MOBILITY AWARE ROUTING PROTOCOL IN AD-HOC NETWORK

Suman Halder[1], Partha Pratim Meta[2] and Sukla Banerjee [3]

RCC Institute of Information Technology, Kolkata, India
[1]sumanhalder.rcciit@gmail.com
[2]partha028@gmail.com
[3]sukla.banerjee@gmail.com

ABSTRACT

A Mobile Ad-hoc Network (MANET) is a collection of mobile nodes that communicate and collaborate with each other without reliance on any pre-existing infrastructure. In MANETs, wireless links are subject to frequent breakages due to nodes high mobility. While several routing protocols such AODV and DSR have been designed for MANETs, many of operate efficiently under low network mobility conditions and do not adapt well with high mobility conditions. Therefore, considering mobility is a demanding task that should be performed efficiently and accurately. Here, we proposed novel mobility-aware routing protocol based on the well known Ad-hoc On Demand Distance Vector (AODV) routing protocol called: MA-AODV (Mobility Aware Ad-hoc On Demand Distance Vector) in an attempt to improve the handling of high mobility factor in ad-hoc networks. MA-AODV protocols perform periodic quantification of nodes mobility for the sake of establishing more stable paths between source/destination pairs, hence, avoiding the frequent link breakages associated with using unstable paths that contain high mobile nodes.

KEYWORDS

MANET, Ad-Hoc, AODV, MA-AODV, Mobility Aware.

1. INTRODUCTION

A network is defined as a connection (may be wired or wireless) between a group of people or system or organization who tend to share their information collectively for their own purpose. And here each people or system or organization involve in the network is known as a node. An ad-hoc network is a self configuring network that connects mobile nodes which have no physical connection. This mobile node may be routers and/or hosts.

Routing is a process to find an efficient way through which two nodes can communicate with each other. Now routing protocol defined as the set of rules which are used at the time of routing means finding the path. Actually routing involves two major steps first one is finding the best path and the second one is after inventing the path sends the message from source to destination node. In Ad-hoc network there is no fixed topology as the nodes are mobile and multipath propagation and path loss is possible. Hence a dynamic routing protocol is needed to route data

Natarajan Meghanathan, et al. (Eds): ITCS, SIP, JSE-2012, CS & IT 04, pp. 17–31, 2012.

DOI : 10.5121/csit.2012.2103

and information from one node to another. There are many number of routing protocols have been developed to accomplish this task. Some of this protocols are DSDV (Destination Sequenced Distance Vector), AODV (Ad-hoc On-demand Distance Vector), Zone routing protocol (ZRP) etc.

As in MANET the nodes are mobile so the topology (topology of network actually referred to the structure if the network) is always changing and there could be a multipath propagation or path loss are interference by other signal may occurred. So in this case we need a dynamic routing protocol. There are two types of routing protocol – static and dynamic. Static routing protocol is applicable where the network topology is fixed for example in case of LAN connection where a physical connection exists between the nodes. And Dynamic routing protocol is applicable where there is no any fixed topology such MANET. There are several routing protocol is exist for Mobile Ad-Hoc network. They are mainly divided in three groups namely- 1.Proactive routing protocol. 2. Reactive routing protocol. 3. Hybrid protocol. There are a number of protocols are available under this category of protocols. We can explain by a picture.

In Ad-hoc network routing is challenging because in ad-hoc network there is no physical connection between nodes so routing is very challenging in case of ad-hoc network. The various are described in brief. Ad-hoc network use some sort of radio frequencies for the transition of data or information. So there is a chance of get obstruction by various things like wall or something like that. Interference due to weather or other frequency devices is also possible in that case. Multipath propagation is at the time of propagating of a signal from source to destination node it may possible that on the way of propagation of signal there is some obstacle which make the signal propagate in the path beyond the direct line of sight due to reflection, refraction, diffraction and scattering. Path loss is also a major problem in wireless communication. Path loss is reduction of propagated signal strength as it moves far from sender node. Path loss can be determined from the ratio of power of transmitted signal to the received signal. Interference is an another problem of wireless communication because when a signal passing through air there may be a number of signals present there which can interfere with each other and destroy the original signal. Security is one of the biggest problems in case of wireless communication. As wireless communication include no any wire for communication then the signal can be accessed by an unauthorized person very easily which is obviously not a secured way to send information. Again in ad-hoc network the nodes can be moveable that's why the position of the nodes is changeable randomly and also can join or leave the network. The main challenges in re-active routing protocols are 1) **High route discovery latency**: Re-active routing protocol required high latency time in route finding. This means that it does not discover a route until a flow is initiated. This route discovery latency result can be high in large-scale mesh networks. 2) **No reuse of routing info:** Re-active lacks an efficient route maintenance technique. The routing info is always obtained on demand, including for common cause traffic also. 3) **It is vulnerable to misuse:** The messages can be misused for insider attacks including route disruption, route invasion, node isolation, and resource consumption.

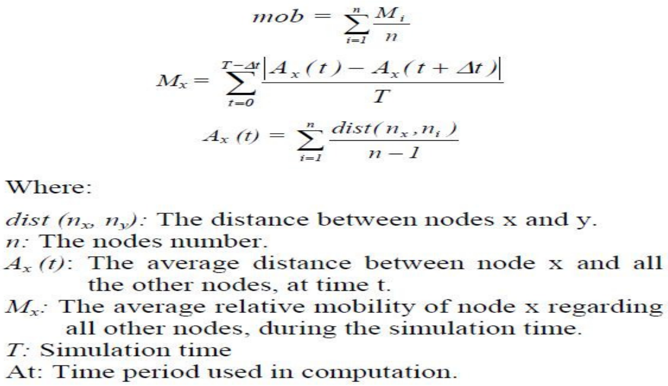
2. RELATED WORKS

According to the proposal of Idrees et al. [9] in mobile ad-hoc networks, mobility of nodes plays a vital role in the overall throughput of the network. As all nodes in an ad-hoc network also work as a router. So if a node is moving on a regular basis then it must not act as a router because it can cause path breakage very often. But there is no mechanism that an individual node can have any information about the mobility of other nodes in the network. Mobility awareness about the dynamic network topology can reduce this problem. They introduce a lightweight mobility aware agent on each node of the network that helps nodes to know about the mobility status of other nodes in the network. To achieve agent based mobility awareness all agents in the network broadcast their current location in the network periodically or semi periodically. Agents receiving this information update their knowledgebase for the mobility information of their neighbors i.e. stationary or in motion; that further helps to know about the direction of moving nodes. The status of those nodes, which are moving away from a particular node, is different from those, which are coming towards it. This movement is considered relative, if node-A is moving away from a static node- B then agent placed at node-A predicts it and can exploit this information for rerouting and link establishment. Placement of agents at network nodes help in many ways e.g. it can use different environmental statistics of network for example speed of a particular node can help for future predictions of network topology. In perfect agent based mobility aware network, parameter like high mobility of a node is recorded that helps in different intelligent decisions. In AODV the mobility status of any node is not considered at all while establishing a route between two nodes. So if a mobile node moves in such a way that it is no longer able to help in transferring data, a route error is generated causing a reroute discovery process. Same thing happen again and again establishing routes after quick intervals resulting in the reduction of overall throughput of the network. Mobility aware agent based ad hoc network prefers static nodes in route discovery. It also restricts highly mobile nodes to be the part of routes. More static nodes in the path help in establishing a route for a longer period, resulting in less reroute discoveries. Furthermore, network statistics helps in its future prediction. The Ad hoc On-Demand Distance Vector routing algorithm is initiated whenever a particular node wish to send data to some other node. AODV allows mobile nodes to obtain routes quickly for new destinations, and does not require nodes to maintain routes to destinations that are not in active communication. When links break, AODV causes the affected set of nodes to be notified so that they are able to invalidate the routes using the lost link. Although AODV allows mobile nodes to inform about link breakage in a timely manner, but their proposed scheme helps in reducing this link breakage by using the concept of mobility aware agent at each node. Following is the enhanced route discovery process of AODV using the concept of mobility aware agent. Agent at the source node broadcast a RREQ single when it determines that it needs a route to a destination and does not have one available in its knowledgebase. This can happen if the destination is previously unknown to the agent or if a previously valid route to the destination expire or it is marked as invalid. When an agent on a node receives RREQ, it records previous hop address; even it is received after the expiry time. Agent on second node generates RREP if it is itself destination or it has an active route to destination. This RREP is sent to best neighbor. In backward path best neighbor is that node which is not moving more frequently. This process of the selection of best neighbor process is done at each intermediate agent based nodes. In this way it is tried to establish a path between two nodes having maximum number of less mobile nodes. During the selection of best neighbor. The key advantage of using this protocol is, instead of using the high mobile node, it is using static node to establish the connection between sources to destination. So, it reduces the possibility of link break as well as the overhead of establishing connection between nodes. But the main problem in their proposal is that in most of the cases nodes are mobile in ad-hoc network. So it is hard to find static node and creating connection using them. Also proposed scheme is not applicable where all nodes are mobile in nature.

Yaser et al. [10] exploits two ideas. Their first protocol is Per Hop Mobility Aware AODV (PH-MA-AODV) and the second is the Aggregate Mobility Aware AODV (Agg-AODV). Their MA-AODV protocols offer major contributions and improve the performance of the original AODV protocol. ***i. PH-MA-AODV Protocol:*** In the traditional AODV, the source node initiates a route discovery process for the sake of finding the intended destination node. The same RREQ packet may pass through several paths (routes) to find its way towards the destination. Upon receiving the RREQ, the destination node replies back choosing the first route it gets, and consequently, other routes with possibly better metrics and Quality of Service (QoS) are dropped. Their contribution makes the traditional AODV conscious to the mobility when choosing the best route. In other words, in PH-MA-AODV, each node computes its own mobility periodically. Then, while initiating the Route Discovery process, each node decides to whether participate in the discovery process and thus relay the RREQ further or not. Therefore, the overall selected route is stable and more reliable. The main advantage of this protocol is instead of using the static node as proposed by Idrees et al. [9] it use the comparatively less mobile node to create the connection between source and destination node. Possibility of link breakage is also low. But the main drawback of this protocol is as each node has the rides to take the decision whether to participate in routing or not it is comparatively insecure and has a high possibility of compromising. An extra overhead for each node for taking decision whether to participate or not in routing. ***ii. Agg-AODV Protocol:*** The mechanism of PH-MA-AODV is partially applied in this approach in the sense that periodically, each node computes its own mobility. However, unlike PH-MAAODV, where the routing decision is made by the individual node as to whether participates in the route discovery process (based on that node's mobility), in Agg- AODV, upon receiving the RREQ packet, if the recipient node is not the intended destination, it adds its own mobility to the RREQ packet and forwards it further towards the destination. The destination node is responsible to store the aggregated value of mobility along the path from itself to the source, and to compare this value with future aggregated values that are obtained from other available paths towards the same source. If there are more than one active path between the source and the destination, the destination chooses the path whose aggregated mobility value is the least among all paths. It is worthy to note that upon receiving the RREQ packet for the first time through a specific path, the destination has no other choice but to select this path (regardless to its aggregated mobility value) to send RREP back to the source. However, the decision to whether continue using this path depends on the aggregated mobility of the subsequent paths. The main advantage of this protocol is as it also uses the less mobile nodes to communicate it has all the advantages as previous. One more advantage is that, here source takes the decision of choosing path. So it is more secure comparatively to the previous. But the main problem is this method is not aware of the direction of movement of nodes. It may choose the node which has less mobility value but the direction is outwards from the source. In that case the node has the possibility of leaving the network.

Nadjib et al. [12] presented an algorithm where mobility of a node has been calculated by the following way. According to the author, Mobility is an important parameter for MANETs routing protocols evaluation and depends on several parameters. These parameters were represented by node speed, or by the pause time when using the random way-point model. These representation, however, are meaningless and do not reflect the topology changes. Nodes may move either in high speed or a low pause time, but towards the same direction without causes any topological change. On the other hand, node may have a low speed or a high pause time, but they move away from each other, resulting in important topological change. This illustrates the weakness of this mobility representation.

A more explanation that better express the network topological change was proposed by Larsson et al [11]. Their proposal is based on relative node's movement, and represents by a parameter called mobility factor (mob) that's depends on both, the nodes speed and the movement pattern (direction). It is given by this following formula:



$$mob = \sum_{i=1}^{n} \frac{M_i}{n}$$

$$M_x = \sum_{t=0}^{T-\Delta t} \frac{|A_x(t) - A_x(t+\Delta t)|}{T}$$

$$A_x(t) = \sum_{i=1}^{n} \frac{dist(n_x, n_i)}{n-1}$$

Where:

dist (n_x, n_y): The distance between nodes x and y.
n: The nodes number.
$A_x(t)$: The average distance between node x and all the other nodes, at time t.
M_x: The average relative mobility of node x regarding all other nodes, during the simulation time.
T: Simulation time
At: Time period used in computation.

In this implementation, this parameter (called mobility factor) is computed for an each interval of time say, t = 0, t =At, t = 2At, ….. t = T. The main advantage of this proposal is this procedure conceder the mobility for each nodes and relative mobility among them. It also tries to takes the direction of movement in consideration. But the main disadvantage of this proposal is as it is just a logical implementation of this formulation. So, it is hard to implement.

3. NETWORK MODEL AND ASSUMPTIONS

Here we are assuming that all nodes are GPS enable and they are connected to each other with bi-directional antenna and range of each antenna is equal. Each node time to time requests the GPS device to get their position in current time (that is co-ordinate in geographical domain). So initially we are considering that each node of the network is aware of their current location with respect to geographical co-ordinate.

At any instant of time, an ad-hoc network can be described by a graph of the nodes (routers+ hosts). Two nodes are connected (i.e., have an arc between them in the graph) If they can communicate directly using their radios. Since one of the two may have a powerful transmitter than the other, it is possible that A is connected to B but B is not connected to A [Figure. 1]. However, for simplicity, we will assume all communication is symmetric.

4. METHODOLOGY AND ALGORITHM

Our objective is to minimize the problems caused by mobility of the nodes in AODV (mainly link breakage problem). So below we are proposing a methodology where an optimal static path is chosen instead of choosing shortest path all the time to send data from source to destination.

To meet our requirement we slightly modified the HELLO message of AODV which is broadcasted by each node among its neighbors to maintain its ROUTING TABLE. According to our assumption each node has its position co-ordinate (Altitude, Longitude, and Height). We are simply adding these three position parameters with the HELLO message field and broadcasting it among neighbors. When the HELLO message will reach to its neighbor node, the neighbor node will copy these three parameters to its ROUTING TABLE. The procedure will be followed by each and every node of the network. By this way each node is getting the information about the position of neighbors at free of cost for a certain interval of time.

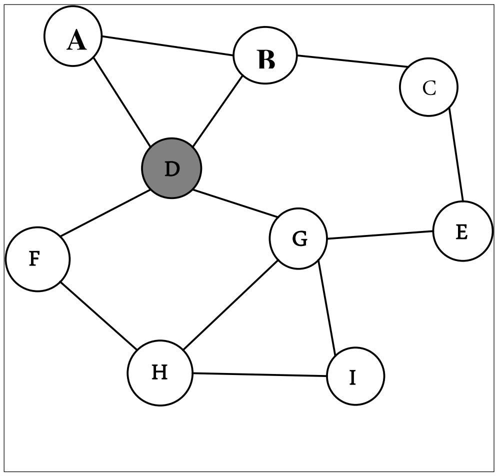


Figure 1: Graphical representation of an Ad-Hoc Network

According to [Figure 1] A, B, F and G are the neighbors of node D. After receiving the HELLO messages from neighbors (that is A, B, F, and G), node D updates the position field with its neighbor's corresponding position co-ordinate. A possible snap of D's ROUTING TABLE is given below. The use of last two fields will be discussed later. So node D has all the information about the position of its neighbors. By following the same procedure every node in the network maintains its ROUTING TABLE.

Table 1: ROUTING TABLE for node D

Dest.	Next Hop	Distance	Active Neighbors	Position (X_N,Y_N,Z_N)	Flag	Velocity
A	A	1	F,G	X_A,Y_A,Z_A	-1	
B	B	1	F,G	X_B,Y_B,Z_B		
C	B	2	F			
E	G	2				
F	F	1	A,B	X_F,Y_F,Z_F	1	
G	G	1	A,B	X_G,Y_G,Z_G		
H	F	2	A,B		0	
I	G	2	A,B			

Where,

X_N : Altitude of node N
Y_N : Longitude of node N
Z_N : Height of node N

4.1. Route Discovery Procedure:

Whenever a node (source) tries to send data to another node (destination) it will first look into its ROUTING TABLE. If source doesn't find entry for destination it constructs a special ROUTE REQUEST packet and broadcast it.

The format of ROUTE REQUEST packet is as follows,

Source Address	Request ID	Destination Address	Source Sequence	Destination Sequence

Here ***Source Address, Request ID*** used to identify the request uniquely. As source broadcasts the ROUTE REQUEST message to the network, so it is possible that same request may come to a node by different path. Here we have slightly modified the AODV algorithm. When a node receives the same ROUTE REQUEST message by more than one path, unlike AODV it will not simply discard the massage which has came later. It keeps track all the neighbors node by which the ROUTE REQUEST message appear, though the root may be a longer one. For example, if a ROUTE REQUEST message from node A [Figure 1] reached to node G by the following 3 root ADG, ABCEG and ADFHG. Then G will keep a reverse path entry for these three neighbors i.e. D, E and H in its local memory. Every node will follow the same procedure and maintain the record in a local database which is called REQUEST TABLE.

The REQUEST TABLE format is as follows,

Source Address	Request ID	Req. From Neighbor

Here, ***Source Address, Request ID*** used to identify the ROUT REQUEST uniquely among the several request made by other node also at the same time. These fields are copied from the ROUTE REQUEST packet. Rest of the procedure is same as AODV.

4.1.1. Action taken by a node when ROUTE REQUEST packet appears to it:

```
Step 1: Copy Destination Address from route request.
Step 2: If Destination Address = node address
        Step 2.1: Generate route reply.
        Step 2.2: Unicast route reply to the neighbor.
        Step 2.3: Discard route request.
Step 3: Else
        Step 3.1: Update REQUEST TABLE.
        Step 3.2: Broadcast route request to neighbours.
```

4.1.2. Action by a node to update REQUEST TABLE when it receives a ROUTE REQUEST packet:

```
Step 1: copy (Source Address, Request ID) from route request.
Step 2: Search for same entry (Source Address, Request ID) in REQUEST TABLE.
Step 3: if (Source Address, Request ID) doesn’t match the entry in REQUEST TABLE then,
        Step 3.1: Copy ROUTE REQUEST (Source Address, Request ID) to REQUEST
        TABLE (Source Address, Request ID).
        Step 3.2: Copy Address of the neighbor node from which route request appear to
        REQUEST TABLE (Req. From Neighbor)
        Comment: Enter (Source Address, Request ID) and address of the neighbor node (from
        which the request appears) into the corresponding fields of REQUEST TABLE (that is
        Req. From Neighbor).
        Step 3.3: Broadcast route request.
Step 4: If (Source Address, Request ID) matches with existing table entry then,
        Step 4.1: Copy Address of the neighbor node from which route request appear to
        REQUEST TABLE (Req. From Neighbor).
        Comment: Add address of the neighbor node (from which the request appears) to the
        corresponding field of REQUEST TABLE (last field) for the same (Source Address,
        Request ID) field value of REQUEST TABLE.
```

Step 4.2: Discard **route request**.

In AODV after appearing the ROUTE REQUEST message to the destination node, it generates a ROUTE REPLY message and unicast the packet to the source through the shortest path. Our algorithm mainly deals with this portion i.e. choosing of path by which the ROUTE REPLY message reaches to the source node. Before going in details let go back to **Table 1** to understand the remaining two fields. Our aim is to select a path (to transfer data) which is static in nature i.e. link breakage is minimum. To meet our requirements to send back the ROUTE REPLY message to the source, here we are considering those nodes which are converging to each other and have low velocity.

Being GPS enable each node knows the current location of its own and also knows the location of its neighbors from its ROUTING TABLE (which is time to time update by the HELLO MESSAGE). This piece of information is used to check whether the nodes are converging or diverging from its neighbor.

So from the two co-ordinates (one is the node's position co-ordinate and another is the position co-ordinate of its neighbor) it will calculate the Euclidian distance. Let explain it with an example, let at t1 time position co-ordinate of node D is (X_{Dt1},Y_{Dt1},Z_{Dt1}) and position co-ordinate of G is (X_{Gt1},Y_{Gt1},Z_{Gt1}). So distance among D and G at time t1 is:

$dist(t1)=sqrt((X_{Dt1} - X_{Gt1})^2 + (Y_{Dt1} - Y_{Gt1})^2 + (Z_{Dt1} - Z_{Gt1})^2)$ [*sqrt*: Square root function]

let after some time (at t2 time) position of D is (X_{Dt2},Y_{Dt2},Z_{Dt2}) and position of G is (X_{Gt2},Y_{Gt2},Z_{Gt2}). So distance among D and G at time t2 is:

$dist(t2)=\mathrm{sqrt}((X_{Dt2} - X_{Gt2})^2 + (Y_{Dt2} - Y_{Gt2})^2 + (Z_{Dt2} - Z_{Gt2})^2)$

So there velocity of converging or diverging is:

velo(t1,t2)=(*dist(t1)* + *dist(t2)*) / (t2 – t1). These *velo(ti,ti+1)* is the entry for the **Velocity** field of ROUTING TABLE. By following the same procedure a node can calculate its distance from its neighbor as well as its velocity at the time *ti,* where i=1, 2, 3, 4… and so on. Now if *dist(t1)* > *dist(t2)* then the node D and G have came to each other at time interval (t2 – t1). So D and G have converged to each other for that time instant. On the other hand if *dist(t1)* < *dist(t2)* then they have diverged from each other. By this way each node gets the information whether it is converging or diverging from its neighbor. **Flag** in ROUTING TABLE is used to keep track whether a neighbor is converging or diverging. Initially flag value is set to 0 if a neighbor of a node diverge from it then it set to +1 otherwise it set to -1. Let explain it with an example, if we consider the above figure [Figure 1], here node D has four neighbors A, B, F and G. Among them suppose node A and G are converging to each other. So, D will set its flag value to -1 for node A. On the other hand if node D and F are diverging to each other then it will set its flag value to 1 for neighbor F.

Format of the ROUTE REPLY message:

Source Address	Request ID	Destination Address	Destination Sequence	Lifetime

4.1.3. Action taken by a node when ROUTE REPLY message appear to it:

```
Step 1: Copy (Source Address, Request ID) from ROUTE REPLY.

Step 2: Search REQUEST TABLE for same entry (Source Address, Request ID).

Step 3: If (Source Address, Request ID) match found then,
    Step 3.1: Copy all address from REQUEST TABLE (Req. From Neighbor).
    Comment: Collect address of all neighbor nodes which are store in Req. From Neighbor field of REQUEST table for the same (Source Address, Request ID) field value.
    Step 3.2: For all neighbor node collected at previous step.
    Comment: The node wills conceder its ROUTING TABLE to calculate the following logic.
    Step 3.2.1: If no of converging neighbor node = 0
        Step: 3.2.1.1: Select the diverging node which has minimum velocity.
    Comment: If all neighbors are diverging from the node then choose one of the diverging node which has minimum velocity.
    Step 3.2.2: If no of converging neighbor node =1
        Step 3.2.2.1: If velocity of converging neighbor node < 2 * (minimum velocity among the neighbors)
            Step 3.2.2.1.1: Select the neighbor node to send route reply.
        Step 3.2.2.2: Else
            Step 3.2.2.2.1: Select the diverging node which has minimum velocity.
    Comment: If only one converging neighbor node is available then select this neighbor node to send the route reply message if the velocity of the neighbor node < 2 * (minimum velocity among the entire neighbor node). Otherwise choose the diverging node which has minimum velocity.
    Step 3.2.3: If no of converging neighbor node >1
        Step 3.2.3.1: For all converging neighbor node If velocity < 2 * (minimum velocity among the neighbors)
            Step 3.2.3.1.1: Select the converging node which has lower velocity.
        Step 3.2.3.2: Else
            Step 3.2.3.2.1: Select the diverging node which has minimum velocity.
    Comment: If more than one converging node is available then choose the converging node which has minimum velocity if the velocity of all of them < 2* (minimum velocity among the neighbors). Otherwise select the node which satisfies the above condition. If no one among the converging neighbor node satisfies the condition then select one of the diverging nodes which have minimum velocity.

Step 4: Else
    Step 4.1: Discard route reply.
```

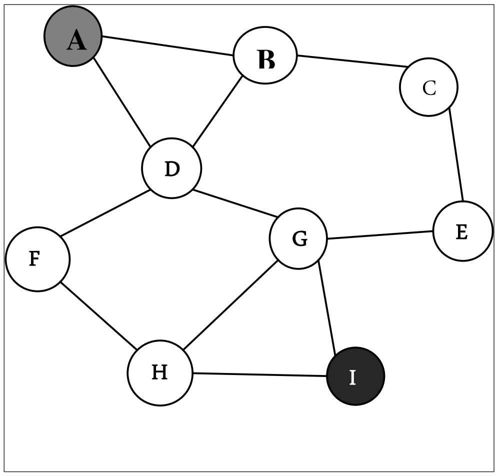


Figure 2: Graphical representation of an Ad-Hoc Network

These above procedure will be followed by each node while selecting the path for sending the ROUTE REPLY message to the source node. Let explain it with example. In [Figure 2] we assume that node a want to send data to node I. To do this A will broadcast the ROUTE REQUEST packet to its neighbor. Here we assume the ROUTE REQUEST packet reached to node I by two paths one is ADGI and another is ADFHI. So node I has two entries in ***Req. From Neighbor*** field of its REQUEST TABLE for the same (***Source Address, Request ID)***, one is G and another is H. Now if we assume the situation that, G and I are diverging to each other and H and I are converging to each other. So, when node I send the route reply message to the source, instead of choosing the neighbor G it will choose node H, as node H are converging in nature. Now when route reply will reach to node H, it also executes the same logic to choose its neighbor where it wills unicast the route reply message. These procedure will followed by each node until source node (that is node A) reached.

5. Experiment Results

We have simulated our proposed algorithm using JAVA Applet. The simulation procedure simultaneously executes two algorithms one is Mobility Aware Routing Protocol (MARP) which we have proposed here and another is Non Mobility Aware Routing Protocol (NMARP).

We have simulated these two algorithms in an 800 x 800 Applet window for different no of node of different velocity. A node is represented by a name (like n[0], n[1], n[2],…etc) and their position co-ordinate in applet window (like (150,300), (200,550), (700,400),… etc). A circle around a node represents its antenna range. When circle of two nodes intersects with each other that means they are within the range of each other, so they are connected and can communicate to each other. For example, if the circle of n[0] and n[5] intersect to each other that means they are connected and can exchange data to each other.

Our aim was to establish an optimistic static path from source to destination by which source can transmit data to the destination. In Applet Window a line between nodes represents this path.

5.1. Simulation Result:

We have simulated this two algorithm (MARP and NMARP) separately for different number of node. Below are the different scenarios for both the algorithm,

5.1.1. Scenario1:

No of Node: 9 Source Node: n[0] Destination Node: n[6]

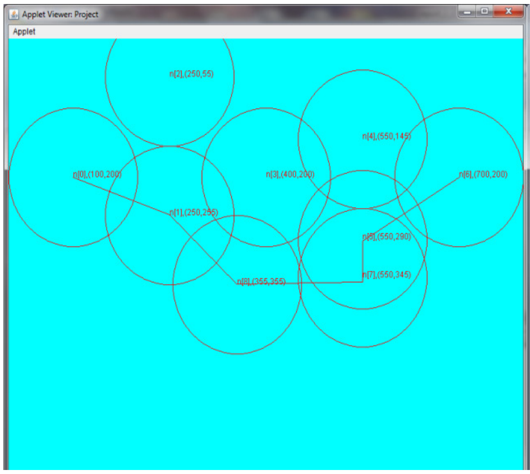


Figure 3: Showing Data Routing Path from Source to Destination for Non Mobility Aware Routing Protocol (NMARP)

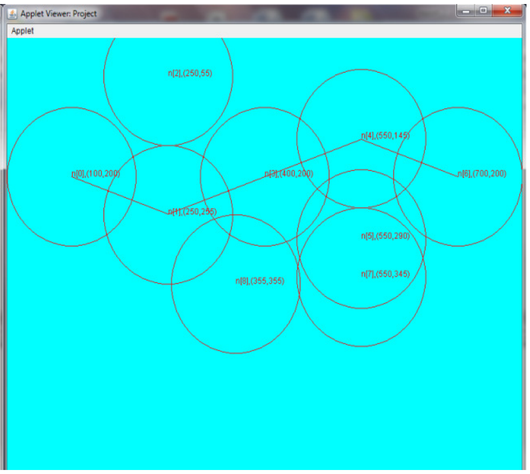


Figure 4: Showing Data Routing Path from Source to Destination for Mobility Aware Routing Protocol (MARP)

5.1.2. Scenario2:

No of Node: 11 Source Node: n[0] Destination Node: n[6]

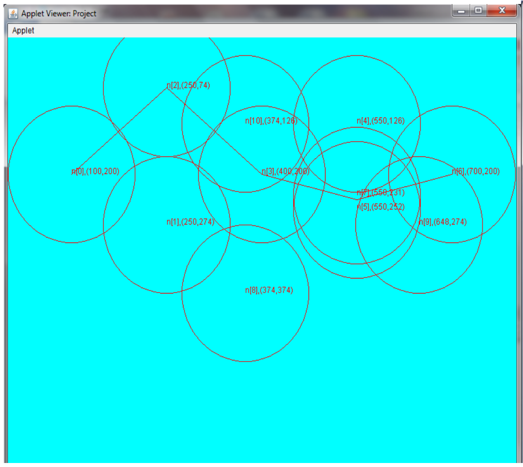


Figure 5: Showing Data Routing Path from Source to Destination for Non Mobility Aware Routing Protocol (NMARP)

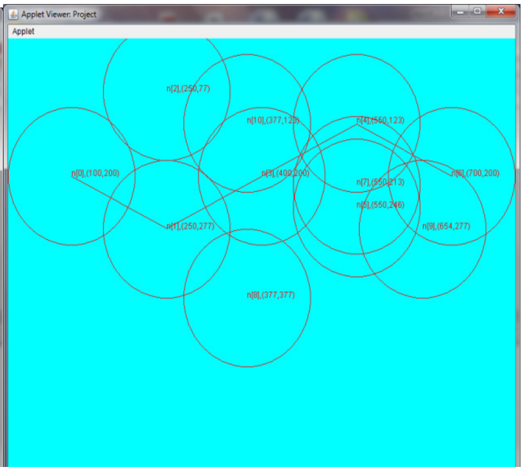


Figure 6: Showing Data Routing Path from Source to Destination for Mobility Aware Routing Protocol (MARP)

5.1.3. Scenario3:

No of Node: 13 Source Node: n[0] Destination Node: n[6]

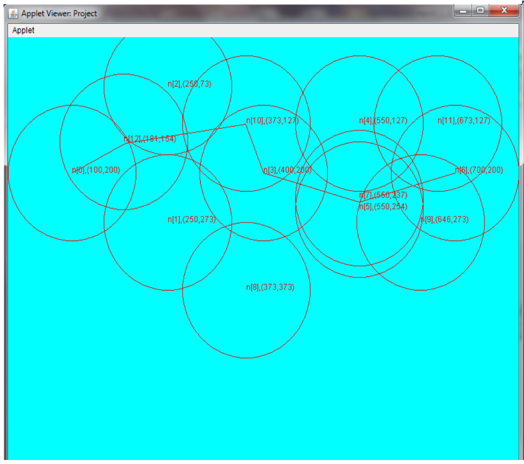


Figure 7: Showing Data Routing Path from Source to Destination for Non Mobility Aware Routing Protocol (NMARP)

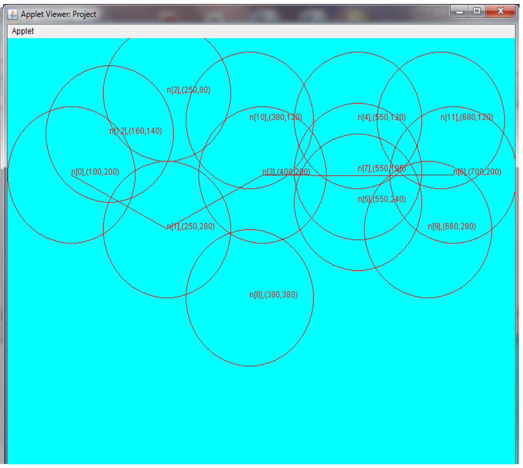


Figure 8: Showing Data Routing Path from Source to Destination for Mobility Aware Routing Protocol (MARP)

Based on such different scenarios we have calculated several statistical values. The table below represents these values:

Table 2: Comparison of no. of link break between NMARP & MARP for different no of nodes

No. of Node	No. of Link Break for NMARP	No. of Link Break for MARP
4	1	0
7	2	1
8	3	2
9	5	2
10	9	4
11	9	6
12	11	6
13	13	9

NMARP: Non Mobility Aware Routing Protocol.

MARP: Mobility Aware Routing Protocol.

Based on the above statistics we have generated the Line Graph and the Bar Graph.

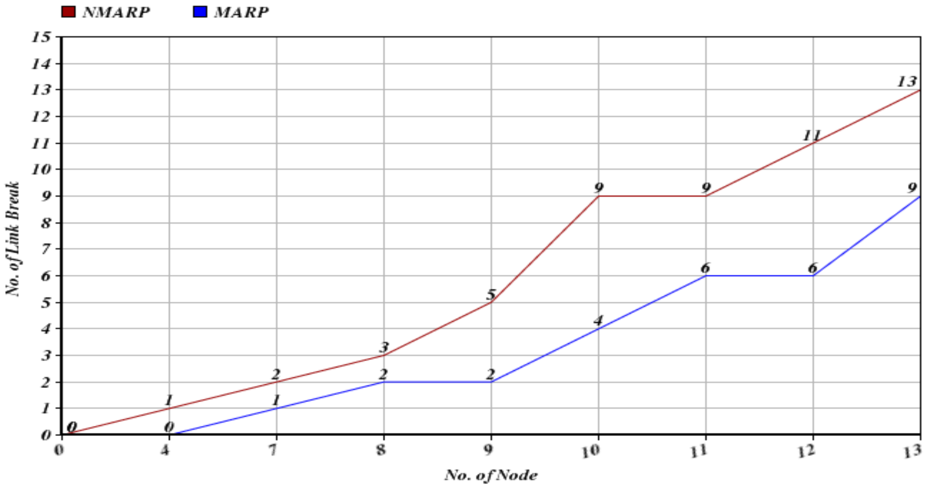


Figure 9: Graph Showing the Statistics for both NMARP and MARP

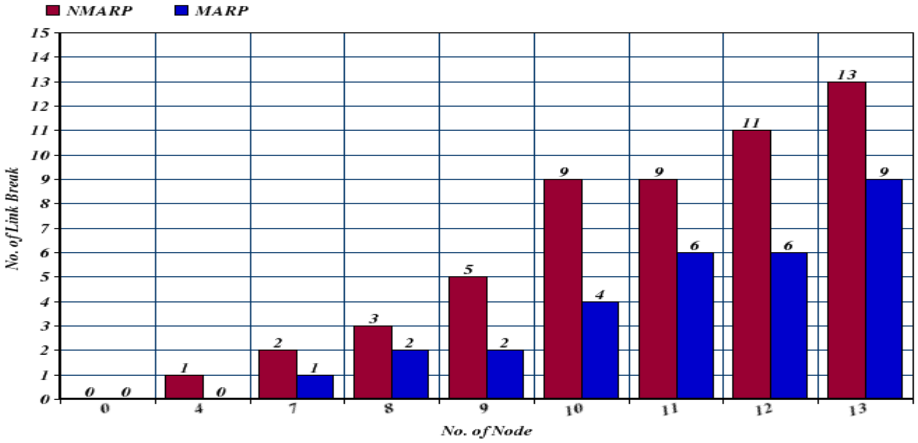


Figure 10: Bar Graph Showing the Statistics for both NMARP and MARP

From the above statistics it can be easily understand that frequency of link break in Mobility Aware Routing Protocol (MARP) is much lower than the Non Mobility Aware Routing Protocol (NMARP). If we consider this above two graphs then we can see that both the Line Graph and Bar Graph values of Mobility Aware Routing Protocol (MARP) lies below the Line Graph and Bar Graph values of Non Mobility Aware Routing Protocol (NMARP). So from the above comparative studies which have done for different no of nodes, it can be easily conclude that the Mobility Aware Routing Protocol (MARP) which we have proposed, always selects comparatively static path than the other protocols.

Advantages of Using This Protocol:

Frequency of link break among nodes is much lower in this protocol. So this protocol gives more static path between source and destination where nodes are mobile in nature.

As the rate of link break is much lower, so the rate of broadcasting of messages (Like HELLO Message, ROUTE REQUEST Message, ROUTE REPLY Message, ERROR Message, etc.) is also low in this protocol.

As the protocol use more static path to send data, so the frequency of broadcasting of messages (Like HELLO Message, ROUTE REQUEST Message, ROUTE REPLY Message, ERROR Message, etc.) is low in this case. So the energy loss for sending these messages is also low in this protocol.

This protocol used more stable path to send data from source to destination. So, it is very much suitable for sending large data where continuous connection is required among source and destination.

6. CONCLUSIONS AND FUTURE WORK

So in this routing protocol we emphasize on more stable path rather than shortest path all the time and as the protocol reduces the probability of link breakage the rate of broadcasting of ROUTE REQUEST, ROUTE REPLY, HELLO, ERROR messages are also reduce. The protocol reduce the topological changes, on the other hand it will also minimize the overhead of broadcasting messages. This protocol can be very efficient at the time of sending the large data where continuous connection among the source and destination is more preferable. If we consider the two graphs in previous section then we can see that both the Line Graph and Bar Graph values of Mobility Aware Routing Protocol (MARP) lies below the Line Graph and Bar Graph values of Non Mobility Aware Routing Protocol (NMARP). So from the above comparative studies which have done for different no of nodes, it can be easily conclude that the Mobility Aware Routing Protocol (MARP) which we have proposed, always gives stable path and selects comparatively static path than the other protocols.

In future we would like to perform extensive experiment with our algorithm using standard network simulator like Glomosim or NS2 or OPNET modeler. We also like to enhance our algorithm for other existing reactive routing protocols like ABR, DSR, TORA, CDRP, etc (other than AODV).

REFERENCES

[1] http://ntrg.cs.tcd.ie/undergrad/4ba2.05/group11/index.html

[2] http://www.acorn.net.au/telecoms/adhocnetworks/adhocnetworks.cfm

[3] http://monet.postech.ac.kr/research.html

[4] Routing Protocols in Mobile Ad-hoc Networks by Krishna Gorantala. SWEDEN, June 15, 2006.

[5] Ad Hoc Networks Routing Protocols and Mobility by Djamel Djenouri, Abdelouahid Derhab and Nadjib Badache. CERIST research center, USTHB University, ALGERIA.

[6] Wireless Ad-hoc Networks by Lu Han. October 8, 2004.

[7] AODV Routing Protocol by Georgy Sklyarenko. Seminar Technische Informatik, Freie University at Berlin, Germany.

[8] Ad hoc On Demand Distance Vector (AODV) Routing Protocol by Dr. Baruch Awerbuch & Dr. Amitabh Mishra. Johns Hopkins, 2005.

[9] Enhancement in AODV Routing Using Mobility Aware Agents by Idrees, M. Yousaf, M.M. Jaffry, S.W. Pasha, M.A. Hussain. IEEE 2005 International Conference on Emerging Technologies September 17-18, Islamabad.

[10] Mobility Aware Routing Protocols for Mobile Ad hoc Networks by Yaser Khamayseh, Omar M. Darwish, Sana A. Wedian. Published in: 2009 Fourth International Conference on Systems and Networks Communications, IEEE Computer Society Washington, DC, USA. ISBN: 978-0-7695-3775-7

[11] Routing Protocol in Wireless Ad-Hoc Networks by Larsson T. and Hedman N. Wireless Personal Multimedia Communications, September 2001.

[12] Ad Hoc Network Routing Protocols and Mobility by Nadjib Badache, Djamel Djenouri. The International Arab Journal of Information Technology , Vol. 3, No. 2, April,2006.

[13] Survey of Ad-hoc Routing Protocols by Jason Winnebeck, Benjamin Willis, Travis Thomas. Team Adhocracy Presentation3 – April 23, 2007.

Authors:

Mr. Suman Halder is a B.Tech Graduate in Computer Science and Engineering from West Bengal University of Technology, Kolkata, India.

Ms. Sukla Banerjee is an M.Tech in Information Technology. She is an Asst. Professor of RCC Institute of Information Technology, West Bengal, India

Mr. Partha Pratim Meta is a B.Tech Graduate in Computer Science and Engineering from West Bengal University of Technology, Kolkata, India.