2011 International Conference on Information and Network Technology
IPCSIT vol.4 (2011) © (2011) IACSIT Press, Singapore

Study of DSR Routing Protocol in Mobile Adhoc Network

Sangeeta Biswal[1], Suneeta Mohanty[2] and Dambarudhar Seth[3]

[1]School of Computer Engineering, KIIT University, Bhubaneswar, India
tejomayeenath@gmail.com
[2]School of Computer Engineering, KIIT University, Bhubaneswar, India
suneetamohanty@gmail.com
[3]School of Electronics Engineering, KIIT University, Bhubaneswar, India
d_seth2002@yahoo.co.in

Abstract. An adhoc network is a collection of wireless mobile nodes dynamically forming a temporary network without the use of any preexisting network infrastructure or centralized administration. By presenting a detailed study of DSR with its characteristics and performances, this study generalizes the lessons learned from DSR so that they can be applied to the many other new routing protocols that have adopted the basic DSR framework. The improvement is done in route reply (RREP) method. During route reply when more than one route replies are about to reach to source there are high chances that they create congestion at the last point. This congestion is a cause for possible collisions. We tried to reduce this possibility of collisions. We propose to include delay in RREP when RREP is one hop away from the source. We did simulation for the proposed scheme in NS-2. Simulation results revealed that the proposed scheme performs better than original DSR for collision.

Keywords:- MANETs, DSR, Routing Protocol;

1. Introduction

An adhoc network is a collection of mobile nodes that are capable of communicating with each other without the aid of any established infrastructure or centralized administration. The mobile nodes in an adhoc network moves randomly resulting in a dynamic topology. They are self organized, dynamically changing multi-hop networks**.**

Routing protocols design for MANETs is a very active research area and many proactive and reactive protocols have been proposed [3]. Proactive protocols find routes between all source-destination pairs regardless of the actual need for such routes. The more traditional proactive protocol can reduce the needed time to get a route by inducing a high routing load over the network.

In Reactive routing, when a source node needs to send data packets to some destination then it checks for route availability. If no route exists, it performs a route discovery procedure to find a path to the destination. Hence, route discovery becomes on demand. Therefore the Reactive routing techniques, also called on-demand routing. The route discovery typically consists of the network-wide flooding of a request message. Once a route has been established, it is maintained by some form of route maintenance procedure until either the destination becomes inaccessible or until the route is no longer desired. Dynamic Source Routing (DSR) is a type of Reactive routing protocol.

2. Dynamic Source Routing (DSR)

Sangeeta Biswal. Tel.: +91- 9439164954; fax: + 91671-2603469
E-mail address: sangeetabiswal@gmail.com

The Dynamic Source Routing (DSR) protocol is a simple and robust routing protocol designed specifically for use in multi-hop wireless ad-hoc networks of mobile nodes. The Dynamic Source Routing protocol (DSR) is based on source routing, which means that the originator of each packet determines an ordered list of nodes through which the packet must pass while traveling to the destination. The DSR protocol consists of two basic mechanisms: Route Discovery and Route Maintenance.

2.1 Route Discovery

Route discovery is used only when a source node attempts to send a packet to a destination node and does not already know a route to it. To initiate the Route Discovery, the source node transmits a "Route Request" with a unique ID as a single local broadcast packet. When some intermediate node receives this Route Request, at first it determines whether it has seen the Route Request or not. If the node has already seen the Route Request earlier, it will discard the packet; otherwise it will check its Route Cache whether there is a route to the destination of the packet. If it has the route to target in its routing cache, it returns a "Route Reply" to the initiator of the Route Discovery, giving a copy of the accumulated route record from the Route Request; otherwise it transmits the Route Request until the Route Request is received by the target .

Route Maintenance

DSR protocol implements the route maintenance mechanism while communicating the packets from source to destination. But when the communication link between the source and the destination is broken or else a

change in network topology is noticed. It will lead to failure of the communication between source node and destination node. In this scenario DSR protocols uses the route mechanism, to detect any other possible known route towards the destination to transmit data. If the route maintenance fails to find an alternative known route to establish the communication then it will invoke the route discovery to find the new route to destination.

2.2 Advantages of DSR

- Routes maintained only between nodes who need to communicate
  - reduces overhead of route maintenance
- Route caching can further reduce route discovery overhead
- A single route discovery may yield many routes to the destination, due to intermediate nodes replying from local caches

Disadvantages of DSR

- Packet header size grows with route length due to source routing
- Flood of route requests may potentially reach all nodes in the network
- Potential collisions between route requests propagated by neighboring nodes
  - insertion of random delays before forwarding RREQ
  - Increased contention if too many route replies come back due to nodes replying using their local cache
  - Route Reply Storm problem
- Stale caches will lead to increased overhead

2.3 Optimizations

Several optimizations to this basic protocol have been proposed and have been evaluated to be very effective by the authors of the protocol [1]. Some of them are:

a) Data Salvaging – If an intermediate node encounters a broken link and has an alternate route to the destination in its cache, it can try to salvage the packet by sending it via the route from its cache.
b) Gratuitous Replies – When a node overhears a packet not addressed to itself, it checks to see if the packet could be routed via itself to gain a shorter route. If so, the node sends a gratuitous reply to the source of the route with this new, better route.
c) Route Snooping – A node that overhears a data packet and does not have the route indicated in the packet's header in its own cache, adds the route to its cache for future use.

Drawback of DSR

Certain features of DSR have previously been shown that degrades its performance [5, 7].

(i) Intermediate-Node Replies – When a route request packet is heard by a node that is not the destination itself, but has a route to the destination in its cache, a route reply packet is sent back to inform the source of the route. Intermediate-Node replies make the route learning process faster because all route requests do not need to travel all the way to the destination if one of the intermediate nodes already has the desired route. They also reduce the number of RREQs transmitted.

However, the drawback with Intermediate-Node replies is that the route in the intermediate node could be arbitrarily old when the route request is heard. Hence, the source has no guarantee of the validity of a route even immediately after the route reply is received. In a highly mobile network, intermediate-node replies are very likely to be invalid. When a source receives a bad route reply, it tries to send the waiting data packet along the route. Upon failure of one of the links along the route, a route error packet is propagated back to the source. The source then issues a new route request, starting the process all over again. So, an invalid route reply from an intermediate node can be very costly in terms of time and bandwidth.

(ii) Data Salvage - When a node in the path of a route experiences a link failure, it checks its own route cache table to see if it can salvage the data packet by sending it via an alternate route from its route cache. Data Salvage can be useful in relatively static networks because the time spent in learning a new route can be saved when it is possible to salvage a packet using a route from an intermediate node.
However, in a network in which the nodes are highly mobile, it is likely that the route in the intermediate node's cache was older, and hence, also invalid. Trying to salvage a data packet by using another bad route would result in a waste of time and bandwidth.

To turn off intermediate-node replies*,* DSR can be modified such that an intermediate node that hears a route request broadcasts it further, without replying to it even if a route to the destination exists in its cache. Only the destination nodes to route requests are allowed to reply. The idea behind this is that it might be a good idea for a route request to travel a few hops further to the destination, and get a reply that is assured to be both fresh and valid.

To turn off Data Salvage, DSR can be modified such that an intermediate node that experiences a link failure does not try to salvage the data. Instead, it sends a route error back to the source, and drops the data packet.

Methods to improve performance

To improve the performance of the DSR, following techniques can be used.

(i)A limit can be put on the Replies from Destination

In the original implementation of DSR, a destination node replies to every route request packet that it hears. This, however, results in a lot of unnecessary route replies when the same route request is heard by a destination multiple times. This can also result in 'bad' routes being added to the route cache of the source. For instance, consider 2 route requests that take the same number of hops, but different paths to reach the destination at different times. The request that reaches the destination late possibly took a path that was more congested. Instead of being discarded, this request is also replied to, and because it had the same hop count as the previous request, it is added to the top of the route cache of the source. Hence, when a data packet is to be sent, a congested route is tried before the route that was not congested.

Hence DSR can be modified such that destination nodes will reply only if

a) The last route request from the given source was older than the current one, or
b) The last route request was made at the same time (the same route request took different routes to the destination) but the current request took fewer hops than the last one.

The destination now sends a route reply only if it is a new route request or a better route for a route request to which it has already replied.

(ii) Preference can be given to 'fresher' routes in cache

When a node learns multiple routes to a destination, DSR stores them in its cache table ordered by hop count. This, however, often results in the use of older, invalid routes over routes that are fresh, and valid. While hop count may be a good metric to determine the order of routes in the cache, it should probably not be the only criteria.

The cache structure of DSR was modified such that it would maintain routes to a particular destination in the following order:

a) A route with a later request time is given preference over a route with an earlier request time.
b) If the request times of two routes are the same, then a route with shorter hop count is given preference over a longer route.
c) If both the request time and the hop count of two routes are the same, then a route with a later reply time is given preference over a route with an earlier reply time.

Simulation

In this part we will discuss about network simulator (ns-2), simulation topology and environment, metrics of evaluation of the simulation result and we will compare results of proposed scheme with original DSR.

Implementation Summary

The proposed schemes were implemented making use of a simulator, Network Simulator (ns-2.31).

Network Simulator

The NS-2 (Network Simulator) [3] is a discrete event driven simulator used for implementation and simulations of various network protocols. It is freely distributed, open source and is widely used for academic and research from the past 10 years. It is a de facto standard in networking research. It is currently maintained by ISI (Information Science Institute). NS-2 is used for modeling network component like

- Traffic models and applications: Web, FTP, telnet, audio, sensor nets.
- Transport protocols: TCP and UDP
- Routing and Queuing: IP, Mobile IP, Static routing, Distance Vector routing, multi- cast, adhoc routing, Drop tail etc
- Link layer: Point to point connection, Local Area Networks, wireless links, and satellite links.

Simulation Result

The results are computed by tracing the output files generated by ns-2.31 simulator during simulation making use of awk script. Results are plotted by varying pause times.

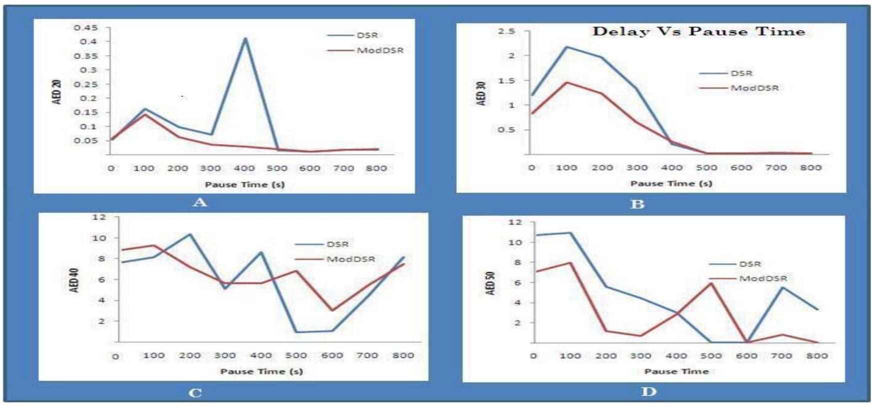

Fig. 1: "Average End to End Delay with 70 nodes 20,30,40,50 connection respectively"

Figure 1 shows Average end-to-end delay comparison of our extended DSR with traditional DSR. DSR and extended DSR have almost identical delay from 10 and 20 connection at low and high pause time, with 40 connections DSR outperform extended DSR but in 20,30,50 connection extended DSR outperform DSR at low and high pause time.

Conclusion

When we compare the traditional DSR with our Extended DSR then we find that in my extended DSR, there are less no of collision in comparison to traditional DSR and collision is important factor in DSR.

3. Acknowledgements

I have great pleasure in expressing my heartiest gratitude to my project guide Prof. Mrs Suneeta Mohanty for her able guidance, valuable suggestions, constructive criticism and for providing a clear direction to the project from time to time. Thanks are also due to Prof. Dambarudhar Seth for his continuous encouragement and guidance has been a constant source of inspiration throughout this study.

4. References

[1] S. Corson, and J. Macker. Mobile ad hoc networking (manet): Routing protocol performance issues and evaluation considerations, 1999. DSR *Rfc by D. Johnson Rice University Y. Hu UIUC D. Maltz Microsoft Research February* 2007.

[2] C. Perkins, and P. Bhagwat. Highly Dynamic Destination Sequenced Distance-vector Routing (dsdv) for Mobile Computers*, Comp. Commun. Rev.,* Oct. 1994, pp. 234–44.

[3] K. Fall, and K. Varadhan. The network simulator ns-2: Documentation. http://www.isi.edu/nsnam/ns/ns-documentation.html.

[4] B. Bellur, R.Ogier, and F. Templin. Topology Broadcast Based on Reverse-Path Forwarding (tbrpf), Internet draft, draft-ietf-manet-tbrpf- 01.txt, work in progress, Mar. 2001.

[5] M. K. Marina, and S. R. Das. Performance of Route Caching Strategies in Dynamic Source Routing. In Proceedings of Int'l Workshop on Wireless Networks and Mobile Computing (WNMC), 2001.

[6] V. Park, and M. Corson. A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks.*Proc.INFOCOM*'97.

[7] T. Goff et al. Preemptive Routing in Ad Hoc Networks. *In Proceedings of the 7thAnnual International Conference on Mobile Computing and Networking (Mobicom),* 2001.