RESEARCH ARTICLE OPEN ACCESS

Optimization of Link Cache in DSR over MANETs

Mahantesh.H.M [1], Shivakeshi.C [2], Naveen Kumar.B [3], Pampapathi.B.M [4]
Assistant Professor [1], [2], [3] & [4]
Department of Information Science and Engineering [1] & [2]
Department of Computer Science and Engineering [3] & [4]
Rap Bahadur Y. Mahabaleswarappa Engineering College
VTU, Ballari
Karnataka - India.

ABSTRACT

On-demand routing protocol for wireless ad hoc networks is one, that searches for and attempts to discover a route to some destination node only when a sending node originates a data packet addressed to that node. In order to avoid the need for such a route discovery to be performed before each data packet is sent, such routing protocols must cache previously discovered route. The cache structure called link cache can be created by adding individual links returned in route reply to the graph of the node. In a link cache, Cache timeout policy is required to be implemented due to time varying topology of the adhoc network, caused by node's mobility, which results in stale routes in the cache over period of time. Cache timeout policy in link cache gives a timeout that may be adaptive to remove the stale routes from the cache. Stale routes is a big issue in link cache structure, where individual links are combined together to find out best path between source and destination. In Adaptive timeout mechanism, a timeout value is assigned on the stability of the link endpoints. The timeout can be calculated based on the elapsed time since the link was last used and the last time the link was observed.

Keywords:-DSR, MANETs, LINK CACHE, RREQ, RREP

I. INTRODUCTION

The term MANETs (Mobile Ad hoc NETworks) refers to a set of wireless mobile nodes that can communicate and move at the same time. No fixed infrastructure is required to allow such communications; instead, all nodes cooperate in the task of routing packets to destination nodes. This is required, since each node of the network is able to communicate only with those nodes located within its transmission radius R, while a source node S and a destination node D of the MANET can be located at a distance much greater than R. When S wants to send a packet to D, the packet may have to cross many intermediate nodes. For this reason, MANETs belong to the class of multi-hop wireless networks. Routing in ad hoc networks traditionally uses the knowledge of the instantaneous connectivity of the network, which emphasis on the state of the links. This is the so-called topology-based approach, the associated routing protocols can be generally classified into two categories, periodic (also called as proactive or table- driven), on-demand (also called as reactive).

Networks using periodic protocols, attempt to maintain the knowledge of every current route to every other node by periodically exchanging routing information, regardless of whether the routes are being used for carrying packets. Each node maintains the necessary routing information, and each node is responsible for propagating topology updates in response to instantaneous connectivity changes in the network. Example of such routing protocol is Destination-Sequenced Distance-Vector (DSDV) routing.

The on-demand protocols create routes only when necessary for carrying traffic. As a result, a route discovery process is a prerequisite to establishing communication between any two nodes, and a route is maintained as long as the communication continues. One of the examples of an on-demand protocol is Dynamic Source Routing (DSR).Topology of the network changes dynamically, routing is difficult in MANETs. So it's difficult to determine path from source to destination. A different approach in using a protocol is to calculate a path when strictly needed for data transmission. If data traffic is not generated by nodes, then the routing activity is totally absent.

II. DYNAMIC SOURCE ROUTING PROTOCOL

The DSR enables the network to be completely self-organizing and self-configuring, requiring no existing network infrastructure or administration.

Since the nodes in the MANETs move about or join or leave the network, all routing information is automatically determined and maintained by the DSR routing protocol. The DSR protocol allows nodes to dynamically discover a source route across multiple network hops to any destination in the ad hoc network. Each data packet which has to send carries complete, ordered list of nodes through which the packet will pass, allowing packet routing to be trivially loop-free and avoiding the need for up-to-date routing information in the intermediate nodes through which the packet is forwarded. By including this source route in the header of each data packet, other nodes forwarding any of these packets may also easily cache this routing information for future use. The DSR protocol provides highly reactive service to ensure successful delivery of data packets in spite of node movement.

The DSR protocol is composed of two mechanisms: route discovery and route maintenance.

A. Route Discovery

When source node originates a new packet addressed to destination node, the source node places source route in the header of the packet. This source route specifies the sequence of hops that the packet is to follow to reach destination. Normally, the sender will obtain a suitable source route by searching its "Route Cache" (routes previously learned), but if no route is found in its cache, it will initiate the Route Discovery phase to dynamically find a new route to destination node. In this case, the source node is called as the "initiator" and the destination node as the "target" of the Route Discovery. For example, suppose a node S is attempting to discover a route to node D, the Route Discovery phase initiated by node S in this example would proceed as follows:

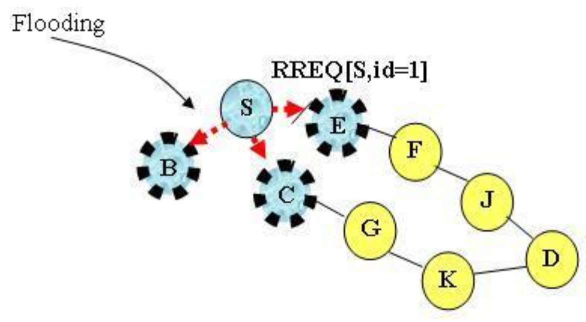

Figure 1. Route discovery initiated by source node S

The above Figure 1 shows that, to initiate the Route Discovery node S transmits a Route Request (RREQ) packet as a single local broadcast, which is received by (approximately) all the nearest neighbor nodes of S, that is the nodes B,E,C. Route Request packet contains the initiator ,target, and unique request identification, determined by the source. Route Request packet also contains a route record, listing the address of each intermediate node through which this particular copy of the Route Request packet has been forwarded. In this example, the route record initially lists only node S. When other node receives this Route Request (such as node B,E,C), if it is the target of the Route Discovery, it returns a "Route Reply" to the initiator of the Route Discovery, 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 the node receiving the Route Request has recently received Route Request packet with the same request id from initiator and target address, then this node discards the request. Otherwise, this node appends its own address to the route record in the Route Request and propagates it by transmitting it as a local broadcast packet (with the same request id). In this example, node B,E,C broadcast the Route Request, which is received by nodes G,F; the nodes G and F will broadcast the Route Request packet, and is received by nodes J,K. Then node J, K will in turn broadcasts then resulting Route Request is received by nodes D, this is shown in Figure 2.

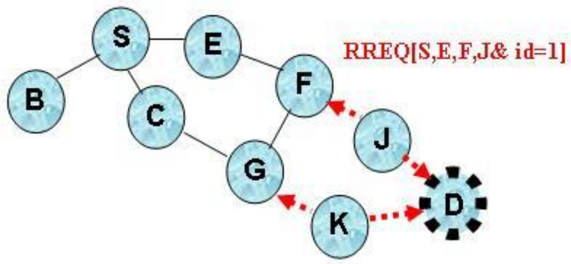

Figure2. Node J has broadcasted the Route Request Packet

In returning Route Reply(RREP) , the node D simply reverse the sequence of hops in the route record that it is trying to send in the Route Reply. This is shown in Figure 3.

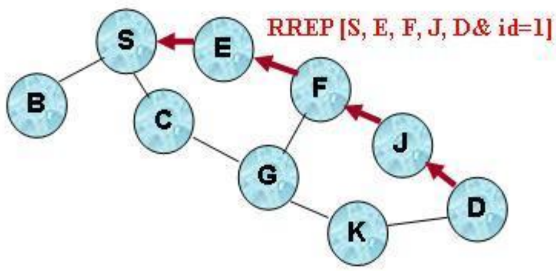

Figure3. Route Reply sent by node D

B. 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. For example a situation illustrated in Figure 4 shows that, node S originated packet to the node D, using the intermediate nodes E, F, J. In this case, node S responsible for the receipt of packet at E, node E is responsible for receipt at F, and then J is responsible for receipt at D. But the node J is unable to get a confirmation of the packet delivery from node D, then it sends Route Error (RRER) packet to the source node S. When RRER packet is received by intermediate nodes F, E if these nodes contain that route to the node D, then nodes will update their routing tables by removing links from cache.

When initiating a Route Discovery, the sending node saves a copy of the original packet in a local buffer called the "Send Buffer". The Send Buffer contains a copy of each packet that cannot be transmitted by this node because it does not have a source route to the destination. In response to a single route discovery, a node may learn and cache multiple routes to any destination. Nodes may also learn routing information from any packets that they forward In particular, routing information may be learned from a ROUTE REQUEST, ROUTE REPLY, or ROUTE ERROR packet, or from the source route in the header of a data packet.

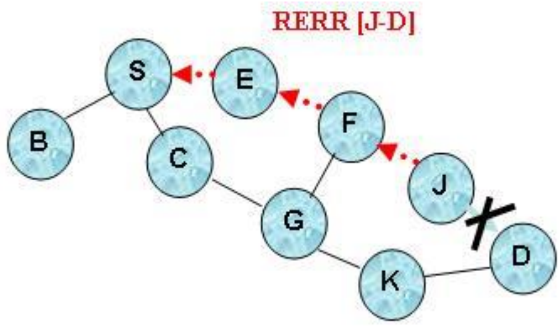

Figure4. Example for route maintenance

C. Route cache

Route cache in DSR protocol is used to store the routes that have learned from the source route or during route discovery phase, to avoid unnecessary route discovery operation each time a data packet is to be transmitted. Because the reinitiating of a route discovery mechanism in on demand routing protocols is very costly in terms of delay, battery power, and bandwidth consumption, which can cause long delay before the first data packet sent. The performance of DSR protocol mainly depends on an efficient implementation of route cache. Using DSR, each node must maintain a Route Cache, containing routing information needed by the node. A node adds information to its Route Cache as it learns of new links between nodes in the ad hoc network; for example, a node may learn of new links when it receives a packet carrying a Route Request, Route Reply, or DSR source route. Likewise, a node removes information from its Route Cache as it learns that existing links in the ad hoc network have broken. For example, a node may learn of a broken link when it receives a Route Error packet.

III. CACHING STRUCTURE

Cache structure is very important for developing a route cache strategy in DSR protocol. In the cache structure, there are two kinds of caches are used. First, a path cache which represents a complete path (a sequence of links) that caches the routes separately. Second, a link cache that represented when a node caches each link individually, adding it to a graph of a node. In case of the path cache, when the source node A adds a new route A-B-F-G-H to its cache, it has to add the whole path as independent entry in its cache. While in case of the link cache when the source node A adds a new route A-B-F-G-H to its cache, it has to add the link B-F only. A path cache is simple to implement, and guarantees that all routes are loop-free, since each individual route from a ROUTE REPLY is loop-free. To find a route in a path cache, the sending node can simply search its cache for any path that leads to the intended destination node. On the other hand, to find a route in link cache, a node must use a much more complex graph search algorithm. However, in path cache, it does not effectively utilize all of the potential information that a node learns about the state of the network. While in link cache, it has a conventional graph data structure, in which all the links are learned from different route discoveries or from the header of any packets can be combined together to form a new routes in the route cache of the network. Therefore, this is not possible in path cache due to the separation of each individual path in the cache. Performance of link cache is good compared to path cache, because link cache can delete only a broken link when the link causes a path to break. While in path cache whole route is deleted when link failure occurs.

Cache capacity can be divided into two halves: first half is called as "Primary Cache", which represents paths that have been used by the current node. Second half is called as "Secondary Cache", which represents paths that have not yet used. When the source node attempts to add new path that have learned and not yet been used in the cache, old paths in the secondary cache are removed due to limitation of the cache capacity in the network. While with primary cache, old paths are removed more actively due to the operation of route maintenance. Cache timeout policy is required to be implemented in link cache due to time varying topology of the ad hoc network caused by node's mobility that can cause stale routes in the cache. Deriving proper cache timeout policy is important role for ensuring cache freshness. Cache timeout policy in link cache gives a timeout that may be static or adaptive to remove the stale routes from the cache. Stale routes is a big issue in link cache structure, where individual links are combined together to find out best path between source and destination. However, cache timeout policy is not possible to setup timeout in path cache structure due to the limitation capacity of the storage space. There are two kinds of cache timeout are used: static timeout approach and adaptive timeout approach. Static timeout is assigned the same timeout value for every link cache entry; each link is removed from its cache after specific value of time has elapsed since the link was added to the cache. In contrast, adaptive timeout is assigned a timeout value based on the stability of the link endpoints. The timeout can be calculated based on the elapsed time since the link was last used. The following Figure 5 shows the various caching strategies.

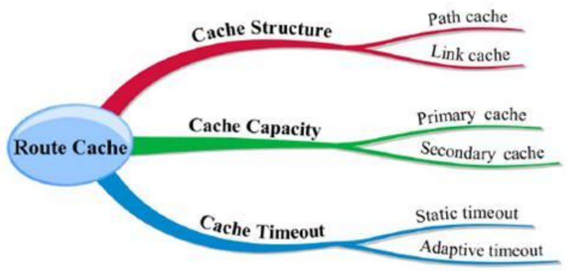

Figure 5. Route Cache strategies

Replies from caches provide dual performance advantages. First, they reduce route discovery latency. Second, without replies from caches, the route request packet will reach all nodes in the network. Cached replies quench the query flood early, thus saving on routing overheads. However, without an effective mechanism to remove stale cache entries, caches may contain routes that are invalid. Then, route replies may carry stale routes. Attempted data transmissions using stale routes incur overheads, generate additional route error packets and can potentially pollute other caches when a packet with a stale route is forwarded.

A. Path Cache System

The Dynamic Source Routing protocol is unarguably the most referred protocol for mobile networks. The effectiveness of the protocol is not unconnected with its caching mechanisms, that is, the ability to store already discovered routes in the nodes caches. This capability of the DSR protocol has been shown to reduce the overall control overhead involved in packets data transmission.

In DSR, the initiator of a Route Discovery represents a complete path (a sequence of links) leading from that node to the destination node. By caching each of these paths separately, a path cache can be formed

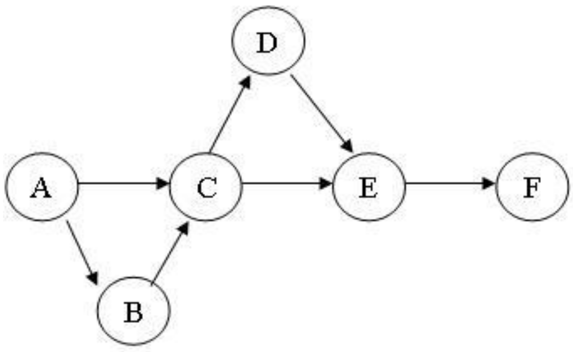

Figure 6: Typical view of MANET

Path cache of the node A:

A ⟶ C
A ⟶ C ⟶ D
A ⟶ C ⟶ D ⟶ E

The problem with the path cache is that, when a link A ⟶ C is broken then there will be no route to the nodes D and E exists in path cache.

Limitation of Path Cache System

- The Stale routes causes packet losses if packets cannot be salvaged by intermediate nodes
- The stale routes increases packet delivery latency
- Use static time out mechanisms
- If the cache size is set large, more stale routes will stay in caches because FIFO replacement becomes less effective

B. Link Cache System

The Link Cache can be formed by adding a link to the graph of the node returned in Route Reply. In link cache, links learned from different Route Discoveries or from the header of any overheard packets can be merged together to form new routes in the network, but this is not possible in a path cache.The following figure 6 shows the example for a link cache:

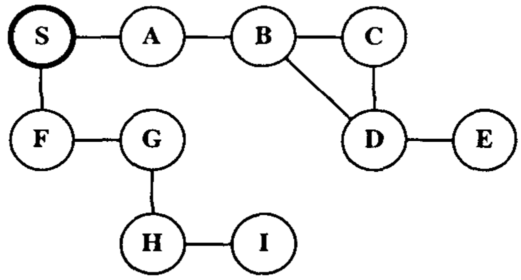

Figure 6 link cache for node S

Each node keeps a table recording its perceived stability of other node. When a link enters cache, timeout for the link will be assigned; if the link is used again before the timeout then timeout of the link is multiplied by some factor that can be configured as per the requirement. This increases the time out of the link. Once this timeout reaches zero, links will be eliminated from link cache

The information maintained in link cache:

- Each node will have topological view of the network.
- Each node maintains time out for the links

The algorithm also has the following attractive properties:

- Distributed: It is scalable with network size because algorithm uses only local information and communicates with neighbourhood nodes.
- Adaptive: It notifies the nodes that have cached a failure link to update their caches.
- Proactive on-demand: Proactive cache updating is triggered on-demand, without periodic behaviour.

The Link Cache system also identifies the shortest path to destination, when source receives multiple routes to the destination. In optimization of the dynamic source routing, each node collect the all information about which node discover which link through forwarding packet.

The advantages of the Link Cache system are:

- Routes maintained only between nodes that need to communicate.

- No periodic activity, thus less overhead in packet delivery.
- A single route discovery allows nodes replying from local caches yield many routes to the destination,
- A route is established only when it is required and hence the need to find routes to all other nodes in the network
- It also reduces the time taken to find the path to given destination from source node

IV. CONCLUSION

This paper presents concept called link cache, this link cache provides topology of the network. The main use of link cache is to avoid unnecessary route discovery process during the broadcasting of the route request packet. The link cache scheme affects on the performance of on-demand routing protocols in ad hoc networks. A link cache with an appropriate timeout mechanism could make use of the available route information more efficiently, thus improve the protocol performance. However, without an appropriate timeout mechanism, a link cache may cause dramatically increased route errors and consequently severe performance degradation. The adaptive link lifetime estimation scheme aims at tracking the "optimal" link lifetime under various node mobility conditions.

REFERENCES

[1] J. Broch, D. Maltz, D. Johnson, Y.-C. Hu, and J. Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In Proc. 4th ACM MobiCom, pp. 85–97, 1998.

[2] K.Fall and K. Varadhan, Eds. ns notes and documentation. The VINT Project, UC Berkeley, LBL, USC/ISI, and Xerox PARC, 1997.

[3] G. Holland and N. Vaidya. Analysis of TCP performance over mobile ad hoc networks. In Proc. 5th ACM MobiCom,pp. 219–230, 1999.

[4] Y.-C. Hu and D. Johnson. Caching strategies in on-demand routing protocols for wireless ad hoc networks. In Proc. 6th ACM MobiCom, pp. 231–242, 2000.

[5] IEEE Computer Society LAN MAN Standards Committee. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, IEEE Std 802.11-1997. The IEEE, New York, New York, 1997.

[6] D. Johnson and D. Maltz. Dynamic Source Routing in ad hoc wireless networks. In Mobile Computing, T. Imielinski and H.Korth, Eds, Ch. 5, pp. 153–181, Kluwer, 1996.

[7] Y.-C. Hu and D. Johnson. Ensuring cache freshness in on-demand ad hoc network routing protocols. In Proc. 2nd POMC,pp. 25–30, 2002.

[8] D. Johnson, D. Maltz, and Y.-C. Hu. The Dynamic Source Routing for mobile ad hoc networks, IETF Internet Draft. http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-10.txt July 2004.

[9] W. Lou and Y. Fang. Predictive caching strategy for on-demand routing protocols in wireless ad hoc networks. WirelessNetworks, 8(6): 671–679, 2002.

[10] D. Maltz, J. Brooch, J. Jetcheva, and D. Johnson. The effects of on-demand behavior in routing protocols for multi-hop wireless ad hoc networks. IEEE J. on Selected Areas in Communication, 17(8):1439–1453, 1999.

[11] M. Marina and S. Das. Performance of routing caching strategies in Dynamic Source Routing. In Proc. 2nd WNMC, pp. 425– 432, 2001.

[12] C. Perkins and E. Royer. Ad hoc on-demand distance vector routing. In Proc. 2nd WMCSA, pp. 90–100, 1999.

[13] C. Perkins, E. Royer, and S. Das. Ad hoc On-demand Distance Vector (AODV) Routing, RFC 3561. http://www.ietf.org/rfc/rfc3561.txt July 2003.

[14] C. Perkins, E. Royer, S. Das, and M. Marina. Performance comparison of two on-demand routing protocols for ad hoc networks. IEEE Personal Communications, 8(1): 16–28, 2001.

[15] The Monarch Project. Mobile networking architectures. http://www.monarch.cs.rice.edu/.

[16] X. Yu and Z. Kedem. A distributed adaptive cache update algorithm for the Dynamic Source Routing protocol. In Proc. 24th IEEE INFOCOM, March 2005. (An earlier version appeared as NYU CS Technical Report TR2003-842, July 2003.)