Comparison of DYMO protocol with respect to various quantitative performance metrics

Narendran Sivakumar
Masters in Software Engineering
Malardalen University
nsr09002@student.mdh.se

Satish Kumar Jaiswal
Masters in Software Engineering
Malardalen University
skr09003@student.mdh.se

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. Dynamic Mobile Ad hoc Network On-demand (DYMO) routing is one such protocol that is 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. In this paper, we have compared the Dynamic Mobile Ad hoc Network On-demand routing protocol with existing routing protocols. We use the implementation based on the specification given in Dynamic Mobile Ad hoc Network On-demand Internet Engineering Task Force to (a) Evaluate the protocol with respect to various quantitative performance metrics like jitter, throughput and delay. (b) To compare this with existing Ad hoc routing protocols. Our findings reveal that DYMO and Ad hoc on Demand Distance Vector (AODV) tend to have a higher packet delivery ratio whereas the jitter experienced by Destination-Sequenced Distance Vector (DSDV) is the highest. Also, the throughput of DYMO, AODV, Dynamic Source Routing (DSR), and DSDV was found to be closer to each other, but the effect of jitter on throughput in DYMO is much lesser than other protocols. The delay experienced by DYMO is the lowest at 0.0278s. This paper is intended for audience having prior knowledge about network routing protocols and its related quantitative performance metrics.

Keywords: *Dynamic MANET on Demand Routing, AODV, DSR, DSDV, Network Simulator 2.*

1. INTRODUCTION

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. Each device in network can act as a router and thus, must forward the traffic that is not related for its own use. MANETs are also capable of handling topology changes and malfunctions in nodes through network reconfigurations [1]. A brief classification of Ad-hoc routing protocols is given in figure 1.

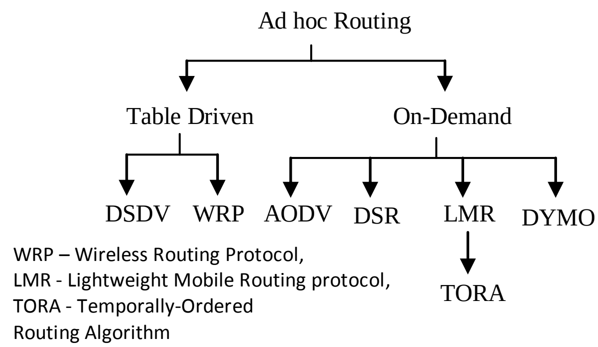

Figure 1: Classification of Routing Protocols in MANETs

In Table-based protocol [9], each node is capable of maintaining a routing table that contains routes to all nodes in the network. Nodes must be able to exchange messages periodically with routing information to keep the routing tables up-to-date [2]. Because of the dynamic nature of ad hoc networks, a considerable number of routing messages may have to be exchanged in order to keep routing information updated. Since at all times, the routes to all destinations are ready to use, initial delays before sending data are small. In [6] and [9], the authors have exposed the limitations of table-driven protocols with respect to overhead, network congestion caused, usage of bandwidth, battery and network resources.

In On-demand protocols [3], nodes compute the routes and maintain routing information only when it is needed, thereby establishing routes as and when required by the source. The route maintenance procedure is responsible for maintenance of the routes from the moment they are established. The routes are maintained as long as the route is required and the destination is accessible along every possible path from the source [4].The route maintenance procedure was designed to overcome the wasted effort in maintaining unused routes. As reactive routing protocols flood the networks to discover the route, they are not optimal in terms of bandwidth utilization, but they scale well in the frequency of topology change [5].

Thus this strategy is suitable for large, high mobility networks. In [6], authors Ian D. Chakeres and Elizabeth M. Belding-Royer illustrate new mechanisms like snooping and netfilter operations to overcome the problem of high bandwidth utilization and topology change.

Section Overview: The paper is organized into four sections. The second section deals with overview of DSDV,DSR and AODV routing protocol and the third deals with the introduction to DYMO routing protocol and a brief overview of its working. The fourth section deals with overview of NS2 simulator and Tracegraph analyser used for routing protocol evaluation. The fifth section deals with performance comparison of routing protocol with respect to Packets Transmission, Jitter, Throughput and Delay. The fourth section deals with the conclusions derived from DYMO routing protocol simulation and last section contains the list of all references.

2. Overview of DSDV, DSR and AODV routing Protocols

Destination-Sequenced Distance Vector (DSDV) is a hop-by-hop distance vector protocol based on the classical Bellman-Ford mechanism. In DSDV, each node maintains a routing table which contains an entry for destination node in the network [14]. The routing table contains entries such as the next hop address, metric or the number of hop counts, and the sequence number. Sequence numbers are assigned by destination node for identification of the routes. DSDV tags each route with a sequence number and considers a route X more favorable than Y if X has a greater sequence number, or if the two routes have equal sequence numbers but X has a lower metric. This was done so that the routing tables have the latest updated path. The sequence number for a route is updated every time a new route discovery is initiated. When a broken link is encountered, the sequence number is set to infinity and it is broadcasted to all the nodes so that the routing tables of the node containing the broken link can be updated to infinity and the link is discarded. The sequence number of every route is assigned by the destination and it is incremented for every route discovery operation. Thus in case of mobile ad-hoc networks, the sequence numbers enable DSDV to maintain up to date routing information at the nodes ensuring the consistency of routing data across all routing tables. Both periodic and triggered route updates are initiated by DSDV to maintain consistency of routing information. In case of periodic updates, fresh route discovery operations are initiated after the elapse of fixed interval of time. Triggered route updates are initiated whenever a node encounters a broken link which can be a result of sudden network topology change or communication link failure.

Dynamic source routing (DSR) is based on source routing where the source specifies the complete path to the destination in the packet header. All intermediary nodes along the path simply forwards the packet to the next node as specified in the packet header [14]. This means that intermediate nodes only need to keep track of their neighboring nodes to forward data packets. The source on the other hand, needs to know the complete hop sequence to the destination. This eliminates the need for maintaining latest routing information by the intermediate nodes as in DSDV. In DSR, all nodes in a network cache the latest routing information. When more than one route to the destination is found, the nodes cache all the route information so that in case of a route failure, the source node can look up their cache for other possible routes to the destination. If an alternative route is found, the source node uses that route; else the source node will initiate route discovery operations to determine possible routes to the destination. During route discovery operation, the source node floods the network with query packets. Only the destination or a node which already knows the route to destination can reply to it, hence avoiding the further propagation of query packets from it. If a broken link is detected by a node, it sends route error messages to the source node. The source node on receiving error messages will initiate route discovery operations. Unlike DSDV, there are no periodically triggered route updates.

Ad Hoc On-Demand Distance Vector (AODV) routing protocol [14] shares DSR's on-demand characteristics in that it also discovers routes on an "as needed'' basis via a similar route discovery process. However, AODV adopts a different mechanism to maintain routing information. It uses traditional DSDV routing tables, with one entry per destination. AODV uses sequence numbers maintained at each destination to determine whether the routing information is latest and up to date and to prevent looping of packets across routes. Just like DSDV, during route discovery AODV floods broadcast route request packets to its neighboring nodes. Each route request contains the source node id, the destination node id, the sequence numbers of the source and the destination, the hop count and the broadcast id. Every time a route discovery operation is initiated, the source sequence number is incremented. AODV depends on sequence numbers to avoid count to infinity problem [15]. Like DSR, a response message is generated once the request packet reaches the destination or a node which knows the route to the destination. This response message contains the latest route information from the source to the destination. AODV also has a feature called "precursor list" maintained by every node. It contains the list of neighbor nodes that are most likely to use the current node for routing packets. Routing table entry of every node contains a list of predecessor nodes that most likely to use that entry to route data packets. This becomes essential when a broken link is encountered. On encountering such broken links, all the predecessor nodes will be notified with route error messages.

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

The DYMO routing protocol is a recently proposed protocol defined in Internet Engineering Task Force (IETF) Internet-Draft [7] and this paper uses the terminologies described in that draft. It is currently in its seventeenth version. Using Ad hoc on Demand Distance Vector (AODV) as the basis, DYMO borrows "Path

Accumulation” from Dynamic Source Routing [8] and removes unnecessary Route Reply (RREP), precursor lists and Hello messages (Route exploration messages), thus simplifying AODV [4]. It retains sequence numbers, hop count and Route Error (RERR) messages from AODV [7].

3.1. 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. This protocol has two basic operations - route discovery and route management [7]. The core of the route maintenance and route maintenance is based on IETF Internet-Draft DYMO specification.

Route Discovery - In this operation, the originating node initiates flooding of Route Requests (RREQ) throughout the network to find the target node, where each intermediate node records the route to the originating node. On receiving the RREQ, the target node responds with a Route Reply (RREP) which is sent in a unicast, hop-by-hop fashion towards the originating node. On receipt of RREP by originating node from the target node, the routes between the originating node and the target node are established in both directions.When a node receives an RREQ, it processes the addresses and associated information found in the message. The information for a node is compared with the corresponding entry in the routing table of the node, if one exists. The information about the originator found in the RREQ is processed first, but subsequent entries are processed the same way:

• If the routing table does not contain an entry for the originator, one is created. The next hop entry is the address of the node from which the RREQ was received. Likewise, the next hop interface is the interface on which the RREQ was received.

• If an entry exists, the sequence number and hop count found in the RREQ is compared to the sequence number route and hop count in the table entry to check if the information in the RREQ is stale or should be disregarded.

• If an entry exists and is not stale or disregarded, the entry is updated with the information found in the RREQ.

Route Maintenance - In order to respond to the changes in network topology, nodes maintain their routes and monitor the links over which the network traffic flows. When a received packet is to be forwarded to some other node where the route is unknown or broken, the source of the packet is notified by sending Route Error (RERR) that indicates the current route is broken.

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.

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 a RERR message once.

Source packet on receiving the RERR performs route discovery to deliver the remaining packets. Sequence numbers are used by DYMO to ensure loop freedom so that the order of DYMO route discovery messages is determined without the need for prior routing information.

3.2. 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. SIMULATION ENVIRONMENT

Network Simulator (NS) is an object-oriented, event driven simulator which is suitable for designing new protocols, comparing different protocols and traffic evaluations.

NS uses languages C++ and Tool Command Language (TCL) [10] to perform two different kinds of runtime requirements. The detailed simulation of protocols necessitates a programming language that can effectively manipulate packet headers, bytes and efficiently implementing algorithms that runs on large data sets. All the above mentioned requirements stipulate a higher runtime speed where as error reporting and correction are not that important. For these tasks C++ is used as it runs faster hence suitable for protocol development. Other tasks such as simulation of various traffic scenarios, configuration of simulation and network parameters, network topology requires a programming language that allows configuring these tasks and simulation faster and simpler. So TCL is used as it provides a faster and an interactive way to change configuration information or varying simulation parameters. The front-end module, written in TCL, describes the network topology with the number and positioning of nodes, creation of agents on top of nodes and scheduling of agents to generate traffic. TCL simulation modules define the network (number of nodes, links, position, topography) and graphically represent the traffic flow between nodes. The simulator supports class hierarchy in C++ and similar class hierarchy in TCL interpreter. The C++ class hierarchy called compiler hierarchy and the TCL hierarchies called interpreted hierarchy are closely linked to each other and share a one-to-one correspondence. The objects that were instantiated in interpreter hierarchy results in the instantiation of a corresponding object in the compiler hierarchy. The root of these two hierarchies is the TCLObject class.

4.1 Simulation tools

The simulation module created using TCL makes use of two tools to simulate the implementation and evaluate its performance:

- NAM (Network Animation Model) – NAM [12] is a TCL based animation tool that can be used for viewing network simulation traces. It supports various network topologies and displays packet level animation. It works based on the configurations (see Section 2.2) and commands specified in the TCL module.
- Tracegraph [11] – It is a trace files analyzer in NS 2, which reads trace files written by the DYMO implementation and displays the summarized data in the form of a graph.

4.2 Simulation configuration

The following are the configurations set as per the assumed simulation context:

- Channel type – Wireless
- Network Interface type – Physical wireless
- Routing protocol – DYMO
- Interface queue type – Priority queue
- Queue Length – 50 packets
- Number of nodes in topography – 6
- X and Y Dimensions of topography – 500*400 sq.m
- Time of Simulation end – 150 s.

4.3 Simulation module

The simulation module, after deploying the configurations mentioned in Section 2.2, performs the following steps:

- It creates an instance of the Simulator class defined in the NS library.
- Creates and opens in write mode trace files.
- Creates a topography object with dimensions as specified in Section 2.2.
- Creates given number of nodes using Node class.
- Positions nodes in required places in the topography using XY coordinate.
- Schedules movement of certain nodes at certain time intervals.
- Creates required number of agents and attaches them to the nodes.
- Connects source and destination nodes.
- Schedules start and stop of the simulation at required time.
- Writes trace file at the end of simulation.

5. SIMULATION RESULTS

The simulation analysis of four protocols DYMO, AODV, DSR, and DSDV primarily focuses on a few simulation parameters (Packet Transmissions, Jitter, Throughput, and Delay).

5.1 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. 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. Packet loss can be caused by a number of factors, including signal degradation over the network, oversaturated and highly congested network links, corrupted and faulty packets rejected, faulty networking hardware.

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.

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.

5.2. Packet Transmission:

The packet transmission details of the four protocols generally indicate that the protocols DYMO [3] and AODV [8] tend to have a higher packet delivery ratio which is a ratio of number of packets transmitted to number of packets dropped or lost (*see table1*) whereas the packet delivery ratio of DSR and DSDV tend to be much lower than the other protocols. The losses suffered by DSR and DSDV may have happened in response to a dynamic changing topology. Each routing protocol requires a robust Route Discovery and Route Maintenance to cope with the dynamic changing topology. DYMO and AODV have such high packet delivery ratio, as both are reactive (route cache) as well as incorporating some features of table driven protocols (routing table).

Table1 [Simulation Statistics]

PARAMETERS	DYMO	AODV	DSR	DSDV
Number of generated packets	3757	3700	3777	3038
Number of sent packets	3619	3561	3605	2911
Number of forwarded packets	470	482	509	374
Number of dropped packets	140	148	220	142
Number of lost packets	610	781	776	650
Minimal packet size	24	28	28	28
Maximal packet size	1072	1072	1104	1072
Average packet size	282.8582	297.8778	288.7804	304.9219
Packets dropping nodes	0 1 2 3 4 5	0 3 5	0 1 2 3 4 5	0 1 5
Minimal delay (CN, ON, PID)	0.000640471(0,1,340)	0.000640471 (0, -1,0)	0.000640472 (5,-1,0)	0.027135538 (0,5,120)
Maximal delay (CN, ON, PID)	1.001006484 (0, -1,0)	2.271632516 (0,5,62)	6.084161062 (0,5,98)	1.077742469 (0,5,3042)
Average delay	0.02780103516	0.04171916814	0.04842961264	0.9835345393

CN:<current node number> **PID**: <packet ID> **ON:** <other node number> **TIL:** <time interval length> (for throughput graphs)

5.3 Jitter

The Jitter experienced by these protocols cannot to be quantified and they can only be bounded within a delay range. The jitter experienced by DSDV (*see figure 4*) is the highest and can be given a range between 0.0 to 0.05 and to a small extent to 0.06 with peak jitter at 0.08.The jitter experienced by DSR (*see figure 5, next page*) though within the range between -0.01 to 0.01, is highly indeterminate. The jitter experienced by AODV (*see figures 3, 7*) is somewhat moderate within the range and -0.01 to 0.01. The jitter experienced by DYMO (*see figures 2, 6*) is the lowest and can be given a range between 0.0 to 0.01and to a small extent from 0.005 to 0.01. It is highly discernible from the graphs that DYMO routing protocol has the lowest jitter range followed closely by ADOV and other protocols DSR and DSDV with a higher range.

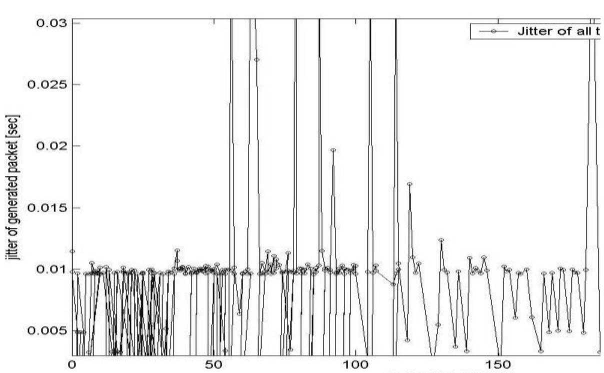

Figure 2 [DYMO Simulation - Jitter]

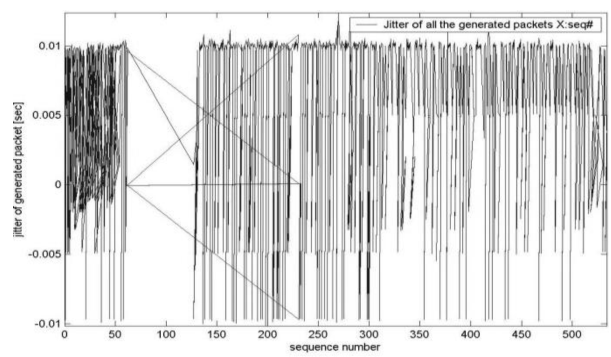

Figure 3 [AODV Simulation - Jitter]

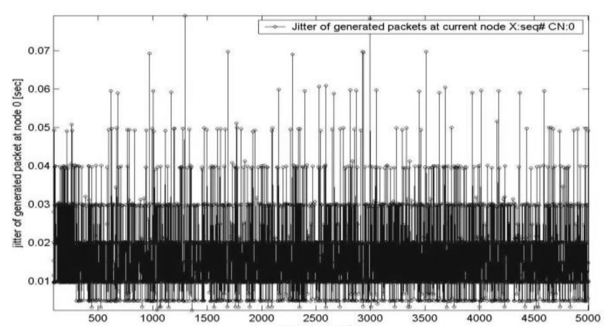

Figure 4 [DSDV Simulation - Jitter]

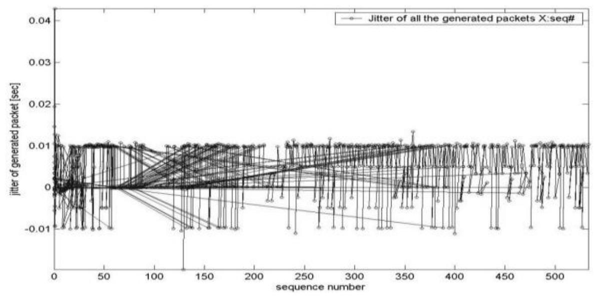

Figure 5 [DSR Simulation - Jitter]

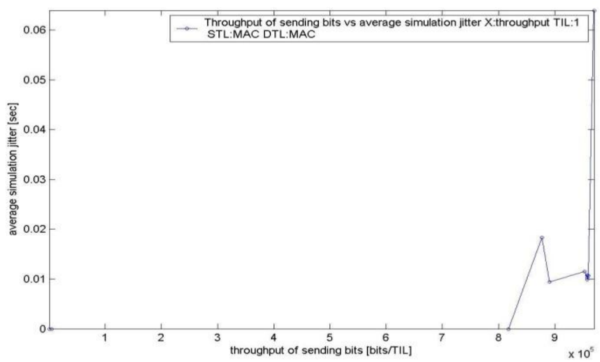

Figure 6 [DYMO Simulation - Throughput vs. Jitter for packets sent]

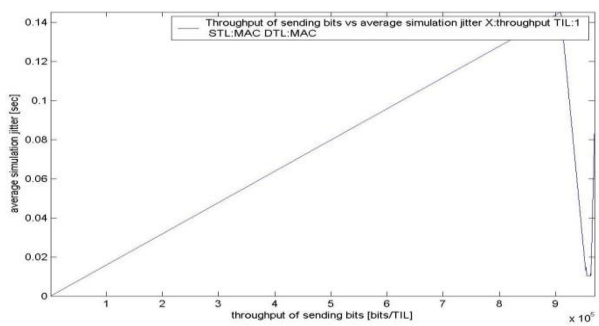

Figure 7 [AODV Simulation - Throughput vs. Jitter for packets sent]

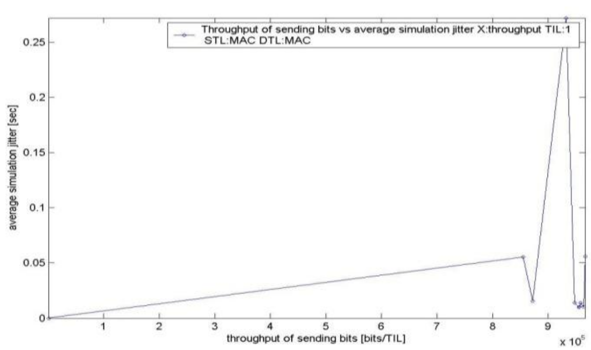

Figure 8 [DSR Simulation-Throughput vs. Jitter packets sent]

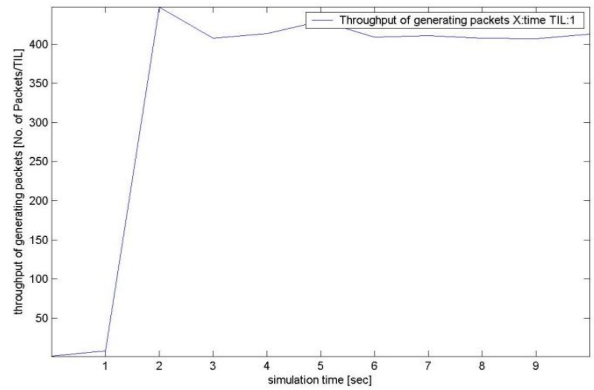

Figure 9 [DYMO Simulation - Throughput of DYMO]

5.4 Throughput

The throughput of the protocols can be defined as percentage of the packets received by the destination among the packets sent by the source. The throughput of DYMO (*see figure 9*), AODV, DSR, and DSDV are closer to each other, but the effect of jitter on throughput is much starker among the protocols. The effect of throughput on jitter in DYMO (*see figure 6*) is much less than other protocols. The effect of throughput on jitter for sent packets is very minimal at the beginning, peaking out at the end for DYMO. This indicates that although the jitter increases at higher throughput for the sent packets, jitter sets in only after the throughput (*see figure 6*) of 800,000 bits time interval length, whereas in AODV, DSR (*see figures 7, 8 resp.*) jitter sets immediately even at zero throughput and in DSDV at 400,000bits. It can be established that by analyzing effect of throughput on jitter for the received packets in DYMO, the jitter although prevalent in the initial stages settles down to a minimal level as the throughput increases thereby having the least influence over DYMO protocol. The same may not apply for AODV, DSR, and DSDV where jitter has very high influence over throughput.

5.5 Delay

The delays experienced by the protocols are a crucial factor contributing to jitter and can adversely affect the performance of the protocol. The delay experienced by DYMO (*see table 1*) is the lowest at 0.0278s which is nearly half of the delay AODV (*see table 1*) experienced which is at 0.0417s. The peak delay experienced by DYMO is far lower than AODV, DSR and DSDV at 1.001s. The peak delay experienced by DSR is the highest at 6.0841s. This establishes the fact that since the average delay experienced by DYMO is significantly lower than ADOV, DSR, and DSDV thereby felt the least influence of jitter among the other protocols. The effect of delay on throughput is an important factor, as delays may adversely affect the throughput. The effect of delay on throughput in DYMO (*see figure 10*) is higher in the beginning; it reduces drastically as throughput increases and settles down at a minimal level in the end. DYMO is the only protocol among AODV, DSR and DSDV where the delay decreases sharply as the throughput increases, on the contrary for protocols AODV, DSR, DSDV(*see figures 11, 12, 13 resp.*) the delay increases as the throughput increases with AODV and DSDV having the sharpest increase, thereby explaining the increased jitter experienced by these protocols.

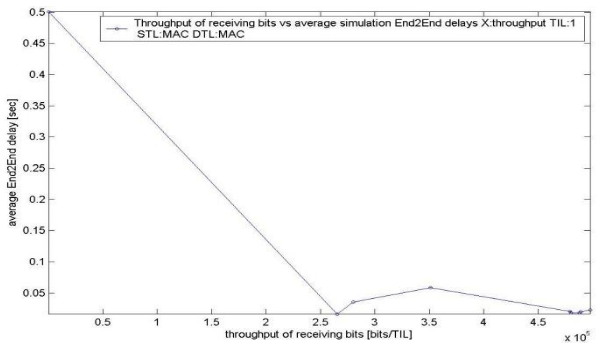

Figure 10 [DYMO Simulation - Throughput vs Delay]

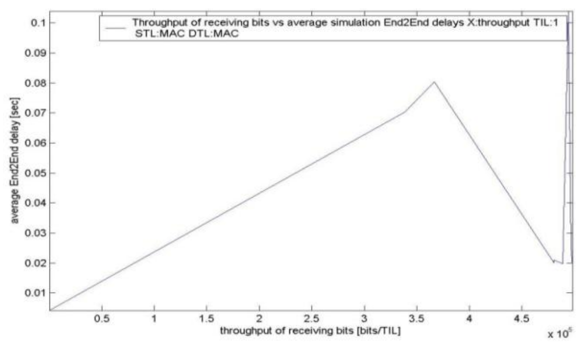

Figure 11 [AODV - Throughput vs. Delay]

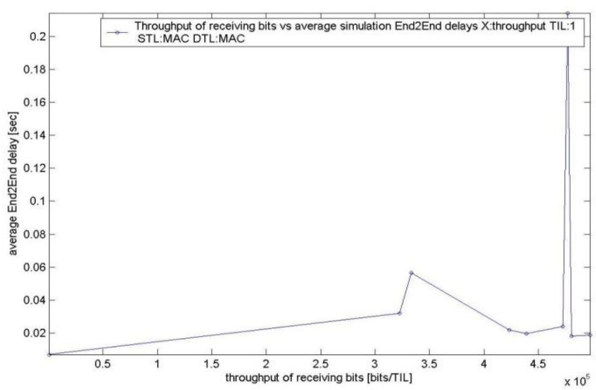

Figure 12 [DSR Simulation - Throughput vs. Delay]

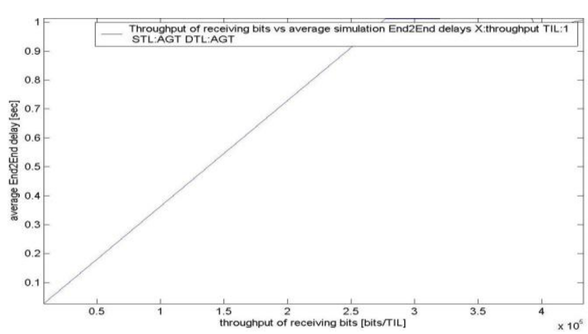

Figure 13 [DSDV Simulation - Throughput vs. Delay]

6. CONCLUSIONS

In this paper, we have reviewed and implemented DYMO Routing Protocol for Ad hoc networks and have evaluated this newly proposed protocol in comparison with the existing protocols. We established that DYMO provides better performance than others when compared in a given network topology with respect to Quality of Service (QOS) parameters, i.e., throughput, jitter, delay, latency. The throughput of DYMO and AODV protocol are quite similar however, as the throughput increases the jitter also increases for AODV, DSR and DSDV routing protocols. The jitter experienced by DYMO is the lowest at 0 to 0.1 seconds halving the jitter experienced by AODV. This is significant for the fact that, as the variations of packet delay or jitter becomes more predictable, the routing mechanisms can factor in that delay to determine whether the packet is lost or not. Instead of a mobile node waiting for 0.2 seconds for AODV packet, it needs to wait for 0.1 seconds to determine whether the packet is lost or not. Thus saving crucial time that can be utilized to initiate fresh route discovery operations. The delay experienced by DYMO is 0.0278s which again halves the delay experienced by AODV. The delay experienced by mobile nodes becomes an important factor for these routing protocols as the increase in throughput also results in increase in delay. This is true for AODV, DSR and DSDV protocols as evident from the detrimental effect of delay on throughput observed in the simulation. However for DYMO routing protocol, the delay reduces as the throughput increases and becomes close to zero for the throughput of 500000 bits per time interval length.

Based on simulation analysis, it is established that DYMO and AODV, owing to their hybrid characteristics, i.e., of both reactive and proactive protocols, exhibit lesser delay and consequently more throughput, lesser packet loss and jitter. 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. Whereas the difference in performance between the other three protocols is typically unremarkable, DYMO succeeds against them by a significant margin as is manifested in the graphs.

Our implementation of the DYMO specification can be further extended twelfth version of IETF for future implementations including MANET Neighborhood Discovery Protocol (NHDP), a newer version of generalized MANET packet and message format, and the three additional kinds of timeouts. The latest version of IETF draft also requires DYMO to support Simple Internet Draft [13] where a sub network of all DYMO routers connect with internet using a single Internet DYMO Router (IDR).

REFERENCES

[1] Ad hoc routing protocol list. *http://en.wikipedia.org/wiki/List_of_ad-hoc_routing_protocols*. Last accessed March 2007.

[2] 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.

[3] 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.

[4] Ian D. Chakeres and Elizabeth M. Belding-Royer. *The utility of hello messages for determining link connectivity*. In Proceedings of the 5th International Symposium of Wireless Personal Multimedia Communications (WPMC) 2002, volume 2, pages 504–508, Honolulu, Hawaii, October 2002.

[5] Ian D. Chakeres and Elizabeth M. Belding-Royer. *AODV routing protocol implementation design*. In ICDCSW '04: Proceedings of the 24th International Conference on Distributed Computing Systems Workshops - W7: EC (ICDCSW'04), pages 698–703, Washington, DC, USA, 2004. IEEE.

[6] Ian D. Chakeres and Elizabeth M. Belding-Royer. *AODV implementation design and performance evaluation*. International Journal of Wireless and Mobile Computing (IJWMC),2005.

[7] Ian D. Chakeres and Charles E. Perkins. *Dynamic MANET on demand (DYMO) routing protocol. Internet-Draft Version* 06, IETF, October 2006, (Work in Progress).

[8] Sumit Gwalani, Elizabeth M. Belding-Royer, and Charles E. Perkins. *AODV-PA: AODV with path accumulation*. In IEEE International Conference on Communications (ICC' 03), volume 1, pages 527–531, Anchorage, Alaska, May 2003. IEEE.

[9] 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

[10] NS, *http://www.isi.edu/nsnam/ns/doc/node5.html*

[11] Tracegraph,http://www.iterasi.net/openviewer.aspx?sqrlitid=vv3uxzdreucu7drikilcja

[12] NAM, *http://www.isi.edu/nsnam/nam/*

[13] Ian D. Chakeres and Charles E. Perkins. *Dynamic MANET on demand (DYMO) routing protocol. Internet-Draft Version* 17, IETF, October 2006, (Work in Progress). http://tools.ietf.org/html/draft-ietf-manet-dymo-17

[14] Amitava Mukherjee, Somprakash Bhandyopadhyay, Debashis Saha. *Location Management and Routing in Mobile Wireless Networks*, Artech House Mobile Communication Series, ISBN 1-58053-355-8

[15] Janne Lindqvist, Counting to Infinity, http://www.tml.tkk.fi/Studies/T-110.551/2004/papers/Lindqvist.pdf