An Analysis On Performance Evaluation Of DSR In Various Placement Environments

Rashmika N Patel
Affiliation to computer science and Engg Department L.D.Collage of Engg, Ahmedabad, Gujarat ,India
patel_rashmi1986@yahoo.co.in

***Abstract*—In this paper, the aim is to discuss Dynamic source Routing protocol in detail and evaluated its performance in three different placement environments namely Random, Grid and Uniform. This paper presents the simulation results for QOS metrics namely Average end-to-end delay, Throughput and Packet delivery ratio by varying network size. Base on analysis of simulation result the performance of DSR is better in Uniform Environment**.

***Index Terms*—DSR, MANETs, NS-2, Random, Grid and Uniform**

I. Introduction

The Dynamic Source Routing protocol (DSR) is a simple and
efficient routing protocol designed specifically for use in multi-hop wireless ad hoc networks of mobile nodes[1 2]. Using DSR, the network is completely self-organizing and self-configuring, requiring no existing network infrastructure or administration .The Random, Grid and Uniform environment have feature of supporting mobile users and resources in a seamless, transparent, secure and efficient way. It has the ability to deploy underlying ad-hoc networks. The rest of the paper is organized as follows. The operation of Dynamic Source Routing (DSR) is summarized in section 2. The environment detail is given in section 3.The simulation environment is described in section 4. I present results in section 5 and conclude with section 6.

II. DSR PROTOCOLDESCRIPTION

A. Overview and Important Properties of the Protocol

The DSR [3 4] protocol is composed of two mechanisms that work together to allow the discovery and maintenance of source routes in the ad hoc network:
(1)Route Discovery is the mechanism by which a node S wishing to send a packet to a destination node D obtains a source route to D. Route Discovery is used only when S attempts to send a packet to D and does not already know a route to D.
(2) Route Maintenance is the mechanism by which node S is able to detect, while using a source route to D, if the network topology has changed such that it can no longer use its route to D because a link
along the route no longer works. When Route Maintenance indicates a source route is broken, S can attempt to use any other route it happens to know to D, or can invoke Route Discovery again to find a
new route. Route Maintenance is used only when S is actually sending packets to D.

B. Basic DSR Route Discovery

When some node S originates a new packet destined to some other node D, it places in the header of the packet a source route giving the sequence of hops that the packet should follow on its way to D.Node A is attempting to discover a route to node E. To initiate the Route Discovery, A transmits a ROUTE REQUEST message as a single local broadcast packet, which is received by (approximately) all nodes currently within wireless transmission range of A. Each ROUTE REQUEST message identifies the initiator and target of the Route Discovery, and also contains a unique request id, determined by the initiator of the REQUEST. Each ROUTE REQUEST also contains a record listing the address of each intermediate node through which this particular copy of the ROUTE REQUEST message has been forwarded.

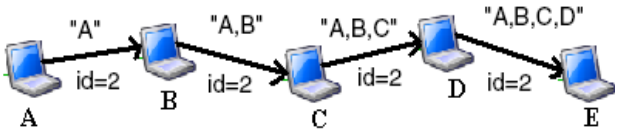

Fig.1: Node A is the initiator, and node E is the target.

C. Basic DSR Route Maintenance

When originating or forwarding a packet using a source route, each node transmitting the packet is responsible for confirming that the packet has been received by the next hop along the source route; the packet is retransmitted until this confirmation of receipt is received. If the packet is retransmitted by some hop the maximum number of times and no receipt confirmation is received, this node returns a ROUTE ERROR message to the original sender of the packet, if C is unable to deliver the packet to the next hop D, then C returns a ROUTE ERROR to A, stating that the link from C to D is currently “broken.” Node A then removes this broken link from its cache; any retransmission of the original packet is a function for upper layer protocols such as TCP. For sending such a retransmission or other packets to this same destination E, if A has in its Route Cache another route to E it can send the packet using the new route immediately.

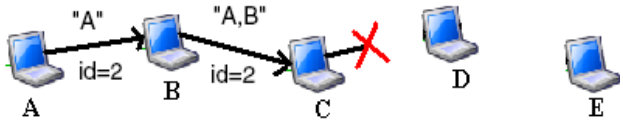

Fig.2: Node C is unable to forward a packet from A to E over its link to next hop D.

D. Additional Route Discovery Features

(1)Caching Overheard Routing Information

A node forwarding or otherwise overhearing any packet may add the routing information from that packet to its own Route Cache. In particular, the source route used in a data packet, the accumulated route record in a ROUTE REQUEST, or the route being returned in a ROUTE REPLY may all be cached by any node. Routing information from any of these packets received may be cached. node A is using a source route to communicate with node E. As node C forwards a data packet along the route from A to E, it can always add to its cache the presence of the "forward" direction links that it learns from the headers of these packets, from itself to D and from D to E.

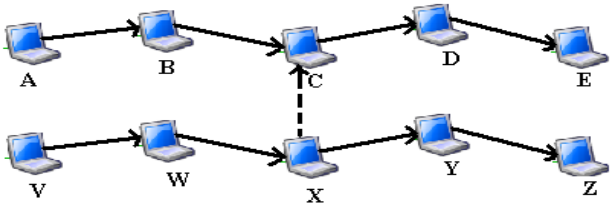

Fig.3: Node C is forwarding packets to E and overhears packets from X.

However, the "reverse" direction of the links identified in the packet headers, from itself back to B and from B to A, may not work for it since these links might be uni-directional.Same as node V in Figure 3 is using a different source route to communicate with node Z. If node C overhears node X transmitting a data packet to forward it to Y (from V), node C should consider whether the links involved can be known to be bi-directional or not before caching them. If the link from X to C (over which this data packet was received) can be known to be bi-directional, then C could cache the link from itself to X, the link from X to Y, and the link from Y to Z. If all links can be assumed to be bi-directional, C could also cache the links from X to W and from W to V.

(2) Replying to ROUTE REQUEST using Cached Routes

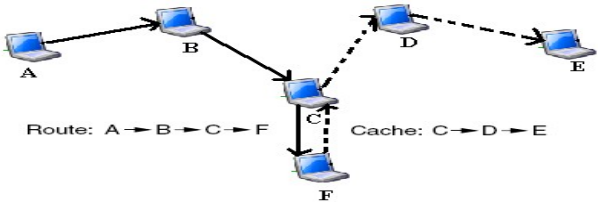

Fig.4: A possible duplication of route hops avoided by the Route Discovery limitation on replying to ROUTE REQUESTs from the Route Cache.

A node receiving a ROUTE REQUEST for which it is not the target, searches its own Route Cache for a route to the target of the REQUEST. If found, the node generally returns a ROUTE REPLY to the initiator itself rather than forwarding the ROUTE REQUEST. In the ROUTE REPLY, it sets the route record to list the sequence of hops over which this copy of the ROUTE REQUEST was forwarded to it, concatenated with its own idea of the route from itself to the target from its Route Cache. Node F in this case could attempt to edit the route to eliminate the duplication, resulting in a route from A to B to C to D and on to E, but in this case, node F would not be on the route that it returned in its own ROUTE REPLY. DSR Route Discovery prohibits node F from returning such a ROUTE REPLY from its cache for two reasons. First, this limitation increases the probability that the resulting route is valid, since F in this case should have received a ROUTE ERROR if the route had previously stopped working. Second, this limitation means that a ROUTE ERROR traversing the route is very likely to pass through any node that sent the ROUTE REPLY for the route (including F), which helps to ensure that stale data is removed from caches (such as at F) in a timely manner.

(3) ROUTE REQUEST Hop Limits

Each ROUTE REQUEST message contains a "hop limit" that may be used to limit the number of intermediate nodes allowed to forward that copy of the ROUTE REQUEST. As the REQUEST is forwarded, this limit is decremented, and the REQUEST packet is discarded if the limit reaches zero before finding the target. We currently use this mechanism to send a no propagating ROUTE REQUEST (i.e., with hop limit 0) as an inexpensive method of determining if the target is currently a neighbor of the initiator or if a neighbor node has a route to the target cached (effectively using the neighbors' caches as an extension of the initiator's own cache). If no ROUTE REPLY is received after a short timeout, then a propagating ROUTE REQUEST (i.e. with no hop limit) is sent.

E. Additional Route Maintenance Features

(1) Packet Salvaging

After sending a ROUTE ERROR message as part of Route Maintenance, a node may attempt to salvage the data packet that caused the ROUTE ERROR rather than discarding it. To attempt to salvage a packet, the node sending a ROUTE ERROR searches its own Route Cache for a route from itself to the destination of the packet causing the ERROR. If such a route is found, the node may salvage the packet after returning the ROUTE ERROR by replacing the original source route on the packet with the route from its Route Cache. The node then forwards the packet to the next node indicated along this source route. For example, in Figure 2, if node C has another route cached to node E, it can salvage the packet by applying this route to the packet rather than discarding the packet.

(2) Automatic Route Shortening

Source routes in use may be automatically shortened if one or more intermediate hops in the route become no longer necessary. This mechanism of automatically shortening routes in use is somewhat similar to the use of passive acknowledgements. In particular, if a node is able to overhear a packet carrying a source route (e.g., by operating its network interface in promiscuous receive mode), then this node examines the unused portion of that source route. If this node is not the intended next hop for the packet but is named in the

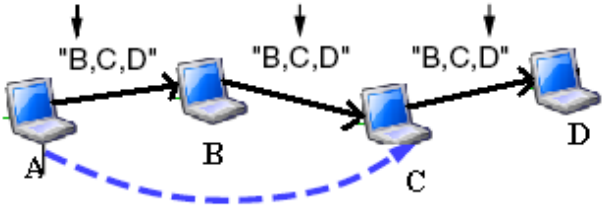

Fig.5: Node C notices that the source route to D can be shortened, since it overheard a packet from A intended first for B.

later unused portion of the packet's source route, then it can infer that the intermediate nodes before itself in the source route are no longer needed in the route. For example, Figure 6 illustrates an example in which node C has overheard a data packet being transmitted from A to B, for later forwarding to C; the arrow pointing to one node in the source route in each packet indicates the intended next receiver of the packet along the route.

(3) Increased Spreading of ROUTE ERROR Messages

When a source node receives a ROUTE ERROR for a data packet that it originated, this source node propagates this ROUTE ERROR to its neighbors by piggybacking it on its next ROUTE REQUEST. In this way, stale information in the caches of nodes around this source node will not generate ROUTE REPLYs that contain the same invalid link for which this source node received the ROUTE ERROR. For example, in the situation shown in Figure 2, node A learns from the ROUTE ERROR message from C, that the link from C to D is currently broken. It thus removes this link from its own Route Cache and initiates a new Route Discovery (if it doesn't have another route to E in its Route Cache). On the ROUTE REQUEST packet initiating this Route Discovery, node A piggybacks a copy of this ROUTE ERROR message, ensuring that the ROUTE ERROR message spreads well to other nodes, and guaranteeing that any ROUTE REPLY that it receives (including those from other node's Route Caches) in response to this ROUTE REQUEST does not contain a route that assumes the existence of this broken link.

III. DETAIL OF VARIOUS ENVIRONMENT

The Dynamic Source Routing protocol (DSR) and evaluated its performance in three different placement environments namely Random, Grid and Uniform.

The Grid can be viewed as a distributed, high performance computing and data handling infrastructure, that incorporates geographically and organizationally dispersed, heterogeneous resources and provides common interfaces for all these resources, using standard, open, general purpose protocols and interfaces. Mobile Grid computing is a form of wireless distributed computing whereby a "super and virtual computer" is composed of acluster of networked, loosely-coupled computers, acting in concert to perform very large tasks. This technology has been applied to computationally-intensive scientific and to solve mathematical problems.

In the random waypoint mobility model, each mobile node begins at a random location and moves independently during the simulation. Each node remains stationary for a specified period that we call the pause time and then moves in a straight line to some new randomly chosen location at a randomly chosen speed up to some maximum speed. Once reaching that new location, the node again remains stationary for the pause time, and then chooses a new random location to proceed to at some new randomly chosen speed, and the node continues to repeat this behavior throughout the simulation run. We have found that this model can produce large amounts of relative node movement and network topology change, and thus provides a good movement model with which to stress DSR or other ad hoc network routing protocols.

IV. SIMULATION ENVIRONMENT

NS-2 is chosen as the simulation tool among the others simulation tools because NS-2 supports networking research and education. Ns-2 is suitable for designing new protocols, comparing different protocols and traffic evaluations. NS-2 is developed as a collaborative environment. It is distributed freely and open source.

A large amount of institutes and people in development and research use, maintain and develop NS-2. This increase the confidence in it. Versions are available for FreeBDS, Linux, So laris, Windows, Mac OS X. NS-2 also provides substantial support for simulation of TCP, UDP, routing and multicast protocols over wired and wireless networks. We run the simulation in Network Simulator (NS-2) accepts as input a scenario file that describes the exact motion of each node and the exact packets originated by each node, together with the exact time at which each change in motion or packet origination is to occur. The detailed trace file created by each run is stored to disk, and analyzed using a variety of scripts.

Here simulation environment include The field configurations 1000 m x 1000 m fieldArea,simulation time is 150sec,nodes are 5,10,20,40 and node placement is Random,Grid,Uniform and pause time is 30 sec and maximum speed is12mps and traffic is CBR and packet

size is 512 bytes. Here continuous bit rate (CBR) traffic sources are used. The source-destination pairs are spread randomly over the network. The number of source-destination pairs and the packet sending rate in each pair is varied to change the offered load in the network.

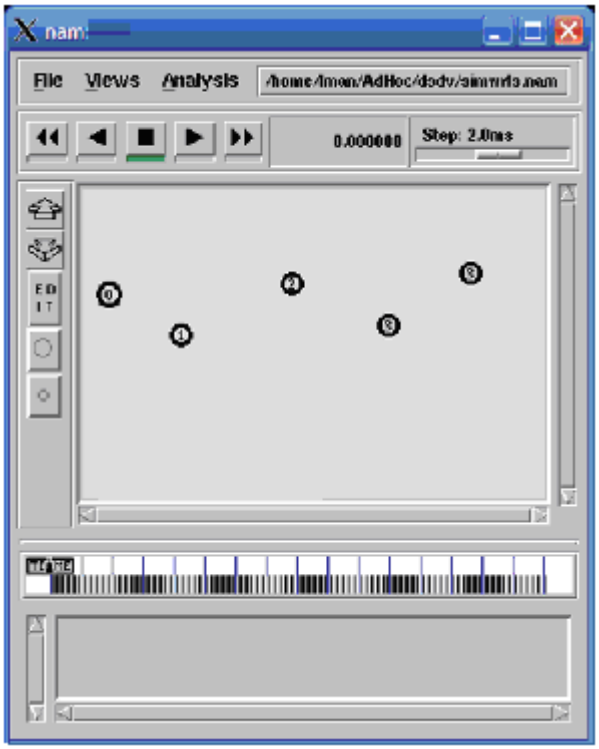

Fig. 6: The topology of network at time 0

V. RESULT AND DESCUSSION

A. Packet Delivery Ratio

PDF(Packet Delivery Fraction) is the ratio between the number of packets originated by the application layer sources and the number of packets received by the sinks at

$$PktDelivery\% = \frac{\sum_{i}^{n} CBRrecv}{\sum_{i}^{n} CBRrecv} \times 100. \quad (1)$$

the final destination. It will describe the loss rate that will be seen by the transport protocols, which in turn affects the maximum throughput that the network can support. The variation of Packet Delivery ratio with varying the number of mobile nodes is shown in the Figure.

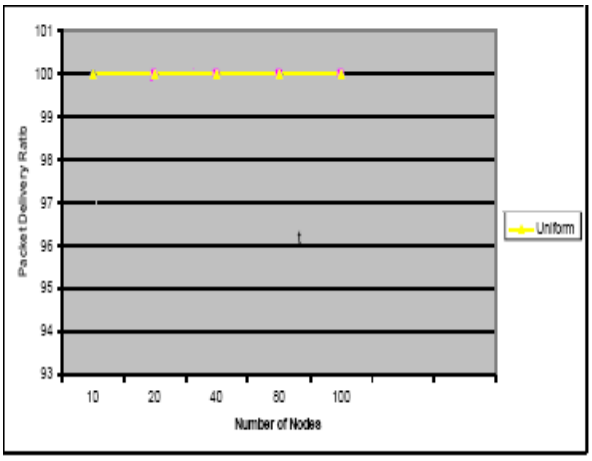

Fig.7: Packet Delivery Ratio with Number of nodes in uniform environment

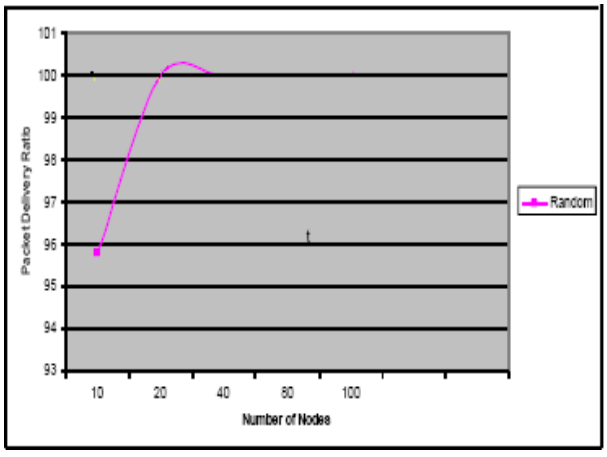

Fig.8: Packet Delivery Ratio with Number of nodes in random environment

B. Average End-to-end delay

End-to-end delay indicates how long it took for a packet to travel from the source to the application layer of the destination. The delay is affected by high rate of CBR packets as well. The buffers become full much

$$Avg\_End-to-End\_Delay = \frac{\sum_{i}^{n} (CBRsentTime - CBRrecvTime)}{\sum_{i}^{n} CBRrec} \quad (2)$$

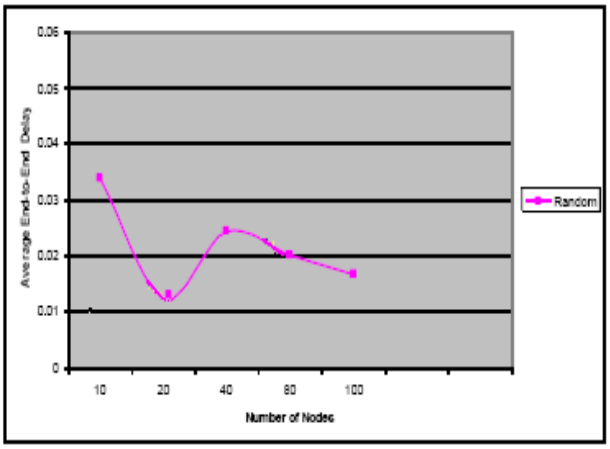

Fig.9: Average End-to-End Delay with Number of nodes in random environment

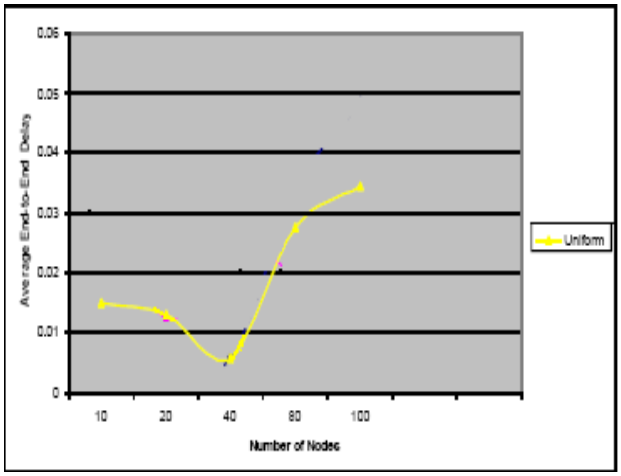

Fig.10: Average End-to-End Delay with Number of nodes in uniform environment

quicker, so the packets have to stay in the buffers a much longer period of time before they are sent.

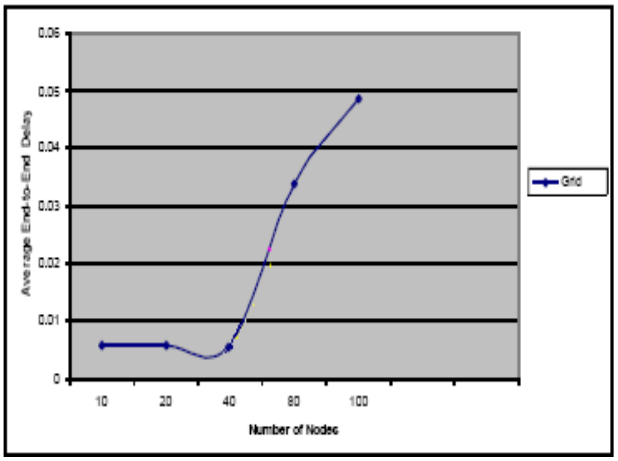

Fig.11: Average End-to-End Delay with Number of nodes in grid environment

C. Throughput

The total amount of data a receiver R actually receives from the sender divided by the time it takes for R to

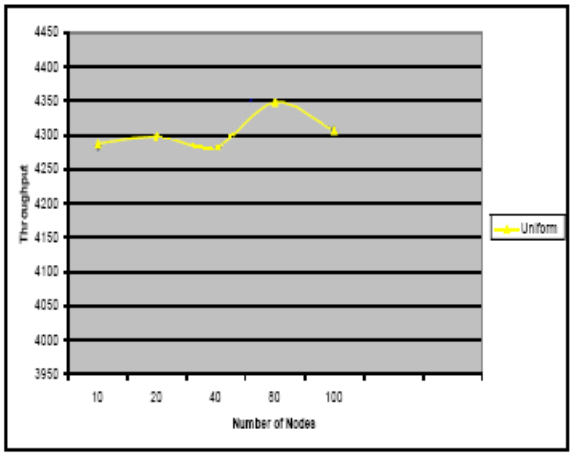

Fig.12: Throughput with Number of Nodes in uniform environment

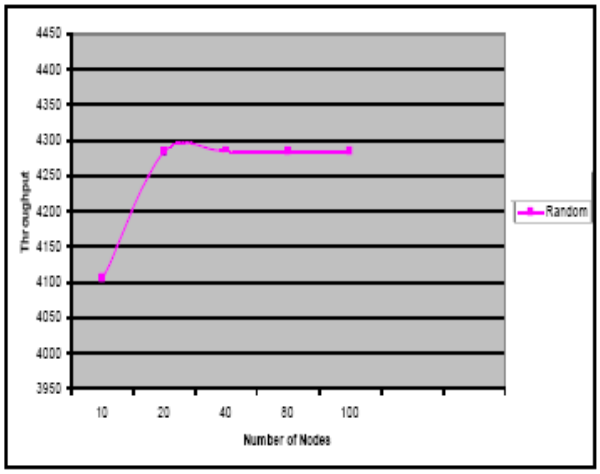

Fig.13: Throughput with Number of Nodes in random environment

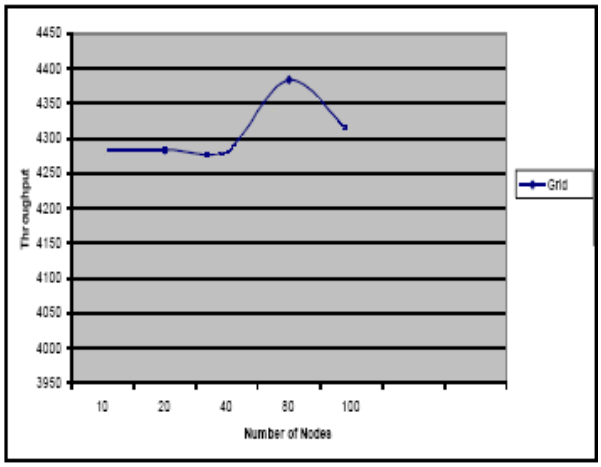

Fig.14: Throughput with Number of Nodes in random grid environment

Conclusions

The simulation results shows that DSR achieves better performance in Uniform Environment. The performance of DSR is studied by placing the nodes in various arrangements one of our future research studies is the study of the behavior of DSR in various environments with various mobility models.

References

[1] Royer, E.M., 1999. A review of current routing protocols for ad hoc mobile wireless networks.IEEE Personal Communications, pp: 46-554.

[2] Vaduvur Bharghavan, Alan Demers, Scott Shenker, and Lixia Zhang. MACAW: A Media Access Protocol forWireless LAN's. In Proceedings of the ACMSIGCOMM'94 Conference, pages 212–225. ACM, August 1994.

[3] D .Johnson and D. Maltz . Dynamic Source Routing in ad-hoc Wireless Networks in Computer Communication Review - Proceedings of SIGCOMM 96 Aug-1996

[4] D.B. Johnson and D.A. Maltz, "Dynamic Source Routing in Ad hoc Wireless Networks", Mobile Computing, Kluwer Academic Publishers, pp. 153-181, 1996.