An Enhanced DSR Protocol for Improving QoS in MANET

Khushboo Gupta
Assistant Professor
Department of CS/IT
Vidya College of Engineering,
Meerut- India

Neetu Bansla
Assistant Professor
Department of CS/IT
Vidya College of Engineering,
Meerut- India

Rajneesh
Assistant Professor
Department of ECE
Shivalik College of Engineering,
Dehradun- India

ABSTRACT

Ad hoc network is a network without centralized administration in which different users can communicate and exchange information. In such a structure, all the nodes participate in order to achieve the network and ensure the travel of the information. Hence, multihopping techniques are used to achieve this task. The communication reliability within an ad hoc network and how the different nodes act are managed by routing protocols. Nowadays, different types of protocols exist. Nevertheless, the source routing ones, based on information known at the source of the communication, seem to attract more studies. Source routing protocols had shown interesting results in realistic scenarios in areas such as military battlefields or airport stations.

This Paper deals with DSR Protocol and is focused on the multipath aspect of this routing protocol. Since, it is necessary to understand that multipath techniques enhance reliability and can ensure security. We have simulated a new multipath algorithm. The solution had been evaluated with the network Simulator 2. Since we want to know how our protocol reacts in different mobility cases, the random waypoint model which allows us to present relevant results, due to the fact this situation is taken into account.

Simulation results show that the multipath protocol behaves better than DSR, the main actual reactive protocol. The Proposed protocol MSR performs well in high mobility by using much less overhead than DSR. Additionally, it is interesting to see that DSR without any modifications manage poorly in high mobility situation.

Keywords

Ad hoc Network; DSR Protocol; MSR Protocol; NS2 Network Simulator; QOS; MANET; etc

1. INTRODUCTION

An ad hoc network is a wireless network formed by wireless nodes without any help of infrastructure. In such a network, the nodes are mobile and can communicate dynamically in an arbitrary manner. The network is characterized by the absence of central administration devices such as base stations or access points. Furthermore, nodes should be able to enter or to leave the network easily. In these networks, the nodes act as routers. They play an important role in the route discovery and route maintenance from source to the destination or from a node to another one. This is the principal challenge [1] to such a network. If link breakages occur, the network has to stay operational by building new routes. The main technique used is the multi-hopping which increase the overall network capacity and performances. By using multi-hopping, one node can deliver data on behalf of another one to a determined destination.

In addition to this, multipath routing is proposed as an alternative to single shortest path routing to distribute load and alleviate congestion the network. In multipath routing, traffic bound to a destination is split across multiple paths to that destination. In other words, multipath routing uses multiple "good" paths instead of a single "best" path for routing.

The objective of this work is to implement a multipath protocol based on an existing one. We have chosen DSR for its reliability and because it seems to impose itself as the best in the ad hoc network field. To build this new algorithm, multipath concept should be clearly assimilated. This new algorithm has been evaluated and compared with other algorithm which had shown a qualitative behavior.

2. LITERATURE REVIEW

The DSR protocol consists of two mechanisms: Route Discovery and Route Maintenance. Route discovery is initiated by a source whenever the source has a data packet to send but does not have any routing information to the destination. To establish a route, the source floods the network with request messages carrying a unique request ID. When a request message reaches the destination or a node that has route information to the destination, the node sends a route reply message containing path information back to the source. In order to reduce overhead generated the ―"route cache" at each node records routes that a node has learned and overheard during this route discovery phase.

Route Maintenance is the mechanism by which a sender S of a packet detects network topology changes that render useless its route to the destination D (e.g., when two nodes listed in the route have moved out of range of each other). When Route Maintenance indicates a source route is broken, S is notified with a ROUTE ERROR packet. The sender S can then attempt to use any other route to D already in its cache or can invoke Route Discovery again to find a new route [2].

3. MULTIPATH SOURCE ROUTING PROTOCOL

Multipath Source Routing introduced, tries to find multiple disjoint paths from a given source to a destination while guaranteeing that these paths altogether can satisfy a given end-to-end reliability (Figure. 3.1). First, source routing is so flexible that messages can be forwarded on arbitrary paths, which makes it very easy to dispatch messages to multiple paths without any demanding path calculation at the intermediate hops. Second, the on-demand nature of DSR helps to reduce the routing storage and routing computation greatly [3, 4].

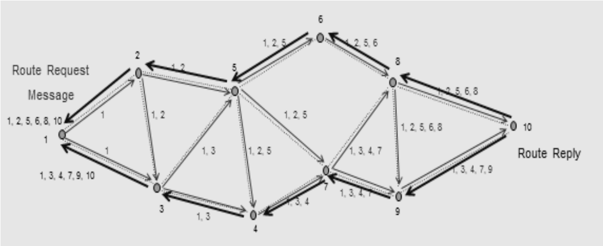

Figure 3.1: Multipath Source routing with disjoint-paths between Source and Destination

3.1. Path Finding

Each route discovered is stored in the route cache with a unique route index. So it is easy to pick multiple paths from the cache [5]. In multipath routing, path independence is an important property because a more independent path set can offer more aggregate physical resources between a node pair. (When those resources are not shared, it is less likely that the performance of one path would affect the performance of others). To achieve high path independence, disjoint paths are preferred in MSR. There is no looping problem in MSR, as the route information is contained inside the packet itself; routing loops, either short- or long-lived, cannot be formed as they can be immediately detected and eliminated.

3.2. Packet Forwarding and load balancing

Since MSR uses source routing, intermediate nodes need not do anything except to forward the packet as indicated by the route in its header, thus adding no more processing complexity than that in DSR. All the work for path calculation is done in the source hosts. In MSR, source nodes are responsible for load balancing .Within an ad hoc network, which is always an autonomous system acting as a stub network, there is less heterogeneity in some sense when compared to WAN and MAN.

For instance, in WAN or MAN, the maximal band widths that every node can obtain vary little; so do the round-trip delays. Therefore, we assume the bandwidth–delay product is a constant. Thus, the available bandwidth is inversely proportional to the RTT (Round Trip Time), so the traffic can be distributed among multiple paths proportional to the available bandwidth. The principle is inherently simple but reasonable in wireless networks. In wire line networks, due to the very different bandwidths, delay cannot be a definite indicator of the available bandwidth.

When distributing the load, the weighted round-robin scheduling strategy is used. To aid load balancing and to decouple the interlayer dependence of delay measurement, a network layer probing mechanism is employed. Probing is also an enhancement to the DSR route maintenance mechanism. Normally, in DSR, a link breakage can be noticed only when a Route Error message is returned. However, in the wireless mobile environment, there is a nontrivial chance that the Route Error message cannot reach the original sender successfully. Although, “as a last resort, a bit in the packet header could be included to allow a host transmitting a packet to request an explicit acknowledgment from the next-hop receiver”, we found that probing one path constantly only to test its validity is not cost effective. Therefore, the function of probing in our MSR is twofold: to obtain the path delay status and to test the validity of active paths [6].

4. SYSTEM DESIGN AND IMPLEMENATATION

4.1. Main Design Criteria

Simulator Chosen: We have chosen to work with NS2 [9]. NS2 is available under Linux, with a GPL license. Some standard algorithms are already implemented in this simulator, and DSR is one of these.

NS2 is a network simulator; built with C++ and TCL. As every simulator, the main purpose is to simulate different networks, to test different protocols [6], and to find the limitations of each. It has been developed in the California University, by LBL, Xerox PARC, UCB, and USC/ISI through the VINT project supported by DARPA.

The simulator is composed of two parts:

- *The TCL code:* it is used to communicate with the simulator, and permits to define different simulation parameters
- *The C++ code:* it is the main part of the project, because it defines how the simulator has to behave.

Algorithms chosen: The implementation part is an important part of the project. By implementing the different solutions, we can test them, find some improvements and understand why one works better than another.

One representative algorithm is DSR and the other is the multipath aspect of the same protocol MSR. This would give a broad picture of which type of the chosen algorithm performs well in which environment. The specific algorithms chosen within each category were DSR for traditional and MSR for multipath

4.2. Implementation

Network Scenarios: In order to do conduct the tests in a controlled way, we define a common scenario for both DSR and MSR, by varying relevant parameters such as the terrain size, rate of data sent max-speed, packet size node density and pause time.

Some tools have been developed to build these scenarios. For example, if we want to have a random model with several nodes, it is possible to use ’setdest’. It is a tool that generates random positions and random speeds for a number of nodes. By doing this, it is easy to use different random models and to test a protocol [7, 8].

Evaluation parameters: Below are the parameters used to evaluate the performance of DSR and MSR:

- A choice of 50 random moving nodes on a squared 1000 m by 1000 m area has been used.
- Mobility model for nodes as the random waypoint propagation mobility model.
- Simulation time is 250s.
- 3 sets of velocity have been used: low mobility, medium mobility and high mobility.
- Data traffic is generated by constant bit rate (CBR) sessions
- Radio propagation, we use two-ray signal propagation model

Routing Metrics:

- *Packet delivery fraction*: Ratio of data packets received by the destinations to the packets sent by the source. (Number of packet receives / number of packet sends)

- *Average end-to-end delay of data packets*: The time taken for the packet to reach the destination, it includes queuing at the interface queue, delay during route discovery (ARP) (sum of delay experienced by each packet of the flow)/number of packets).
- *Throughput:* It is the amount of data transferred successfully over a link from one end to another in a given period of time. (Number of bits transferred /Observation duration)

5. SIMULATION AND RESULTS

Simulation is carried with 50 nodes randomly distributed in a field of 1000 × 1000. A constant bit rate of 0.1Mbytes with a packet size of 1500 Bytes is used. Mobility models for pause time (0 to 100 s) are generated with .tcl scripts in ns2 [9]. The simulation time is set to 250s.

Results have been organized according to the velocity. First low, then medium and high mobility will be exposed. Three graphics for each speed has been produced to evaluate the overall performance: To evaluate MSR, a comparison has been done between reactive ad hoc protocols, using DSR To evaluate these two protocols, scenario with random mobility models and random traffic models have been used.

A set of scenario has been produced, a total of 132 scenarios have been studied. A summary of the number of scenario done is shown in the table below.

Table 5.1: Scenario summary of DSR and MSR

Speed	Traffic	DSR	MSR
1m/s	1src-1dst	11	11
1m/s	10src-10dst	11	11
10m/s	1src-1dst	11	11
10m/s	10src-10dst	11	11
20m/s	1src-1dst	11	11
20m/s	10src-10dst	11	11

5.1. Low Mobility (1M/SEC)

DSR is known to perform well in low mobility environment, in this set of scenario close to the velocity of a pedestrian (1m/s or 3.6 km/h); DSR and MSR will be analyzed.

- *1 source 1 destination continuously sending data*: These scenarios are based on 1 source and 1 destination continuously sending data (CBR over UDP) in a slow mobility environment.

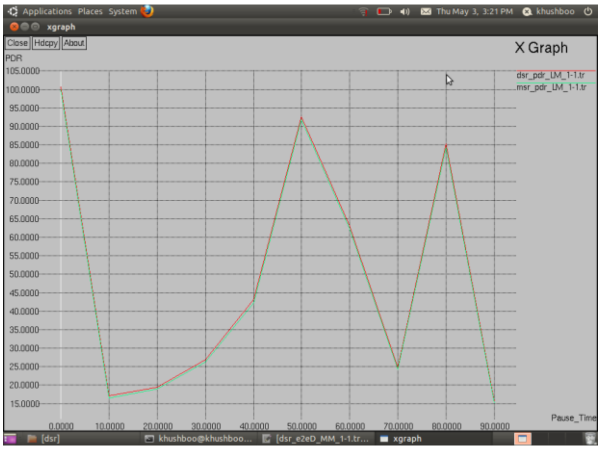

Figure 5.1: Packet delivery ratio (1m/sec 1src-1dst)

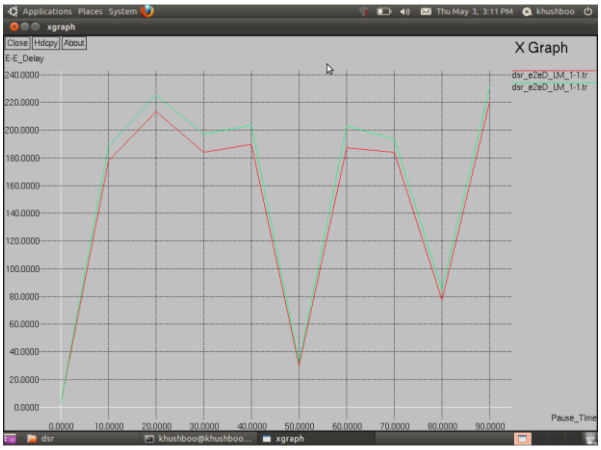

Figure 5.2: Average end to end delay (1m/sec 1src-dst)

As seen on figure 5.1, MSR in low mobility seems to gives worse results than DSR. MSR performs nearly 3 to 5% below DSR.

The average end to end delay curves (figure 5.2) are similar for DSR and MSR but MSR always introduced an additional delay between 10 ms up to a maximum of 350 ms. These results show that the mechanisms implicated to enhanced DSR toward MSR are not sufficient by themselves to increase DSR's performance since it lower the results from 3 to 5% in that specific case.

- These scenario below are based on 10 sources and 10 destinations randomly sending CBR traffic over UDP) on a low mobility environment.

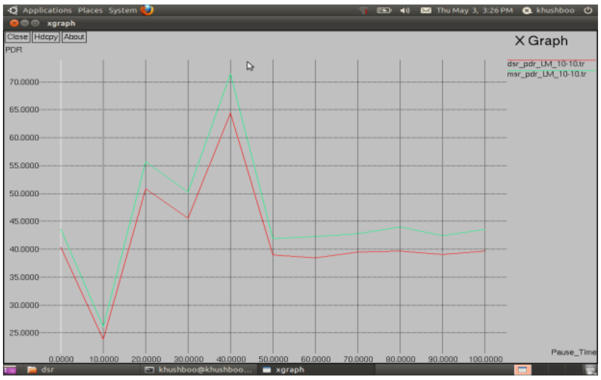

Figure 5.3: Packet Delivery fraction 1m/s -10src-10dst

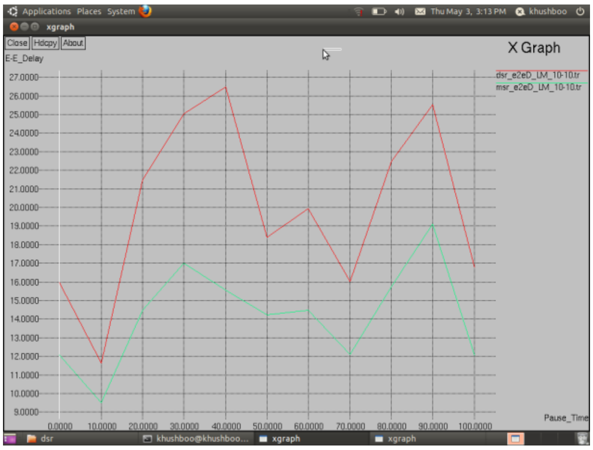

Figure 5.4: Average end to end Delay 1m/s - 10src-10dst

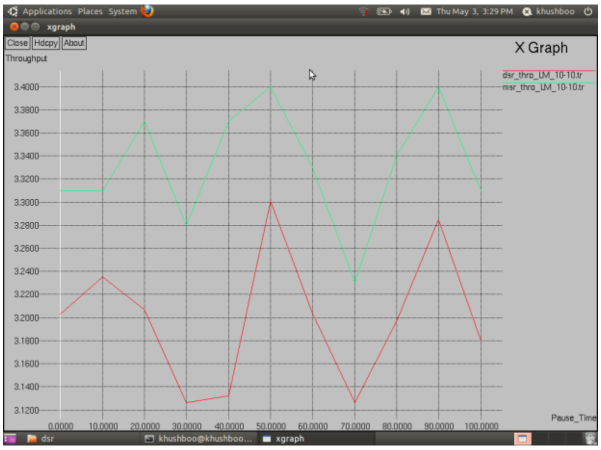

Figure 5.5: Throughput in Kbps 1m/s - 10src-10dst

In this scenario 10 sources and 10 destinations are used. Data packets are send in a random manner, which is much more realistic than using 1 source and 1 destination continuously sending data. In this scenario, the MSR’s PDR is 1 or 2% higher than DSR.

In this topology the curve is quite similar for DSR and MSR, but MSR’s end to end delay is significantly reduces at some pause time (10 s 50s 70 s and 100 s).

Finally, the throughput is always higher with MSR (5 to 10%) than with DSR.

5.2. Medium Mobility (10m/Sec)

DSR is known to perform well in low mobility environment, but in high mobility DSR is not the best protocol to use. Therefore in order to have a more accurate view of how DSR and MSR are performing on various speeds, a medium velocity is studied. 10m/s or 36 km/h is close to the speed of a fast bicycle.

- *1 source 1 destination continuously sending data*: The graphics shown below have been produced with 1 source and 1 destination continuously sending data (CBR over UDP) in a medium mobility environment.

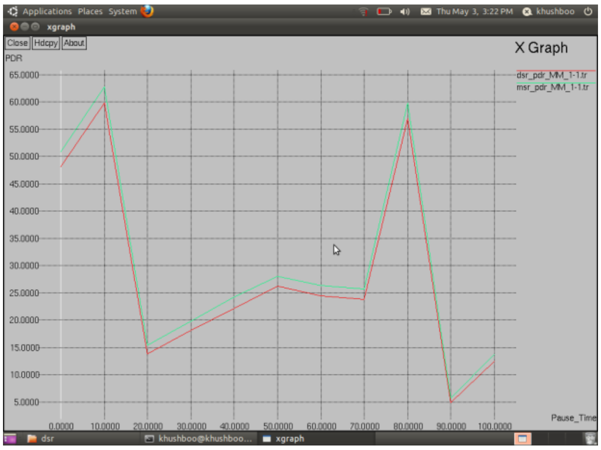

Figure 5.6: Packet Delivery fraction 10m/s - 1src-1dst

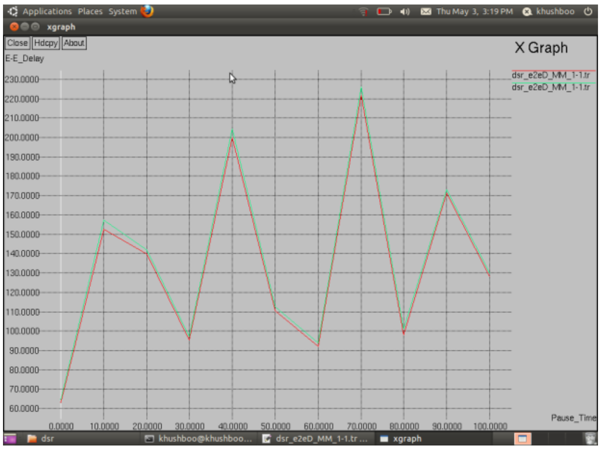

Figure 5.7: Average end to end Delay 10m/s - 1src-1dst

In medium mobility motion, MSR performs better than DSR in a range of 3 to 5.

- *10 sources 10 destinations randomly sending data:* In a medium mobility environment 10 sources and 10 destinations randomly sending data (CBR over UDP) have been studied, and graphics have been produced as shown below.

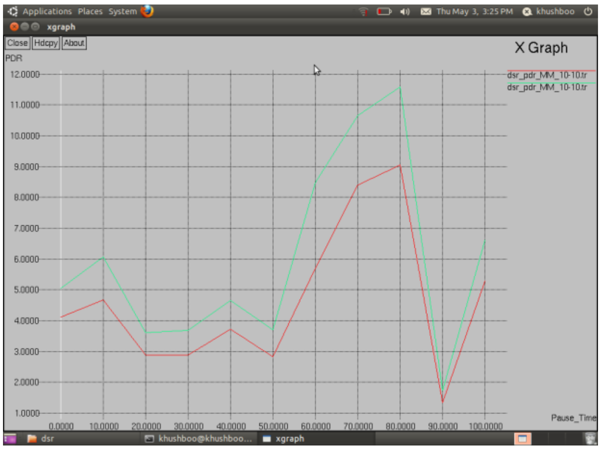

Figure 5.8: Packet Delivery fraction 10m/s - 10src-10dst

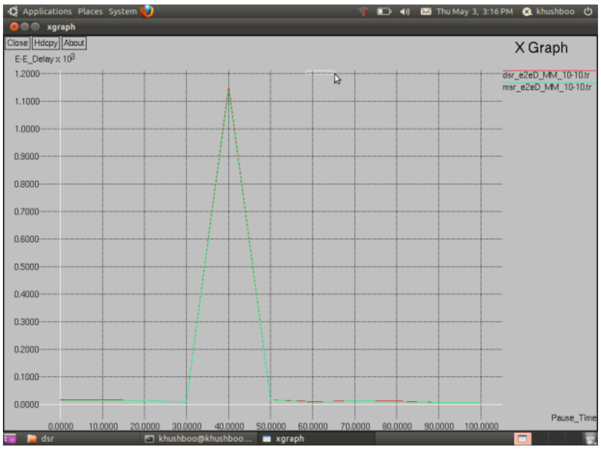

Figure 5.9: Average end to end Delay 10m/s - 10src - 10dst

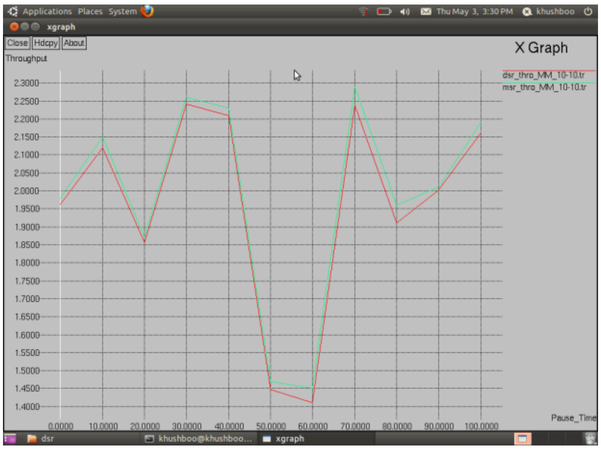

Figure 5.10: Throughput 10m/s- 10src-10dst

In medium mobility motion, MSR performs better than DSR. MSR is 20 to 25% higher than DSR for PDR (figure 5.8). The average end to end delay (figure 5.9) is always higher both with DSR and MSR. Due to the routing used with MSR, the throughput (figure 5.10) is increased with MSR.

5.3. High Mobility (20m/Sec)

DSR, which has been designed to be used in low mobility environment, has been simulated in a high mobility environment. In this section the improvement of MSR over DSR will be shown, for a high mobility velocity of 20m/s or 72 km/h, which is similar to the speed of a car.

- *1 source 1 destination continuously sending data*: In a high mobility environment, the graphics shown below have been produced with 1 source and 1 destination continuously sending data (CBR over UDP).

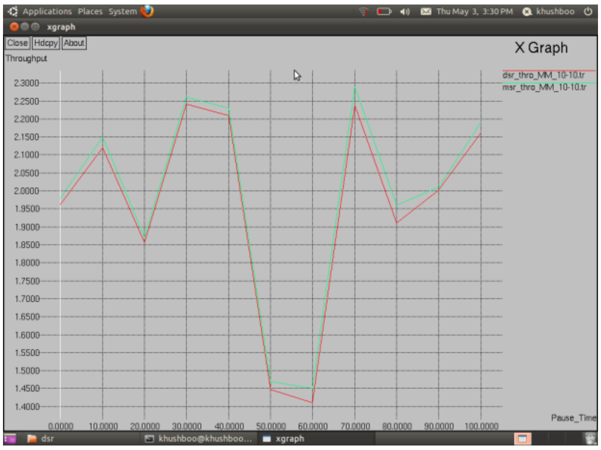

Figure 5.11: Packet Delivery fraction 20m/s - 1src-1dst

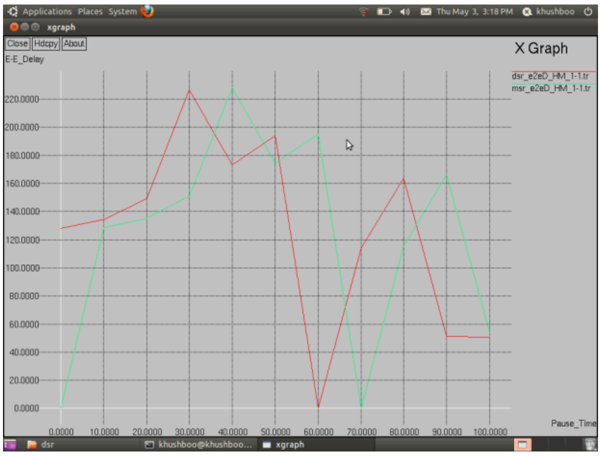

Figure 5.12: Average end to end Delay 20m/s- 1src-1dst

DSR is known to perform better in low mobility environment, with 1 source and 1 destination it is clearly shown on graphics that MSR improves the packet delivery ratio (figure 5.11). As seen on the medium mobility movement scenario, the PDR of MSR increases gradually from 3 to 5% in medium mobility to 8 to 10% in high mobility scenario. Adding multipath to DSR improves the packet delivery ratio in high mobility scenario. For Both DSR and MSR the average end to end delay (figure 5.12) is highly correlated, but MSR always introduces an additional delay up to 350 ms.

- *10 sources 10 destinations randomly sending data:* To produce the graphics shown below 10 sources and 10 destinations randomly sending data (CBR over UDP) have been used.

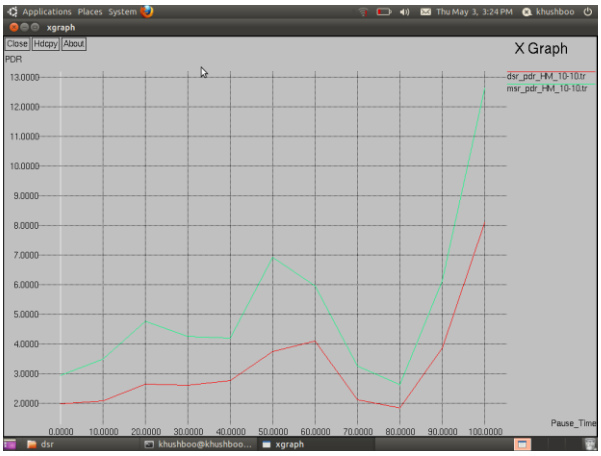

Figure 5.13: Packet Delivery fraction 20m/s - 10src-10dst

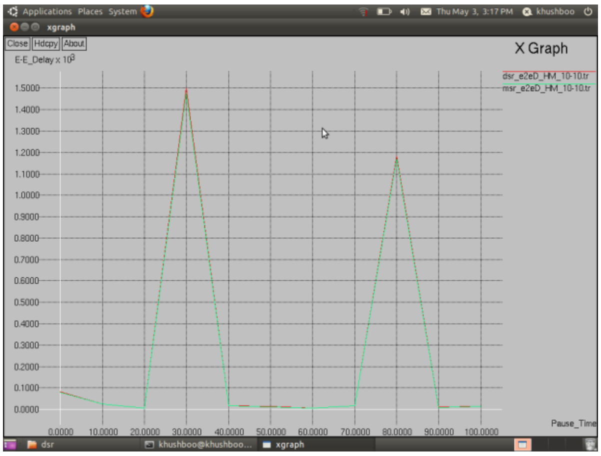

Figure 5.14: Average end to end Delay 20m/s - 10src-10dst

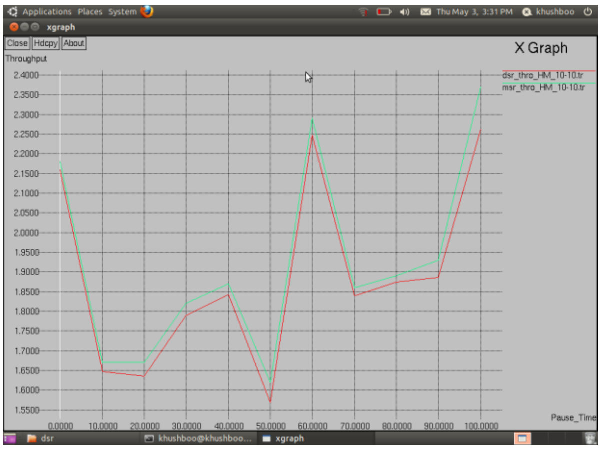

Figure 5.15: Throughput 20m/s - 10src - 10dst

In high mobility environment the performance of MSR is clearly seen. The PDR (figure 5.13) is up to a maximum 30% higher than in DSR.

The average end to end delay (figure 5.14) is almost the same as in case of DSR.

The throughput (figure 5.15) still remains higher with MSR by 20 than with DSR.

6. CONCLUSION AND FUTURE WORK

Simulations have shown that MSR protocol can be considered as reliable. The routing metric packet delivery fraction stays above 30%, even in high mobility case.

Besides, the average end to end delay is improved on the overall significantly along with multiple flows scenario and increased mobility speed and at least highly similar to DSR otherwise.

Characteristics and results for MSR were achieved after an extensive design part in which the algorithm has been tested. Indeed, it is that gradual validation of the algorithm and its implementation that permitted to reduce routing overhead. Design has been a key part to get reliable results.

This research work could be continued by, for instance, developing the multipath aspect of our protocol. It could be achieved by splitting data packets from the source to the destination; the whole message would not be transmitted by the same path or the same nodes all the time. Another solution could be to enforce reliability adding some redundancy code; in that case, it would allow not sending again packets in case one link breaks.

7. REFERENCES

[1] Khushboo Gupta and Vaishali Sikka, “Design Issues and challenges in Wireless Sensor network”, International Journal of Computer Applications (0975 – 8887) Volume 112 – No 4, February 2015

[2] D. Johnson, D. Maltz and Y-C. Hu, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks," IETF Internet-Draft, Apr. 2003.

[3] L. Almazaydeh, E. Abdelfattah, M. A. Bzoor and A. Rahayfeh, "Performance Evaluation of Routing Protocols in Wireless Sensor Networks," International Journal of Computer Science and Information Technology, Vol. 2, No. 2, Apr. 2010.

[4] R. R. M. V.C . Patil, Rajashree. V. Biradar and S. R. Sawant, "On-Demand Multipath Routing Protocols for Mobile Ad Hoc Networks Issues and Comparison," International Journal of Wireless Communication and Simulation, vol. 2, no. 1, pp. 21-38, 2010.

[5] B. Yosi, F. Sharoni and F. Moran, “Dynamic multipathallocation in Ad Hoc networks,” *Computer Journal*, vol.54, no. 2, pp. 197-212, February 2011.

[6] A. Agarwal, K. Gupta, KP. Yadav, “A novel energy efficiency protocol for WSN based on optimal chain routing” In Proceedings of the 10th INDIACom; INDIACom-2016; IEEE Conference ID: 37465

[7] B. Brian, S. Shaya, “Mobile ad hoc network broadcasting: A multi-criteria approach”, International Journal of Communication Systems, vol. 24, no. 4, pp. 438-460, April2011.

[8] Sachin Dnyandeo Ubarhande, "Performance Evolution of AODV and DSR Routing Protocols in MANET Using NS2", International Journal of Scientific & Engineering Research Vol 3, Issue 5, May.2012

[9] The network simulator-ns 2.35. http://www.isi.edu/nsnam/ns