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

Route Cache Update Mechanisms in DSR Protocol – A Survey

Naseer Ali Husieen[+], Osman B Ghazali, Suhaidi Hassan, Mohammed M. Kadhum

InterNetWorks Research Group, Universiti Utara Malaysia, Malaysia

Abstract. Recently, there has been a growing interest in Mobile Ad Hoc Network (MANET). Ad hoc network becomes popular since it can provide useful personal communication in certain applications such as battlefield, academic, and business without any support where no fixed infrastructure exists. All mobile nodes communicate with each other direct or through intermediate node using any routing protocol. This paper has focused mainly on Dynamic Source Routing (DSR) protocol regarding route cache. The route cache is used in DSR protocol to store all the routes are learned from the source node and to avoid unnecessary route discovery process. With high mobility environment and high load network traffic stale routes will be generated. These stale routes can mainly affect the performance of DSR protocol which cause long delay, increase the packet loss, increase the overhead, and decrease the performance of TCP protocol. Thus efficient update mechanism required in the cache of DSR protocol. This paper will review different strategies of the route cache and the drawbacks of the pervious mechanism for updating the routes in the cache of DSR protocol.

Keywords- MANET, DSR, Route cache.

1. Introduction

Mobile ad hoc network is collections of wireless nodes that can allow people and devices to communicate with each other without help of an existing infrastructure, e.g., disaster recovery environments [1]. A MANET is self configuring and self-organize into arbitrary and temporary, mobile node can be work as router or host. MANET day after day gets new applications ranging from military applications for connecting soldiers in battlefields and social or business application such as Public and Personal Area Networks, other applications are recently under development will also gain from mobile ad hoc network advantages such as telemedicine, weather conditions report, and disaster environment such as in seism. All the examples above of use predict for some envisioned mobile ad hoc network to increase in range to reach the threshold of thousands of nodes per system (commercial or military). One of the most challenges in mobile ad hoc network is routing protocols. The primary goal of routing protocols is to find the path between the source and destination and deliver the data packets in timely manner. In ad hoc networks, high mobility, limited computing capability, and low bandwidth characteristics of mobile nodes make routing data is one of the most difficult issues.

There are two categories of routing protocols in mobile ad hoc network have been proposed. First, on-demand (or reactive) protocols such as (DSR [2], AODV [3], and TORA [4]), in which node establish the route before sending packets; on other hand node will maintain a route only when it's needed. Second, periodic (or proactive) protocols such as DSDV[5], in which the nodes will periodically exchange the routing information and periodically exchange routing information and try always to know the current route to the destination. In general, reactive protocols have been shown outperform those based on periodic mechanisms due to their reduced overhead and ability to react quickly as route changes [6-9], which is motivated us to carry out and investigate more on demand protocols.

[+] Corresponding author. Tel.: + 60175283740; fax: +604-9284054
E-mail address: naseerali@internetworks.my

2. Dynamic Source Routing

Dynamic Source Routing (DSR) is a reactive and simple protocol developed at Carnegie Mellon University (CMU) by Johnson et al. in 1996 [10]. The key characteristic of DSR is based on the concept of source routing. The advantages of using source routing concept in DSR protocol, first is to provide flexibility, simplicity, and correctness of DSR protocol. Second, is to allow the packet routing to be trivially loop-free and to avoid the need for updating routing information in the intermediate nodes, through which packets are forwarded as stated in [11]. The DSR protocol is composed of two main mechanisms that work together to allow the discovery and maintenance of source routes in the ad hoc network. All aspects of protocol operate entirely on demand allowing routing packet overhead of DSR to scale up automatically.

Route Discovery: When a source node *S* wants to send a packet to the destination node *D*, it obtains a route to *D*. This is called Route Discovery. Route Discovery is used only when *S* attempts to send a packet to *D* and has no information on a route to *D*. **Route Maintenance:** When there is a change in the network topology, the existing routes can no longer be used. In such a scenario, the source S can use an alternative route to the destination *D*, if it knows one, or invoke Route Discovery. This is called Route Maintenance

3. Route Cache in DSR Protocol

Route cache strategy in DSR protocol was proposed by Johnson et al. in 1998 [12]. It is used to store the routes that have learned from the source node and to avoid unnecessary route discovery operation each time a data packet is to be transmitted. Because that reinitiating of a route discovery mechanism in on demand routing protocols is very costly in term of delay, battery power, and bandwidth consumption due to flooding of the network, 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 as stated in [13-15].

The first study of effect the route cache in the performance of DSR protocol was stated in Maltz et al. [12]. From their observation, the majority of Route Replies (REPP) packets are based on route cache, and only 59 percent of (REPP) packets carry correct routes. In addition, they have also discovered that even (REPP) packets from the destination are not 100 percent correct, since routes may break down while (REPP) packet sent back to the source node. Thus, efficient route maintenance is important for all reactive protocols with route cache. A route caching is the major approach to decrease the flooding of the network by avoiding route discovery operation as much as possible. Thus, route cache is a vital component in DSR protocol. As examined by [14, 16-21], the current route cache in DSR protocol does not have any mechanism to update the routing information in the route cache to distinguish the fresh routes from stale routes. However, if there is a mechanism that can expire the routes or determine the freshness of routes in the route cache, it will improve the DSR performance significantly. Specially, with high mobility environments and high load network traffic, a route cache may contain stale routes that can affect the performance of DSR protocol. The stale routes in the route cache of DSR protocol can have several adverse effects such as packet loss, long delay, increase the routing overhead during generation of RERR packet, and decrease the performance of TCP protocol as stated in [12]. According to Hu and Johnson [13] route cache can be divided into three different strategies as shows in Fig. 1.

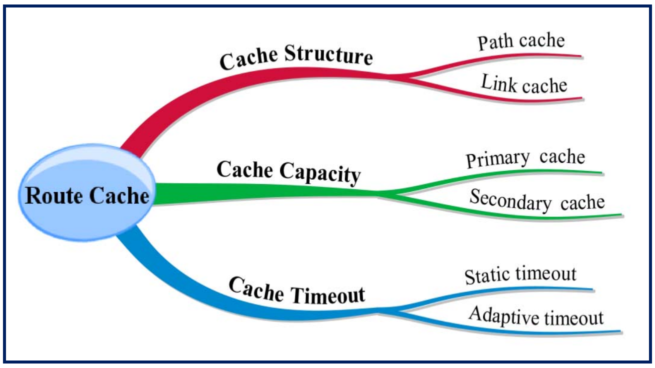

Fig. 1: Route Cache Strategies

3.1. Cache Structure

Cache structure is very important for developing a route cache strategy in DSR protocol as stated in [15, 22]. 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 represents when a node caches each link individually, adding it to a graph of links.

In case of the path cache structure 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, since rest of the links already exist in its cache.

However, in path cache, it does not effectively utilize all of the potential information that a node learns about the state of the network and there is no sharing in the data structure even when two paths share a number of common links as stated in [13, 21]. 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 overhead 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. In addition, several studies have been shown that link cache outperforms path cache when the network load is high as examined by [15, 20, 21-23]. Because of the link cache can delete only a broken link when the link causes a path to break. While in path cache whole the route which link failure occurred will be deleted

3.2. Cache Capacity

According to Hu and Johnson [13] cache capacity is also important cache for choice in designing a cache strategy for DSR protocol. For link cache, it can store all the links that are discovered in the cache, since there is fixed maximum N x N links that may exist in an ad hoc network with N nodes; these stored routes require less storage space. While in the path cache, the maximum storage space is required much larger, since a path is stored separately.

Cache capacity can be divided into two halves: first half-called “Primary Cache”, which represents paths that have used by the current node. Second, half-called “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 into 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.

3.3. Cache Timeout

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 over period of time as stated in [13, 19, 24-26]. 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.

According to Hu and Johnson [13] 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 and the last time the link was observed

4. Related Work

Several studies have been proposed for updating routes in DSR protocol and remove the stale routes as proposed by [14-15, 18-20, 26].

Marina and Das [14] addressed three main problems with current route cache in DSR protocol: Incomplete Error Notification, No Expiry, and Quick Pollution. Based on these problems they have developed three mechanisms namely Wider Error Notification, Timer- Based Cache, and Negative Cache. Wider Error Notification is based on the fast and wide propagation of Route Error (RERR) packet to increase the speed and the size of (RERR) packet, route error propagation packets are transmitted as broadcast packets at the MAC layer. When a node receives a (RERR) packet that containing link failure information, a node updates its route cache so that all source routes the contained link failure are truncated at the point of failure. There are two drawbacks of using this mechanism, first, a node that detects link failure does not know which neighbors have cached the link and cannot notify all nodes that need to be notified. Second, this mechanism is based on broadcast technique, which can introduce overhead to the node that does not cache a broken link, and some of the nodes cached a broken link but might not receive a notification because broadcast in unreliable. In addition, these three mechanisms were considered static timeout scheme in which a fixed time value assigned same value to every link. After a link stays for specific time, it is deleted from the link cache. However, the weakness of static timeout scheme that it cannot adapt to the network changes rapidly in the topology of the network.

Lou and Fang [15] proposed Adaptive Link Cache Scheme to remove the stale routes in the cache of DSR protocol and compare it with path cache structure. Adaptive Link Cache Scheme is a combination of link cache and adaptive timeout policy. The main concern of using Adaptive Link Cache Scheme is to track the optimal link lifetime with different node mobility levels situations. Through their results, it shows that with high load traffic network, Adaptive Link Cache mechanism outperforms the path cache and the stale routes are removed from the cache. This mechanism uses heuristics with ad hoc parameters to predict the lifetime of a link. However, heuristics cannot accurately estimate timeout of the link because topology changes are unpredictable.

Xin Yu [18] proposed Distributed Adaptive Cache Update Algorithm. The main goal of this algorithm is to update the route cache of DSR protocol by using proactive cache update instead of adaptive timeout mechanism in link cache structure to remove the stale routes in the cache and to collect the extra information about how the routing information distributed through the network. In the Distributed Adaptive Cache Update Algorithm, four fields where added to the path cache structure in each node, for each route a node maintains the information about these four fields. First and second fields represent the source and the destination node for current route. Third field represents the number of data packets that delivered to the destination node by using current route to know how the routing information is synchronized among all the nodes on the route. Finally, the field "Routing Table" that represents each node maintains the routing information about which neighbor has learned for this route. However, Distributed Adaptive Cache Update Algorithm is based on the path cache, which cannot effectively utilize all of the routing information that a node learns about the state of the network. In addition, cache timeout is not used, but in some cases, mobile nodes can be unreachable and they do not remove the stale route from their caches.

Y. Raghavendra et al. [19] proposed approach called "Active Packet" to solve the stale route in the route cache of DSR protocol. This approach based is on the link cache structure, which an active packet roams around the network, and collects the routing information that has obtained by network topology. This approach has two phases: first phase is updating the route cache and second phase is gathering all the information about the network by checking the payload of Active Packet. With Active Packet, it can update the routing information in the cache and it is possible to add a new route to the cache. Active Packet is generated frequently by choose randomly from the nodes. However, it is not described how the node can be selected and how it is guaranteed that only one node generates active packet in a specific time. Therefore, this approach increases the overhead significantly without any benefit, which leads to decrease the performance of DSR protocol.

J. Chen et al. [20] proposed Tiding Active Packets (TAP) mechanism to improve the Active Packet Method in [19]. TAP is composed of the Active Packets and Route Error (RERR) flooding to update the stale routes in the cache of DSR protocol. As stated in [20], Tiding Active Packets has three phases: first phase is the Topology Collection, in which, Active Packet makes two visits to all the nodes. In the first visit, Active Packet travels to each node to collect the information of the neighbor nodes. After finished the information collection by Active Packet, its starts the second visit, through which, all the nodes get all the topology information from Active Packet and use this information in the route discovery process. Second phase is the Path Calculation that represents a node that calculates its path from the source to the destination before it needs to forward a data packet. Third phase is the Topology Maintenance, in which a node has responsibility to ensure the transmission of data packet successfully to the next hop by using a local data cache. Tiding Active Packets mechanism have tested with low mobility 0 m/s to 1 m/s only, and low load traffic network. However, with high mobility environments and high load network traffic, a route cache may contain stale routes that can affect the performance of DSR protocol.

Hu and D. Johnson [22] proposed Epoch Number mechanism to avoid the stale routes in the cache of DSR protocol. Epoch Number mechanism prevents a node from relaying on stale link after heard that the link is broken. Epoch Number mechanism allows the node to hear both of link broken and route discovery on the same link to sequence two events to find out which event can be occurred before others and it is not based on any ad hoc mechanisms such as short-lived negative cache mechanism. By using Epoch Number mechanism, the invalid information in the route cache is reduced in order to have enough information to make sure that new link correct and sequence with link discovery and link breakage information. However, the authors did not address how is to quickly remove the stale routes from the route cache and relaying stale link is only one aspect of the cache staleness issue.

5. Conclusion

The main use of route cache in DSR protocol is to avoid unnecessary route discovery process during the broadcasting of the route request packet. In this paper, we have presented the different strategies of the route cache in DSR protocol and the drawbacks of the previous mechanisms for updating the routes in the cache of DSR protocol. With high mobility situations and high load traffic network stale routes will be generated in the route cache which is big issue in DSR protocol. These stale routes can cause increase packet loss, long delay and reduce the efficiency of the performance of DSR protocol. Therefore efficient mechanism for updating the routes in the cache of DSR protocol is needed.

References

[1] J. Jain, M. Fatima and R. Gupta, “Overview Challenges of Routing Protocol and Mac Layer in Mobile Ad-hoc Network,” Journal of Theoreticla and Applied Information Technology, JATTT, 2009.

[2] C.E. Perkins and E.M. Royer, “Ad-Hoc on-Demand Distance Vector Routing,” in Second IEEE Workshop on Mobile Computing Systems and Applications (WMCSA), pp. 90–100, Feb.1999.

[3] M.S. Corson and A. Ephremides, “A Distributed Routing Algorithm for Mobile Wireless Networks,” Wireless Networks, vol. 1, pp. 61–81, Feb.1995.

[4] C.E. Perkins and P. Bhagwat, “Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers,” in Proceedings of the SIGCOMM ’94 Conference on Communications Architectures, Protocols and Applications, pp. 234–244, Aug.1994.

[5] J. Broch, D.A. Maltz, D.B. Johnson, Y.C Hu, and J.G. Jetcheva, “A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols,” in Proceedings of the Fourth Annual ACM/IEEE International Conference on obile Computing and Networking (MobiCom), pp. 85–97, Oct. 1998.

[6] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark, “Scenario-based Performance Analysis of Routing Protocols for Mobile Ad-hoc Networks,” in Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom), pp. 195–206, Aug. 1999.

[7] D.A. Maltz, J. Broch, J. Jetcheva, and D.B. Johnson, “The Effects of On-Demand Behavior in Routing Protocols for Multi-Hop Wireless Ad Hoc Networks,” IEEE Journal on Selected Areas in Communications, Special Issue on “Wireless Ad Hoc Networks,” vol. 17, pp. 1439–1453, Aug. 1999.

[8] S.R. Das, C.E. Perkins, “Performance Comparison of Two On-demand Routing Protocols for Ad Hoc Networks”, in Proceedings of the IEEE Computer and Communications Societies (INFOCOM), pp. 3-12, 2000.

[9] P. Prabhakaran and R. Sankar “Impact of Realistic Mobility Models on Wireless Networks Performance, “IEEE,

2006.

[10] D.B. Johnson and D.A. Maltz. “Dynamic Source Routing in Ad Hoc Wireless Networks,” in Mobile Computing, edited by Tomasz Imielinski and Hank Korth, Kluwer Academic Publishers, chapter 5, pp. 153–181, 1996.

[11] D.B. Johnson, D.A. Maltz, Y. Hu, “Dynamic Source Routing Protocol for Mobile Ad-hoc Networks (DSR),” [Online]. Available: http://tools.ietf.org/html/draft-ietf-manet-dsr-10, [Accessed: Jul.2004].

[12] D.B. Johnson, D.A. Maltz, J. Broch, “The Dynamic Source Routing protocol for Mobile Ad Hoc Networks,” in Mobile Adhoc Network (MANET) Working Group. IETF. [Online]. Available: http://www.monarch.cs.rice.edu/internet-drafts/draft-ietf-manet-dsr-00.txt. [Accessed: Mar. 1998].

[13] Y. Hu, D.B. Johnson, “Caching Strategies in On-Demand Routing Protocols for Wireless Ad Hoc Networks,” in Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MobiCom), Aug. 2000.

[14] M.K. Marina and S.R. Das, “Performance of Route Caching Strategies in Dynamic Source Routing,” in Proceedings of the 2nd Wireless Networking and Mobile Computing (WNMC), pp. 425-432, Apr. 2001.

[15] W. Lou and F. Yuguang, “Predicative Caching Strategy for On- Demand Routing Protocols in Wireless Ad Hoc Networks,” Wireless Networks, vol. 8, pp. 671-679, 2002.

[16] A.B.R, C.L. Reddy, and P.S. Hiremath, “High Performance Modified DSR with Power Consumption Optimization for Mobile Ad hoc Network,” International Journal of Computer Science and Network Security (IJCSNS), vol. 7, pp. 169-178, May. 2007.

[17] Abdullah Gani, Qi Han, Nor Badrul Anuar, Omar Zakaria, “Enhancing DSR Protocol Performance in Mobile Ad Hoc Network,“ Wseas Transactions on Communications, vol. 8, pp. 227-238, Feb. 2009.

[18] Xin Yu, “A Distributed Adaptive Cache Update Algorithm for the Dynamic Source Routing protocol,” IEEE Transactions on Mobile Computing, vol. 5, pp. 609-626, Jun. 2006.

[19] Y. He, C.S. Raghavendra, S.Berson, B. Braden, “Active Packets Improve Dynamic Source Routing for Ad Hoc Networks”, Proceeding IEEE Conference on Open Architectures and Network Programming (OPENARCH), 2002.

[20] Jiaxu Chen, Yazhe Tang, Dian Fu, and Heng Chang, “On the Improving Strategies upon the Route Cache of DSR in MANET,” Springer, vol. 6406, pp. 445–458, 2010.

[21] Dimitri Marandin, “Improvement of Link Cache Performance in Dynamic Source Routing (DSR) Protocol by Using Active Packets,” Next Generation Teletraffic and Wired Wireless Advanced Networking, Springer, vol. 4712, pp. 367–378, 2007.

[22] Gavin Holland, Nitin Vaidya, “Analysis of TCP performance over Mobile Ad Hoc Networks,” Proceeding 5th annual ACM/IEEE International Conference on MOBICOM, pp. 219–230, 1999.

[23] Adrian J. Cahill , Phillip L. De Leon , Cormac J. Sreenan, “Link Cache Extension for Predictive Routing and Repair in Ad Hoc Wireless Networks,” Proceedings of the 4th IEEE Conference on Mobile and Wireless Communication Networks, pp. 53-57, 2002.

[24] Shukla, “Ensuring Cache Freshness in On-demand Routing Protocols for Mobile Ad Hoc Network: A Cross-layer Framework,” 4th IEEE Costumer Communications and Networking Conference (CCNC), pp. 264-268, 2007.

[25] J.A.G. Garrido, D. Marandin, “A Lincage Invalidation Mechanism for Dynamic Source Routing (DSR) in Ad Hoc Networks,” 18th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'07), pp. 1-5, 2007.

[26] Y.-C. Hu and D. Johnson, “Ensuring Cache Freshness in On- Demand Ad Hoc Network Routing Protocols,” Proceeding 2nd Workshop Principles of Mobile Computing, (POMC ’02), pp. 25-30, 2002.