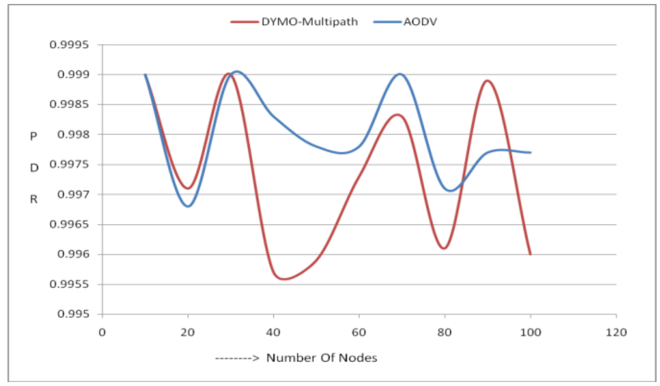

Figure 12 Excel graph for Packet Delivery Ratio vs nodes

ANALYSIS: In this Excel graph the X-axis is for number of nodes and Y-axis contains the packet delivery ratio. Red line is for DYMO-MULTIPATH and Blue line is for AODV. From this graph, it is analyzed that both the protocols perform same till 30 nodes. From 30 to 60 nodes DYMO shows decreasing performance because of the multipath concept as there are large number of ways in comparison to unicast routing protocol. From 60 to 80 nodes the trend of the protocols is same with AODV performing slightly better. Beyond 80 nodes the performance of DYMO improves significantly.

CONCLUSION

The DYMO routing protocol is designed for mobile ad hoc networks in small, medium, and large node populations. DYMO handles all mobility ranges. DYMO can handle various traffic patterns, but is most suited for sparse traffic sources and destinations. DYMO is designed for network where trust is assumed, since it depends on nodes properly forwarding traffic to the next hop toward the destination on behalf of the source. DYMO routing protocol with excellent performance is simple, compact, easy to implement and highly scalable characteristics, and is a very promising protocol.

In this work DYMO-Multipath is presented, a multipath algorithm based on DYMO routing protocol. Simulations show that as the node increases, throughput decreases. The new routing Protocol DYMO-Multipath works well on the average throughput parameter. As the no of nodes increases the performance of DYMO is better than AODV. The second parameter is jitter. As the number of nodes increases, jitter increases. As compared to AODV,

ISSN 2229-5518

DYMO multipath gives less jitter. The last parameter is Packet delivery ratio. It is analyzed that both the protocols perform same till 30 nodes. From 30 to 60 nodes DYMO shows decreasing performance because of the multipath concept as there are large number of ways in comparison to unicast routing protocol. From 60 to 80 nodes the trend of the protocols is same with AODV performing slightly better. Beyond 80 nodes the performance of DYMO improves significantly.

Based on simulation analysis, it is established that DYMO and AODV, DYMO-multipath perform better than AODV exhibit lesser jitter and consequently more throughput and lesser packet loss as we increase the number of nodes.. It is also clear that DYMO, though a derivative of AODV is more efficient than the latter since it takes advantage of its salient features carefully pruning its weaknesses.

In the future, we intend to examine its performance when RERRs are allowed to be transmitted for unused paths and investigate the effect of invalidation period for alternative paths.

Acknowledgement

The assistance of University Institute of Engineering & Technology (UIET), Panjab University, Chandigarh and Prof. Savita Gupta, UIET, for her valuable guidance and insightful comments on the paper, are noted with gratitude.

References

[1] IETF Working Group: Mobile Adhoc Networks(manet).http://www.ietf.org/html.charters/manet-charter.html.

[2] Ad hoc routing protocol list. http://en.wikipedia.org/wiki/List_of_ad-hoc_routing_protocols.

[3] I. Chakeres, E. Belding-Royer, UC Santa Barbara, etc. Dynamic MANET On-demand (DYMO) Routing [Zl Intemet-Draft,draft-ietfmanet- dymo-OO (work in progress) ,January 2005.

[4] I. Chakeres, CenGen, WiChorus, etc. Dynamic MANET On-demand (DYMO) Routing[Z]. Intemet-Draft, draft-ietf-manet-dymo-21 (work in progress) , July 26, 2 010.

[5] I. Cidon, R. Rom, and Y. Shavitt, "Analysis of multi-path routing," IEEE/ACM Transactions on Networking, vol. 7, no. 6, pp. 885-896, Dec. 1999.

[6] K. Kuladinithi, M. Becker, C. Gorg, and S. Das, "Radio disjoint multi-path routing in manet," in Center of Excellence in Wireless and Information Technology 2005 Conference (CEWIT 2005), Stony Brook, Dec. 2005.

[7] D. Xu, Y. Chen, Y. Xiong, C. Qiao, and X. He, "On the complexity of and algorithms for finding the shortest path with a disjoint counterpart, "IEEE/ACM Transactions on Networking, vol. 14, no. 1, pp. 147-158,Feb. 2006.

[8] A. Nasipuri, R. Castaneda, and S. R. Das, "Performance of multipath routing for on-demand protocols in mobile ad hoc networks," ACM/Kluwer Mobile Networks and Applications (MONET), vol. 6, no. 4,pp. 339-349, Aug. 2001.

[9] K. Wu and J. Harms, "On-demand multipath routing for mobile ad hoc networks," in Proc. European Personal and Mobile Communications Conference 2001 (EPMCC '01), Vienna, Austria, Feb. 2001, paper No.21.1.

[10] N.Sharma and S.Rana " Provisioning of Quality of Service in MANETs Performance Analysis &Comparison (AODV and DSR)", 2nd International Conference on Computer Engineering and Technology IEEE 2010.

[11] J.Benkovi, I.Kotuliak,P.Trúchly " Comparison of DSR and DYMO routing protocols in building Environments", 51st International Symposium ELMAR-2009, 28-30 September 2009, Zadar, Croatia

[12] The VINT Project. The network simulator - ns-2. http://www.isi.edu/nsnam/ns/.

Archie Budhiraja (Author-1) is currently pursuing masters degree program in information technology in University Institute of Technology, Panjab University, Chandigarh, India,PH-91-9896065825.
E-mail:archie.budhiraja@gmail.com

Roopali Garg (Author-2) is Assistant Professor in the department of information technology in University Institute of Technology, Panjab University, Chandigarh, India, PH-91-9417724824. E-mail: roopali.garg@gmail.com