International Journal of Electronics and
Communication Engineering (IJECE)
Vol.1, Issue 1 Aug 2012 46-54

International Academy of Science, Engineering and Technology
Connecting Researchers; Nurturing Innovations

PROPOSED PATH FORMATION AND MAINTENANCE ALGORITHM FOR BETTER EFFICIENCY IN AODV

[1]VIKAS BHATNAGAR & [2]MANOJ KUMAR SINGH

[1]M. Tech Scholar, MIT, Moradabad, India

[2]Asst. Prof., MIT, Moradabad, India

ABSTRACT

MANETs workspace was initiated by a simple protocol of AODV that tries to find out the path to destination node dynamically, using circuit switching procedure. After the circuit of nodes for data transfer become known to source, transmission of data through it starts. Since MANETs have the inherent property of nodes moving here and there, this circuit breaks very often during data transfers. Here some resolutions are provided for these limitations that will provide a way to harness the attributes in an efficient way. The initial working scenario of AODV is deeply analyzed and optimizations were made in it to solve the problems of path breaking by making multiple paths towards destination, either from the source or from middle nodes, without incurring much overhead. Beside this, some local maintenance from nodes are also offered to cater the link loss problems.

KEYWORDS: MANET, AODV, Path Break, Local Recovery, Multi Path, Path Recovery.

INTRODUCTION

MANETS [1] are characterized by the property of moving nodes, with fast moving topologies and nodes entering and leaving the network anytime. For routing the packets various algorithms have been designed which are categorized according to the essence they depend on.

Current research work is heavily oriented towards designing better algorithms, as well as removing the limitations of the previous ones. Such works always remove some sort of problems but help develop other limitations as side effect. According to the platform on which they are developed, routing algorithms can be categorized as:

1. Pro-active.
2. Reactive.
3. Hybrid

Proactive [3] routing algorithms need that the routing table of the nodes should have data path being inserted before they were being inserted in the network. Manually we have to find the path from source to the destination which is then inserted in node's routing table [4]. We can also create conditions where one node is always forced to transmit data to only given specific nodes. Such algorithms work well only in the conditions where knowledge exist at hand which nodes are permanent in network, as well as if there is very little mobility in the network. If the network is highly dynamic, then it is cumbersome for the MANET admin to insert the new nodes at given places with routing table updates for the nearby nodes. Examples of this type include DSDV [5], WRP [6].

On the other hand, Reactive [7] protocols do not require initial routing table to work for. Their routing tables are initialized to no routes. When a node finds data to send to other node, it dynamically looks for connectivity to the destination node. Due to this feature, this protocol becomes highly adaptive to ever changing topologies and location changes of the nodes in MANETs. If the path breaks before the completion of data transaction, such protocols make use of remedial methods to overcome the problems. Example of this type includes DSR, AODV [2][5] and ABR [6].

Hybrid [7] protocols are mixture of proactive as well as reactive protocols. They try to acquire the best features of both the protocols. The better attributes of each protocol, depending on the scenarios of the network, are chosen and applied, so that the overall burden on the resources of the MANETs is limited. Example of this type includes TORA [6], ZRP [8].

AODV ROUTING

Adhoc Ondemand Distance Vector routing is a reactive protocol that is a mixture of DSR and DSDV protocols. The route discovery procedure as well as route maintenance phases are similar to DSR protocol while the hop to hop data transfers as well as route updating phases are similar to DSDV protocols.

AODV offers quick adaptation to dynamic link conditions, lower processing needs and memory overhead, lower network utilization, and determines unicast routes to destinations. It uses destination sequence numbers to ensure loop freedom at all times, avoiding problems associated with classical distance vector protocols. It enables dynamic, self-starting, multi-hop routing between participating mobile nodes wishing to set up and maintain an ad hoc network. AODV allows mobile nodes to obtain new routes speedily for new destinations. AODV offers quick convergence even in the case of path breakage. When paths breaks, all the nodes which are on the side of the source are conveyed message that path breaks has occurred and delete the related entries from the routing table.

Route Requests (RREQs), Route Replies (RREPs), and Route Errors (RERRs) are the packet types utilized by AODV. UDP is used to transfer such packets and normal IP header processing applies. For broadcast, the IP limited broadcast address (255.255.255.255) is used.

Before starting the data transfer, the source generates a RREQ packet that is broadcasted to the nearby stations. Such a RREQ packet contains RREQ, id reserved bit, destination IP address, destination sequence number, source IP address and source sequence number. It also consists of some other fields also. This packet when received by a node is checked in the routing table of the node. The destination sequence number is compared from packet to that of routing table. If the packet sequence number is greater than the one in the node routing table, packet is forwarded to establish a new route. If this is not true, then it means that the node has a more active route to the destination and in response, a RREP packet is generated and unicasted towards the source. When a node receives multiple RREQ originated from same source as well as for same destination, then it acknowledges by forwarding only the first RREQ and discarding rest of all, received by it. This way lot of overhead is overcome at each level of intermediate nodes. When the destination node receives a RREQ, it processes the information in it and generates a RREP packet which is unicasted towards the source, and will traverse only the nodes that are contained in the RREQ packet received by the destination. Each node that gets a RREQ packet makes an entry in RREQ packet before forwarding it to next nodes, and makes an entry in routing table [9] about the path from its last node. In the course of RREP packet back to source, the nodes make an entry of the backward path in their routing table. When the source gets the RREP packets it analyze the information in it and start sending data through the sequence of nodes given in it. If during transmission, any node steps out of the scope of the nearby node, and is indulge in transmission, then a RERR packet is transmitted by the last node in path, towards the source. Getting a RERR packet the source again initializes a RREQ packet for a new path. The limitation of AODV is that in case of path break, overhead is incurred to a very large extent. A whole new process of RREQ is initiated in case of losing an intermediate node. Another limitation is that the nodes can detect each other broadcasts. Along with this, if a valid route is expired, how much time is used to detect the expiry of that link.

PROPOSED SOLUTION

The path development phase in AODV is considered optimized and not much is changed in its working as any advancement made to it completely changes the algorithm to some different, already developed algorithms. Whatever changes that can be suggested and implemented in path development phase are very rare to find as very lass scope is find in this scenario. The solution not only works in the development phase of the path but also takes care of all the work done by individual nodes in developing route to destination. Apart from this if a node displaces its position then very less overhead in incurred in maintaining the path or creating a new one. In the very first phase of development of the path, optimizations can be done with the RREQ packets. Normally a RREQ packet is discarded by the nodes when it is already received and processed previously, forwarded by some other node. This is feasible because it prevents a lot of overhead on the network. But, what if, the destination node receives more than one RREQ packet. Default AODV routing algorithm discards any duplicate RREQs at destination also. This algorithm, make changes to default scenario little bit to optimize the whole process. Why discard a packet at destination when it can prove to be a helping agent in making a backup path for the same route. We can buffer a limited numbers of RREQ packets at destination node, depending upon a given attribute of our choice. If we are looking for time constraint then we can gather RREQ packets whose latency is very near to first arrived RREQ packet. If not this, we can have number of hops as the decision making attribute in caching the RREQ packets. Upon defining the attribute and caching the packets, a RREP packet is generated for each of the RREQ packets. During this whole procedure, there may be RREQ packets arrived which may contain lots of nodes common in most of the routes. Creating double paths with same nodes will just increase the burden on the resources of the nodes, by adding same entries in the routing table. We can delete such entries in most of the intermediate nodes on the basis of destination node and next node similarity but will defer us in removing entries from the node in which the next node is different. This concept is shown in Fig 1 where the entries from routing table are deleted by nodes B and C but not by D, which has two branches going towards the destination. This check will be at intermediate nodes routing table when they find same entries multiple times. We will then define a threshold value for such routes. Only those routes will be considered that will contain at least Ψ % of different nodes starting from the source. This value of Ψ will vary for dense networks to loosely coupled networks, and will be decision making factor for the optimization of the algorithm. Fig. 1 and Fig 2 show a scenario where a destination node gets multiple RREQ packets but discard only some and process others. It must be noted that in Fig. 2 any one node from E or F will be chosen as they have lots of common intermediate nodes.

The algorithm for such a process is stepped down:

1. Start at source, broadcast RREQ.
2. At intermediate nodes, verify destination sequence no. as usually done in AODV
   If dest. Seq. No. (Node) < dest. Seq. No. (packet)
   Update routing table entry with this path and broadcast this packet
   Else send RREP to source
3. If node gets more than one RREQ from same source then:
   If node!=destination
   Discard RREQ
   If node=destination
   Process RREQ
4. Compare all RREQ at destination on some attribute value like latency or number of hops and selects RREQs to process and discard rest.
5. Create RREP for all collected RREQ and process them towards source.
6. At intermediate nodes, for multiple entries by RREP from same destination, preserve the entries coming from different nodes, delete rest duplicate entries.
7. At source node after getting multiple RREP
   Forward data using single path.
   Forward data using multiple paths.
8. If RERR is received by source for some path:
   Delete the entry of concerned path from routing table
   Consider the other path as primary or as needed.

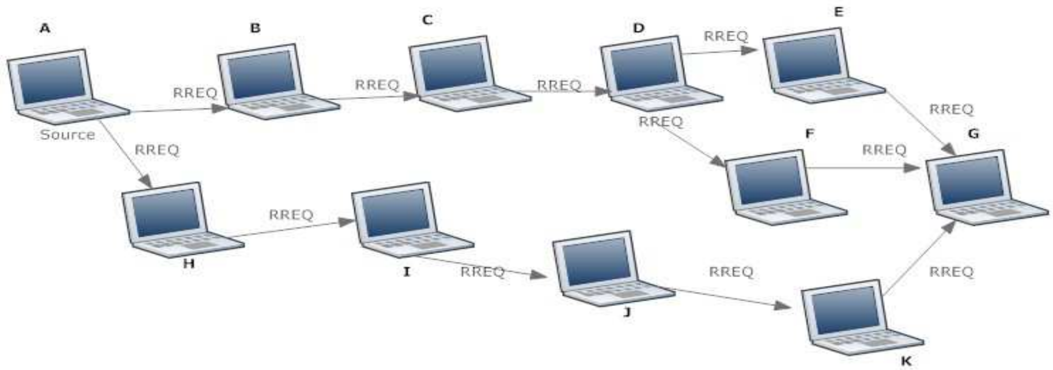

Fig. 1 Multiple RREQ processing at Destination (G)

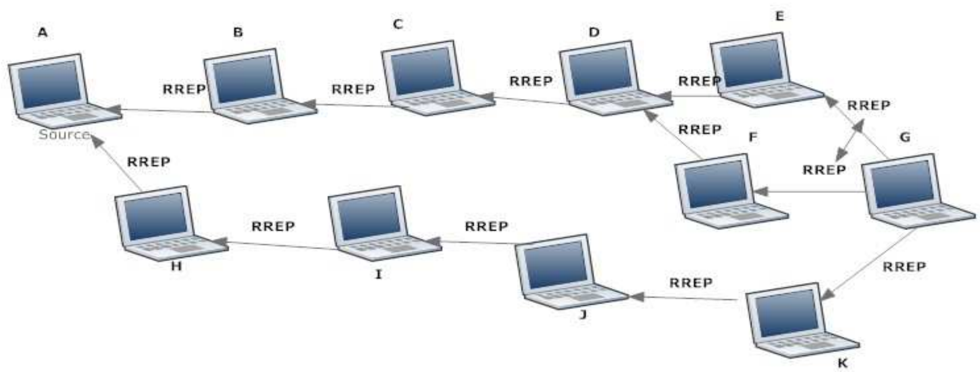

Fig. 2 Sending multiple RREP for source

Advantage of this process lies in the fact that there may be completely different paths from source to destination, which never crosses each other at any node as depicted in the figure. We can consider this path as a backup path and processes it for the time when original path gets disrupted. As soon as source gets RERR message from some node, it deletes the primary route to the destination and starts using next route from the routing table to send the data as shown in Fig. 3. If this is also found to be down, another path, if exists, can be considered for data transmission. Now the probability of path disruption for any route is already low then it can be said easily that the probability of getting all routes down from source to destination is very less. Not only this, the source can send data from more than one path so that the resource utilization is uniform throughout the network nodes. This whole process is initiated by the RREQ packet from the source.

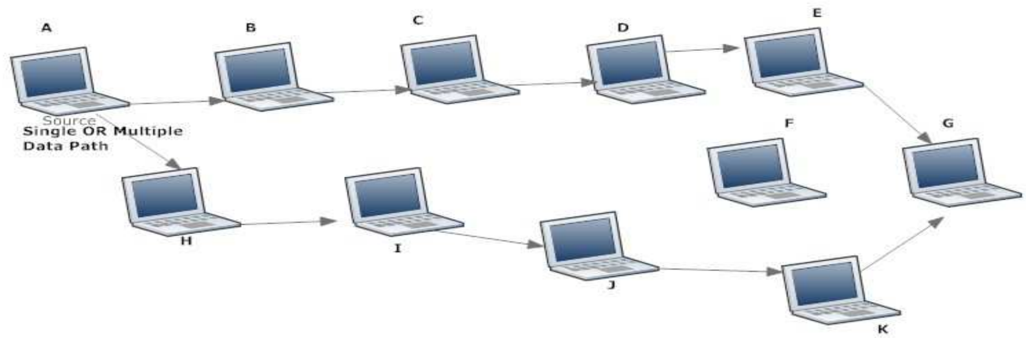

Fig. 3 Using single or multipath for data transmission from source to destination

Another event can also be associated with this process, not from the source but the destination node. If we do not opt for processing of multiple paths from source to destination using RREQ packet then we can try to create a backup route from destination to source for recovery path. The workaround is again done by the destination. When it receives multiple RREQ packets it analyses the very first packets and discards rest of all. Since this RREQ packet contains list of nodes traversed, the destination node constructs a RREQ packet for the source and broadcasts in the network. Again this packet is constructed in such a way that it seldom tries to create path using the original path created by the source node as shown in fig. 4.

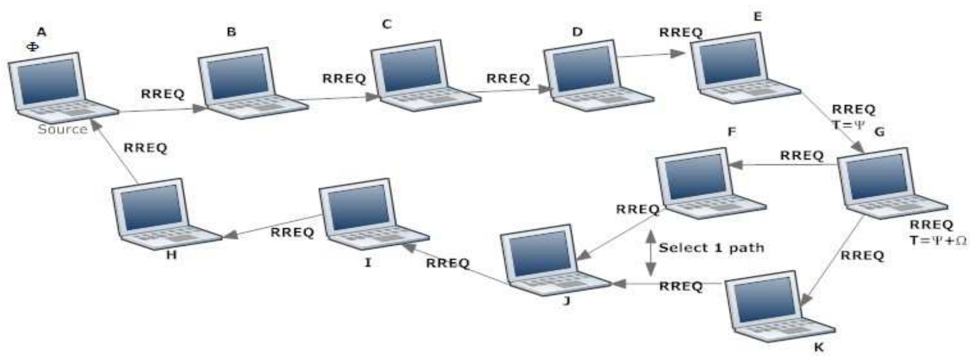

Fig. 4 RREQ process from destination upon receiving RREQ from Source

It is not that it will never construct a path with nodes in parent RREQ but its emphasis will be to gather least number of common nodes in both the paths. If such a packet reaches source node with all different nodes, then we have backup path for our default route. The main difference between this process and the last process is that the latter process is going to work well where we are having different upstream and downstream values of transmission rates.

These all algorithms need path development features to be initiated either by source or destination. Since the source or destination does not initially know how much it is going to use the resources, why should the source or destination waste time and resources in creating paths for bad times. Let the path break occur and then try something at runtime to cope with the situation. When a node moves out of the scope of the last node in the forward data path, then only the last node is aware of this change. So he will be the only object that can cope with this situation of disruption. The very first thing that it normally does is to send the RERR messages back to source. Now, we will change its work and make it a real decision making node in such a scenario. When noted about a movement in the next node, if the current node is aware of the second node in the path then it can try to find a fresh path to that node. For this there will be a little change to be made in the packet format of RREP message as shown in the fig 5. A next to next field will be added that collects the information of the second node in the path. It is to be noted that such a change is to be made in packet format of RREP as this node information will only flow at this stage of path formation. The routing table update is shown in Table 1. When a path is found to the second node, this change in path is appended to the original path deleting the old redundant entry. This update can either be forwarded to the source node or can be kept to be implemented at local node as shown in Fig. 6.

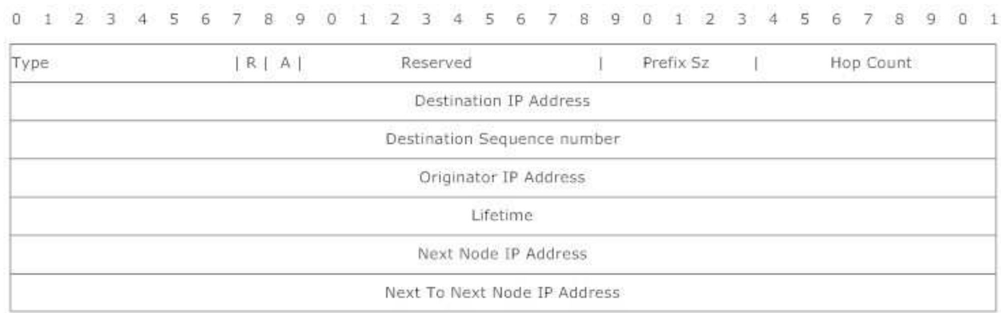

Fig. 5 Next to next field added to RREP packet

Table 1 Advanced Routing Table at a given node for local recovery

Destination Node	Next hop	Next 2 Next Hop	No. Of Hops	Destination Seq. No.	Active Neighbour Nodes	Expiration Timer
-	-	-	-	-	-	-

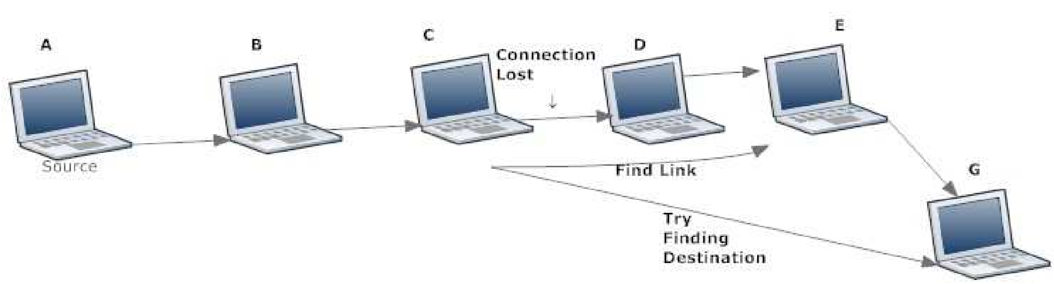

Fig. 6 finding path from last active node to second in line node or destination

If for some reason we are not opting for a path to the next node, to save the default packet format, then we can try searching the destination node from the last node after which the next has moved out of scope. This way we will not mess up with the packet format and there can be less overhead on path forming nodes. When the node gets a fresh new path to the destination, it appends that path to the original path, removing the older non usable path. In this process this has to be remembered that the last node will forward the RREQ packets only towards the destination node and not towards the source side. The default RREQ process will be applied in finding route from node to destination.

CONCLUSIONS

A lot of work that has been done in this field are innovative in nature, but none of them were able to completely rectify the problems occurring due to dynamic nature of the MANETs. Also the resource utilization also poses great difficulties in implementing good algorithms. My proposal may be said to be innovative to the very basic level of implementation, and can may be proved to be effective in different working environments and not limited to any one or all environments.

FUTURE WORK

We have just provided the algorithms for the advancements that can be made in the initial processes of the AODV routing protocol. Since each of them are just collection of words and their feasibility in real or simulated world is still not known, it is a matter of time to prove their applicability of acceptance in the wireless world. Also if we try implementing each of them in a single process them we will be able to get remedies to lot many security problems like black hole etc.

REFERENCES

1. An Overview of MANETs Simulation Luc Hogie 1,4 Universit´e du Luxembourg Luxembourg, Electronic Notes in Theoretical Computer Science, www.elsevier.nl/locate/entcs
2. RFC 3561 Ad hoc On-Demand Distance Vector (AODV) Routing C. Perkins Nokia Research Center University of California, Santa Barbara S. Das, University of Cincinnati, July 2003.
3. Unghee Lee A Proactive Routing Protocol for Multi-Channel Wireless Ad-hoc Networks June 14, 2006 Blacksburg, Virginia
4. Hongmei Deng, Wei Li, and Dharma P. Agarwal, “Routing Security in Wireless Ad Hoc Networks”, University of Cincinnati, IEEE Communications magazine, October 2002
5. Routing Protocols in Mobile Ad-hoc Networks, Krishna Gorantala Master’s Thesis in Computing Science June 2006
6. S. Murthy and J. J. Garcia-Luna-Aceves, “An Efficient Routing Protocol for Wireless Networks,” ACM Mobile Networks and App. J., Special Issue on Routing in Mobile Communication Networks, Oct. 1996, pp. 183–97.
7. Elizabeth M. Royer and Chai-Keong Toh, “A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks”, IEEE Personal Communications, pages 46-55, April 1999.

8. Performance Evaluation of ZRP over AODV and DSR in Mobile Adhoc Networks Using Qualnet SreeRangaRaju, Department of Telecommunication Engineering, Bangalore Institute of Technology, Bangalore, India European Journal of Scientific Research, ISSN 1450-216X Vol.45 No.4 (2010), pp.651-667

9. Royer E.M. Perkins C.E. Ad-hoc on-demand distance vector routing. Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, p.90, 1999.