International Journal of Recent Trends in Engineering, Vol 2, No. 2, November 2009

Performance Improvement Techniques for Dynamic Source Routing Protocol in Mobile Ad Hoc Networks

Prof. M.Neelakantappa[1] Dr.B.Satyanarayana[2] Dr. A.Damodharam[3]
[1] Professor, BITS , Kunool,AP,India
Email:m_neelakanta@yahoo.com
[2] Professor & Head, SKUniversity, Anantapur,AP,India
Email: bachalasatya@yahoo.com
[3] Director,UGCASC,JNTUniversity,Hyderabad,AP,India
Email: damodarama@gmail.com

***Abstract*— A Mobile Ad hoc NETwork (MANET) is a temporary wireless network environment where in nodes that are in mobility establishes the network with out aid of any fixed infrastructure. Routing in the MANET is a major challenging problem to solve, because of its dynamic topology and infrastructure less nature, namely Dynamic Source Routing (DSR) is one of the widely used routing protocols for MANETS protocol. It was proven that, several of the optimizations proposed on the DSR protocol, tend to hurt the performance especially in the case of high node mobility and low traffic load. In this paper the performance issues has been studied extensively. Taking DSR with certain optimizations turned off as a base, three intuitive techniques are proposed to improve the performance of DSR. Using the simulations, it was shown that the proposed techniques provide significant performance improvements for various network densities and traffic load.**

***Index Terms*—Mobile Ad hoc Network, routing protocols, Packet Delivery rate, GloMoSIM.**

I. INTRODUCTION

Recent advances in technology have provided portable computers with wireless interfaces that allow networked communication among mobile users .The resulting computing environment, which is often referred to as mobile computing, no longer requires users to maintain a fixed and universally known position in the network And enables almost unrestricted mobility .A Mobile Ad hoc NETwork (MANET)is a special type of wireless mobile network[1,4] in which a collection of mobile hosts with wireless network interface may form a temporary network, without aid of any established infrastructure or centralized administration. The application ranges from civilian to disaster recovery and military.

Routing in the MANET faces special challenges because of its infrastructure less network and its dynamic topology. The tunnel-based triangle routing of mobile IP works well only for fixed infrastructure network to support the concept of “home agent”. But when all hosts move, such a strategy cannot be directly applied. Traditional routing protocols for wired networks like distance vector or link state are no longer suitable for ad hoc wireless networks. In an environment with mobile hosts as routers, changes in network topology may be slow and this process could be expensive due to low bandwidth.

Routing protocols for MANETS[13] can be roughly divided into *proactive* and *reactive*. In proactive routing, each host continuously maintains complete routing information of the network. Both link state and distance vector belong to proactive routing. The reactive scheme, invokes a route determination procedure only on demand through a query/reply approach. Dynamic source routing protocol (DSR)[1] is a reactive routing protocol. The source determines the complete path for each routing process. The approach consists of two steps, route discovery and route maintenance. Route discovery allows any host to dynamically discover a route to a destination host. Each host also maintains a route cache in which it catches source routes it has learned. Unlike regular routing-table based approaches that have to perform periodic routing updates, route maintenance only monitors the routing process and informs the sender of any routing errors.

The Dynamic Source Routing (DSR) [1,4] is one of the widely used routing protocols for MANETs. Several previous studies indicate that some of the route gathering techniques and optimizations proposed in the original protocol actually hurt the performance in many situations and make DSR under perform another commonly used routing protocol—ad hoc on demand distance vector (AODV) [2]. Because of source routing, however, DSR is considered to be desirable from security aspect [6]. Several previous studies indicate the benefit of turning off some of the "optimization" features of DSR to improve its performance [7]. In this paper, we show that with these modifications, DSR’s performance is significantly improved especially at high traffic loads. Using simulations through GloMoSIM, we show that these features improve DSR's performance.

II. BASIC DSR PROTOCOL

A. *Overview of DSR*

Route Discovery and Route Maintenance of DSR are all operate on-demand. In particular, unlike other protocols, DSR requires no periodic packets of any kind at any level within the network. This entirely on-demand behavior and lack of periodic activity allows the number of overhead packets caused by DSR to scale all the way down to zero, when all nodes are approximately stationary with respect to each other and all routes needed for current communication have already been discovered. As nodes begin to move more or communication patterns change, the routing packet overhead of DSR automatically scales to only that needed to track the routes currently in use.

We can use the following formula[5] to denote *MANET* G:

$$G= (N, V) \qquad (1)$$

N denotes the set all nodes of G, V denote the set all links of G. Among the elements of set N, when 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 **.** Normally, s will obtain a suitable source route by searching its Route Cache of routes previously learned, but if no route is found in its cache, it will initiate the Route Discovery protocol to dynamically find a new route to d**.** We call ***s*** the initiator and d the target. For example, Figure 1 shows an example of Route Discovery, in which a node a is attempting to discover a route to node e**.** To initiate the Route Discovery, a transmits a ROUTE REQUEST [5] message as a single local broadcast packet, which is got by all nodes currently within wireless transmission range of ***a.***

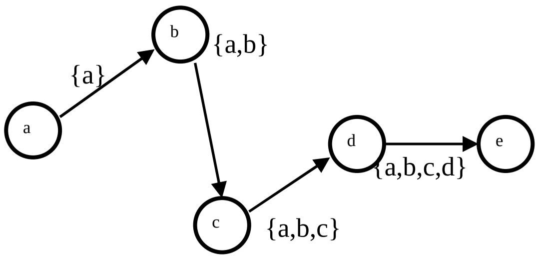

Fig 1: Example of route Discovery with same request ID

Each ROUTE REQUEST contains a record listing the address of each intermediate node through which this particular copy of the ROUTE REQUEST message has been forwarded. This route record is initialized to an empty list by the initiator of the Route Discovery.

When another node receives a ROUTE REQUEST, if it is the target of the Route Discovery, it returns a ROUTE REPLY message to the initiator of the Route Discovery, giving a copy of the accumulated route record from the ROUTE REQUEST; when the initiator receives this ROUTE REPLY, it caches this route in its Route Cache for use in sending subsequent packets to this destination. Otherwise, if this node receiving the ROUTE REQUEST has recently seen another ROUTE REQUEST message from this initiator bearing this same request id, or if it finds that its own address is already listed in the route record in the ROUTE REQUEST message, it discards the REQUEST. Otherwise, this node appends its own address to the route record in the ROUTE REQUEST message and propagates it by transmitting it as a local broadcast packet. In returning node e replying back to a**,** node e will examine its own Route Cache for a route back to a**,** and if found, will use it for the source route for delivery of the packet containing the ROUTE REPLY. Otherwise, e may perform its own Route Discovery for target node a**,** but to avoid possible infinite recursion of Route Discoveries.

B. security and performance issues

Certain features of DSR hurt its performance or make it vulnerable to security attacks.

- *No Expiration of Routes*: Without an effective mechanism to remove excessively old (*stale*) entries, route caches may contain broken or non-minimum hop routes. Using stale routes causes loss of data packets (low delivery rate) and wastes network bandwidth. Route replies from intermediate nodes and snooping data packets exacerbate this problem by polluting caches with stale routes [6, 9].
- *Intermediate-Node (IN) Replies*: Intermediate-node replies make the route learning process faster because all route requests do not need to travel all the way to the destination. Without route freshness indication, however, it results in polluting caches with stale routes when node mobility is high and data transmissions are infrequent [5, 8].

  When a source receives the 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, which then issues a new route request, starting the process all over again.
- *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 [9].

  Data Salvage can be useful in relatively static networks, in which routes remain stable for relatively long periods of time. However, in a MANET, 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. Also, a malicious node may misroute data packets without risking its detection under the guise of data salvaging.
- *Gratuitous Replies*: When a node overhears a packet addressed to another node, 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.

  Like data salvaging, gratuitous replies can be of limited benefit when the routes are fresh and nodes are not malicious. Otherwise, this feature degrades performance, security, or both.

III. PROPOSED TECHNIQUES OF DSR

We analyzed the performance of the original DSR and the impact of turning off some the optimizations discussed above. We made 3 optimizations to turn off, namely intermediate node replies, data salvaging and gratuitous. Taking this version of DSR as Base DSR we devised three new techniques to further improve its performance.

The throughput achieved by Base DSR is still less compared to other routing protocol like AODV at various loads for the example network configuration. So, to improve the performance of DSR further, we evaluate three simple, intuitive routing modifications based on our observations of other protocols.

A. Limiting Replies from Destination

In the original implementation of DSR, a destination node replies to every route request packet it receives. 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.

We modified DSR such that destination nodes will reply to a route request only if (a) the last route request from that source was older than the current one or (b) the last route request has the same timestamp (the same route request took different routes to the destination) but the current request took fewer hops. This ensures that replies are sent only for fresh request packets and multiple replies are sent only if they improve route hop count. This feature can be easily implemented using request and reply timestamps in route request and reply packets.

B. Giving Preference to Fresher Routes

The original DSR keeps multiple routes to a destination ordered by hop count. This ensures that routes used are minimum hop count routes, but also ensures that a stale 1-hop route overrides a fresher 2-hop route to the same destination.

We modified the route cache such that it maintains routes to a particular destination in the following order:

- A route with a later request time is given preference over a route with earlier request time;
- If the request times of two routes are the same, then a route with shorter hop count is given preference over a longer route;
- 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.

C. Keeping Only One Route per Destination

If routes are ordered by freshness, and the first route fails, it is very likely that the older routes stored in the cache will also fail. By trying all the routes in the cache before sending a new route request, a lot of time and bandwidth is wasted. In this technique, only one route determined to be the main route by freshness or hop count is kept in the cache. The current trend is to keep multiple routes and switch to a new one as soon as one fails. Keeping multiple routes improves throughput and reduces overhead when the network is congested and alternate routes are fresh. Since our interest is in uncongested networks, like AODV, which keeps only one route per destination performs well at low traffic, DSR also benefit from this feature.

IV.SIMULATION ENVIRONMENT

All simulations were run on the GloMoSIM network simulator [10]. The modifications were made to the implementation of DSR written for GloMoSIM. A 100 node network in a field size of 1000m x 1000m was used. The mobility model used was random waypoint [11] in a square/rectangular field. In random waypoint, each node starts its journey from its current location to a random location within the field. The speed is randomly chosen to be between 1-19 m/sec. Once the destination is reached, another random destination is targeted after a specified pause. We used 0-second pause time, which results in continuous node mobility in our simulations.

Twenty-five CBR (constant bit-rate) over UDP connections (distinct sources and destinations) were used to generate traffic by injecting 512-byte packets with average inter-packet time varied according to the load rate desired. For each configuration, the network is simulated for 600 seconds. We used *delivery rate*, the percentage of injected packets that are delivered to destinations, used to analyze the performance.

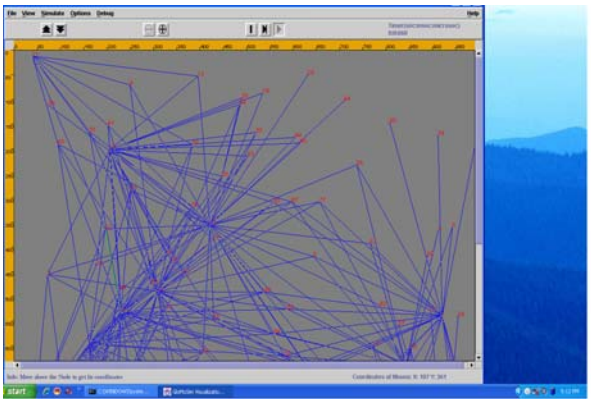

Fig 2: Snapshot of Simulation

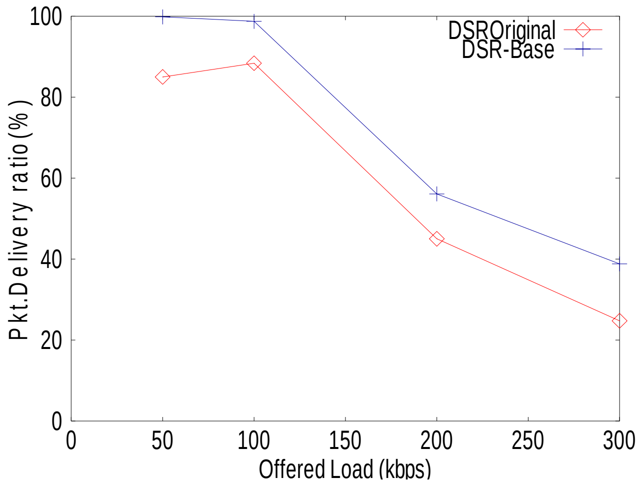

Fig3: Comparison of DSR with DSR all the three Optimizations off

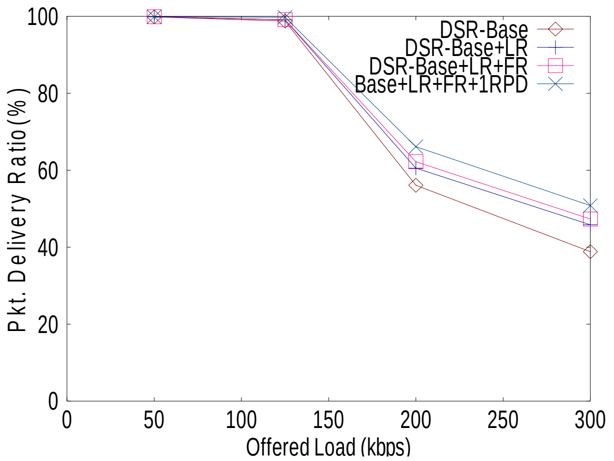

Fig4: Comparison of DSR Base with proposed techniques

At low loads, the delivery rate gives a measure of route correctness rather than load balancing or other issues of the protocol. In addition to 'Base DSR', we simulated the three proposed techniques applied to the base DSR. Figure 3 gives comparison in delivery ratios of original DSR and 'Base DSR' (original DSR with IN replies, DS and gratuitous replies turned off). Figure 4 gives delivery rates of 'Base DSR' and combinations of the three proposed techniques applied to the base DSR. (LR indicates limited replies, FR routes sorted by timestamps, and 1R one route per destination.)

Applying all three proposed techniques, denoted 'Base + LR + FR + 1RPD' in the graph, achieves the best performance until the network starts to saturate for high loads (>250 Kbps). At these high loads, most routes are congested. In such a scenario, congested links could be wrongly identified as 'broken', resulting in route errors and route requests propagating throughout the network.

Keeping only 1 route increases the routing overhead (Figure 4), and hurts the performance for high loads. At lower loads, there is enough network bandwidth to absorb the additional control traffic caused by 1RPD option.

Compared to the original DSR, the combination of base DSR with LR, FR and 1RPD options improves delivery rates by a significant factor various transmission loads.

V.CONCLUSIONS

DSR is a widely used routing protocol for mobile ad hoc networks, but has very low delivery rates and poor performance in lightly loaded networks with high node mobility. Several of the modifications proposed in the literature such as turning off intermediate node replies improves the performance somewhat.

This paper presents three intuitive techniques—limiting replies sent by destination, keeping only one route per destination, and preferring fresher routes over shorter ones—to further improve the performance of DSR. While multiple routes may benefit at higher traffic loads, keeping only one route per destination helps sender nodes gather routes when the topology changes. Without using any complicated strategies, our proposed techniques perform significantly better than previously proposed modifications at various traffic loads (50-300 Kbps).

REFERENCES

[1] D. Johnson, D. Maltz and Y. Hu. The dynamic source routing Protocol for mobile ad hoc networks. IETF MANET Working Group, draft 2003. http://www.ietf.org/internetdrafts/draft-ietfmanet-dsr03.txt,

[2] Samir Das, Charles Perkins, Elizabeth Royer. Performance Comparison of On-demand Routing Protocolss for MANETSIEEE,INFO2000. http://www.ietf.org/internetdrafts/dsr.txt,aodv.txt

[3] Theodoros Salonidis, Pravin Bhagwat, Leandros Tassiulas, Richard LaMaire. Distributed Topology Construction of Bluetooth Personal Area NetworksJEEE, INFOCQM2001.

[4] Zygmunt J. Haas, Guest Editorial Wireless Ad Hoc Networks,IEEE journal on selected areas in commn's, vol.17,no8,1999

[5] D. Johnson and D. Maltz. Dynamic Source Routing in Ad Hoc Wireless Networks. In Mobile Computing, edited by Tomasz Emilienski , Kluwer ,Academic Publishers, 1996.

[6] M. K. Marina and S. R. Das. Performance of Route Caching Strategies in DSR. In Proceedings of International Workshop on Wireless Networks,Mobile Computing,2001.

[7] D. Chakraborty and A. Joshi, "GSD: A novel group-based Service discovery protocol for MANETS", In IEEE Conf. on Mobile and Wireless Comm'ns Networks, Sept' 2002.

[8] S. Helal, N. Desai, V. Verma, and C. Lee, "Konark – A Service Discovery and Delivery Protocol for Ad-Hoc Networks", in Proceedings of the 3rd IEEE,WCNC, 2003.

[9] D. De Couto, D. Aguayo, J. Bicket, and R. Morris. A High-Throughput Path Metric for Multi-Hop Wireless Routing. In Proceedings of MobiCom

[10] X. Zeng, R. Bagrodia, and M. Gerla, "Glomosim: A library for parallel simulation of large-scale wireless nets in *Workshop on Parallel & Distributed Simulation*, 1998.

[11] T. Camp, J. Boleng, V. Davies. A survey of mobility models for Ad Hoc Network Research. WCMC: Special Issue on MANETS: Research&Apps. vol. 2, no. 5, 2002.

[12] C. E. Perkins. Mobile IP: Design Principles and Practices. Addison Wesley, 1997.

[13] Jochen Schiller. Mobile Communications. Pearson Education,2004.

[14] D.J.Goodman. Wireless Personal Communications Systems.AddisonWesley,2002.