OPTIMAL PATH FOR MOBILE AD-HOC NETWORKS USING REACTIVE ROUTING PROTOCOL

Akshatha. P. S, Namita Khurana, Anju Rathi
Faculty, Department of Computer Science, KIIT College of Engineering, Gurgaon, India

ABSTRACT

Reactive protocols don't maintain routing information or routing activity at the network nodes if there is no communication. Reactive protocols determine a route to some destination only when somebody wants to send a packet to that destination. The route discovery usually occurs by flooding the route request packets throughout the mobile ad-hoc networks. Our approach is using reverse route calculation in RRQ packets and reverse route calculation in RRP packets to obtain optimal path communication between sender nodes to destination node for mobile ad-hoc networks.

KEYWORDS: *MOBILE AD-HOC NETWORKS, REACTIVE ROUTING PROTOCOL*

I.INTRODUCTION

1.1 Mobile ad-hoc network

Mobile ad-hoc networks are self organizing and self configuring multi hop wireless networks where the structure of the network changes dynamically because of mobility of nodes [1]. A MANET can be a standalone network or it can be connected to external networks (Internet).The main two characteristics of MANET are mobility and multi hop and hence multi hop operation requires a routing mechanism designed for mobile nodes. In mobile ad-hoc networks where there is no infrastructure support as is the case with wireless networks, and since a destination node might be out of range of a source node transmitting packets; a routing procedure is always needed to find a path so as to forward the packets appropriately between the source and the destination [1]. Within a cell, a base station can reach all mobile nodes without routing via broadcast in common Wireless networks. In the case of ad-hoc networks, each node must be able to forward data for other nodes.Therefore the requirements of the protocol for MANET are loop free paths, optimal path, dynamic topology maintenance etc.

1.2 Reactive Routing Protocol

Reactive routing protocol is an on-demand routing protocol for mobile ad-hoc networks. The protocol comprises of two main functions of route discovery and route maintenance. Route discovery function is responsible for the discovery of new route, when one is needed and route maintenance function is responsible for the detection of link breaks and repair of an existing route. Reactive routing protocols, such as the AODV [4], the DSR [5], do not need to send hello packet to its neighbor nodes frequently to maintain the coherent between nodes. Another important feature of reactive routing protocol is that it does not need to distribute routing information and to maintain the routing information which indicates about broken links [3]. Both the neighbor table and routing information would be created when a message needed to be forwarded and nodes maintain this information just for certain lifetime. When communication between two nodes completes, nodes discard all these routing and neighbor information. If another message needs to be forwarded, same procedure continues.

II. OUR APPROACH TO FIND THE OPTIMAL PATH

We calculate optimal path between source node and destination node by two steps, which are executed both in forwarding RRQ (route request) packets and RRP (route reply) packets.

2.1 Reverse route calculation in RRQ

Each node will create a route table called reverse route table when it receive a RRQ. This reverse route table is different from other route tables. It records and indicates the route to the source node, not the destination node. Furthermore, node will calculate the distance every time, and most importantly, this distance is the key fact to choose the shortest path to the source node [2]. First of all, when a node receives RRQ, it will create a route entry which indicates the next hop (the node forwarding the RRQ) to the source node and calculate the distance between this next hop node and the source node. Second, this node will also make the similar decision when it receives RRQ, update route table or discard RRQ [3]. For convenience, we use two variables (*First* and *new*) to indicate how to make reverse route calculation in RRQ. The *first* is distance the node calculates at the first time when it receives RRQ or the distance at current time. The *new* is distance the node calculates when it receives RRQ again. Once an intermediate node receives a RRQ, the node sets up a reverse route entry for the source node in its route table. Reverse route entry consists of <Source IP address, Source seq. number, number of hops to source node, IP address of node from which RRQ was received>. Using the reverse route a node can send a RRP to the source. Reverse route entry also contains life time field. RRQ reaches destination, In order to respond to RRQ a node should have in its route table unexpired entry for the destination and sequence number of destination at least as great as in RRQ (for loop prevention). If both conditions are met & the IP address of the destination matches with that in RRQ the node responds to RRQ by sending a RRP. If conditions are not satisfied, then node increments the hop count in RRQ and broadcasts to its neighbours. Ultimately the RRQ will make to the destination. Let us consider the temporary topology of mobile ad-hoc network as shown below in Fig. 1.

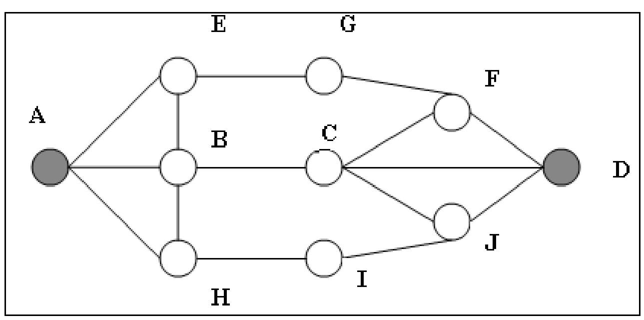

Fig.1 Temporary topology of mobile ad-hoc network

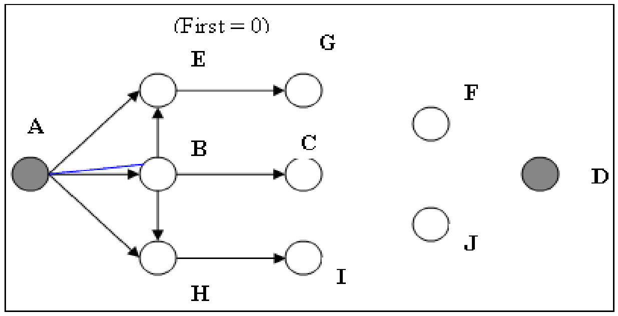

Fig.2 Node creates reverse route entry and calculates distance in RRQ

[*Note: new
RRQ]

In Fig. 2, when node A broadcasts RRQ to node B and E, node B and E will create a reverse route entry which indicates the next hop to the source node when packet arrives at node B and E. Besides, node B and E would calculate the distance between forwarding node and source node. In this situation, the next hop to source node for node B and E is node A and the *first* for node B and E is 0, because node A is both the forwarding node and source node. And then, when node B forwards the RRQ to node E, node E will calculate the *new* which is the distance between forwarding node (node B) and the source node (A). Then, node E will compare the *new* with *first* (the first distance when node E receives RRQ from node A). Since *new* > *first*, the node discard this RRQ.

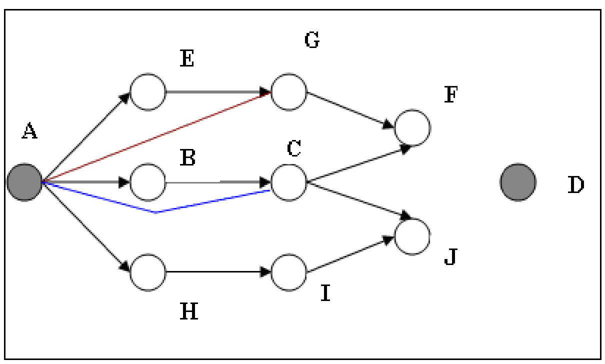

Fig.3a) Update route table in RRQ

[* Note: new
first
RRQ]

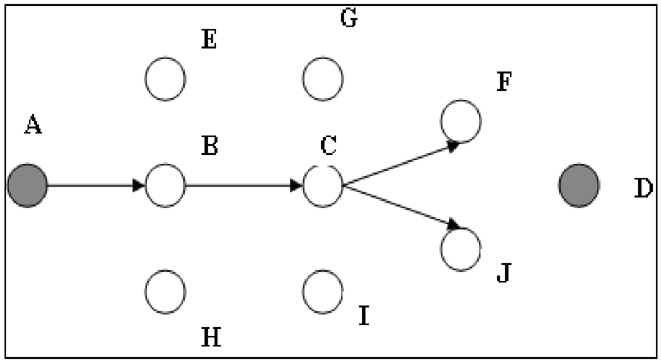

Fig.3b) The result of updating route table in RRQ

[* Note: → RRQ]

As shown in Fig. 3(a), node F creates reverse route entry when it receives RRQ from node G and select node G as the next hop to the source node (A). The same process happen when node F receives the same RRQ again from node C. Node F calculates the *new* between the forwarding node (node C) and the source node (A). Since *new* < *first*, node F updates the route table and select node C as the next hop to the source node. Fig. 3(b) is the finally route after node C broadcasts RRQ.

2.2 Reverse route calculation in RRP

We use similar calculation mechanism to get the optimal path in forwarding RRP. The only difference is that the distance we calculate in RRP is from the node forwarding RRP to the destination node.

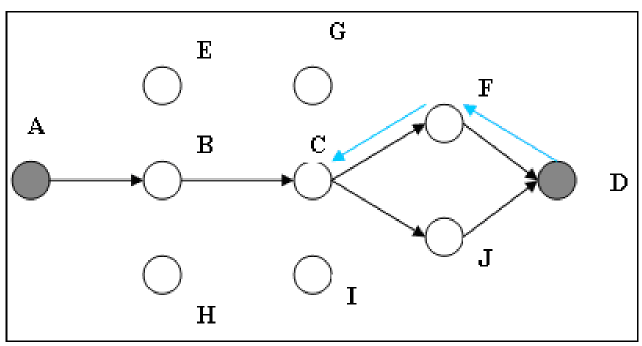

Fig.4 Node creates reverse route entry and calculates distance in RRP

[*Note: → RRQ
→ RRP]

As shown in Fig. 4, the destination node (node D) receives the RRQ from node F and then creates the RRP and unicasts it to node F. Node F forwards this RRP to node C according to the route table created by forwarding the RRQ.

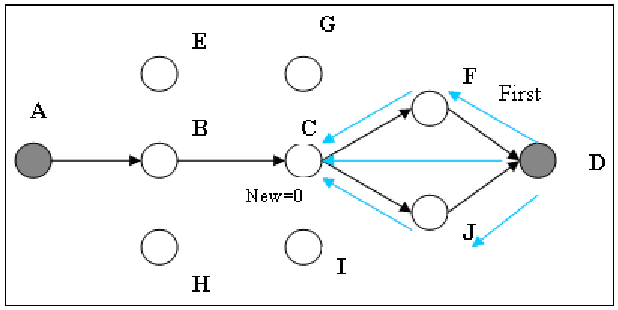

Fig .5a) update route table in RRP

[*Note: RRQ
RRP]

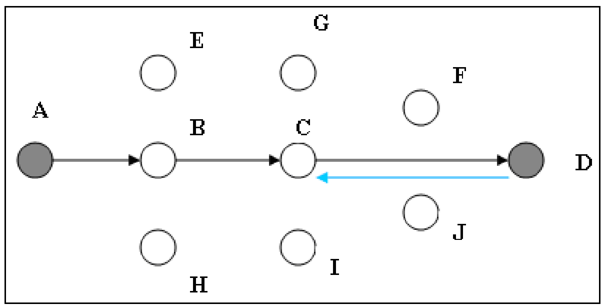

Fig.5b) Optimal path communication between A to D

When node C receives the RRP from node F, it creates the route entry and calculates the fir*st*, which indicates the next hop is node F when the message whose destination node is node D arrives at node C. And then, when node C receives RRP from node D, it will calculates the *new* and finds that *new* < *first*, as shown in Fig. 5(a) and (b), node C updates the route table, and then finally optimal path is found.

III. CONCLUSION AND FUTURE WORK

In this paper, we propose a reactive routing protocol, consists of two steps to find the optimal path: first, we calculate the shortest path to the source node and create reverse route table, second, we filter these paths to obtain optimal path communication for mobile ad-hoc networks by calculating distance to the destination node.

As future work, we will make some measurement to increase the reliability of the reactive routing protocol, especially on how to fix the link when new node joints in the mobile ad-hoc network or when a node dies in the mobile ad-hoc network.

REFERENCES

[1] Krishna gorantala, “Routing protocols in Mobile ad- hoc networks”, master’s thesis report, June 15, 2006, Ume° a University, Sweden

[2] Andrew S. Tanenbaum, “Computer networks”, fourth edition, ISBN 81-7758-1655-1, published by arrangement with Pearson education, lnc. And Doring Kindersley publishing lnc.

[3] Rong Ding and Lei Yang,” A reactive geographic routing protocol for wireless sensor networks” Beihang University, China.

[4] C. Perkins and E. Royer, “Ad-hoc on-demand distance vector routing,” in *Proc. 2nd IEEE Workshop on Mobile Computing Systems and Applications*, 1999, pp. 90–100.

[5] D. Johnson and D. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing, Imielinski and Korth, Eds. Boston, MA:Kluwer Academic*, 1996, vol. 353.
[6] Mauve, M., Widmer, J., Hartenstein, H. (December 2001). "A Survey on Position-Based Routing in Mobile Ad Hoc Networks".
[7] Bibliography of wireless ad-hoc networks available at http://w3.antd.nist.gov/wctg/manet bibliog.html.
[8] F. Baker, "An outsider's view of MANET, "Internet Engineering Task Force document 17 March 2002.
[9] L. M. Feeney, "A Taxonomy for Routing Protocols in Mobile Ad Hoc Networks," Swedish Institute of Computer Science Technical Report T99/07, October 1999.
[10] Z. J. Haas, et al., eds., Special Issue on Wireless Ad Hoc Networks, *IEEE J. on Selected Areas in Communications*, Vol. 17, No. 8 (August 1999).

Authors Biography

Akshatha.P.S was born at Kolar district, Karnataka, India in 1983. She has done her B.Tech in SJCIT, Karnataka. She is now pursuing M.Tech in Lingaya's university, Faridabad. Her research interests are Computer Networks, Database Management System.

Namita Khurana was born at Hansi, Haryana, India in 1981. She has done her graduation in 2001 from the Kurukshetra University, M.C.A in 2004 from G.J.U University Hisar, M.Phil in 2007-08 from C.D.L.U, Sirsa & pursuing M.Tech from Karnataka State University. Her research interests include Soft computing , Artificial Intelligence.

Anju Rathi was born at Faridabad, Haryana, India in 1981. She has done her graduation in 2002 from the Maharishi Dayanand University, M.C.A in 2005 from M. D. University, Rohtak & M. Tech from M. D. University, Rohtak. Her Research interests include Genetic Algorithm, Artificial Intelligence and Software Engineering.