Intelligent Caching in on-demand Routing Protocol for Mobile Adhoc Networks

Shobha.K.R., and K. Rajanikanth

***Abstract*—**An 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 routes previously discovered. This paper presents an analysis of the effect of intelligent caching in a non clustered network, using on-demand routing protocols in wireless ad hoc networks. The analysis carried out is based on the Dynamic Source Routing protocol (DSR), which operates entirely on-demand. DSR uses the cache in every node to save the paths that are learnt during route discovery procedure. In this implementation, caching these paths only at intermediate nodes and using the paths from these caches when required is tried. This technique helps in storing more number of routes that are learnt without erasing the entries in the cache, to store a new route that is learnt.

The simulation results on DSR have shown that this technique drastically increases the available memory for caching the routes discovered without affecting the performance of the DSR routing protocol in any way, except for a small increase in end to end delay.

***Keywords*—**Caching, DSR, on demand routing, MANET.

I. Introduction

CACHING is an important part of any on-demand routing protocol for wireless ad hoc networks. In a Mobile ad hoc network (MANETS) [1], [2], [3] all nodes cooperate in order to dynamically establish and maintain routing in the network, forwarding packets for each other to allow communication between nodes not directly within wireless transmission range. Rather than using the periodic or background exchange of routing information common in most routing protocols, an on-demand routing protocol 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, an on-demand routing protocol must cache routes previously discovered. Such caching then introduces the problem of proper strategies for managing the structure and contents of this cache, as nodes in the network move in and out of wireless transmission range of one another, possibly invalidating some cached routing information.

Several routing protocols for wireless ad hoc networks have used on-demand mechanisms, including Temporally-ordered routing algorithm (TORA) [7], Dynamic Source Routing protocol (DSR) [4], Ad Hoc On Demand Distance Vector (AODV) [5], Zone routing protocol (ZRP) [6], and Location-Aided Routing (LAR) [8]. For example, in the Dynamic Source Routing protocol [4] in its simplest form, when some node S originates a data packet destined for a node D to which S does not currently know a route, S initiates a new Route Discovery by beginning a flood of request packets through the network. When a copy of this request packet reaches either D or another node that has a cached route to D, this node then returns to S the route discovered by this request. Performing such a Route Discovery can be an expensive operation, since it may cause a large number of request packets to be transmitted, and also add latency to the subsequent delivery of the data packet that initiated it. But this Route Discovery may also result in the collection of a large amount of information about the current state of the network that may be useful in future routing decisions. In particular, S may receive a number of replies in response to its Route Discovery flood, each of which returns information about a route to D through a different portion of the network; a node may also learn the information about the state of the network by eavesdropping on the Route Discovery packets from other nodes. By caching and making effective use of this collected network state information, the amortized cost of Route Discoveries can be reduced and the overall performance of the network can be significantly improved.

In this paper, effects of intelligent caching algorithm on the performance of DSR were analyzed by changing the threshold distance at which the paths are cached.

II. Overview of the DSR Protocol

The Dynamic Source Routing protocol [4] has been used in this paper to illustrate the effects of intelligent caching strategies in on demand routing protocols, since DSR operates *entirely* on-demand and thus clearly shows the caching behavior. DSR is composed of two mechanisms that work together to allow the discovery and maintenance of source routes in the ad hoc network. *Route Discovery* is the mechanism by which a node S wishing to send a packet to a

Shobha.K.R is with Telecommunication Engineering Department of M.S Ramaiah Institute of Technology, Bangalore-560054, Karnataka, India (phone: 91-80-23600822; fax: 91-80-23603124; e-mail: shobha_shankar@yahoo.com).

K. Rajanikath is with Information Science Department of M.S Ramaiah Institute of Technology, Bangalore-560054, Karnataka, India (e-mail: principal@msrit.edu).

destination node D obtains a source route to D. Route Discovery is used only when S attempts to send a packet to D and does not already know a route to D. *Route Maintenance* is the mechanism by which node S, while using a source route to D, is able to detect when the network topology has changed such that it can no longer use its route to D because a link along the route no longer works. When Route Maintenance indicates a source route is broken, S can attempt to use any other route it happens to know to D, or can invoke Route Discovery again to find a new route for subsequent packets that it sends. Route Maintenance is used only when S is actually sending packets to D. This section describes the basic operation of Route Discovery and Route Maintenance, although a number of optimizations to this basic operation exist [4, 9] that are not discussed here due to space limitations.

To initiate a new Route Discovery for a node D (the target of the Route Discovery), S transmits a ROUTE REQUEST packet, which is received by other nodes located within direct wireless transmission range of S. Each node that receives the ROUTE REQUEST packet appends its own address to a record in the packet and rebroadcasts it to its neighbors, unless it has recently seen another copy of the ROUTE REQUEST for this Route Discovery or it finds that its address was already listed in the route record in the packet. The forwarding of the ROUTE REQUEST terminates when it reaches node D; this node then returns a ROUTE REPLY packet to S, giving a copy of the accumulated route record from the ROUTE REQUEST, indicating the path that the ROUTE REQUEST traveled to reach D. The forwarding of the ROUTE REQUEST also terminates when it reaches a node that has in its cache a route to D; this node then returns a ROUTE REPLY packet to S, giving the route as a concatenation of the accumulated route record from the ROUTE REQUEST together with this node's own cached route to D. The returned source route from the ROUTE REPLY is cached by S for use in sending subsequent data packets.

Route Maintenance is performed by each node that originates or forwards a data packet along a source route. Each such node is responsible for confirming that the packet has been received by the next hop along the source route given in the packet; the packet is retransmitted (up to a maximum number of attempts) until this confirmation of receipt is received. This confirmation may be provided at no cost to DSR, either as an existing standard part of the MAC protocol in use (such as the link-level acknowledgement frame defined by IEEE 802.11), or by a passive acknowledgement. If neither of these confirmation mechanisms are available, the node transmitting the packet may set a bit in the packet header to request a DSR-specific software acknowledgement be returned by the next hop. If this confirmation is not received after some maximum number of local retransmission attempts, this node returns to the original sender of the packet a ROUTE ERROR message, identifying the link over which the packet could not be successfully transmitted. When receiving the ROUTE ERROR, this original sending node removes this broken link from its cache. In addition to returning a ROUTE ERROR message, this node may also attempt to salvage the original packet [10], [11], [20] if it has a route to the intended destination of the packet in its own cache. If so, the node replaces the original source route on the packet with the route from its cache and forwards the packet along that route; otherwise, the node discards the packet since no correct route is available.

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 or that they can overhear through optionally operating their network interface hardware in promiscuous mode; 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.

III. Related Work

A. CacheData and CachePath

The CacheData [18] scheme considers the cache placement policy at intermediate nodes in the routing path between the source and the destination. The node caches a passing-by data item locally 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. Since CacheData needs extra space to save the data, it should be used prudently. A conservative rule is proposed as follow: 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). Each MH does not independently perform the caching tasks such as placement and replacement. CachePath [18] is also proposed for redirecting the requests to the caching node. In MANETs, the network topology changes fast and thus, the cached path may become invalid due to the movement of MHs.

B. Neighbor Caching

The concept of Neighbor Caching (NC) [16] 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. 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 neighbor to improve the caching performance. However, it lacks the efficiency of the cooperative caching protocol among the MHs.

C. Node Caching

This is a novel approach to constrain route request broadcast based on node caching [17]. 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.

D. 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 an MH may not be retrieved by other MHs and then usefulness of the cache is reduced.

The MHs also decide 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 paper, proposes a novel cooperative caching scheme called Group Caching (GC) [22] 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 the appropriate group member to execute the caching task in the group; this reduces redundancy of cached data objects.

IV. Intelligent Caching

Intelligent caching is a technique in which, a node not only saves the path discovered during route discovery for itself but also for others who are located close to it. This technique reduces the number of route request packets unnecessarily circulating in the network, when the path it requires is present in its neighborhood. This concept is tried on DSR for saving the data paths discovered by a node that has data to transmit in the network. In DSR each and every node overhears the route request and route reply packet circulating in the network and learns the path to different nodes in the network. This route information is replicated in multiple nodes occupying large amount of cache. In order to make efficient utilization of cache, these paths are stored at nodes called cache nodes which are located at intermediate points between source and destination. A threshold value is selected to determine the selection of caching nodes. The path identified is first cached at the node which is at the predetermined threshold level from the destination node. Then the current cached node is taken as reference to compute the nodes at which the path will be cached next, till the source node is found. The amount of cache available in the non caching nodes (i.e nodes between the two caching nodes) is made known to the selected caching nodes so that they can save the paths in the neighborhood if their cache memory is full.

Consider node N1 (source) wants to send some data to the node N11 (destination) in the network shown in Fig 1. The source first checks for a route to the destination in its route cache. If it finds a route then it sends the data directly to the destination N11. But when it does not have a route to the destination, it broadcasts a route request to all the nearby nodes which in turn forwards it to the destination N11. While broadcasting, if any of the intermediate nodes N2, N3, N4, N5, N6, N7, and N10 have the route to the destination, they concatenate the route, to the route from the source and send the route reply to the source node. When no intermediate nodes have a route to the destination, each of them just append their IP addresses to the packet header before forwarding the route request. Now before forwarding the route request, the node will enter the route into its route cache. In DSR all the intermediate nodes cache the path. In this implementation the number of nodes caching the path is limited based on a threshold value TH.

The performance of intelligent cache is affected by the threshold value TH as a path is only cached by a node when its HOPCOUNT and SEGLEFT values are greater than TH.

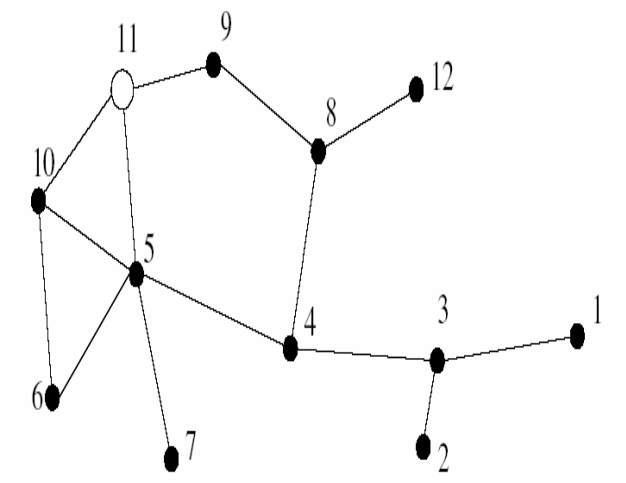

Fig. 1 Ad hoc network

A small TH means more intermediate nodes cache the path, but too many nodes caching may increase the delay because the cached paths are not very reliable and also use up a lot of memory which is expensive. A large TH means only some nodes cache the path and hence save memory. However, if TH is too large, many nodes do not cache the path because of the high threshold which might increase the overhead and delay. So a careful selection of threshold helps in efficient utilization of the cache.

The performance of the system was analyzed for different values of threshold and the results are as presented below.

V. Simulation Results

The simulations were performed using Glomosim [19]. The mobility scenarios were randomly generated using random waypoint model. The distributed coordination function (DCF) of IEEE 802.11 for wireless LANs used, was the MAC layer protocol.

In this simulation, 300 nodes were used in a 1000x1000 meter rectangular region for 300 seconds simulation time. Initial locations of the nodes were obtained using a uniform distribution. It was assumed that each node moves independently with the same average speed. With the random waypoint mobility model, a node randomly selects a destination from the physical terrain and moves in the direction of the destination with a uniform speed chosen between the minimal and maximal speed. After it reaches its destination, the node stays there for a *pause time* and then moves again. In this simulation, the minimal speed was chosen to be 2m/s and maximal speed was 15m/s and pause time was varied from 2 to15 seconds. The simulated traffic was constant bit rate (CBR).

The analysis of the variations in the overhead was done by varying the density of the network at different mobilities (i.e. pause time) of 2s, 8s and 15s.

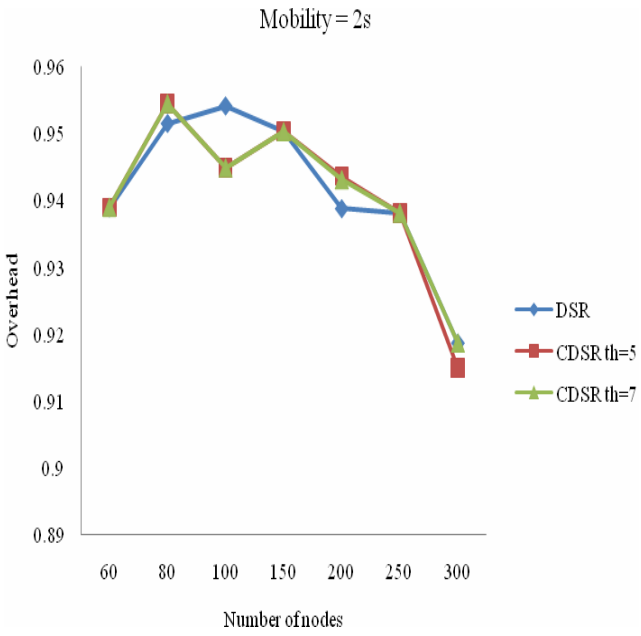

Fig. 2 Number of nodes v/s overhead

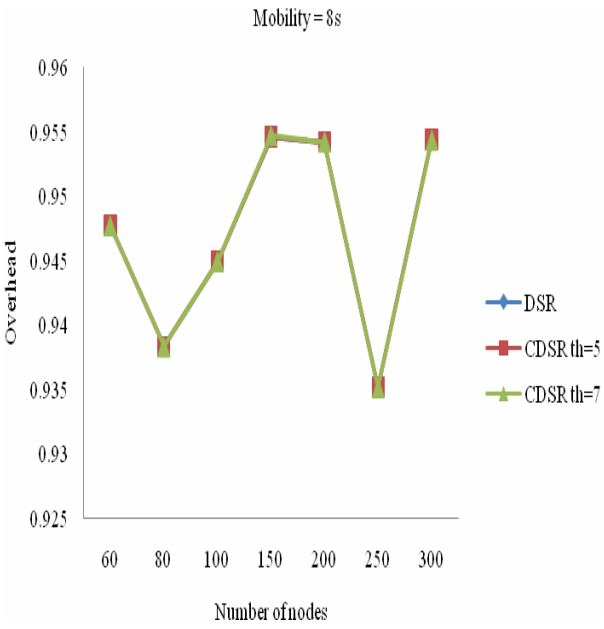

Fig. 3 Number of nodes v/s overhead

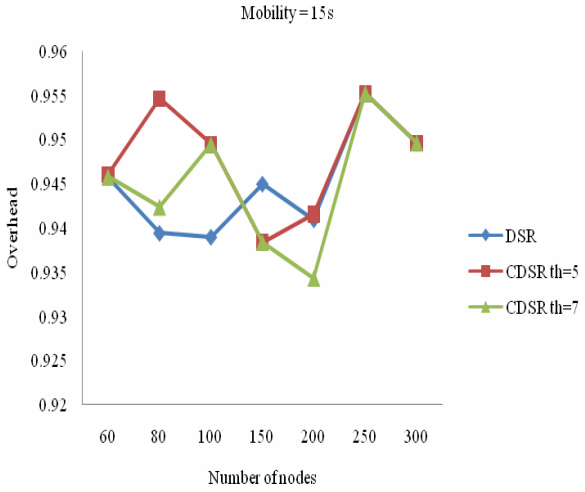

Fig. 4 Number of nodes v/s overhead

From the above three graphs for number of nodes v/s overhead shown in Figs. 2,3 and 4 it can be seen that the technique implemented is suitable for larger number of nodes (>100).For lesser number of nodes, the overhead increases as the mobility and threshold values are increased. This is because the nodes are located at far off distances and it takes time to retrieve the information from the caching nodes. For larger number of nodes in the same terrain dimension, the overhead has improved (decreased or remained the same), therefore this technique is not deteriorating the performance of the system for higher number of nodes. Practically ad hoc networks are implemented with larger number of nodes(>100). Hence the implementation of this technique reduces the frequent deletion of paths from cache and makes space for new routes in the neighborhood cache.

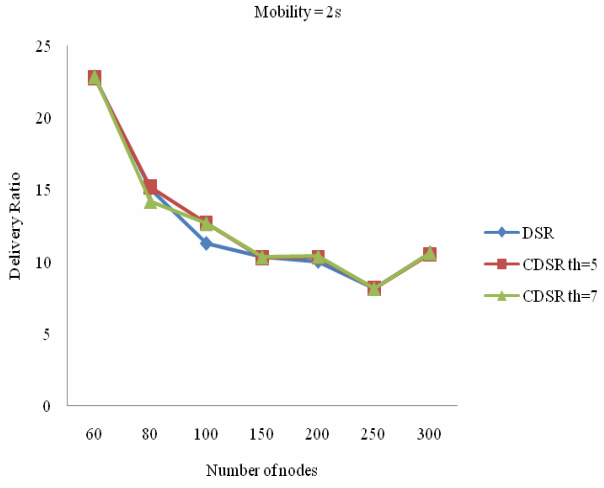

Fig. 5 Number of nodes v/s delivery ratio

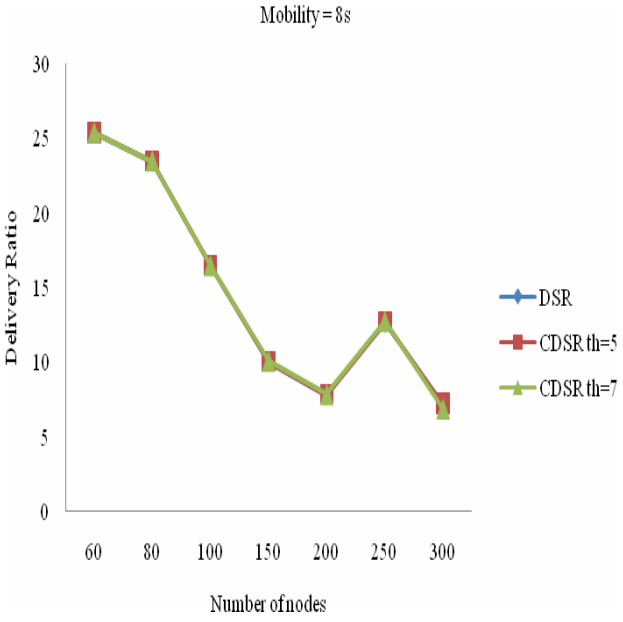

Fig. 6 Number of nodes v/s delivery ratio

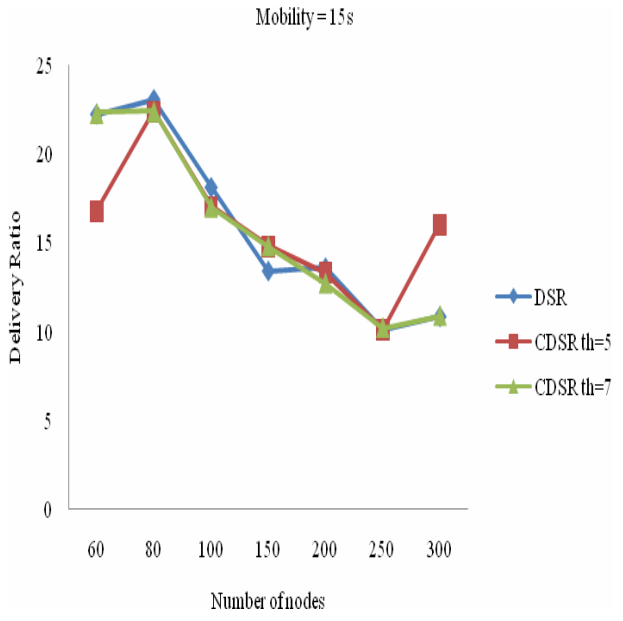

Fig. 7 Number of nodes v/s delivery ratio

From the above three graphs for number of nodes v/s delivery ratio in Figs. 5,6 and 7 it can be seen that the technique implemented is suitable for larger number of nodes (>100) since the delivery ratio has improved (increased or remained the same) in this region. Hence by reducing the amount of caching, this technique has not deteriorated the performance of the system.

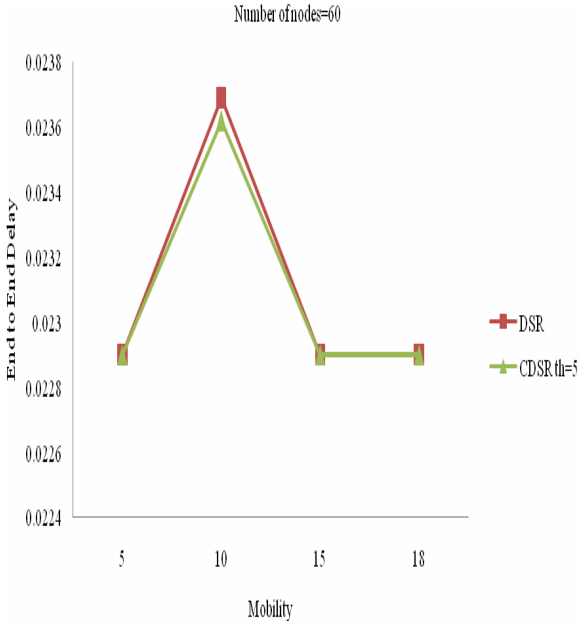

Fig. 8 Mobility v/s end to end delay

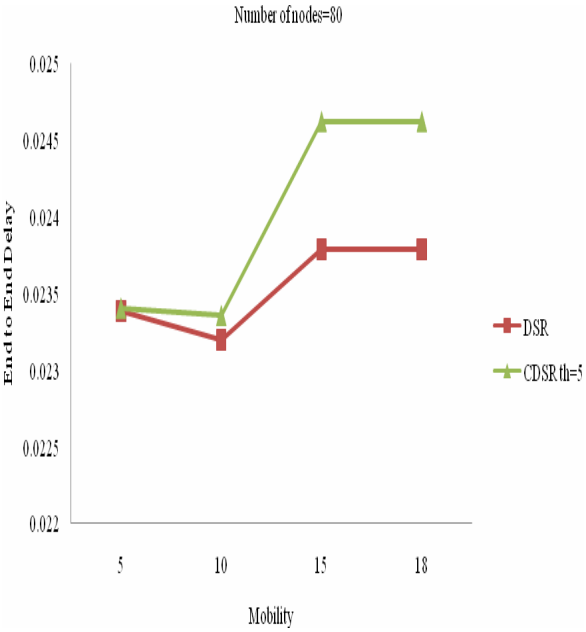

Fig. 9 Mobility v/s end to end delay

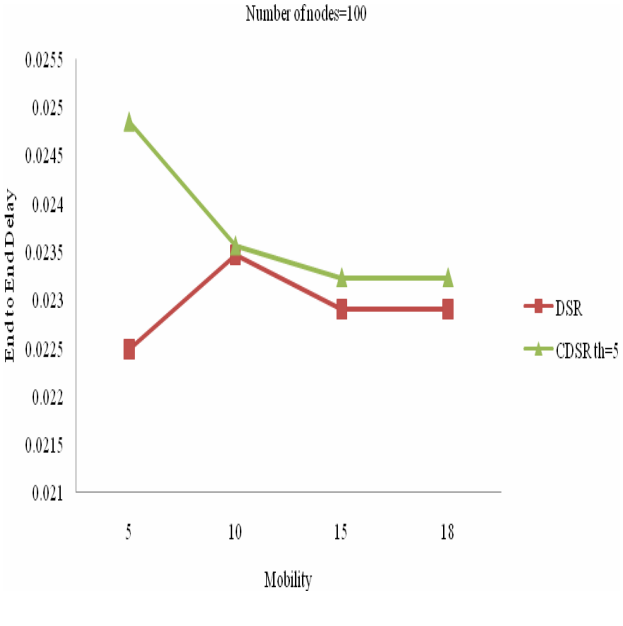

Fig. 10 Mobility v/s end to end delay

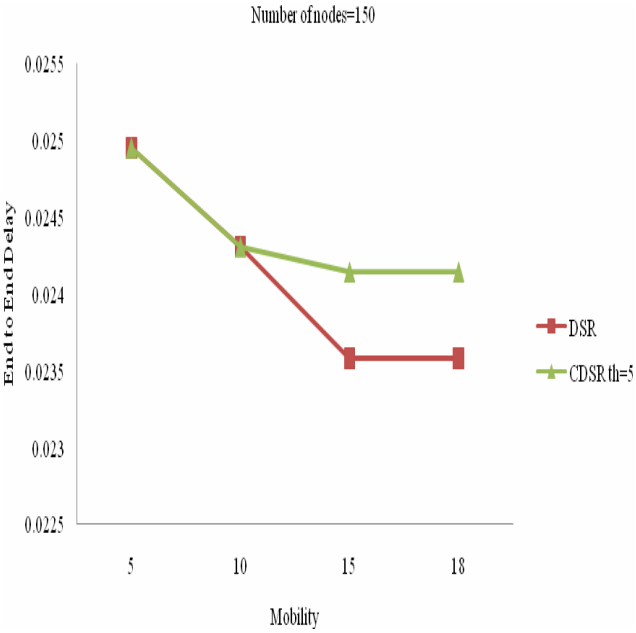

Fig. 11 Mobility v/s end to end delay

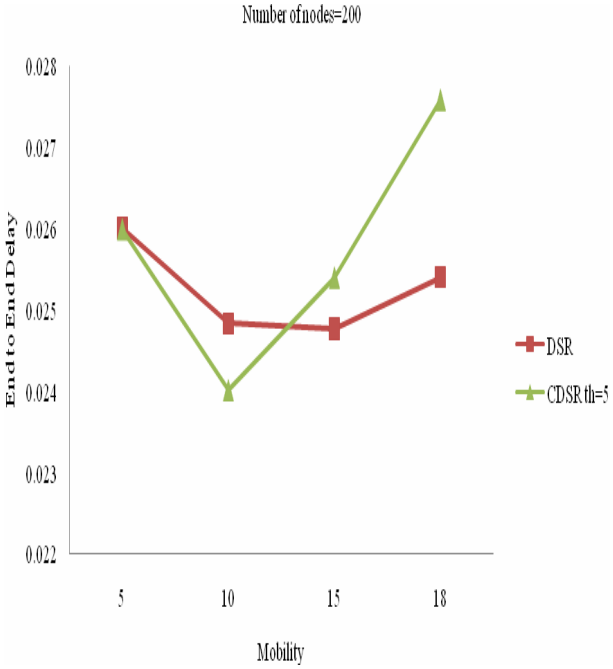

Fig. 12 Mobility v/s end to end delay

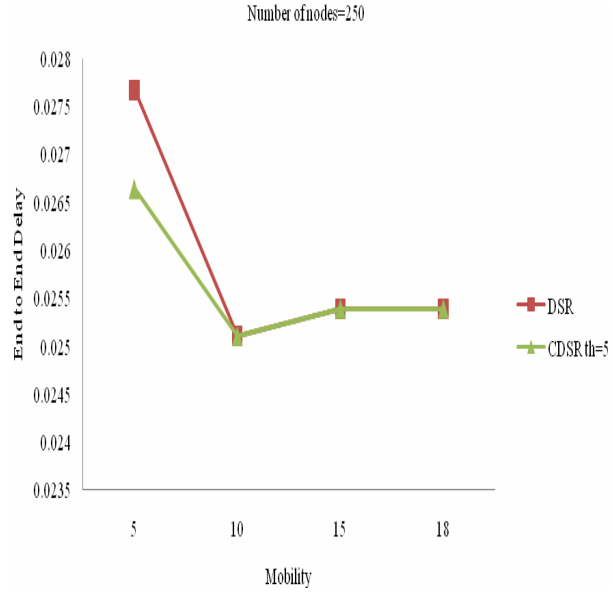

Fig. 13 Mobility v/s end to end delay

From the above graphs for Mobility v/s end to end delay in Fig 8, 9, 10, 11,12,13,14 it can be seen that as expected, the end to end delay in this technique has slightly increased. This is because; lesser number of intermediate nodes is caching the paths that are discovered during route discovery. When a node has data to transmit, it needs some time to get this path from the neighborhood as a result of which route discovery will take more time, increasing the total end to end delay; but this increase is not very significant. So this technique is able to utilize the cache efficiently, reduce the overhead slightly without affecting the end to end delay drastically.

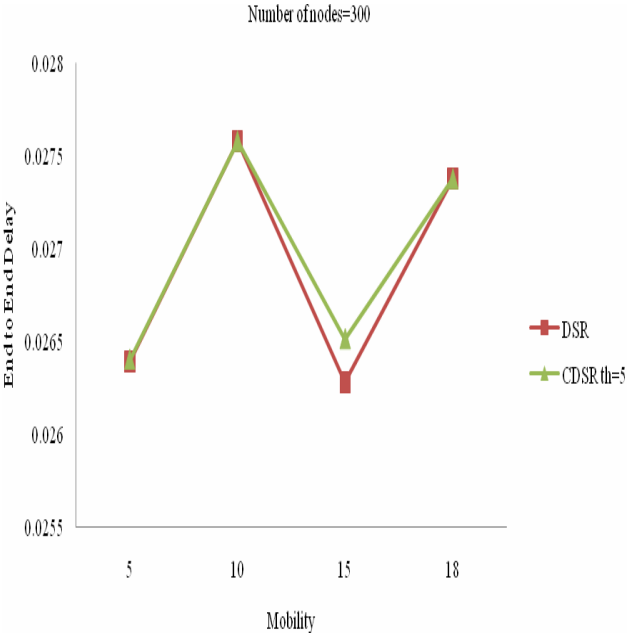

Fig. 14 Mobility v/s end to end delay

VI. Conclusion

A number of on-demand routing protocols for wireless ad hoc networks have been proposed, including TORA [7], DSR [4], AODV [5], ZRP [6], and LAR [8], and earlier detailed simulation work has shown that such protocols can have excellent performance [19]. One key to achieving this type of performance is the design of an appropriate caching strategy for the protocol, which can make effective use of the state information about the network collected by the protocol as part of the process of discovering routes to other nodes. Caching is important in order to avoid the overhead of discovering a new route before sending each data packet, but caching also brings with it the risk and associated expenses of retaining routing information in a cache after the information is no longer valid due to changes in different nodes' position or changes in the wireless propagation environment.

This paper has presented an analysis of the effects of intelligent caching strategy for on-demand routing protocols in wireless ad hoc networks. The simulation results have shown that this technique drastically increases the available memory for caching the routes discovered without affecting the performance of the DSR routing protocol in any way except for a small increase in end to end delay. So it can be concluded that intelligent caching technique helps in saving lot of cache memory which can be used for saving additional routes which in turn helps in efficient utilization of the cache for enhancing the overall performance of the routing protocol.

References

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

[2] Asis Nasipuri, Mobile Adhoc networks, Department of Electrical & Computer Engineering, The University of North Carolina at Charlotte, Charlotte, NC 28223-0001.
[3] C. E. Perkins, Ad hoc Networking, Addison-Wesley, 2001.
[4] D. Johnson, Rice University;Y.Hu,UIUC and D.Maltz ,Microsoft Research The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4, February 2007 http://www.ietf.org/rfc/rfc4728.txt .
[5] 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.
[6] Z. Haas, M. Pearlman, and P. Samar, Zone routing protocol(ZRP), Internet Draft, Internet Engineering Task Force, Jan.2001, http://www.ietf.org/internet-drafts/draft-ietf-manet-zoneierp-00.txt.
[7] S. Bradner, Temporally-ordered routing algorithm (TORA) Routing IETF, Internet Draft, draft-ietf-manet-tora-spec-04.txt, RFC 2026, July 2001.
[8] Y. Kuo, and N. H. Vaidya, "Location-Aided Routing (LAR) Mobile Ad Hoc Networks," in Proceedings of the International Conference on Mobile Computing and Networking (MobiCom'98), Oct.1998.
[9] David B. Johnson and David A. Maltz, " Dynamic Source Routing in Ad Hoc Wireless Networks," In Mobile Computing, chapter 5, pages 153 – 181.Kulwar Academic Publishers,1996.
[10] Scott Carsom and Joseph Macker, Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations, internet draft, January 1999 http://www.ietf.org/rfc/rfc2501.txt .
[11] M. Marina and S. Das, "Performance of route caching strategies in dynamic source routing," in Proceedings of the 2nd Wireless Networking and Mobile Computing Workshop, Apr. 2001.
[12] 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, Wireless Networks 8, 671–679, 2002.
[13] X. Y. Hong, K. X. Xu, and M. Gerla, "Scalable Routing Protocols for Mobile Ad Hoc Networks," IEEE Network, July-Aug.2002, pp. 11–21
[14] K. X. Xu, X. Y. Hong, and M. Gerla, "An Ad Hoc Network with Mobile Backbones," Proc.IEEE ICC'2002, vol.5, Apr.–May 2002, pp. 3138–43.
[15] Roberto Beraldi and Roberto Baldoni, "A Caching Scheme for Routing in Mobile Ad Hoc Networks and Its Application to ZRP," IEEE transactions on computers, vol. 52, No. 8,August 2003.
[16] 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.
[17] Sunsook Jung, Nisar Hundewale,and Alex Zelikovsky, "Node Caching Enhancement of Reactive Ad Hoc Routing Protocols," IEEE Wireless Communications and Networking Conference, 2005 .
[18] 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.
[19] Jorge Nuevo,Comprehensible Glomosim Tutorial Compilation, Inrs - Universite Du Quebec. Mail to:Nuevo@Inrs-Telecom.Uquebec .
[20] 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 Proceedings of IEEE/ACM MOBICOM'98, pages 85–97, October 1998.
[21] Mieso K. Denko and Jun Tian, University of Guelph, Guelph, Ontario, N1G 2W1,Canada, "Cooperative Caching with Adaptive Prefetching in Mobile Ad Hoc Networks," IEEE 2006.
[22] 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.