Effective Cache Management for the Dynamic Source Routing Protocol

[1]Prof. Miss. Sharada D.Patil ,[2]Prof. Gaurav R. Patel
ASSISTANT PROFESSOR,
Gangamai College of Engg.,Nagaon,Dhule.
Department of Electronics &Teleconnumication
me.sharada.patil@gmail.com
gpgaurav09@gmail.com

Abstract: Ad hoc network is group of wireless nodes to establish a network without any fixed infrastructure or centralized supervision/management. In such a network, topology changes dynamically and due to limitations of bandwidth, transmission range and power routing becomes an important issue. A lot of work has been done in field of routing in ad-hoc network since 1990. Dynamic Source Routing protocol (DSR) provides simple and efficient routing for multihop ad-hoc network of mobile nodes. On-demand routing protocols use route caches to make routing decisions. Due to mobility, cached routes easily become stale. To address the cache staleness issue, prior work in DSR used heuristics with ad hoc parameters to predict the lifetime of a link or a route. We proposeproactively disseminating the broken link information to the nodes that have that link in their caches. We define a new cache structure called a cache table .Each node maintains in its cache table the information necessary for cache updates. When a link failure is detected, the algorithm notifies all reachable nodes that have cached the link. We conclude that proactive *cache* updating is key to the adaptation of on-demand routing protocols to mobility.

Index Terms- Mobile ad-hoc networks, On-demand routing protocols, Mobility, cache updating , NS-2.

*****

I. INTRODUCTION

In versatile specially appointed system [3, 4] unique hubs (for instance cellular telephones, portable PCs, pda , and different remote gadgets) speak with each other with no altered framework and the media utilized for correspondence is remote. "Ad hoc" signifies "can take Different structures" can be portable hubs standalone or Networked .In versatile impromptu system, portability variable is a vital issue because of which versatile hubs are unreservedly permitted to join and leave the system in the meantime, and these developments (changing in positions of hubs) are flighty. As there is no altered framework thusly it can work as a different system or can be a piece of existing/vast system.

Versatility displays a crucial test to steering conventions Routing conventions for specially appointed systems can be characterized into two noteworthy sorts: proactive and on-interest. Proactive conventions endeavor to keep up a la mode directing data to all hubs by intermittently spreading topology overhauls all through the system. Conversely, on interest conventions endeavor to find a course just when a course is required. To lessen the overhead and the dormancy of starting a course revelation for every parcel, on-interest directing conventions use course Caches. Because of portability, reserved courses effortlessly get to be stale. Utilizing stale courses cause bundle misfortunes, and expansions idleness and overhead. In this paper, we research how to make on-interest steering Protocols adjust rapidly to topology changes. This issue is critical on the grounds that such conventions use course stores to settle on directing choices; it is testing since topology changes are incessant.

II. THE DYNAMIC SOURCE ROUTING PROTOCOL

A. Overview of DSR

Dynamic Source Routing (DSR) is a directing convention for remote lattice systems. It is like AODV in that it builds up a course on-interest when a transmitting portable hub demands one. Be that as it may, it utilizes source steering as opposed to depending on the directing table at every halfway gadget.

Dynamic Source Routing Protocol (DSRP) is an on-interest, source steering convention whereby all the steering data is kept up (constantly upgraded) at versatile nodes.DSR permits the system to be totally self-arranging and self-designing, without the requirement for any current system base or organization so we consider in taking after area about the DSR convention review.

DSR Protocol Overview:-

DSR Protocol work in two phases:

1. Route discovery
2. Route maintenance

B.Benefits of Dynamic Source Routing Protocol

The Dynamic Source Routing Protocol has the following advantages:

- The Node have the information about the networks

- The DSR reduce the Packet loss and latency time
- The Node maintains the Route Status and Path for data transfer
- The Node automatically handles the Cache Updating Process
- Use On-Demand and Adaptive type of protocol for communication

C. Route Caching in DSR

DSR uses path caches [1] or link caches [3] in a path cache; a node stores each route starting from itself to another node. In a link cache, a node adds a link to a topology graph, which represents the node's view of the network topology. Links obtained from different routes can form new routes. Thus, link caches provide more routing information than path caches. A
node learns routes through forwarding ROUTE REPLIES and data packets, or by overhearing packets, since ROUTE REQUESTS are broadcast packets and thus links discovered may not be bi-directional [6]. Due to the same reason, when a node forwards a ROUTE REPLY, it caches only the links that have been confirmed by the MAC layer to be bi-directional ,which are the downstream links starting from the node to a destination

When forwarding a data packet, a node caches the upstream links as a separate route. After initiating a Route Discovery, a source node may learn many routes returned either by intermediate nodes or by the destination; it will cache all those routes. Thus, DSR aggressively caches and uses routing information.

III Cache update algorithm

In this section, we present an overview of our caching strategy. We then give the definition of a cache table and present two algorithms used to maintain the information for cache updates. Finally, we describe our cache update algorithm in detail.

A.Overview:

Quick store redesigning is critical for decreasing the antagonistic impacts of stale courses. It is additionally important to compel reserve upgrade notices to the hubs that have stored a softened connection up request to dodge the overhead of advising different hubs .Thus, our objective is this: when a hub distinguishes a connection disappointment, every single reachable hub whose reserves contain the broken connection will be advised about the connection disappointment.

To accomplish this objective, we make utilization of the data acquired from course disclosures and information transmission. We characterize a store table to assemble and keep up the data fundamental for reserve redesigns. Every hub keeps up two sorts of data for every course in its reserve table: how well the steering data is synchronized among hubs on a course, furthermore, which neighbor hub outside a course has realized which joins through a ROUTE REPLY. By keeping such nearby data, a hub knows which neighbor hub should be advised around a broken connection. A hub getting a reserve overhaul notice utilizes the nearby data kept as a part of its store table to decide and advise the neighbor hubs that have stored the broken connection. Hence, a broken connection data will be immediately engendered to every reachable hub whose reserves contain that connection.

B. The Definition of a Cache Table

A cache table has no capacity limit and thus its size changes as needed.

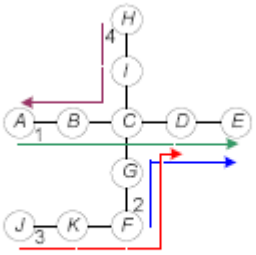


Fig.1. a Network Used for Examples

We use a network shown in Fig. 1 in our examples. We will use S-D for Source Destination,& DP for Data Packets, and replied for Reply Record in the headers of tables describingthe content of caches. Initially, there are no flows and all nodes' caches are empty. Node*A*initiates a route discovery to *E* and a ROUTE REPLY is sent from *E* to *A*. Upon receiving a ROUTE REPLY, eachintermediate node creates a new entry in its cacheFor instance, node *C* creates an entry consisting of four fields: (1) a route containing downstream links: *CDE*; (2) the sourceand destination pair: *AE*; (3) the number of data packets it received from the source node *A*: 0; (4) which neighbor will learn which links through the ROUTE REPLY.

C:

Route	S-D	dP	replyRec
CDE	*A E*	0	*B ← CDE*

When *A* receives the ROUTE REPLY, it adds to its cachetable entry

A:

Route	S-D	DP
ABCDE	*A E*	0

When node *A* uses this route to send the first data packet, it increments *Data Packets*to 1 Each intermediate node receiving the first data packet updates its cache table entry For instance, node *C* increments *Data Packets*to 1, adds the

 ISSN: 2321-8169
834 - 839

upstream links to route *CDE*, and removes the reply Recordentry, as the complete route indicates that the upstream nodes, *A* and *B*, have cached the downstream

	Route	S-D	dP	replyRec
C :	*CDE*	*A E*	0	*B ← CDE*

links, *CDE*.

When node *E* receives the first data packet, it creates a cache table entry

	Route	S-D	DP
E:	*ABCDE*	*A E*	1

When a node on this route receives the second data packet, it increments *Data Packets*to 2 Assume that after transmitting at least two data packets for flow 1, node *C* receives a ROUTE REQUEST from *G* with source *F* and destination *E*. Before sending a ROUTE REPLY to node *G*, node *C* adds a *reply Record*entry to its cache

	Route	S-D	DP	ReplyRecord
C:	*ABCDE*	*A E*	2	*G ← CDE*

Before sending a ROUTE REPLY to node *F*, node *G* creates a cache table entry

	Route	S-D	DP	ReplyRecord
G:	*GCDE*	*F E*	0	*F ← GCDE*

When *F* gets the ROUTE REPLY, it inserts into its cache:

	Route	S-D	DP
F:	*FGCDE*	*F E*	0

When node *C* receives a ROUTE REQUEST from *I* with source *H* and destination *A*, it adds the second*ReplyRecord*entry to its cache

	Route	S-D	DP	ReplyRecord	ReplyRecord
C:	*ABCDE*	*A E*	2	*G ← CDE*	*I ← CBA*

As described in Section III-E.1, a node creates a cache table entry to store a source route if aroute consisting of the downstream links in the source route does not exist in its cache Assume that flow 2 starts. When it reaches node *D*, node *D* adds the second entry to itscache, because the sub-route *CDE* has been completed by flow 1. When receiving the first datapacket, node *D* knows that its upstream nodes have cached the downstream link *DE*.

	Route	S-D	DP
D:	*ABCDE*	*A E*	2
D:	*FGCDE*	*F E*	1

When node F receives a ROUTE REQUEST from node *K* with source *J* and destination *D*, itextends its cache entry

	Route	S-D	DP	ReplyRecord
F :	*FGCDE*	*F E*	2	*K ← FGCD*

By securing a complete way, we keep the relationship between the upstream and the downstream center points, so that a center knows which upstream centers ought to be prompted if a downstream association is broken. In light of the information in the information Packets field, a center point makes sense of if downstream centers ought to be told around a broken association. In light of the information in the answer Record field, a center point knows which neighbors have held which joins through ROUTE REPLIES. Each center point collects the information about which neighbors have held which associate in the center point's store, without the need to know all centers in the frameworks that have put away a particular association. Appropriately, topology spread state is kept in a passed on way.

C.Cache Update Algorithm

We present the cache update algorithm. We define a broken link aforwardor *backward* link. A broken link is a *forward* link for a route if the flow using the route crosses the link in the same direction as the flow detecting the link failure; otherwise, it is a backward link& according to type algorithm applied.

Proposed Work

We propose proactively diffusing the broken association information to the centers that have that association in their stores. We portray another store structure called a store table and present a passed on store update estimation. Each center point keeps up in its store table the information imperative for store upgrades. Exactly when association disillusionment is recognized, the figuring educates each and every reachable center that has held the association in an appropriated way. The above issue can be handled through after modules..

The modules that are included in this project are

- Route Request
- Route Maintenance
- Message Transfer
- Cache Update

Module 1: Route Request: At the point when a source hub needs to send bundles to a destination to which it doesn't have a course, it starts a Route Discovery by TV a ROUTE REQUEST. The hub accepting a ROUTE REQUEST checks whether it has a course to the destination in its reserve. In the event that it has, it sends a ROUTE REPLY to the source including a source course, which is the link of

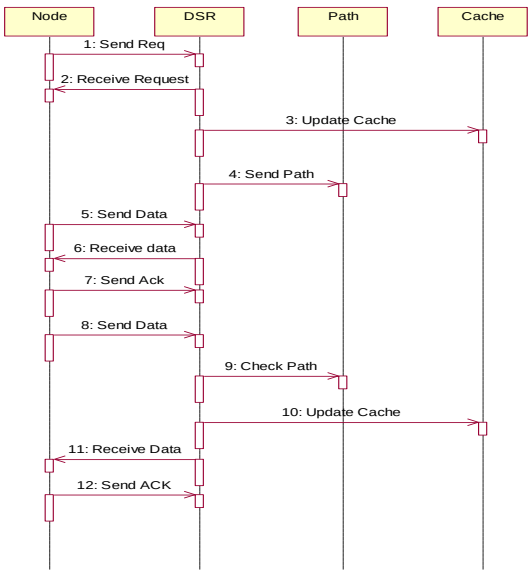
 ISSN: 2321-8169
834 - 839

the source course in the ROUTE REQUEST and the reserved course. In the event that the hub does not have a stored course to the destination, it adds its location to the source course and rebroadcasts the ROUTE REQUEST. At the point when the destination gets the ROUTE REQUEST, it sends a ROUTE REPLY containing the source course to the source. Every hub sending a ROUTE REPLY stores the course beginning from itself to the destination. At the point when the source gets the ROUTE REPLY, it reserves the source course.

Module 2: Message Transfer: The Message transfer relates with that the sender node wants to send a message to the destination node after the path is selected and status of the destination node through is true. The receiver node receives the message completely and then it sends the acknowledgement to the sender node through the router nodes where it is received the message.

Module 3: Route Maintenance: Route Maintenance, the node forwarding a packet is responsible for confirming that the packet has been successfully received by the next hop. If no acknowledgement is received after the maximum number of retransmissions, the forwarding node sends a ROUTE ERROR to the source, indicating the broken link. Each node forwarding the ROUTE ERROR removes from its cache the routes containing the broken link.

Module 4: Cache Updating: When a node detects a link failure, our goal is to notify all reachable nodes that have cached that link to update their caches. To achieve this goal, the node detecting a link failure needs to know which nodes have cached the broken link and needs to notify such nodes efficiently. Our solution is to keepkeep track of topologypropagation state in distributed manner.



Sequence Diagram

Simulation Method:-

We used *ns-2* [2] network simulator. Ns2 is a discrete event simulator targeted at networking research.. Ns-2 is an object-oriented simulator written in C++ and OTcl. Surveys of different mobility models have been done. This includes the Random Waypoint Mobility Model that is used in our work. In this model, a node starts in a random position, picks a random destination, moves to it at a randomly chosen speed, and pauses for a specified pause time.

NS2:-Ns2 is nothing but the Network Simulator version 2.. Network simulator provides substantial support for simulation of TCP, simulation of routing as well as simulation of multicast protocol over wired and wireless (local and satellite) networks.

Features of ns2 are as follows:

Network simulator helps in easy verification of protocols in less time and money, protocols like TCP, UDP, HTTP, routing algorithms etc.

- In Network simulator OTCL is front end and C++ is back end.
- Network simulator is also a ongoing effort of research and development
- A network environment for ad-hoc network
- Wireless channel modules (e.g.802.11
- Routing along multiple paths
- Mobile hosts for wireless cellular networks.

Ns version 2.34 has been used for simulation. How reenactment is completed and how required result information is accumulated from the yield records is examined here.

Reenactment Input: TCL scripts are utilized to determine situations, design has, actuate movement sources and sinks, assemble measurable information and run reproductions

Reproduction Output: The yield of the recreation utilizing ns is a follow record, which contains follows from all layers. New follow position, which incorporates generally all required subtle elements, has been utilized.

Basic settings of NS-2 simulation

Parameter	Value
Channel Type	Wireless Channel.
Network Interface Type	Wireless Phy
MAC Type	802.11
Interface queue (IFQ) Type	Droptail /Priqueue.
Link Layer Type	LL
Antenna Type	Omni Antenna
Maximum no. of nodes	50/100

 ISSN: 2321-8169
834 - 839

Pause Time	0,10,20,40,100
Traffic Size	CBR
Packet Size	512 Bytes
Mobility Model	Random Waypoint

Performance Metrics: We used four metrics:

Packet Delivery Ratio: the fraction of data packets sentby the source that are received by the destination.

Packet Delivery Latency: the average time taken by adata packet to travel from the source to the destination.

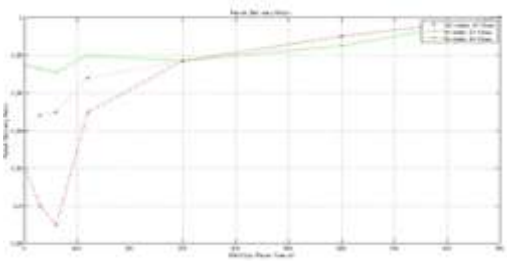
Normalized Routing Overhead: the ratio of the total number of routing packets transmitted to the total numberof data packets received. For the cache update algorithm,routing packets include ROUTE ERRORS used forcache updates.

Good Cache Replies Received: the percentage of ROUTE REPLIES without broken links received by the originated from caches.

Simulation Results

As already outlined we have taken On-demand (Reactive) routing protocols, namely Dynamic Source Routing (DSR). The mobility model used is Random waypoint mobility model because it models the random movement of the mobile nodes is for scen. 25-0,25- 5,25-10,25-20,25-40,25-100. the number of traffic sources was 25-5,25-10,25-15,25-20, the maximum speed of the nodes was set to 20m/s

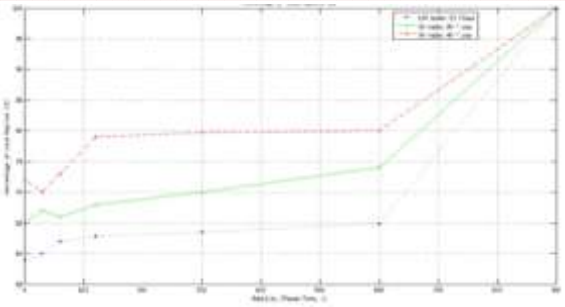
1. Packet Delivery Ratio: Figure show packet delivery ratio the improvement increases as mobility, traffic load, or network size increases We get value of packet delivery ratio which more than referred paper for different nodes values & different flows which is shown in fig.



Graph of Packet Delivery Ratio Vs Mobility

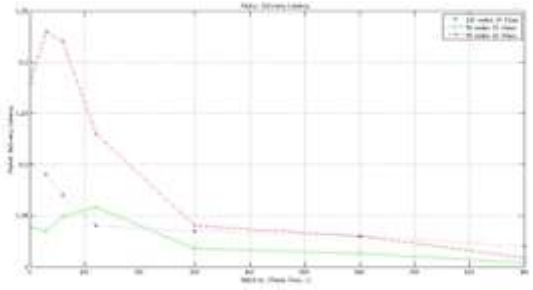
2)**Percentage of good route replies**: Percentage of good route replies sent from caches. is an important measure of cache performance.

These results demonstrate that DSR-Update removes stale routes more quickly than FIFO and predicting timeouts



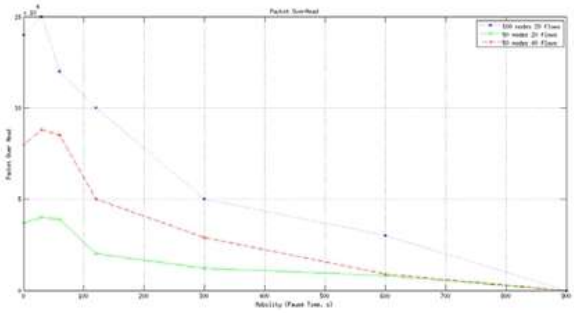
Graph of Percentage of Good Replies Vs Mobility.

3) **Packet Delivery Latency**: DSR-Update has lower latency in most cases of the 100 node scenario



Graph of Packet Delivery Latency Vs Mobility

4) **Packet Overhead**: Our algorithm introduces overhead due to cache update notifications



Graph of Packet Overhead Vs Mobility

After running the simulation for different metrics for different number of nodes & flows weget following result which is shown in table.

Nodes	50 Nodes		100
	20 Flows	40 Flows	
Packet Delivery Ratio	0.980	0.947	0.967
Percentage of Good Replies(%)	72	79.11	67.46
Packet Delivery Latrency	0.031	0.119	0.065
Packet Overhead	22285	43714	84285

The parameters which should be **more** than the one mentioned inreferred paper:

Percentage of Good Replies

Packet Delivery Ratio

CONCLUSION

We define a cache tablefor cache updatesProactive cache updating is more efficient than adaptive timeout mechanisms. Our work combines the advantages of proactive and on-demand protocols: on-demand link failure detection and proactive cache updating. Our solution is applicable to other on-demand routing protocols hence we conclude that proactive cache updating is key to the adaptation of on-demand routing protocols to mobility.

REFERENCES

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

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

[3] Mobile Ad Hoc Networks MANET (By KalinTaskov)http://cse.unl.edu/~ktaskov/projects/CSE990[4] Swedish Institute of Computer Science Laura Marie Feeney http://www.nada.kth.se/kurser/kth/2d1490/05/lectures/feeney_mobile_adhoc_routing.

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

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

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

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

[8] *Distributed Cache Updating for the Dynamic Source Routing Protocol* IEEE transactions on mobile computing, vol. 5, no. 6, June2006.

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

[10] X. Yu and Z. Kedem. A distributed adaptive cache update algorithm for the Dynamic Source Routing protocol. In *Proc. 24th IEEE INFOCOM*, March 2005appeared as NYU CS Technical Report TR2003-842, July 2003.)2006

[11] B. Liang, H, Zygmunt, "Optimizing route-cache lifetime in ad hoc networks." *INFOCOM 2003. (Twenty-Second Annual Joint Conference of the IEEE)*