www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 2 Issue 12 Dec,2013 Page No. 3548-3555

Overcoming Cache Staleness Issue Using Dynamic Source Routing Protocol

Chaitali G. Taral, Dr.P.R.Deshmukh

ME Student, Dept of Computer Science & Engineering
SIPNA College of Engineering & Technology
Amravati (MS), India.
Email- id: chaitalitaral@rediffmail.com
Professor, Dept of Computer Science & Engineering
SIPNA College of Engineering & Technology
Amravati (MS), India.

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. To address the cache staleness issue which leads towards the loss of the packets during delivery of the information in DSR protocol, prior work used adaptive timeout mechanisms. Such mechanisms use heuristics with ad hoc parameters to predict the lifetime of a link or a route. However, heuristics cannot accurately estimate timeouts and as a result of those valid routes will be removed or stale routes will be kept in caches. In this work we will propose proactively disseminating the broken link information to the nodes that have that link in their caches. It is important to inform only the nodes that have cached a broken link to avoid unnecessary overhead. Thus, when a link failure is detected, our goal is to notify all reachable nodes that have cached the link about the link failure which leads towards the minimization of overheads, latency and congestion control of the network.

Keywords - ad-hoc network, DSR, Cache updating, Proactive Protocol, On-demand Routing Protocol.

1. INTRODUCTION

In ad hoc environment, the reduction of overhead, latency and congestion presents a fundamental challenge to routing protocols. Routing Protocols for ad-hoc networks can be classified into two major types:

1. Proactive Protocol and
2. On-demand Routing Protocol

1. Proactive Protocol: It attempt to maintain up-to-date routing information to all nodes by periodically disseminating the network updates throughout the network.
2. On-demand Routing Protocol: Attempts to discover a route only when a route is needed.

Sometimes due to congestion in the path of the network cached routes easily becomes stale and that stale routes becomes the reason for packet lasses and due to the loss of the packet or the information it leads to increase the latency and the overhead.

So, by considering the reasons of staleness in the path of the route and the loss of the packets during the delivery of the information we are attempting to maintain on-demand routing protocol which will be going to detect the broken link information and update the cache with respect to that broken link information. For that we defining a new cache structure called a cache table and present a distributed cache update algorithm. 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 that link in their caches and outperforms DSR with path caches, an adaptive timeout mechanism for link caches.

2. LITERATURE REVIEW & RELATED WORK

On-demand Route Maintenance results in delayed awareness sometimes, because a node is not notified when a cached route breaks until it uses the route to send packets. We classify a cached route into three types:

1. **pre-active**, if a route has not been used;
2. **active,** if a route is being used;
3. **Post-active**, if a route was used before but now is not.

It is not necessary to detect whether a route is active or post-active, but these terms help clarify the cache staleness issue. Stale pre-active and post-active routes will not be detected until they are used. Due to the use of responding to ROUTE REQUESTS with cached routes, stale routes may be quickly propagated to the caches of other nodes. Thus, pre-active and post-active routes are important sources of cache staleness. 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 the following objectives:

(1) How well routing information is synchronized among nodes on a route.
(2) Which neighbor has learned which links through a ROUTE REPLY Each node gathers

such information during route discoveries and data transmission.

Several studies have been proposed for updating routes in DSR protocol and remove the stale routes as proposed by:

Marina and Das 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 packet, route error propagation packets are transmitted as broadcast packets at the MAC layer. When a node receives a 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.

Lou and Fang 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.

3. ANALYSIS OF PROBLEM

Several studies have been proposed for updating routes in DSR protocol and remove the stale routes as proposed by [4-5].

Marina and Das [4] 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 [5] 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. Prior Work in DSR used heuristics with ad hoc parameters to predict the lifetime of a link or a route. Prior researches have proposed to provide link failure feedback to TCP so that TCP can avoid responding to route failures as if congestion had occurred. TCP assumes such losses occur because of congestion, thus invokes congestion control mechanisms such as decreasing congestion windows, raising timeout, etc, thus greatly reduce TCP throughput.

However, after a link failure is detected, several packets will be dropped from the network interface queue; TCP will time out because of these packet losses, as well as for Acknowledgement losses caused by route failures. There is no intimation information regarding about to the failure links to the Node from its neighboring Node's. So that the Source Node cannot able to make the Route Decision's at the time of data transfer.

Limitations of Existing System:

1. The Stale routes causes packet losses if packets cannot salvaged by intermediate nodes.
2. The stale routes increases packet delivery latency before concluding link failure
3. Use adaptive time out mechanism

4. PROPOSED WORK

The main aim of the paper is to get the fast and accurate prediction of the broken link information in the network. The overall framework of the work will be given in the algorithm.

The Main work is found under four Modules are:

Module 1: Request for Route

When a source node wants to send packets to a destination to which it does not have a route, it initiates a Route Discovery by broadcasting a ROUTE REQUEST. The node receiving a ROUTE REQUEST checks whether it has a route to the destination in its cache. If it has, it sends a ROUTE REPLY to the source route. If the node does not have a cached route to the destination, it adds its address to the source route and rebroadcasts the ROUTE REQUEST.

Module 2: Transferring Message via Route

The Message transfer relates with that the sender node wants to send a message to the destination node after the path is selected.

Module 3: Path Maintenance

Path 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: Updation of Cache

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.

The overall logical structure of the work is divided into processing modules and a conceptual data structure is defined as Architectural Design.

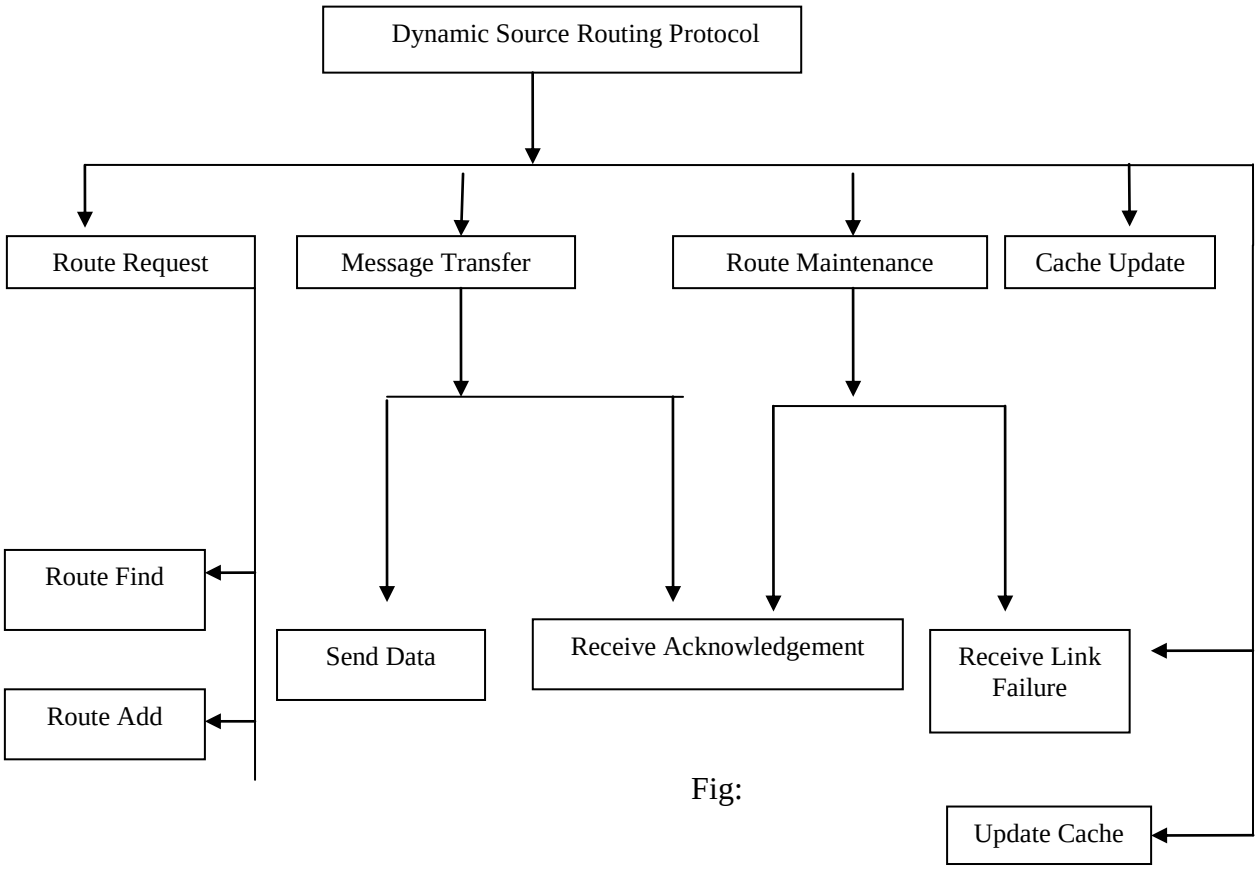

Fig:

Fig : Architecture Design for Overcoming Cache Staleness issue using DSR

The Dynamic Source Routing Protocol starts either when a node detects a link failure or when it receives a notification. In a cache table, a node not only stores routes but also maintain two types of information for each route:

(1) How well routing information is synchronized among nodes on a route; and

(2) Which neighbor has learned which links through a ROUTE REPLY. Each node gathers information during route discoveries and data transmission, without introducing additional overhead. The two types of information are sufficient; because each node knows for each cached link which neighbors have that link in their caches.

To work over this concept we need to design Cache table and need a systematic procedure

which will going to handle all the processing properly. The systematic procedure is called as an algorithm. An algorithm will give the proper way to the proposed work

Algorithm : Basic steps describing the proposed work.

1. Cache Staleness issue
2. Designing of Cache Table Structure
3. Collection of information for cache updated
4. Maintenance of information for cache update
5. Adding of route
6. Finding of route
7. Final Collection & Maintenance of information

Steps to be implemented from an algorithm:

Cache Staleness issue: The first phase of the work will be done by detecting the staleness in the links of the network. This might because due to the congestion in the network.

Designing of Cache Table Structure:
We design a cache table that has no capacity limit. There will be four fields in a cache table entry:

- Route: It stores the links starting from the current node to a destination or from a source to the destination.
- SourceDestination: It is the source and destination pair.
- DataPackets: It records whether the current node has forwarded 0, 1, or 2 data packets. It is 0 initially, incremented to 1 when the node forwards the first data packet, and incremented to 2 when it forwards the second data packet.
- ReplyRecord: This field may contain multiple entries and has no capacity limit.

Collection and Maintenance of information:
We design two algorithms for collection and maintenance of information in cache such as:

1) Add Route :We use add route when a node attempts to add a route to its cache table.

2) Find Route : When a node tries to find a route to some destination.

The algorithm notifies the closest upstream and/or downstream nodes and the neighbors that learned the broken link through ROUTE REPLIES. When a node receives a notification, the algorithm notifies selected neighbors: upstream and/or downstream neighbors, and other neighbors that have cached the broken link through ROUTE REPLIES. Thus, the broken link information will be quickly propagated to all reachable nodes that have that link in their caches.

5. APPLICATON

- The DSR is mainly used in VANET(Vehicular ad-hoc Network).
- The Dynamic Source Routing protocol (DSR) is a simple and efficient routing protocol designed specifically for use in multi-hop wireless ad hoc networks of mobile nodes. DSR allows the network to be completely self-organizing and self-

configuring, without the need for any existing network infrastructure or administration.

- It maintains the Route Information of the network.
- DSR plays an important role in the discovery of the route path for delivery of the data packets.
- DSR is used in different supporting heterogeneous networks as well as interconnecting to the Internet and it supports for the routing of multicast packets in ad hoc networks.

6. CONCLUSION

In this work, we present proactive updates of route caches in an adaptive manner and define a new cache structure called a cache table to maintain the information necessary for cache updates and present a distributed cache update algorithm that uses the local information kept by each node to notify all reachable nodes that have cached a broken link. This will show that proactive cache updating will be more efficient than adaptive timeout mechanisms. Our work will combine the advantages of proactive and on-demand protocols: on-demand link failure detection and proactive cache updating. Our solution will be applicable to other on-demand routing protocols.

REFERENCES

[1] Xin Yu,"Distributed Cache Updating for the Dynamic Source Routing Protocol," IEEE transactions on mobile computing, vol.5,no.6,june 2006.

[2] J. Broch, D. Maltz, D. Johnson, Y.-C. Hu, and J. Jetcheva, "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols," Proc. Fourth ACM MobiCom, pp. 85-97,1998.

[3] ns Notes and Documentation, K. Fall and K. Varadhan, eds., TheVINT Project, Univ. of California, Berkeley, LBL, USC/ISI, andXerox PARC, 1997.

[4] G. Holland and N. Vaidya, "Analysis of TCP Performance over Mobile Ad Hoc Networks," Proc. Fifth ACM MobiCom, pp. 219-230,1999.

[5] Y.-C. Hu and D. Johnson, "Caching Strategies in On-Demand Routing Protocols for Wireless Ad Hoc Networks," Proc. Sixth ACM MobiCom, pp. 231-242, 2000.

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

[7] D. Johnson and D. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," Mobile Computing, T. Imielinski and H. Korth, eds, chapter 5, pp. 153-181, Kluwer, 1996.

[8] Y.-C. Hu and D. Johnson, "Ensuring Cache Freshness in On- Demand Ad Hoc Network Routing Protocols," Proc. Second Workshop Principles of Mobile Computing, pp. 25-30, 2002.

[9] D. Johnson, D. Maltz, and Y.-C. Hu, "The Dynamic Source Routing for Mobile Ad Hoc Networks," IETF Internet Draft,

[10] W. Lou and Y. Fang, "Predictive Caching Strategy for On-Demand Routing Protocols in Wireless Ad Hoc Networks," Wireless Networks, vol. 8, no. 6, pp. 671-679, 2002.