History Based Reliability: A Novel Routing Metric in Mobile Ad Hoc Networks

Yaser Taj*, Karim Faez**
*Department of IT and Computer Engineering, Qazvin Islamic Azad University, Qazvin, Iran
**Department of Electrical Engineering, AmirKabir University of Technology, Tehran, Iran
yaser_taj81@yahoo.com, kfaez@aut.ac.ir

***Abstract*— A mobile ad hoc network has a dynamic topology. A route probably fails if intermediate nodes of the route have high mobility. In order to choose reliable nodes in a route, we define and quantify a novel routing metric called HBR that avoids choosing nodes with frequent failure history. Thus, route failure probability is decreased. We propose a routing scheme called HAODV based on the HBR metric. In HAODV, source node is able to determine its desired minimum reliability of route. HAODV is able to avoid waste of bandwidth caused by unreliable nodes. HAODV finds the most reliable route with a low overhead. Our simulation experiments show that HAODV improves packet delivery ratio when nodes are mobile.**

***Keywords*— Ad Hoc Network, Routing, Reliability, Failure, Mobility**

I. INTRODUCTION

Ad Hoc networks consist of mobile nodes which suffer from distributed non-organized management. Since nodes of such networks move randomly, their topologies always change leading to frequent route failure between source and destination nodes. Generally, routing schemes in ad hoc networks are categorized into Reactive and Proactive schemes. Proactive routing schemes (such as OLSR [1]) create a route table that contains an entry per destination. They always update the table and recalculate distances to all destinations. In reactive schemes (such as DSR [2] and AODV [3]), a new route is calculated between source and destination whenever there is a demand.

A stable route in ad hoc networks is a route which does not fail for an acceptable period of data transfer. In the same way, intermediate nodes of the route must have acceptable stability. An intermediate node is stable where it does not break the route by its movement. Mobility and energy are the main factors affecting route stability in mobile ad hoc networks (Fig. 1). In this paper, we focus on reducing the effects of mobility.

A. Related works

Finding stable routes has always been a challenging issue in wireless mobile networks. The first routing algorithm that analyzes route stability is ABR [4]. This algorithm is based on studying behaviors of mobile nodes in the network. This study concludes that mobile nodes (similar to human beings) stop for a few moments before starting a new move. The method aims at preferring stable links to unstable ones. Another work called SSA [5] follows a similar way. It separates strong links from weak ones. [6] combines SSA and ABR and tries to choose reliable links and increase route lifetime.

[7] proposes a scheme in which every node keeps a history table of its neighbors. The table contains status of neighbors. During the routing process, the table is analyzed and the route request packet is sent to neighbors who are in better conditions. SWORP [8] is a reactive routing scheme that finds the most stable route according to a weight parameter. Weight of a route depends on route expiration, number of errors, and hop count.

[9] focuses on mobility degree and connection time between nodes. Each node is categorized in one of fast, normal, and slow classes, so, a reactive or proactive routing scheme or both is used based on such classification. Every node in the scheme proposed in [10] frequently measures signal strength from its neighbors. When it receives stronger signal from a neighbor, it concludes that the neighbor has come closer.

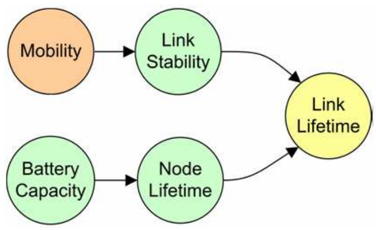

Figure 1. Two main factors affecting link lifetime

B. Our Contribution

We introduce a novel routing metric called HBR in mobile ad hoc networks. Using such a metric, all established routes get higher reliability. HBR tries to choose nodes that have the least failure history. HBR relies on this fact that a node that has broken numerous routes because of high mobility, will probably break new routes if selected as intermediate node since its mobility is expected to remain high for at least a short time. In the existing works on route failure, if a mobile node breaks a route, only neighbors of the node become aware of the route failure instead of the node. This study presents a strategy in which the route breaking node detects the route failure.

The rest of this paper is organized as follows. Section II defines and quantifies HBR. Section III incorporates HBR into

AODV. Section IV contains the simulation experiments of our algorithm. Finally, section V concludes the paper.

II. The HBR Routing Metric

In this section, we analyze the role of a mobile intermediate node in breaking a route and define a novel routing metric that keeps history of route-breaking nodes. The metric is based on the principle that if a node breaks a route because of its own mobility, then it is likely to break more routes for a while. Fig. 2 illustrates an example in which node N is heading south while the other nodes are relatively fixed. Node N breaks multiple routes because its mobility properties (including direction and speed) are different from its neighbors.

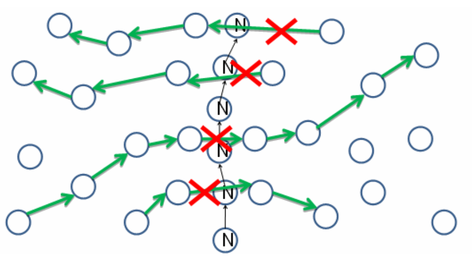

Figure 2. Node N breaks multiple routes while moving.

Therefore, by recording route failure history, we can distinguish unreliable nodes from reliable ones. When a route is to be calculated, we avoid choosing unreliable nodes as intermediate nodes.

A. Failure Coefficient

Let us consider the general example of Fig. 3 in which it contains node A as the source and node B as the destination. There is also node N which is an intermediate node on the route between A and B. All the three nodes are moving in a way that the distance between A and B is fixed. The following cases of node N's mobility can break the route.

1. Towards node A and out of nod B's communication range (Fig. 4).
2. Towards node B node and out of node A's communication range (Fig. 5).
3. Out of both node A's and node B's communication range (Fig. 6).

We want to define a routing metric that takes into account which nodes are responsible for route failure by their mobility. In all the three cases above, node N is responsible but node A or node B may condemn themselves since they are moving too. In order to avoid such mistakes, we define a failure coefficient α for each node involved in a route failure that shows the responsibility of the node in the failure.

Since a link fails due to disconnection of two neighbor nodes, two separate failure coefficients are considered for the two nodes when the link fails. The α assignments for the three cases above are as follows.

1. *α(B)=0.5, α(N)=0.5*
2. *α(A)=0.5, α(N)=0.5*
3. *α(B)=0.5, α(A)=0.5, α(N)=1.0*

In cases 1 and 2, we consider an α of 50 percent for node N because this node is available through at least one of its neighbors, therefore the route would possibly be repaired. In other words, the node was responsible for breaking the route as much as 50 percent. α(N) is considered as 1 in the third case because node N got away from its neighbors in a way that it was not accessible anymore. In other words, it was the main culprit. In addition to node N, other nodes (A, B) are directly influenced by the failure. It is a reason for considering α for A and B.

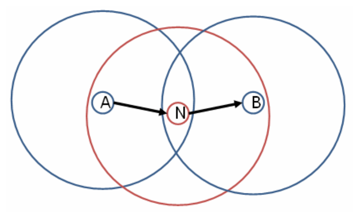

Figure 3. A route between node A and node B through node N.

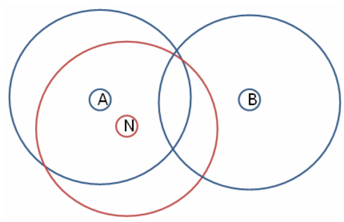

Figure 4. Node N goes out of node B's communication range.

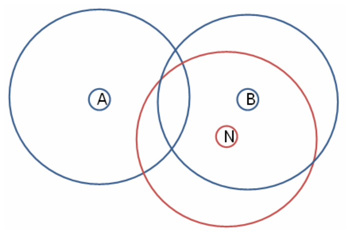

Figure 5. Node N goes out of node A's communication range.

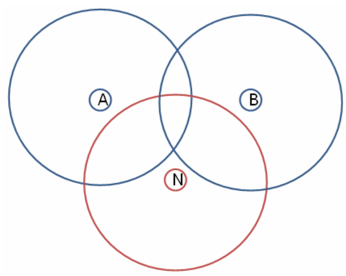

Figure 6. Node N gets away of both node A and node B.

B. Calculating HBR

Every node contains a route failure table to keep its history of route breaking. As table I shows, it includes three fields including 1) Route Identifier, 2) Failure Time, and 3) Failure Coefficient. A record is recorded in this table per failure.

We only consider route failures which are caused by mobility. We assume that the direction of a mobile node averagely lasts for D seconds in the network. To keep up-to-date information in the table, each record is removed after D seconds. The value of D is different based on network properties.

We define HBR (History-Based Reliability) of node, R(node), in (1).

$$R(node) = (n - \Sigma\alpha(node)) / n \quad (1)$$

where n is the number of record in the failure table of the node. $\Sigma\alpha$(node) is the summation of all the n failure coefficients in the table.

Whenever the node breaks a route, its failure table is updated and then R(node) is recalculated. Therefore, during the routing process, it is not necessary to calculate R(node).

A node is selected as intermediate node of a route if

$$R(node) >= Threshold \quad (2)$$

where Threshold reflects the minimum acceptable reliability.

TABLE I. AN EXAMPLE OF ROUTE FAILURE TABLE

Route Identifier	Failure Time	Failure Coefficient
Route1	Time1	0.5
Route2	Time2	0.5
Route3	Time3	1
Route4	Time4	0.5
Route5	Time5	1

III. HBR-BASED AODV ROUTING

In this section, we incorporate HBR as the main routing metric in the AODV [3] routing algorithm and thus create a new algorithm called HAODV (HBR-based AODV). This shows the procedure of incorporating HBR in a routing protocol and makes us able to compare a typical routing protocol with and without HBR. We assume the default behaviors and specifications of the AODV protocol and only present the changes to the routing algorithm.

A. Changes in Route Request/Reply Packets

We add two fields, *Reliability* and *Threshold* into the RReq (or Route Request) packet and a *Reliability* field into the RRep (or Route Reply) packet. When a source node is going to find a route to a destination node, the source creates an RReq packet with a unique sequence number and initially sets the reliability field to 1 and sets the minimum acceptable reliability of the route in the threshold field of the RReq. Then, the source broadcasts the RReq.

When a node receives a new RReq (an RReq that has a newer sequence number than the received RReqs with the same source and destination), it multiplies its R(node) at the reliability field of the RReq and then saves the result into the reliability field of the RReq. At this time, the reliability field shows the reliability of the route from the source node to the current node. If the reliability field is lower than the threshold field, it means that the route is not reliable enough. In this case, the node does not rebroadcast the RReq packet. Otherwise, the node broadcasts the RReq again and let it spread more in that neighborhood of the network.

If the node receives another RReq with the same sequence number, source, and destination of previous RReqs, the node rebroadcasts the new RReq unless the new RReq contains an equal or a lower reliability field than the previous RReqs. This enables the source to find the most reliable route.

The reliability threshold avoids propagation of RReq on routes that have unacceptable reliabilities. The source determines the threshold based on application. Assigning a too-high threshold leads to very limited propagation of RReq in the network and assigning a too-low threshold does not bypass routes with unacceptable reliabilities.

When the destination node receives an RReq, it multiplies its R(node) at the reliability field of the RReq and then saves the result into the reliability field of the RReq. At this time, the reliability field shows the reliability of the route from the source to the destination. If the reliability field is more than the threshold field and no RReq with the same sequence number, source, and destination that has a higher or an equal reliability is replied, then the destination generates an RRep packet and sets the reliability field of it to the reliability field of the RReq. Then, the destination sends this RRep to the source.

B. Route Learning

AODV contains a route learning mechanism in which a typical node *v* learns the next hop to the source of an RReq when it receives the RReq and node *v* learns the next hop to the destination of an RRep when it receives the RRep without trying to find a route. Node *v* can use these learned routes if needed.

We have to change this route learning mechanism since a learned route must contain a correct reliability field. The ways of reliability calculation differs when receiving an RReq from receiving an RRep. An RReq definitely contains the reliability of the route from the source to the receiving node, but an RRep always holds the reliability of the route from the source to the destination. If node *v* learns the reliability of the route from the source to itself when receiving an RReq, the node is able to compute the reliability of the route from the destination to itself when it receives the RRep of the RReq. Node *v* just needs to divide them as (3).

$$Reliability(destination, v) = \frac{Reliability(source, destination)}{Reliability(source, v)} \quad (3)$$

where $Reliability(n_1, n_2)$ is the reliability of the route from node n_1 to node n_2.

IV. Simulation and Comparison

We simulated HAODV (along with AODV and SWORP [8]) using the GloMoSim [11] network simulator. In this section, we present the simulation results and compare HAODV with AODV and SWORP. AODV is an example of routing protocols that do not take reliability into account. SWORP is a recent example of the existing routing protocols that try to find reliable routes.

SWORP and HAODV take advantage of different routing metrics that have no conflict in a way that combining them results in a new routing protocol that surpasses both SWORP and HAODV. However, we want to show the results of considering HBR in this paper.

We define two different scenarios (Table II) and are seeking the following outputs in each scenario. In scenario (1), we change number of nodes while mobility speed is fixed. In scenario (2), we change mobility speed while number of nodes is fixed.

- Routing Control overload
- End-to-end Delay
- Packet Delivery Ratio

A. Simulation Results

Fig. 7 and 8 show the control overhead of HAODV versus number of nodes and mobility. Increase in number of nodes increases requests for routing and thus makes more control overhead. A higher degree of mobility increases route failures and needs more rerouting, thus leads to more control overhead. The curve in Fig. 7 rises rapidly whereas the curve in Fig. 8 rises slowly. Since HAODV is able to avoid unreliable mobile nodes from routes, it needs less rerouting and leads to less control overhead than AODV and SWORP.

Fig. 9 and 10 show the end-to-end delay of HAODV. In addition to hop count, end-to-end delay strongly depends on traffic load balance on network links when traffic load is high. HAODV and SWORP achieve more balance of load than AODV that always finds shortest paths leading to huge load on the center of the network. Employing a load metric [12] in HAODV and SWORP further improves load balance.

Packet delivery ratio shows the average ratio of packets which each destination node receives. This depends on both route reliability and traffic congestion. Increasing number of nodes increases congestion and increasing mobility leads to lower reliability. As illustrated in Fig. 11 and 12, HAODV achieves higher packet delivery ratio than AODV and SWORP.

TABLE II. Simulation Parameters

Parameter	Value	
	Scenario (1)	Scenario (2)
Simulation Duration	100 minutes	
Reliability Threshold	0.3	
Number of Nodes	20 to 50	50
Mobility Speed	10 (m/s)	2 to 14 (m/s)
Mobility Model	Random Waypoint	
Transmission Range	100 m	
Route Lifetime	5 seconds	
Topology	Nodes are uniformly distributed in the network.	
Data Traffic	Every node sends data to a unique destination at the constant rate of 2Kbps.	
Network Size	600m × 600m	

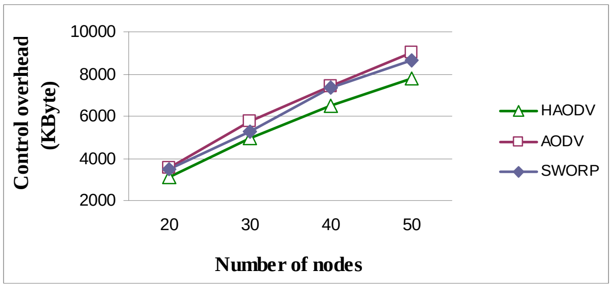

Figure 7. Routing control overhead vs. number of nodes.

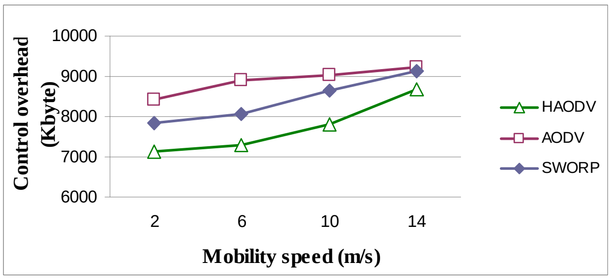

Figure 8. Routing control overhead vs. mobility speed.

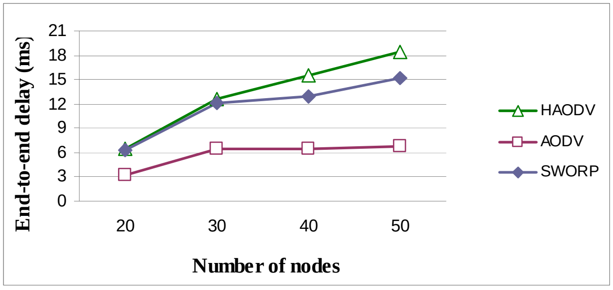

Figure 9. End-to-end delay vs. number of nodes.

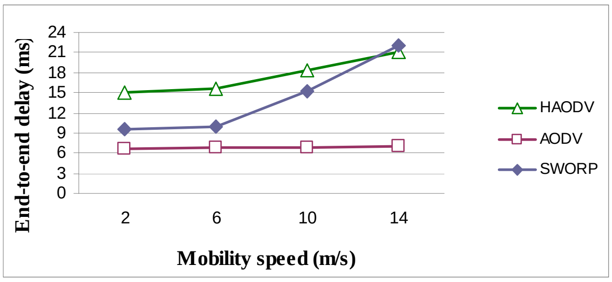

Figure 10. End-to-end delay vs. mobility speed.

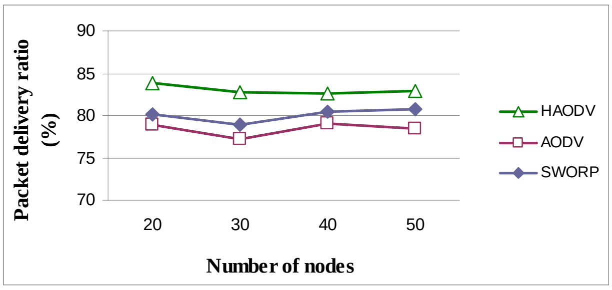

Figure 11. Packet delivery ratio vs. number of nodes.

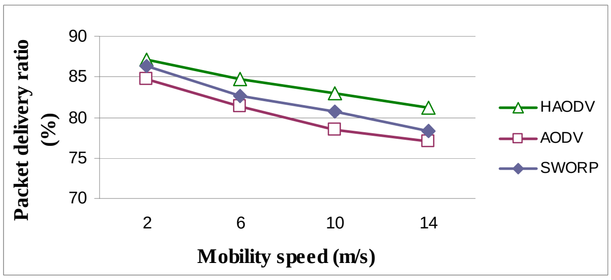

Figure 12. Packet delivery ratio vs. mobility speed.

V. CONCLUSION

In this paper, we introduce a novel routing metric, called HBR, in wireless ad hoc networks that takes into account the reliability of nodes based on their route failure history. We propose a method to quantify this metric. Then we present the way of incorporating this metric in a popular routing protocol, AODV, leading to HAODV.

HAODV shows the following advantages over AODV.

- Source node is able to determine its own minimum reliability of routes.
- An extremely unreliable intermediate node drops data packets at an unacceptable rate. Therefore, the bandwidth consumed for delivering an RReq from such an unreliable node to the destination is wasted. Since HAODV is able to avoid such waste of bandwidth, it leads to less control overhead than the original AODV when there are a number of extremely unreliable nodes in the network.
- A node in HAODV is able to learn a number of routes with correct reliability when another node is routing.
- Generally, the most reliable route with a low overhead is found in HAODV.
- Our simulations show that HAODV improves packet delivery ratio when nodes are mobile.

REFERENCES

[1] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L.Viennot, “OLSR: Optimized Link State Routing Protocol”, IEEE INMIC 2001, pp. 62 - 68, Dec. 2001.

[2] D. B. Johnson, D. A. Maltz, and Josh Broch, “DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks”, in Ad Hoc Networking, edited by Charles E. Perkins, Chapter 5, pp. 139-172, Addison-Wesley, 2001.

[3] C. E. Perkins, “Ad hoc On Demand Distance Vector (AODV) Routing”, Internet Draft, draft-ietfmanet-aodv-01.txt, 1998.

[4] C. K. Toh, “Associativity-based routing for ad hoc mobile networks”, Wireless Personal Communications, 4(2):103–139, March 1997.

[5] R. Dube, C. D. Rais, K.-Y. Wang, and S. K. Tripathi, “Signal Stability based Adaptive Routing (SSA) for Ad-Hoc Mobile Networks”, IEEE Personal Communication, February 1997.

[6] G. Lim, K. Shin, S. Lee, H. Yoon, and J. S. Ma, “Link Stability and Route Lifetime in Ad-hoc Wireless Networks”, Proceedings of the International Conference on Parallel Processing Workshops (ICPPW’02) IEEE, 2002.

[7] S. Medidi, and P. Cappetto, “History-based route selection for reactive ad hoc routing protocols”, 15th IEEE International Conference on Networks, Nov. 2007, Page(s):25 – 30.

[8] N. C. Wang, Y. Huang, and J. Chen, “A stable weight-based on-demand routing protocol for mobile ad hoc networks”, Elsevier Information Sciences 177, 2007, p 5522–5537.

[9] A. Bamis, A. Boukerche, I. Chatzigiannakis, and S. Nikoletseas, “A mobility aware protocol synthesis for efficient routing in ad hoc mobile networks”, Elsevier Computer Networks 52, 2008 p130–154.

[10] B. Ramachandran, and S. Shanmugavel1, “Received Signal Strength-based Cross-layer Designs for Mobile Ad Hoc Networks”, IETE technical review, vol 25, issue 4, 2008, p192-200.

[11] L. Bajaj, M. Takai, R. Ahuja, K. Tang, R. Bagrodia, and M. Gerla, “GloMoSim: A Scalable Network Simulation Environment”, UCLA, Computer Science Department, Technical Report, 990027, 1999.

[12] S. Yang, J. Li, J. Yun, I. Lee, and K. Han, "A Routing Metric for Load Balance in Wireless Mesh Networks", International Conference on MultiMedia and Information Technology, pp.614-617, 2008.