ANALYSIS OF PERFORMANCE OF DSR USING DIFFERENT TYPES OF CACHE IN DYNAMIC ENVIRONMENT

Shobha.K.R and Dr.K.Rajanikanth

M.S.Ramaiah Institute Of Technology, Bangalore, Karnataka, India.
shobha_shankar@yahoo.com
principal@msrit.edu

ABSTRACT

Route caching strategy is important in on-demand routing protocols used in Mobile Adhoc Networks (MANETs). While high routing overhead usually has a significant performance impact in low bandwidth wireless networks, a good route caching strategy can reduce routing overheads by making use of the available route information more efficiently. In this paper, we present the effects of two cache schemes, "link cache" and "path cache" on the performance of on-demand routing protocols through simulations based on the Dynamic Source Routing (DSR) protocol. We have proposed a new simulation environment where the nodes move in random direction with random velocity. This was implemented by modifying the random waypoint model so that the scenario matches to that of the real world. The analysis of the performance of DSR using the link cache has been done in a new simulation environment by using static time out mechanism and a new proposed dynamic timeout mechanism. The effects of different link lifetime values and cache timeout values on the performance of routing protocol in terms of routing overhead and packet delivery ratio are investigated and presented.

KEYWORDS

MANETs, Reactive protocols, DSR, Caching, Link cache, Path cache, Static timeout, Dynamic timeout.

1. INTRODUCTION

An adhoc network is an infrastructure-less multi-hop mobile wireless network. In an adhoc network, there is no fixed infrastructure such as base stations that function as routers as in a wireless cellular network [1][2][3]. Each node in an adhoc network is capable of moving independently and functioning as a router that discovers and maintains routes and forwards packets to other nodes. Due to its self-configuring, self-organizing nature, and its capability to be promptly deployed without any wired base stations or infrastructure support, adhoc networks have been very attractive in tactical and military applications, where fixed infrastructures are not available or reliable and fast network establishment and self-reconfiguration are required. Examples include the tactical communication in a battlefield and the disaster rescue after an earthquake. Recently, due to the availability of wireless communication devices that operate in the ISM band, the interest in ad hoc networks has extended to civilian applications such as on-the-fly setup for conferencing and home-area wireless networks.

Since each node in an adhoc network is capable of moving collectively or independently of other nodes, the network topology can change dramatically. Traditional Internet routing protocols are no longer effective in adhoc networks. It presents a great challenge for a routing protocol to keep up with the frequent and unpredictable topology changes. Based on when routing activities are initiated, routing protocols for mobile adhoc networks may be broadly classified into three basic categories: (a) proactive or table-driven protocols, (b) reactive or on-demand routing protocols, and (c) hybrid routing protocols.

Proactive protocols perform routing operations between all source destination pairs periodically, irrespective of the need of such routes. Proactive protocols have advantages of providing lower latency in data delivery and the possibility of supporting applications that have quality-of-service constraints. These protocols work well under heavy traffic and high mobility conditions as they try to maintain fresh routing information continuously. Their main disadvantage is wastage of bandwidth in sending update packets periodically even when they are not necessary, such as when there are no link breakages, or when only a few routes are needed. The most popular proactive protocols are Destination-Sequenced Distance-Vector Routing (DSDV) [5] and Optimized Link State Routing Protocol (OLSR) [13].

Reactive protocols are designed to minimize routing overhead. Instead of tracking the changes in the network topology to continuously maintain shortest path routes to all destinations, these protocols determine routes only when necessary. The advantage of this on-demand nature of operation is that it usually has a much lower average routing overhead in comparison to proactive protocols. However, it has the disadvantage that a route discovery may involve flooding the entire network with query packets. Moreover, route discovery adds to the latency in packet delivery as the source has to wait till the route is determined before it can transmit. Despite these drawbacks, on-demand protocols receive comparatively more attention than proactive routing protocols, as the bandwidth advantage makes them more scalable. The most popular reactive protocols are Dynamic Source Routing (DSR) [6] and Ad Hoc On Demand Distance Vector Routing (AODV) [7].

Hybrid routing is an approach that is often used to obtain a better balance between the adaptability to varying network conditions and the routing overhead. These protocols use a combination of reactive and proactive principles, each applied under different conditions, places, or regions. For instance, a hybrid routing protocol may benefit from dividing the network into clusters and applying proactive route updates within each cluster and reactive routing across different clusters. Routing schemes that employ proactive route maintenance on top of reactive route discoveries also exist.

Our simulations are based on the reactive routing protocol DSR. DSR is an on-demand routing protocol that is based on the concept of source routing. The protocol is composed of two major mechanisms, i.e. Route Discovery and Route Maintenance, and three types of route control messages, i.e. Route Request, Route Reply, and Route Error. When a source node in an ad hoc network attempts to send a packet to a destination but it does not have a route to that destination in its route cache, it initiates a route discovery process by broadcasting a route request packet. This route request packet contains the source node address, the destination node address, a unique sequence number, and an empty route record. Each intermediate node, upon receiving a route request for the first time, will check in its own route cache. If it has no route to the destination, the intermediate node will add its own address to the route record and rebroadcast the route request. If it has a route to the destination in its route cache, the intermediate node will append the cached route to the route record and initiates a route reply back to the source node. The route reply contains the complete route record from the source to the destination. The intermediate node ignores the other route requests to the same destination by examining the sequence number in the incoming route request. If the node receiving the route request is the destination node, it will copy the route record contained in the route request and send a route reply back to the source. In most simulation implementations, the destination node will reply to all the route requests received as DSR is capable of caching multiple paths to a certain destination and the replies from the destination most accurately reflect the up-to-date network topology.

Due to the node movement, the routes discovered may no longer be valid over time. The route maintenance mechanism is accomplished by sending route error packets. When a link is found broken, a route error packet is sent from the node that detects the link failure back to the source node. Each node, upon receiving the route error message, removes from its cache all the routes that contain the broken link [6]. In DSR, each node transmitting the packet is responsible for confirming that the packet has been received by the next hop along the source route. This can be done by either a link layer acknowledgement (as in IEEE 802.11), or a "passive acknowledgement" (in which the first transmitting node confirms the receipt at the second node by overhearing the second node transmitting the packet to the third node), or a DSR-specific software acknowledgement returned by the next hop. Thus, once a route enters the cache, the failure of the route can only be detected when it is actually used to transmit a packet but fails to confirm the receipt by the next hop.

In this paper, we present the effects of path cache and link cache on the performance of DSR in an adhoc network. A path cache is one in which each cache entry is a node list representing an entire path leading to a certain destination, while a link cache has a conventional graph data structure in which each individual link is referred to as a cached data unit. A link cache has the potential to utilize the obtained route information more efficiently. However, without a good link update mechanism, stale links may cause more route errors and consequently severe performance degradation. In this paper, we investigate link and cache update mechanisms using a static and dynamic timeout mechanism. The performance of this cache schemes obtained from simulations will be presented.

Previous studies have shown that a path cached DSR protocol has good performance in less "stressful" situations, i.e. smaller number of nodes in the network or less traffic in the network or low mobility of nodes [1][13][15]. In this paper, we will show that the performance of DSR under heavy traffic load situation can be improved by incorporating a link cache scheme together with a static or dynamic timeout mechanism.

The remainder of the article is organised as follows: In section 2 we discuss the various caching techniques available for enhancing the performance of routing protocols. Section 3 gives an explanation about the proposed algorithm. Section 4 discusses the simulation environment, simulation parameters as well as results for performance of DSR with path cache and link cache. The main conclusions for this paper are summarized in section 5.

2. RELATED WORK

There are many caching techniques available to improve the performance of on demand routing protocols. A few important ones are discusses below.

2.1 Cache Data and Cache Path

The Cache Data [11] scheme considers the cache placement policy at intermediate nodes in the routing path between the source and the destination. In this technique a node caches a passing-by data item locally only when it finds that the data item is popular, i.e., there were many requests for the data item, or it has enough free cache space. To save space for more data, a node does not cache the data if all requests for the data are from the same node. However, it uses cooperative caching protocol among Mobile Hosts (MHs).This technique helps in serving future requests for data by fetching the data from nearby node instead of the server always. In Cache Path [11], mobile nodes cache the data path and use it to redirect future requests to the nearby node which has the data instead of the faraway data center. In MANETs, the network topology is dynamic and thus, the cached path may become invalid due to the movement of MHs. These two caching techniques help in efficient data access in MANETs.

2.2. Neighbor Caching

The concept of Neighbor Caching (NC) [9] is to utilize the cache space of inactive neighbors for caching tasks. The basic operations of NC are as follows: When a node fetches a data from a remote node, it puts the data in its own caching space for reuse. This operation needs to evict the least valuable data from the cache based on a replacement algorithm if the cache is currently full. With this scheme, the data that is to be evicted is stored in the idle neighbor node's storage. In the near future, if the node needs the data again, it requests the data not from the far remote source node but from the near neighbor that keeps the copy of data. The NC scheme utilizes the available cache space of neighbour to improve the caching performance. However, it lacks the efficiency of the Cooperative caching protocol among the MHs.

2.3. Node Caching

This is a novel approach to constrain route request broadcast based on node caching [10]. The intuition used is that the nodes involved in recent data packet forwarding have more reliable information about its neighbors and have better locations (e.g., on the intersection of several data routes) than other MANET nodes. The nodes which are recently involved in data packet forwarding are considered as cache nodes, and only they are used to forward route requests. The modified route request uses a fixed threshold parameter H. The first route request is sent with the small threshold H. When a node N receives the route request, it compares the current time T with the time T (N) when the last data packet through N has been forwarded. If T -H > T (N), then N does not belong to the current node cache and, therefore, N will not propagate the route request. Otherwise, if T -H ≤ T (N), then N is in the node cache and the route request is propagated as usual. Of course, the node cache cannot guarantee existence of paths between all source destination pairs, therefore, if the route request with the small threshold H fails to find a route to destination, then a standard route request (which is not constrained by cache) is generated at the source. This method showed average decrease by 90% in communication overhead as well as average decrease by 63% in the delay, and average increase by 20% in the delivery ratio.

2.4. Group Caching

There are some challenges and issues such as mobility of MHs, power consumption in battery, and limited wireless bandwidth when caching techniques are employed in MANETs for data communication. Due to the movement of MHs, MANETs may be partitioned into many independent networks. Hence, the requester cannot retrieve the desired data from the remote server (data source) in another network. The entire data accessibility will be reduced. Also, the caching node may be disconnected from the network for saving power. Thus, the cached data in a MH may not be retrieved by other MHs thus reducing the usefulness of the cache. The MHs also decides the caching policy according to the caching status of other MHs. However, the existing cooperative caching schemes in a MANET lack an efficient protocol among the MHs to exchange their localized caching status for caching tasks. This can be overcome by using Group Caching (GC) [12] which maintains localized caching status of 1-hop neighbors for performing the tasks of data discovery, caching placement, and caching replacement when a data request is received in a MH. Each MH and its 1-hop neighbors form a group by using the "Hello" message mechanism. In order to utilize the cache space of each MH in a group, the MHs periodically send their caching status in a group. Thus, when caching placement and replacement need to be performed, the MH selects an appropriate group member to execute the caching task in the group; this reduces redundancy of cached data objects.

2.5. Caching Structure and Cache Timeout

In developing a caching strategy for an on-demand routing protocol for wireless adhoc networks, one of the most fundamental design choices that must be made is the type of data structure that can be used to represent the cache. In DSR, the route returned in each ROUTE REPLY that is received by 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. A path cache is very simple to implement and easily 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 (or prefix of a path) that leads to the intended destination node.

Alternatively, a link cache could also be created, in which each individual link in the routes returned in ROUTE REPLY packets are added to a unified graph data structure. To find the current best path through the graph to the destination node in a link cache, a node must use a much more complex graph search algorithm, such as the well-known Dijkstra's shortest path algorithm. Such an algorithm is more difficult to implement and may require significantly more CPU time to execute. However, a path cache data structure cannot effectively utilize all of the potential information that a node might learn about the state of the network. But in a 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 [17].

Cache timeout policy introduces a number of design choices to consider in a caching strategy. Because a path cache generally has a mechanism for removing entries through a capacity limit, there is no need to implement a timeout for path caches. For link caches, the timeout on each link in the cache must be used and they may be either static or dynamic.

For a static timeout, each link is removed from the cache after a specified amount of time has elapsed since the link was added to the cache. For an adaptive timeout, a node adding a link to its cache attempts to determine a suitable timeout after which the link will be deleted from the cache and this timeout value should be based on properties of the link or the nodes that are the endpoints of the link. Finally it is possible to allow a link that is being used to not expire by increasing its timeout as soon as it is used.

Replies from caches provide dual performance advantages. First, they reduce route discovery latency. Second, without replies from caches the route query flood will reach all nodes in the network (request storm). 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 error packets and can potentially pollute other caches when a packet with a stale route is forwarded or snooped on [17].

The paper [17] has presented an analysis of the effects of different design choices in caching strategies for on-demand routing protocols in wireless ad hoc networks, dividing the problem into choices of cache structure, cache capacity, and cache timeout. The analysis is based on the Dynamic Source Routing protocol which operates entirely on-demand. Using detailed simulations it has been shown that the performance of adaptive caches is comparable to that of well-tuned static caches, and that by utilizing a cache data structure based on a graph representation of individual links, rather than based on complete paths through the network, the routing protocol will be able to make use of the potential information available to it in a better way. It is also shown that caches of unlimited capacity or with no cache timeout perform substantially worse than caches with reasonable capacity or timeout limits. These conclusions were drawn using Random Waypoint mobility model and several other mobility models.

2.6. Link timeout mechanisms

As nodes in an adhoc network are capable of moving independently, a link has a limited lifetime. Current existing link is no longer valid when the two end nodes move out of the transmission range of each other. Due to the on-demand nature of the routing protocol, the link status will not be updated until they are used. However, using an actually broken route will cause a number of route errors to be generated and potential packet loss. So taking advantage of using active links individually has to be combined with a mechanism that removes stale links timely to avoid route errors. A natural choice is to combine a timeout policy to the link cache such that each link is given an appropriate lifetime when it enters the link cache, and is removed when its lifetime is running out. The lifetime estimation becomes a critical issue for such a link cache scheme. The lifetime assigned to the link should properly reflect the expected value of its real lifetime. If this value assigned is too small, links expire too soon before they really break, resulting in more costly route discoveries. On the other hand, if the value is too large, links break early before the timers expire, more route errors will be caused, which would also degrade the overall performance of the protocol [16].

The paper [16] shows the results of simulation on DSR protocol using Random Waypoint mobility model. The performance of the link cache strategy is compared with the conventional "path cache" DSR. The results show that when the number of traffic sources increases, the proposed "link cache" DSR outperforms the "path cache" DSR, with wider performance gap with increasing load. Previous studies have shown that a path cache DSR protocol has good performance in less "stressful" situations, i.e. smaller number of nodes and lower load and/or mobility. The simulation results show that with a link cache, the performance of DSR under more stressful situation can be improved. As the cache organization is a local implementation decision at each node, all the protocol control messages and route discovery and maintenance mechanisms remain the same. It is also suggested that by switching between the two types of cache organizations dynamically in response to the network load condition might provide a good way to improve the overall performance of the DSR protocol.

3. PROPOSED WORK

Considering all the previous work that has been done to improve the performance of routing protocols using caching techniques we have proposed a new technique where a time out is considered for cache as well as link entries made in the cache .We have used a path cache and tried to analyze the performance of DSR in a Dynamic environment where the node moves in a random direction with random velocity.

We have then used a link cache and given a time out for the entire cache as well as for individual links entries in the link cache. We have analyzed the performance of link cache by using different static time outs for cache as well as link and tried to find out an optimum value which can be used for the network selected. We have also analyzed the performance of link cache by using a dynamic time out for both cache and link. This dynamic time out technique is proposed by us. We are calculating the dynamic time out based on the mobility of the node. Once the node is ready to move the random movement time is added to the pause time selected and a dynamic time out is computed and used. This helps in keeping the cache entry valid only for a time period for which the node will remain static at a certain place.

So our contributions can be listed as

1. Proposing a new environment for simulation so that the simulation can be carried out in an environment close to that of real world scenario.
2. Comparing the performance of DSR with path cache and link cache in the new proposed environment.
3. Method of calculating an ideal static time out for cache and link for any given network and any traffic conditions.
4. Proposing a new method for computing dynamic time out for cache and link based on the mobility of the node.

5. Analyzing the performance of DSR when link cache is considered and cache timeout and link timeout is taken dynamically.

4. SIMULATION

The simulation of link cache DSR protocol was implemented within the GloMoSim [4].Glomosim uses supporting software called Parsec [8]. The link layer model is the Distributed Coordination Function (DCF) of the IEEE 802.11 wireless LAN standard. The network simulated consisted of 50 nodes. The simulation area was 700 × 700 m^2. Each simulation was executed for 15 minutes.

Initial locations of the nodes are obtained using a uniform distribution. We assumed that each node moves independently with a random speed later. With the Random Waypoint Mobility [14] model, a node randomly selects a destination from the physical terrain and moves in the direction of the destination with a uniform speed chosen as an average of the minimal and maximal speed. After it reaches its destination, the node stays there for a pause time and then moves again. In our simulation, we have modified the random waypoint mobility model so that the node moves in random direction with random mobility and then stays there for the selected pause time till next random movement. This modification was done so that the movement matches the real world scenario .The traffic used in this simulation was CBR. The source–destination pairs were chosen randomly among the 50 nodes. The number of communication sessions was varied to change the offered load in the network.

We have evaluated the performance of the DSR protocol with link cache taking static lifetime assignments scheme. With static lifetime assignments, we have executed multiple simulations for different values of cache time out and link time out assumed as (3, 1), (12, 2), (50, 5), (100, 10).sec. For each lifetime value, we execute multiple simulation runs with various traffic load conditions (10, 20, 30, 40 sources) and various node mobility levels (pause time = 0, 100, 200, 300, 400, 500, 600, 700, 800, 900 s). For comparison purpose, we also evaluate the conventional path cached DSR protocol. The traffic load and the density of the network are identical across different variations of DSR protocol.

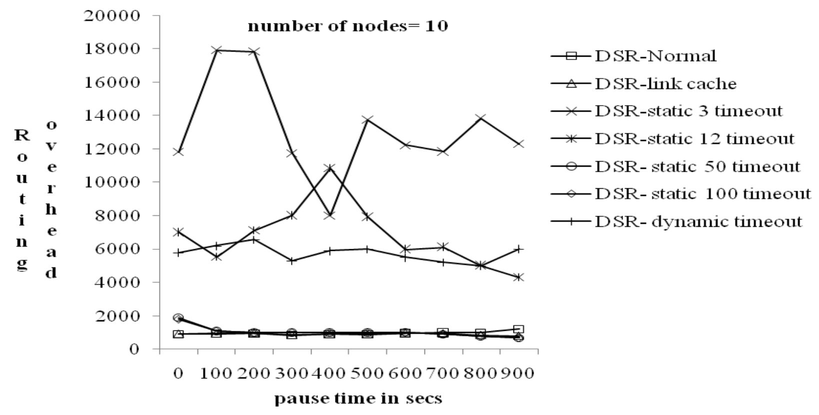

Figure 3. Comparison of routing overhead for a network with 10 nodes

We first examined the effects of different cache schemes on the routing overhead generated. The routing overhead is calculated as the number of control packets transmitted by the protocol. There are three types of routing packets, e.g., route requests, route replies, and route errors. When the lifetime is within an appropriate range, the overall control messages are quite balanced. The graphs in Fig 3 give the comparison of the overall routing overhead generated by each variation of the protocol. We observe that, in general, with an appropriate static link lifetime, the number of control packets used by the link cache scheme is less than or comparable to that used by the path cache scheme. However, without an appropriate timeout mechanism the routing overheads are quite high. The graph shows that when the static time out period assumed is less than the value suitable for the selected network and selected traffic conditions setup in the network the routing overhead increases. This is because the known routes expire even before the links actually break resulting in new Route requests being generated. Whereas if the static time out assumed is larger than the value suitable for the selected network and selected traffic conditions setup in the network the routing overhead increases . This is because the routes are valid even after the links actually break resulting in generation of route error messages caused by the use of stale routes which in turn generates new Route requests. But assuming an appropriate static time out for an Adhoc network where the number of nodes and the traffic conditions keep changing is really difficult as in an real network trial and error cannot be adopted with changing traffic conditions. Graph shows that dynamic time out proposed results in more overhead compared to path cache used by normal DSR as we have plotted the graph considering 10 nodes with less traffic in the network and as expected standard DSR works well in such conditions.

Similar comparisons as shown in Fig 3 are done in Fig 4, but the number of nodes used is 40 and heavy traffic flow is set up in the network for simulation. This graph shows that with static time out, the assumed value should be appropriate to reduce the routing overhead. If inappropriate values are used then the routing overhead increases for the reasons explained for Fig 3.As expected the routing overhead for dynamic time out is less than that of normal DSR as shown in Fig 4 as we have used a network with heavy traffic conditions. Under heavy traffic conditions the link cache with dynamic timeout is updated with latest routes as a result of which the routing overhead decreases.

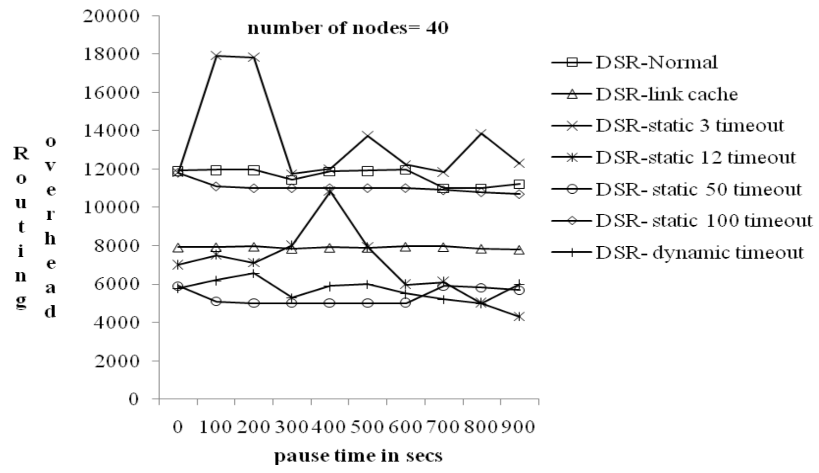

Figure 4. Comparison of routing overhead for a network with 40 nodes.

The above observations indicate that a link cache scheme with appropriate static lifetime estimation can make use of the available route information more efficiently, but it is difficult to fix an appropriate value for a dynamic network. When the network traffic is high, more route replies and more promiscuously overheard route information can be collected. The link cache can maintain more accurate and more up-to-date information of which the link cache scheme can take advantage to reduce the route discovery processes. The reduced routing overhead, especially in heavy traffic situation, contributes to the improved performance of the protocol when dynamic timeout mechanism is used.

Using the next set of graphs we evaluate the application level performance metric, packet delivery ratio (the end-to-end throughput). The packet delivery ratio is the fraction of packets that are received at corresponding destination over those sent at the source. It is the most important metric to evaluate the performance of an ad hoc routing protocol. There are two major situations that a packet may drop. One is due to the stale routes. A stale route in the cache may direct a packet to an actually broken link. When a node fails to forward a packet over a broken link, it will try to salvage the packet by looking up in its own route cache for an alternative route to the packet's destination. If the alternative path does not exist or the packet has already been salvaged before, the packet is dropped.

The other type of packet loss is due to the heavy collisions in the MAC layer that cause a packet drop after failing a certain number of attempts to transmit the packet. A route discovery process can cause a large number of route requests and route replies generated within a short time, thus causing increased interference to data traffic at MAC layer. When the traffic load is high, the packet loss caused by collisions becomes more severe. To reduce the packet loss due to this reason, a large lifetime value is preferred because a large lifetime value could minimize the number of route discovery performed. The graphs in Fig 5,6,7,8 summarize the performance of packet delivery ratio under various network traffic loads and node mobility levels. From these graphs we can observe that the PDR of normal DSR with path cache as well as link cache with inappropriate static time out decreases as the traffic increases. But the performance of link cache with dynamic timeout improves a s traffic conditions in the network increases.

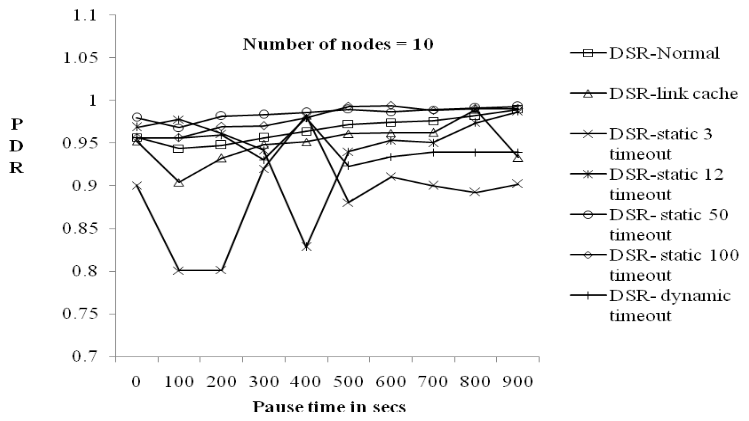

Figure 5. Comparison of PDR for a network with 10 nodes

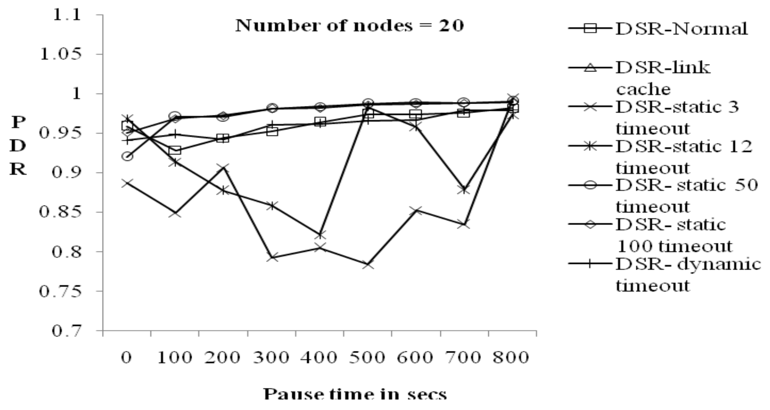

Figure 6. . Comparison of PDR for a network with 20 nodes

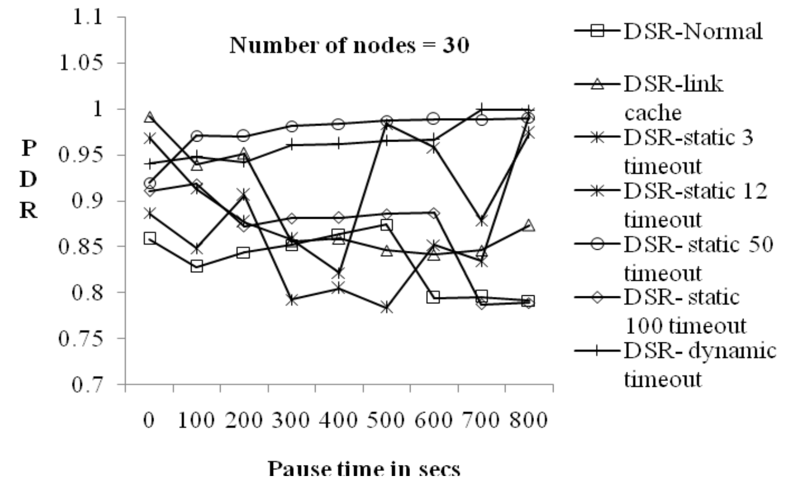

Figure 7. Comparison of PDR for a network with 30 nodes

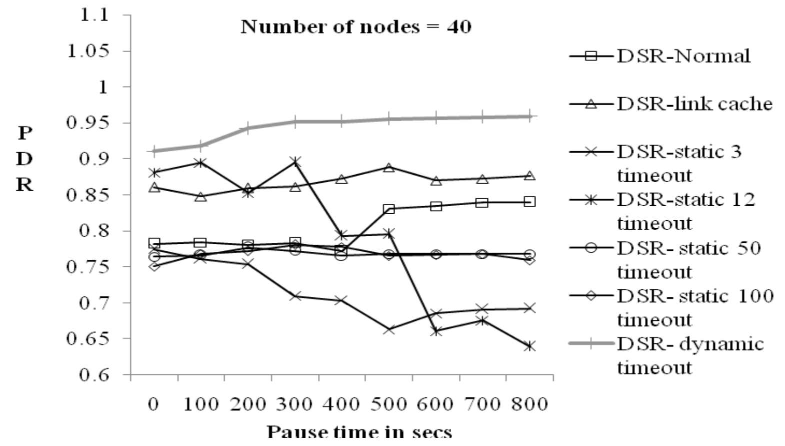

Figure 8. Comparison of PDR for a network with 40 nodes

With the above two analysis we can say that, in general, the link cache scheme performs better than the path cache scheme. A link cache does not always show significant advantage over the path cache, and is dependent on the traffic load. A link cache with dynamic time out performs efficiently under heavy traffic conditions

5. CONCLUSIONS

Previous studies have shown that a path cache DSR protocol has good performance in less “stressful” situations, i.e. smaller number of nodes and lower load and/or mobility. In this paper, we have analysed the effects of the route cache schemes on the performance of on-demand routing protocols in ad hoc networks. We base our simulation on DSR, the well-evaluated on-demand routing protocol in ad hoc networks. We have studied the “link cache” performance with static lifetime assignments. The results indicate that an appropriate timeout mechanism is critical on the performance of such a link cache scheme. A link cache with an appropriate static timeout mechanism is difficult to assume, but if rightly assumed it can make use of the available route information more efficiently and improve the protocol performance. However, without an appropriate static timeout mechanism, a link cache may cause dramatically increased route errors and consequently severe performance degradation. The performance of the proposed dynamic cache strategy was compared with the conventional "path cache" DSR and it was shown that this strategy works well under heavy traffic conditions in the network

All our simulation results show that with a link cache, the performance of DSR under more stressful situation can be improved. As the cache organization is a local implementation decision at each node, all the protocol control messages and route discovery and maintenance mechanisms remain the same, by switching between the two types of cache organizations dynamically in response to the network load condition might provide a good way to improve the overall performance of the DSR protocol.

REFERENCES

[1] Asis Nasipuri, *Mobile Adhoc networks*, Department of Electrical & Computer Engineering,The University of North Carolina at Charlotte, http://www.ece.uncc.edu/~anasipur/pubs/adhoc.pdf.

[2] Internet Engineering Task Force MANET Working Group.Mobile Adhoc Networks (Manet) Charter. Available at http://www.ietf.org/html.charters/manet-charter.html.

[3] C. E. Perkins, *Ad Hoc Networking*, Addison-Wesley, 2001

[4] UCLA, *Glomosim: A scalable simulation environment for Wireless and Wired Network systems*, http://pcl.cs.ucla.edu/projects/glomosim.

[5] C.E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," *Computer Communication Review (*October 1994) 234–244.

[6] D. Johnson, Y.Hu and D.Maltz, "The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4,"February 2007 http://www.ietf.org/ rfc/rfc4726.txt.

[7] C. Perkins and S. Das, "Ad Hoc On Demand Distance Vector (AODV) Routing,"IETF, Internet Draft, draft-ietf-manet-aodv-13, RFC 3561, February 2003.

[8] *Parsec: A parallel simulation environment for complex systems*, Computer, Vol. 29(10), pp. 77-85, Oct. 1998.

[9] Joonho Cho, Seungtaek Oh, Jaemyoung Kim, Hyeong Ho Lee, and Joonwon Lee, "Neighbor caching in multi-hop wireless ad hoc networks," *IEEE Communications Letters*, Volume 7, Issue Nov. 2003 , Page(s):525 – 527.

[10] Sunsook Jung, Nisar Hundewale,and Alex Zelikovsky, "Node Caching Enhancement of Reactive Ad Hoc Routing Protocols," *IEEE Wireless Communications and Networking Conference*, 2005 .

[11] LiangZhong Yin and Guohong Cao, "Supporting cooperative caching in ad hoc networks," *IEEE Transactions on Mobile Computing*, Volume 5, Issue 1, Jan. 2006 Page(s):77-89.

[12] Yi-Wei Ting and Yeim-Kuan Chang,"A Novel Cooperative Caching Scheme for Wireless Ad Hoc Networks: GroupCaching," *International Conference on Networking Architecture and storage*, NAS 2007.

[13] T. Clausen, P. Jacquet, A. Laouiti, P. Minet, P. Muhlethaler, A.Qayyum and, L.Viennot,"Optimized Link State Routing Protocol (OLSR) , " Internet Draft, draft-ietf-manetolsr-05.txt, Nov. 2001.

[14] J. Broch, D. Maltz, D. Johnson, Y.-C. Hu and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocol,"*ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, October 1998, pp. 85–97.

[15] S.R. Das et al., "Comparative performance evaluation of routing protocols for mobile ad hoc networks," *International Conference on Computer Communication and Networks (IC3N)*, October 1998, pp. 153–161.

[16] Wenjing Lou and Yuguang Fang , "Predictive Caching Strategy for On-Demand Routing Protocols in Wireless Ad Hoc Networks , " Wireless Networks Laboratory (WINET), Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611, USA.

[17] Yih-Chun Hu and David B. Johnson, " Caching Strategies in On-Demand Routing Protocols for Wireless Ad Hoc Networks, " *Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking (MobiCom 2000)*, August 6–11, 2000, Boston, MA, USA.

Authors

Shobha.K.R received M.E degree in Digital communication from Bangalore University, Karnataka, India and is currently working towards the Ph.D. She is currently working as Assistant Professor with the department of Telecommunication Engineering, M.S. Ramaiah Institute of Technology, Bangalore. She had served as teaching faculty at BMSCE, Bangalore for 5 years up to 1999. Her research areas include Routing protocols in Mobile Adhoc Networks and Wireless Networks.

Dr.K.RajaniKanth received M.E in Automation and Ph. D degrees from Indian Institute of Science, Bangalore, India. Areas of interest are software engineering, Object Technology, Embedded Systems. Currently working as Professor and Principal at M S Ramaiah Institute of Technology, Bangalore.