Distributed Cache Architecture for Routing in Large Networks

Rakesh.H.Shetty
Department of Computer-Science & Engg.
Shaikh College of Engineering and Technology
Belgaum, Karnataka, India
rakesh.shetty610@gmail.com

Asst. Prof. Kalkundri Ravi
Department of Computer-Science & Engg.
Shaikh College of Engineering and Technology
Belgaum, Karnataka, India
kalkundri.ravi05@gmail.com

Rohit S. Kadagaokar
Department of Computer-Science & Engg.
Shaikh College of Engineering and Technology
Belgaum, Karnataka, India
rohitkadagaokar2012@gmail.com

Amul Dake
Department of Computer-Science & Engg.
Shaikh College of Engineering and Technology
Belgaum, Karnataka, India
amuldake@gmail.com

***Abstract*—On-demand routing protocols use route caches to make direction-finding decision. Due to ad hoc networks, cached routes easily become stale. To address the cache staleness matter, former work in DSR used heuristics with ad hoc parameters to predict the lifetime of a link or a route. The goal of our work is to proactively disseminating the broken link information to the nodes that have that link in their caches. We describe a new cache structure called a cache table and present a distributed cache bring up to date algorithm. Each node maintain in its cache table the information necessary for cache updates. When a connection crash is detected, the algorithm notify all accessible nodes that have cached the link in a distributed manner.. We conclude that proactive cache updating is key to the adaptation of on-demand routing protocols to mobility.**

Keywords— DSR; Cache table; Distributed Cache Architecture

I. INTRODUCTION

Routing protocols for ad hoc networks can be classified into two major types: proactive and on-demand. Proactive protocol try to preserve up-to-date routing information to all nodes by periodically disseminating topology updates all over the network. In contrast, on demand protocols crack to discover a route only when a path is required. To decrease the overhead and the latency of initiating a route discovery for every packet, on-demand routing protocols make use of route Caches. Due to ad hoc, cached routes simply turn into stale. Using stale routes cause packet dead, and increase latency and overhead. In this paper, we study how to make on-demand routing Protocols adapt quickly to topology change [1]. This problem is vital since such protocols make use of route caches to make routing decisions; it is challenging because topology changes are frequent.

To address the cache staleness issue in DSR (Dynamic Source Routing protocol) prior work used adaptive timeout mechanism. Such mechanisms employ heuristics with ad hoc parameters to predict the lifetime of a link or a route.

Sponsored by: Shaikh College of Engineering and Technology, Belgaum, Karnataka, India.

However, a prearranged choice of ad hoc parameters for certain scenarios may not work fine for others, and scenario in the real world are different from those used in simulation. furthermore, heuristics cannot precisely estimate timeouts because topology changes are volatile. As a result, either valid routes will be removed or stale routes will be kept in caches.

In our work, we propose proactively disseminating the broken link information to the nodes that have that link in their caches. Proactive cache update is solution to making route caches adapt quickly to topology changes [1][2]. It is also main to notify only the nodes that have cached a broken link to stay away from unnecessary overhead. Thus, when a link crash is detected, our goal is to notify all reachable nodes that have cached the link about the link failure.

The rest of the paper is organized as follows: Section II describes the Problem Definition for the DCA Section III QOS & Its Importance Section IV Distributed Cache Architecture (DCA) advantages Section V proposed system. Lastly we conclude in Section VI and its Future Scope in Section VII.

II. PROBLEM DEFINATION

1. The Dynamic Source Routing Protocol has the following objectives
2. The DSR reduce the Packet loss and latency time
3. The Node maintains the Route Status and Path information for data transfer and path request
4. The Node automatically handles the Cache Updating Process if any Link failure is happened in the Network

III. QOS & ITS IMPORTANCE

QOS specify how “good” the offered services are.QOS assures that the data packet or message have reach at the destination in an efficient way.Routing is the process of selecting best paths in a network.QOS capable networks identify a route between a source and destination pair. Communication networks face a variety of quality-of-service demands. The main motivation for a QOS unit in a data

ISSN: 2278-0181

network port processor is to control access to available bandwidth and to regulate traffic. A network that can provide these various levels of services requires a more complex structure [2][7]. We put the basis of our proposal on an end-to-end QoS routing architecture that can be easily deployed across real and potentially huge network. Issue, such as the hierarchical structure of large networks, topology aggregation, the diversity of traffic in large networks, and inter-domain and intra-domain topologies are taken into account.

IV. DISTRIBUTED CACHE ARCHITECTURE (DCA) ADVANTAGES

In this work, the distributed cache architecture is to reduce the route computing load. The distributed cache architecture has several advantages are as mentioned below [4].

1. It can scale to very large networks since it has a distributed nature. It has been designed to be easily deployable in networks with multiple domains.
2. A cache content management/replacement technique called cache flushing has been developed. It suits the distributed nature of our cache architecture.
3. Once a route is cached, our distributed cache architecture does not rely on network state updates and operates independently.
4. We have considered simplicity *as* a key design issue for distributed cache architecture and its associated techniques. Therefore, the distributed cache architecture relies on simple but efficient algorithms and techniques so that the added complexity to the network is minimized.

V. PROPOSED SYSTEM

The proposed work deals with a new cache structure called a cache table to maintain the information necessary for cache updates. The presented a distributed cache update algorithm that uses the local information kept by each border node to send message through the route which has maximum bandwidth available. The algorithm enables LSR to adapt quickly to topology changes. The above diagram shows the A Distributed Cache Architecture (DCA) having 2 domains containing few normal nodes and few border nodes. One domain is connected to other domain via their border nodes.
Each border node is containing the information about the network topology and geographical position of the nodes in the network. The proposed system uses Dynamic Source Routing algorithm for implementing distributed cache architecture. In DCA each border node maintains a cache table that stores information about the local network topology in which border node resides.

Table 1: Initial Finding

SL.NO	PROTOCOL PROPERTY	DSR (DYNAMIC SOURCE ROUTING)
1.	Network Overhead	Low
2.	Multihop	Yes
3.	Loop free	Yes
4.	Multiple routes	Yes
5.	Route Discovery	on demand
6.	Route mechanism	Complete route cached
7.	Reactive/ Proactive	Reactive
8.	Routing Overhead	Low

A. System Design:

Description: From above Fig. 1, we depict that, there are 2 domains each contains 4 nodes. Here node B is acting as border node in Domain 1and node E is acting as border node in Domain 2.
On-demand Route Maintenance results in delayed awareness of mobility, because a node is not notified when a cached route breaks until it uses the route to send packet. We categorize a cached route into three types [5]:

1. **Pre-active:** if a route has not been used.
2. **Active**: if a route is being used.
3. **Post-active**: if a path was used earlier than but now is not.

It is not necessary to detect whether a route is active or post-active, but these conditions help out make clear the cache staleness issue. out of date pre-active and post-active route will not be detected until they are use. Due to the use of respond to ROUTE REQUESTS with cached route, out of date route may be fast propagate to the caches of additional nodes. Thus, pre-active and post-active route are important sources of cache staleness.

When a node detects a connection breakdown, our target is to inform all available nodes that have cached that link to update their caches. To accomplish this aim, the node detect a link failure needs to know which nodes have cached the broken link and needs to notify such nodes efficiently. This purpose is extremely demanding because of mobility and the fast propagation of routing information [8][9].

Our solution is to maintain way of topology transmission state in a distributed manner. Topology propagation state means which node has cached which link. How well routing information is synchronized among nodes on a route.

B. Main Modules

The main modules that are included in this work are:

1. *MsgServer:* This module is responsible for transfer of message or data between two nodes.
2. *MsgTab:* This module is used for domain and node selection. It also takes care about database updating, cache snooping and cache flushing.
3. *RouteServer:* This module is responsible for route computing and finding possible routes to the destination.

Sub Modules:

The sub modules that are included in this work are [6]:

1. *Sub Module 1: Route Request*: 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.

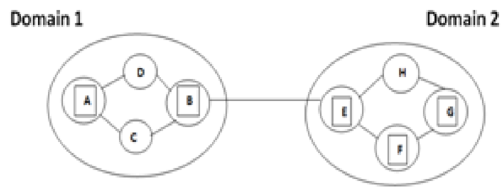

Figure 1: Default Topology.

ISSN: 2278-0181

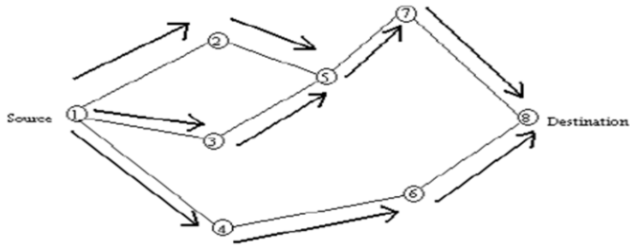

Figure 2: Route Request.

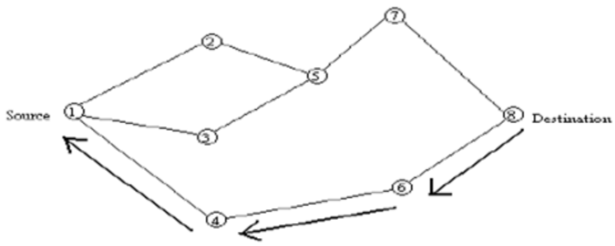

Figure 3: Route Reply.

1. *Sub 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 appropriate. The recipient node receive the message entirely and then it sends the acknowledgement to the sender node through the router nodes where it is received the message.
2. *Sub Module 3: Route Maintenance:* The node forward a packet is responsible for confirming that the packet has been successfully received by the next hop.
3. *Sub Module 4: Cache Updating:* When a node detects a connection breakdown, our objective is to inform all available nodes that have cached that link to update their caches. To attain this aim, the node detecting a link failure needs to know which nodes have cached the broken link and needs to notify such nodes proficiently. Our solution is to keep track of topology propagation state in a distributed manner.

C. *The Definition of a Cache Table*

There are seven fields in a table entry:

1. *Dest:* It stores the destination node.
2. *Bandwidth:* It stores the band width of the destination node
3. *iSegment:* It contains information about internal segment.
4. *eSegment:* It contains information about external node connection.
5. *Address:* It contains the address from source to destination.
6. *Date:* It contains current date.
7. *Time:* It contains current time.

Table 2: Cache Table Format.

Dest	**B/w**	**iSegment**	**eSegment**	**Addr.**	**Date**	**Time**

VI. CONCLUSION

In this work, we have introduced distributed cache architecture to reduce the route computing load caused by the execution of the QoS routing algorithms, assuming bandwidth-based QoS requirements. Considering the distributed nature of cache architecture, to minimize the added complexity to the network, simplicity is a key design issue in our approach. Therefore, we have designed and incorporated simple yet efficient algorithms and techniques. The distributed cache architecture is easily scaled to large hierarchical networks. The cached routes are stored in the form of multiple interconnected segments across several cache elements. Cache snooping was proposed as a distributed technique to alleviate the effects of rapid changes in the network states so that the route computing load is reduced more efficiently. In addition, cache snooping helps to increase the tolerance of QoS routing in the presence of inaccurate network state information caused by long network state bring up to date interval. This funds that the planned distributed cache architecture can also reduce the overhead traffic caused by the frequent distribution of the network state information, while achieving a good performance.

VII. FUTURE SCOPE

1. Establish Key agreement process between the Source and the Destination nodes.
2. Send the messages in the Encrypted format show that the Network hackers are not able to interfere while transmission.
3. Enable file attachments to be sent.
4. Incorporate live audio and video streaming and find QoS enabled route based on the type of application using the network.

ACKNOWLEDGMENT

We are highly thankful to our esteemed institution Shaikh College of Engineering and Technology for building us into good software engineers and sincere thanks to our principal Dr. S. C. Kamate. We express our gratitude to our beloved Prof.Mahesh Marigeri, H.O.D Department of computer science and Engineering who motivated us and helped us in canalizing our efforts in the right direction. We are highly thankful to our guide Asst. Prof. Ravi.U.Kalkundri, Who has been a constant source of inspiration and provided valuable suggestions and supports to successful accomplish this work.

-Rakesh.H.Shetty, -Rohit.S.Kadagaokar, -Amul.Dake.

REFERENCES

[1] M. Rezvan, K. Pawlikowski, and H. Sirisena "A Distributed Cache Architecture with Snooping for QoS Routing in Large Networks".

[2] J. Broch et al. "A performance comparison of multi-hop wireless ad hoc network routing protocols". In Proc. 4th ACM MobiCom, pp. 85–97, 1998.

[3] G. Holland and N. Vaidya. "Analysis of TCP performance over mobile ad hoc networks". In Proc. 5th ACM MobiCom, pp. 219–230, 1999.

[4] D. Johnson and D. Maltz. "Dynamic Source Routing in ad hoc wireless networks". In Mobile Computing, T. Imielinski and H. Korth, Eds, Ch. 5, pp. 153–181, Kluwer, 1996.

[5] 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.

[6] Acharya, R, "A Distributed QOS Routing for Multimedia Traffic." In Proceedings of GLOBCOM'2000, November 2000.

[7] Apostolopoulos, G., Guerin, R, Kamat, S., Tripathi, S., "Quality of service based routing: a performance perspective." In Proceedings of ACM SIGCOMM'98 Conference, September 1998.

[8] Apostolopoulos, G., Tripathi, S., "On reducing the cost of on-demand QOS path computation." In Proceedings of the Sixth International Conference on Network Protocols, October 1998.

[9] "A secure message authentication scheme for VANET using ECDSA" Kalkundri Ravi, S. A. Kulkarni, Department of Computer Science, K.L.S. Gogte Institute of Technology, Belgaum, Karnataka, India.