Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/090151

A New Zone Disjoint Multi Path Routing Algorithm to Increase Fault-Tolerant in Mobile Ad Hoc Networks

Hoda Rafiee Pour[1], *Marjan Kuchaki Rafsanjani*[2,*] *and Hamid Saadat*[3]

[1] Department of Computer, Science and Research Branch, Islamic Azad University, Iran
[2] Department of Computer Science, Shahid Bahonar University of Kerman, Kerman, Iran
[3] Islamic Azad University, Tehran, Iran

Received: 19 Apr. 2014, Revised: 20 Jul. 2014, Accepted: 21 Jul. 2014
Published online: 1 Jan. 2015

Abstract: In Mobile Ad-hoc Networks, there is no fixed infrastructure, router and station and so all network operations such as routing are being done with nodes. Some routing algorithms of Mobile Ad-hoc Networks perform routing in a multi-path manner and simultaneously discover and record a number of paths in a path discovery process. In these algorithms, it is possible to send information to destination concurrently using several paths, in order to balance the load in the network, increase end to end bandwidth, and decrease delay. In this case, selecting node disjoint paths is one of the best options because of increasing error tolerability. Given channel access mechanisms, sending information through completely disjoint paths is not fully independent.

To solve the problem, we can use zone disjoint paths rather than node disjoint paths. Two paths are fully zone disjoint ones if none of the nodes in a path is adjacent to the nodes in the other. Methods have been proposed to discover zone disjoint paths; one of them is using directional antennas but these are not available in most of current facilities. The other method is Zone Disjoint Ad-hoc On-demand Multipath Distance Vector (ZD_AOMDV) which has high overload while routing and additionally we see the very high delay during path discovery phase.

A new method will be proposed in this paper which has been applied in the proposed algorithm ”Improved Zone Multipath Ad-hoc On-demand Distance Vector” (IZM_AODV). In the algorithm, it is possible to discover zone disjoint paths using typical, omnidirectional antennas with minimum overload and delay in path discovery phase. In addition, these paths are used to send information concurrently. The proposed algorithm has been simulated in different scenarios using NS2 software and it has been examined and compared with multipath algorithms (AOMDV and ZD_AOMDV) and basic algorithm (AODV). Routing overloads, sending packets successfully, and end to end delays have been improved compared to existing methods.

Keywords: Mobile Ad-hoc Networks (MANETS), Node Disjoint Paths, Multipath Routing Algorithm, Zone Disjoint Paths

1 Introduction

Today, mobile ad-hoc networks have experienced impressive growth because of their applications and services. The networks are going through a fast development and offered services are increasing continuously too. In the near future, information technology (IT) will be based on wireless communications; since they have dynamic topology and they need not any pre-determined structure or central management as well as all network operation like routing is done by the nodes which work together and because of limited potency of mobile nodes in data sending, the nodes have to use several intermediate nodes; Hence, routing is the most important issue of these networks [1, 2] and so fault-tolerable algorithms should be designed in a way that even if failure occur in the networks or some nodes show inappropriate behavior, the algorithm must be guaranteed a specified rate of packets destined to deliver [3]. To do this, one of the proper methods is using multipath routing algorithms that they maintain some different paths between source and destination. It has some benefits such as increased fault tolerability, reducing end-to-end delay, and increases the bandwidth [4, 5]. Using single path routing will result in poor distribution network load, latency and packet loss due to the failure of path as a result of the mobility of nodes. Hence, the necessity of using multipath routing arises [6]. In the multipath routing, due to the simultaneous use of multiple paths for data transmission, we will see that density and

* Corresponding author e-mail: Kuchaki@uk.ac.ir

load distribution reduce in several directions [7]. To increase tolerability against failure, the more the chosen paths are distinct and have fewer joints, we will achieve better results [3] because in the event of failure (whether in nodes or joints), less paths are destroyed. Most of multipath routing algorithms use node-disjoint paths to send information to a destination and they achieve desirable results about tolerability against failure [8]. In the case, it seems that their sending delay is completely independent because these paths are disjoint. But after reviewing access control layer, it appears that there is an inevitable competition between nodes to take possession of the common channel; on the other hand, Multiple Access with Collision Avoidance (MACA) protocol has been proposed to avoid collisions in access control layer and to resolve classic problems - Hidden Terminal Problem and Exposed Terminal Problem [9], which is used in IEEE 802.11 standard for access to channels. The protocol uses Clear To Send (CTS) and Request To Send (RTS) to make coordination between stations. The station wants to send data to another one and indeed wants to take possession of common channel within its radio range, initially sends an RTS message and determines that it is going to send N bytes to a specific destination. After receiving the message and if the exchange is agreed, the destination station sends a CTS message back to the source and determines that it is ready for receiving N bytes from a specific transmitter source. Meanwhile, other stations which receive the CTS message must remain silent due to respect to receiver station and to avoid any possible collision. Therefore, it is possible nodes in a route postpone their sending because of receiving CTS messages from nodes in other paths. In other words, even though two paths are node disjoint paths, perchance one of them has nodes which are merely adjacent to nodes of other path and for this reason it receives a CTS message from node of other path; then the node has to be silent and in this case delay of sending in a path will be depend on the traffic on the opposite path. To solve this problem, we use zone disjoint paths in the proposed algorithm; i.e. in the concurrent information sending, some nodes are being used that at best they are not adjacent to nodes of another path or they have the least number of neighbors. However, some solutions have been presented in the [10] in which either the network must have directional antennas or path discovery process has to be performed by sending inquiry packets to determine the number of neighbors (the method has high delay and overload). Given this situation, number of sent packets are higher than data packets [11].

In the present paper, the path discovery process of proposed algorithm has been enhanced in a way that neither it need to directional antennas to detect zone disjoint paths nor necessity to send inquiry packets during the path discovery process to find the number of active neighbors. Thereupon, a path discovery process in this algorithm will be done with the least delay and routing overload.

The rest of the paper is composed of the following parts: section 2 reviews related works to multipath routing in mobile ad-hoc networks. Section 3 examines the proposed algorithm in detail and section 4 expresses about simulation results . Finally, section 5 will be about conclusion and future works.

2 Related Works in Multipath Routing

Many works have been performed regarding multipath routing in mobile ad-hoc networks and also many algorithms have been proposed; however, research in this field continues. Some multipath routing algorithms in mobile ad-hoc networks are reviewed in this section. One of the multipath routing algorithms, Ad hoc On-demand Multipath Distance Vector (AOMDV) has been proposed in [12]. The algorithm has been developed based on AODV algorithm [13] and it is proposed to provide stability in the network against potential failures. To this end, some changes will be made in different processes of AODV algorithm. In this algorithm, several connecting disjoint paths will be discovered during the path discovery process. Each node maintains two next steps per destination. If one of the connections disappeared throughout communication, then the intermediate node will use next alternate step. However, the algorithm has a weakness because it is possible that the next alternate step ceased to exist too during communication. And when the next alternate step is required, its connection has been lost. One method has been proposed in [14] to perform multipath routing using zone disjoint paths and directional antennas. In this method, each node alternatively inserts information of its adjacent nodes in a table. The information includes radio signal strength between two adjacent nodes in a specific angle. To do this, the nodes must be equipped with radio directional antennas. After this phase, when a source needs a path to the destination, it first tries to identify all node disjoint paths to the destination. Afterwards it selects zone disjoint paths among these paths. In this mode, the source can repeat computations of identifying zone disjoint paths alternatively and update its routing decision. It should be noted that the advantages and disadvantages of multipath algorithms, which use directional antennas, have been discussed in [14]. The biggest problem with this idea is that most existing equipments are not equipped with directional antennas.

To solve this problem, the ZD_AOMDV routing algorithm has been proposed in [11]. It can identify the paths with least active neighbors without using directional antennas.

ZD_AOMDV algorithm has been developed based on AODV routing algorithm. In general, it can be said that ZD_AOMDV algorithm tries to choose zone disjoint paths from receiving path request packets and it sends the right path to source through these paths. To discover zone disjoint paths between source and destination, a new field

("ActiveNeighborCount") with initial value of zero is created in the header of path request packets. In fact, the field is the number of active neighbors of nodes which previously received the same path request packet and it is likely that source and destination have selected another path which passes through the node to exchange information. If so, sending information through these two paths is interdependent. It is evident in this algorithm that after receiving path request packets and before sending them, each intermediate node asks its neighbors whether they have seen the packet with same specifications. To do this, it sends a PREQ_Query packet to its neighbors and after setting a timer waits specific times for responses of its neighbors. After the timer expires, the node will increase value of ActiveNeighborCount field (within path request path) by the number of positive responses received from neighbors and subsequently send it to all. When the packet reaches its destination, a response path packet is sent to source per path request packet. The source will choose the steps with least ActiveNeighborCount field from path response packets. Then it divides the loads in existent paths on the basis of considered criteria for balancing loads. By detecting zone disjoint paths and sending information from these paths concurrently, ZD_AOMDV algorithm increases failure tolerability, increases successful packet sending rate, and reduce end to end delays but path discovery process is performed to count active neighbors in order to identify paths which have least number of neighbors with high routing overload because it use PREQ_Query and PREQ_Query_Reply packets to calculate number of the path's active neighbors and it results in the fact that finally routing packets will be more than data packets. To solve this problem, we propose our algorithm ("IZM_AODV") which results in significant decreases in routing overload by introducing some changes to the path discovery process.

3 The Proposed Algorithm IZM_AODV

In this section, the IZM_AODV algorithm (enhanced type of zone multipath routing based on AODV) is proposed for mobile ad-hoc networks which is a multipath algorithm with zone disjoint paths and the information will be sent concurrently through these paths.

(a) *Differences of IZM_AODV routing algorithm and AODV algorithm*

Unlike AODV algorithm, on the one hand, the proposed algorithm allows more PREQ packets reach to the destination (for this reason, passed path will be inserted in the header's PREQ packet) and, on the other hand, it allows PREQ packets traverse their path to the destination node in order to count the number of active neighbors in a path correctly.

Unlike AODV algorithm, here PREP packets need ActiveNeighborCount field in their header.In the proposed algorithm, intermediate node needs a PREQ_Seen table to insert properties of received PREQ packet. There is a Counter field in the table and the intermediate node increases its value by one after receiving repetitive path request packet.

Routing tables in the proposed algorithm need ActiveNeighborCount field.

(b) *Procedure of the algorithm*

In the IZM_AODV, source node begins a path discovery process by sending a request packet based on the applied principles for all requests-based routing algorithm when it needs a path to a specific destination. The packet must have the ability to store passed node's addresses.

Once the source generated and sent path request packet, all his neighbors receive the packet. In this case, the intermediate nodes that receive the path request packet, list its specifications in Table RREQ_Seen of their own. There is a field for every RREQ as counter in this table. Each node (whether it is a broadcast node or must be removed) increases the counter field of this PREQ in the PREQ_Seen table by one. The field actually presents the number of the node's active neighbors for the PREQ.

Then, to avoid loops, newness of the path request packet is considered by referring to the header of the packet and examining the field corresponds to addresses of passed nodes. If the intermediate node finds its address in the header of the path request packet, it means that the same packet already has been received and directed and now the node must ignore it, otherwise either the path request packet is a new packet or it has been received from a new path (in each case, the current node must process the packet). Therefore, no delay will be imposed on the path discovery process while broadcasting PREQ packets. In fact, the PREQ packets move toward a destination in conformity with their standard procedure.

After this step, RREQ packets are delivered to the destination respectively and the destination returns a RREP packet to the source for each RREQ packet. There is an ActiveNeighborCount in each PREP packet which initially has zero value.

These PREP packets return to the source through the reverse path of PREQs based on AODV algorithm. Indeed, PREP packets contain carried path of PREQ in their header in order to find a return path to the source.

To count number of active neighbors in each path, in their return path toward the source and after reaching an intermediate node, PREP packets fetch stored value of counter from PREQ_Seen table and add it to their ActiveNeighborCount field.

In addition, these path response packets update the routing table of intermediate nodes while traversing towards the source. To this end, once an intermediate node receives a path response packet and before sending it toward the next node, it creates an entry in the routing table and sets address of its upper-hand node equal to the address of the node sent the same packet. Care must be exercised that ActiveNeighborCount field is stored in the entry in this step. It should be noted that the intermediate nodes will create an entry for destination after receiving each path response packet. Thus, it is possible that a node has already conducted several times a path response packet, receive a number of path request packets; in this case, some entries to the destination will be recorded in the routing table of this node.

The PREP packet has recorded the number of its active neighbors in the ActiveNeighborCount field when it reaches to the source. After receiving the first PREP, the source node sets a timer and waits for remaining PREPs in a specified time. After the timer expires, it sorts received PREPs based on ActiveNeighborCount value in an ascending order. Three steps are selected from the top of the list and concurrently information sending will be started through them. In other words, the steps with least ActiveNeighborCount value are selected. Hence, every intermediate node broadcast its received data packets to the next step with the least ActiveNeighborCount field.

To understand how nodes operate on the algorithm, pseudo codes of the source node, destination node, and intermediate nodes are presented in the Figure 1, Figure 2 and Figure 3 respectively.

1. Broadcast path request packet to all when requiring a path to a specific destination.
2. Wait for returned response path packets from destination.
3. Once first path response packet received from destination, set a timer and wait for remained path response packets in a specified time.
4. After the timer expires, sort received PREPs based on ActiveNeighborCount value in an ascending order.
5. In accordance with the considered criteria for balancing loads, select next steps dependent upon number of required paths from top of sorted list contains path response packets.
6. Start concurrent sending of information to the next step of destination path through nodes which have been selected in Step 5.

Fig. 1: pseudo code for source node in the proposed algorithm.

1. If the received packet was path request packet:
   a) If the received packet is new and acceptable, insert its specification in the PREQ_Seen table and set its counter field equal to zero.
   b) If the received packet is not new (repeated one), increase its counter field in the PREQ_Seen table by one.
   c) In line with path discovery policy, if path request packet is able to be broadcasted, broadcast it to all.
1. If the received packet was path response packet:
   a) Sum ActiveNeighborCount field of the packet and its corresponding counter field from PREQ_Seen table and update path response packet with value of new ActiveNeighborCount field.
   b) Create an entry to routing table per path response packet of destination.
   c) Send the new path response packet toward source in accord with routing algorithm.
1. If the received packet was data packet:
   a) Select steps with least ActiveNeighborCount field from routing table entries.
   b) Send data packets of selected steps to the next step (toward destination).

Fig. 2: pseudo code for intermediate nodes in the proposed algorithm.

1. Once path request packets were received, send per path request packet a path response packet with an equal-to-zero ActiveNeighborCount to the source based on routing algorithm.

Fig. 3: pseudo code for destination node in the proposed algorithm.

To clarify the proposed solution of the algorithm, consider the hypothetical network of Figure 4. In the figure, radio range is specified for each node and the dotted line between two nodes means these nodes are in each other radio range.

Assume that the node S (See Figure 4) wants to send data concurrently through two paths to the node D. To this end, it begins the path discovery process by sending PREQ packets in a comprehensively. PREQ

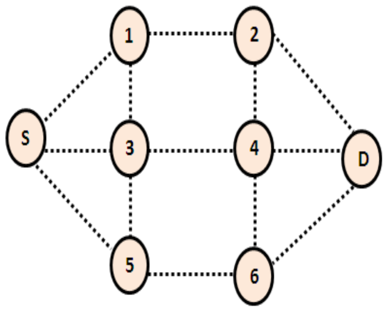

Fig. 4: Topology and nodes' range in a hypothetical network.

packets make their way toward the destination based on AODV algorithm. At the same time, the intermediate nodes update their table after receiving each PREQ of counter field.
Figure 5 presents the network state while all PREQs reach to the destination. To simplify the figure, drawing the rest of the RREQs is avoided.
As can be seen in Figure 5, the PREQ counter in Node 1 equals to 2 in the PREQ_Seen table because the node receives the same PREQ from Nodes 2 and 3 after it received the PREQ from Node S. Hence, it increases the counter of PREQ by two but Node 2 firstly receives PREQ from Node 1 and then just receives it from Node 4.

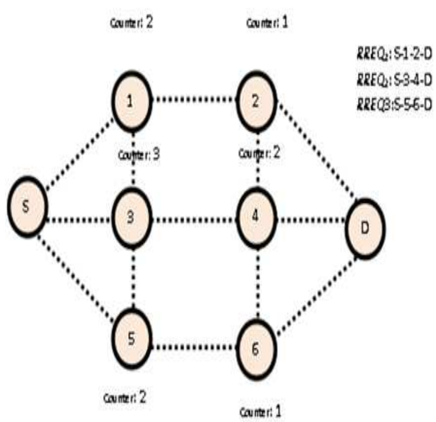

Fig. 5: how to receive PREQ by destination in a hypothetical network.

Therefore, the counter of PREQ is equal to 1 in the PREQ_Seen table. Node 3 also at the outset receives PREQ from Node S and then receives it from Nodes 1, 5, and 4 again. Thereupon, it sets the counter value equal to 3. Once PREQ packets reach destination, destination node begins sending PREQ packets for all of them. Initially, ActiveNeighborCount field in the packet is considered to be zero. On the way back to the source, the packets add the value of the counter field of their equivalent PREQ from intermediate nodes' PREQ_Seen tables. Figure 6 presents the network state when PREP packets return back to the source.
As shown in figure 6, the source examines received PREPs, identifies first and third paths as more appropriate one to send information concurrently, and begins concurrent sending through these two paths.
It is necessary to be said that keeping track of the phases in the algorithm IZM_AODV is just performed as well as AODV algorithm and no changes will be done occur in this phase.

4 Simulation and Results Evaluation

In this section, the results of the performed simulations are presented. The proposed IZM_AODV algorithm (coupled with multipath algorithm ZD_AOMDV, AOMDV, and AODV basic algorithm) is examined and compared throughout these tests using different scenarios in terms of (1) routing overload, (2) average end to end delay, and (3) packet delivery rate against nodes' speed, Pause time, number of nodes, and number of applied paths.

(a)*Simulation Environment*
In the present paper, we have been using NS2 software to simulate [15]. 100 nodes with a radio range of 250 m are used for simulation which placed randomly in an area of 1000 square meters and move randomly. We imposed CBR-type traffic on the network. In these networks, the nodes use the 802.11 protocol within their MAC layer and RADIO-ACCNOISE mode has been used to send/receive data. In addition, the Random Waypoint model has been selected in all cycles for mobility of nodes in a way that each node chooses a point as destination and then move toward a destination while its speed is between maximum and minimum. Once it reached the destination, remains at that point for a specific time (Pause Time) and performs the same operation again. Time of each simulation is considered equal to 300 seconds and according to recorded results, the average number of scenario execution equals to 25.

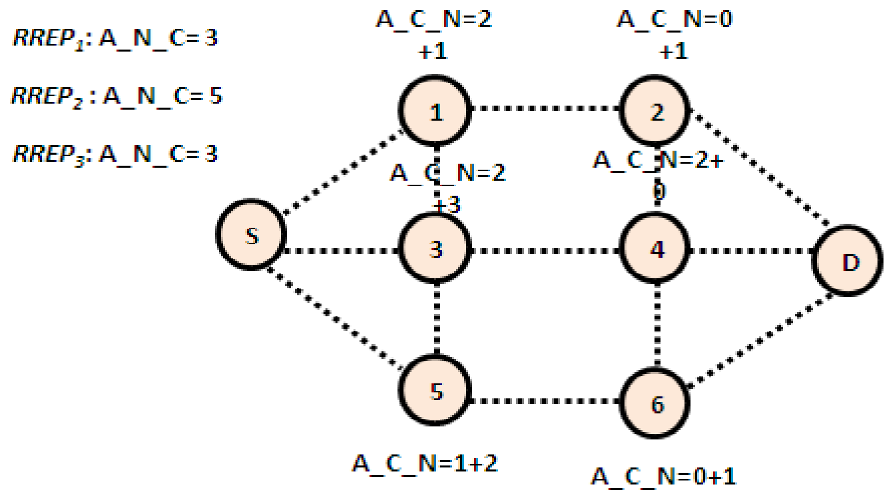

Fig. 6: How RREPs received by the source node in a given network

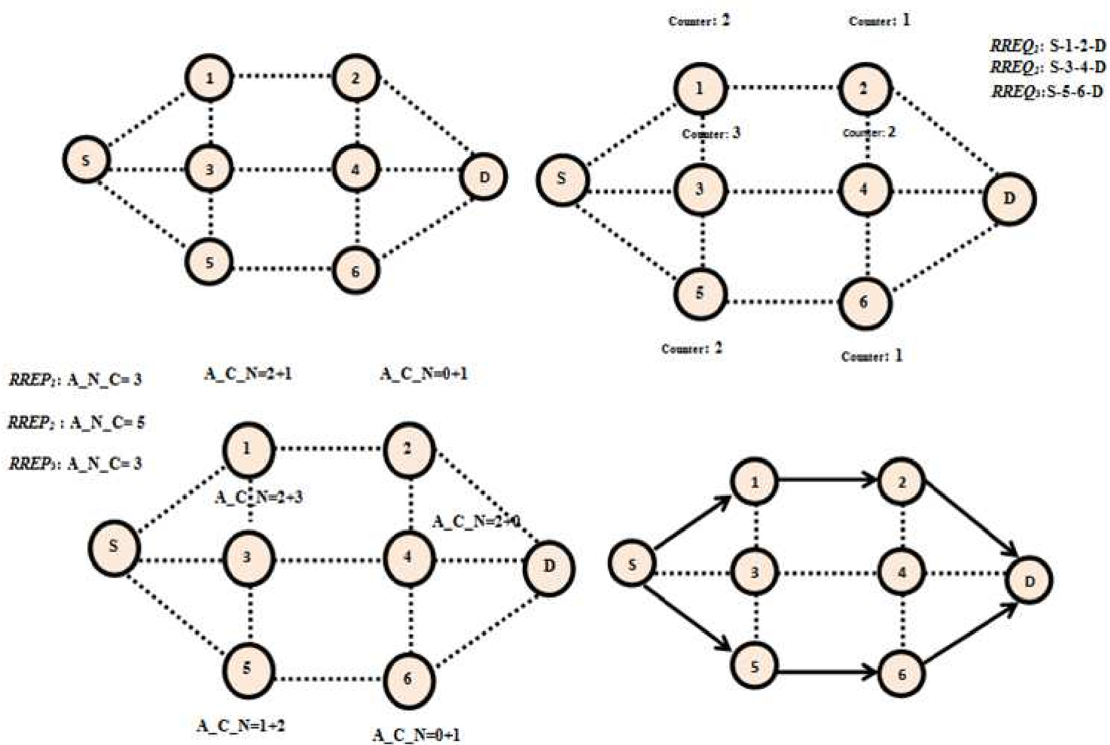

Fig. 7: shows an example of the routing in algorithm IZM__AODV that described above.

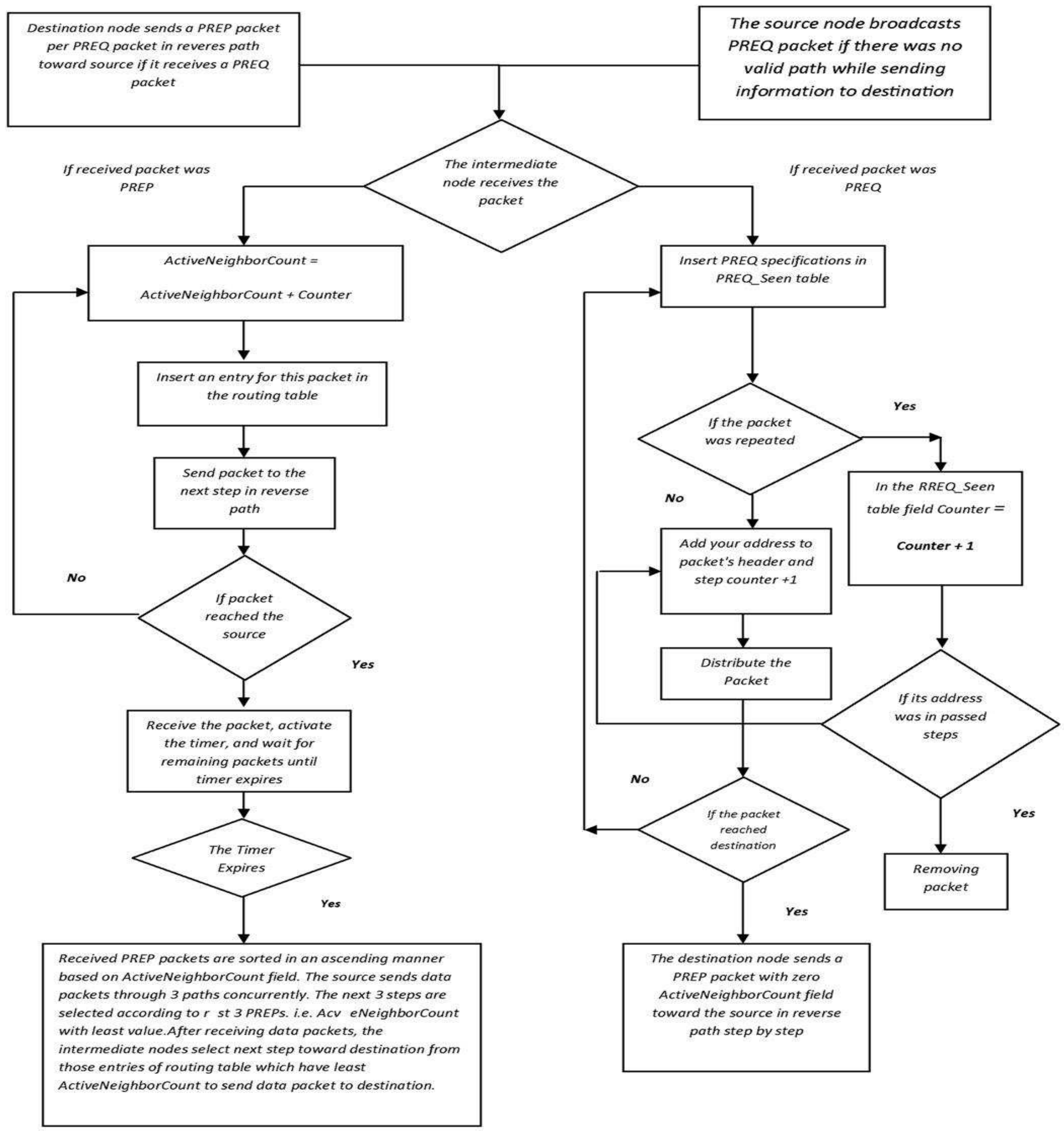

Fig. 8: A flowchart of the routing algorithm IZM_AODV

(b)*Simulation results*

–Routing Overload

Routing Overload is the number of all control packets in the network layer, which were sent during simulation time. They include path request packets, path response packets, packets for declaration of path failure, and special packets of each algorithm.

(i) Routing Overload versus Maximum Speed of Nodes

In all four algorithms, in the event that the speed of the node and thus the number of executing path discovery process are increased, routing overload increases relatively. In the basic AODV algorithm, with the increase of network dynamics, loss of paths, and restarting the path discovery phase, the routing overload increases. Additionally, the number of routing packets in ZD_AOMDV algorithm is more than another two multipath algorithm which is because using PREQ_Query and PREQ_Query_Reply packets in the path discovery phase. However, the routing overload in the proposed algorithm and the AOMDV algorithm are almost identical (figure 9).

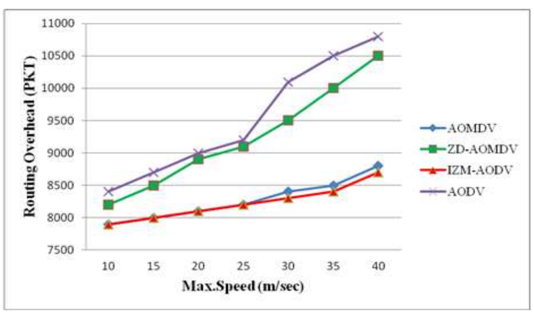

Fig. 9: Routing overload versus maximum speed of the nodes in the four algorithms: IZM_AODV, ZD_AOMDV, AOMDV, and AODV.

(ii) Routing Overload versus Pause Time of the Nodes

Figure 10 shows If the pause time of the nodes increased and node mobility decreased, the routing overload will be decreased. The decline in the AODV and ZD_AOMD algorithm is more stronger. In the AODV algorithm, with more static network, we have the greatest reduction in routing overload.

(iii) Routing Overload versus Number of the Nodes

An increasing number of the nodes highly affects

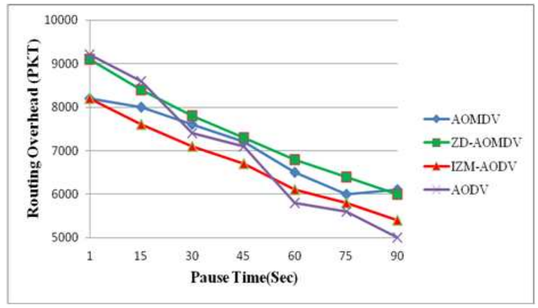

Fig. 10: Routing overload versus pause time in the four algorithms: IZM_AODV, ZD_AOMDV, AOMDV, and AODV.

ZD_AOMDV because of growth in the number of each node's neighbors and thus growth in number of inquiry packets in the path discovery process (figure 11).

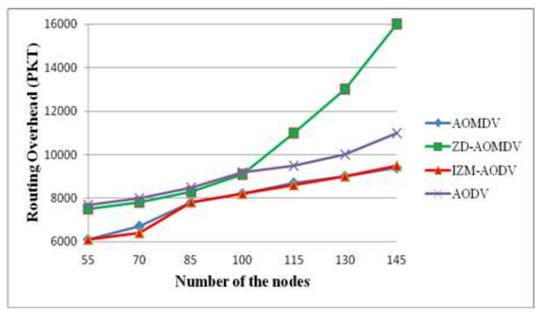

Fig. 11: Routing overload versus number of the nodes in the four algorithms: IZM_AODV, ZD_AOMDV, AOMDV, and AODV.

(iv) Routing Overload versus Number of the used paths

As you see in figure 12, increasing the number of used paths does not affect the routing overload because number of used paths have no effect on the number of control packets used to discover paths.

–Average End to end Delay

(i) Average End to End Delay versus Maximum Speed of the Nodes

If the maximum speed of the nodes increases, the average end to end delay is increased in all three algorithms because of dynamics of network topology and increment of path breaking rate. Since using zone disjoint paths during the data sending phase, average end to end delay in the ZD_AOMDV and IZM_AODV algorithms is less than the average delay in the AOMDV. Finally,

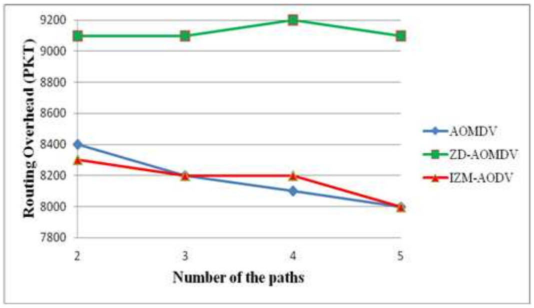

Fig. 12: Routing overload versus number of the paths in the three algorithms: IZM__AODV, ZD__AOMDV, and AOMDV.

due to shorter path discovery phase in IZM__AODV algorithm relative to ZD__AOMDV, end to end delay in the proposed algorithm is slightly less than the delay in the ZD__AOMDV algorithm. And as a result of using several paths, increment of failure tolerability, decreasing the number of restarts in path discovery phase, and concurrent data sending through these paths, delay of the AOMDV is less than the basic AODV algorithm (Figure 13).

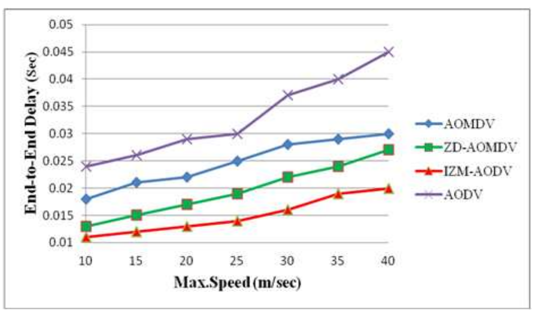

Fig. 13: Average end to end delay versus the maximum speed of the nodes in the four algorithms: IZM__AODV, ZD__AOMDV, AOMDV, and AODV.

(ii) Average End to End Delay versus Pause Time of the Nodes.
With decrement in the pause time and indeed increment of dynamics of network topology, average end to end delay increases in all four algorithms. The remarkable thing is that the more static network and the less packet loss rate, the delay of the algorithms is lesser but once the dynamics and joint loss rate increase in the network, the delay increases too which is due to the increment in the number of path discovery phase execution. If pause time increases, the delay of the basic AODV algorithm will be less than AOMD algorithm and multipath AOMD algorithm because of decrement in the number of path discovery phase execution (figure 14).

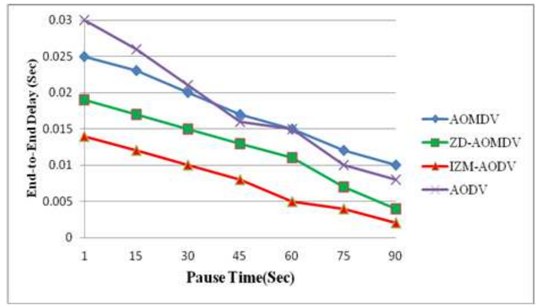

Fig. 14: Average end to end delay versus pause time of the nodes in the four algorithms: IZM__AODV, ZD__AOMDV, AOMDV, and AODV.

(iii) Average End to End Delay versus Number of the Nodes
Compared to the other elements, an increment in the number of the nodes has no tangible effect on the end to end delay. As you see in figure 15, however, in case of increasing in the number of the nodes and actually increment of the network density, average end to end delay decreases in all four algorithms since increment in the number of paths between source and destination.

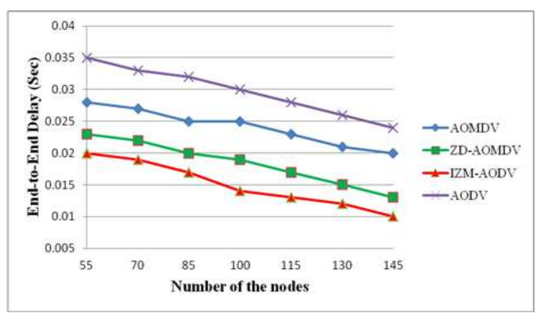

Fig. 15: Average end to end delay versus number of the nodes in the four algorithms: IZM__AODV, ZD__AOMDV, AOMDV, and AODV.

(iv) Average End to End Delay versus Number of the Used Paths
In the event of increment in the number of user paths, average end to end delay increases on account of the used paths close to each other and

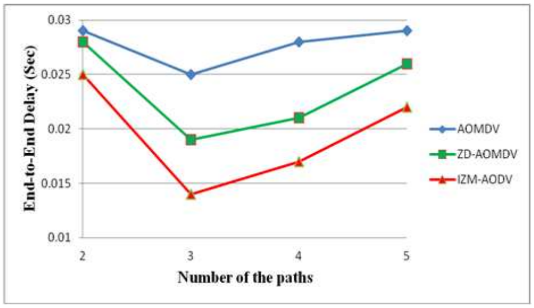

Fig. 16: Average end to end delay versus number of the paths in the three algorithms: IZM_AODV, ZD_AOMDV, and AOMDV.

therefore the neighbors will have more effect on one another (figure 16).

–Delivery Rate of Packets to the Destination

Our mean by delivery rate of packets to the destination is that once the source sends all packets to the destination, which percent of them is delivered to the source properly.

(i) Delivery Rate of Packets versus Maximum Speed of the Nodes

In case of increment in the maximum speed of the nodes, the delivery rate of packets to the destination decreases in all algorithms in consequence of constant path breaking and loss of packets. As can be seen in the figure 17, delivery rate of packets to the destination in the proposed algorithm is slightly more than ZD_AOMDV because of decrement in the delay of path discovery phase as well as decrement in the MAC layer collisions. It should be noted that the proposed algorithm does not use PREQ_Query and PREQ_Query_Reply packets and it helps to decrease MAC layer collisions.

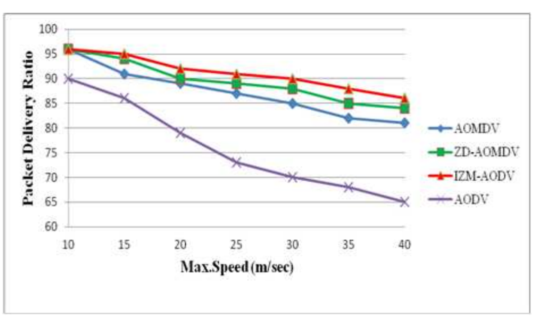

Fig. 17: Delivery rate of packets versus maximum speed of the nodes in the four algorithms: IZM_AODV, ZD_AOMDV, AOMDV, and AODV.

(ii) Delivery Rate of Packets versus Pause Time of the Nodes

In case of increment in the pause time and in fact decrement in dynamics of network topology, delivery rate of packets to the destination increases in all four algorithms (figure 18).

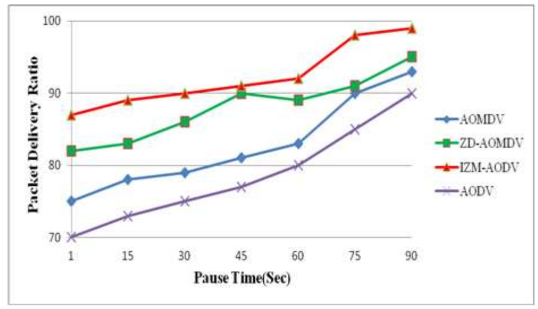

Fig. 18: Delivery rate of packets versus pause time of the nodes in the four algorithms: IZM_AODV, ZD_AOMDV, AOMDV, and AODV.

(iii) Delivery Rate of Packets versus Number of the Nodes

In case of increment in the number of the nodes and indeed increment in the network density, delivery rate of packets to the destination increases in all four algorithms because of more paths in the network (figure 19).

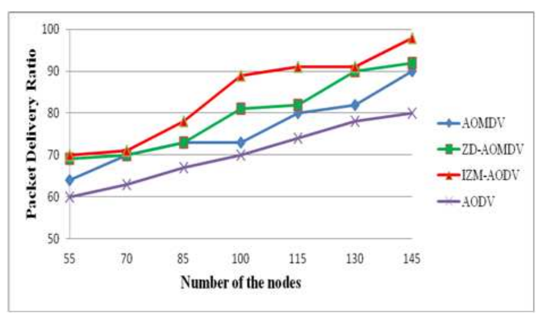

Fig. 19: Delivery rate of packets versus number of the nodes in the four algorithms: IZM_AODV, ZD_AOMDV, AOMDV, and AODV.

(iv) Delivery Rate of Packets versus Number of Used Paths

The three algorithms send data through used paths concurrently. In the event that occur any increase in the used paths, the delivery rate of packets to the destination increases in all three algorithms (figure 20).

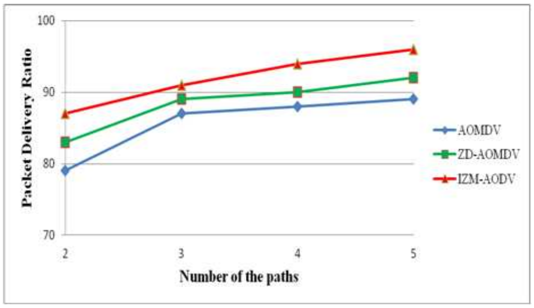

Fig. 20: Delivery rate of packets versus number of paths in the three algorithms: IZM__AODV, ZD__AOMDV, and AOMDV.

5 Conclusion and Future Works

To increase failure tolerability, some routing algorithms perform routing in a multipath manner with paths which have least sharing and some of them perform data sending concurrently to decrease end to end delay. In this way, use of node disjoint paths showed desired results in terms of failure tolerability and so it is a proper option but even sending data through zone disjoint paths is not wholly independent; in fact, sending by way of one path affect other path because of inherent problems in the ad-hoc networks and CSMA/CA protocol. To solve the problem, zone disjoint paths are used to send data. One of the algorithms utilizes zone disjoint paths to send data concurrently is ZD__AOMDV algorithm but its path discovery phase has very high delay and so imposes significant overload to the routing. In the paper, IZM__AODV algorithm has been proposed which run path discovery phase with less delay and it detects zone disjoint paths in a way that routing overload decreases significantly. The simulation results show considerable improvements in terms of end to end delay and routing overload.It is possible to perform new activities in the multipath routing topic, especially about how to select paths and path selection criteria.

The proposed idea in the paper can be implemented using other similar algorithms in the demand-type class and even it is possible to introduce similar ideas about proactive algorithms.

We can add some solutions and so make the proposed method sensitive to service quality. For example, the number of paths to send data concurrently can be determined by the destination over time.

Using fuzzy logic, optimum point in the end to end delay, routing overload, and packet sending rate is reachable.

One of the evident weaknesses of the proposed algorithm is that passed path is stored in the header of the packet which endangers routing security as well as making a high overload. It is hoped that the problem is resolved by introducing a solution.

References

[1] I. Chlamtac, M. Conti and J. Liu, Mobile ad hoc networking: imperatives and challenges, Journal of Mobile Ad hoc Networks, **1**, 13-64 (2003).

[2] T.Sharma, D. Bhattacharyya and K.Tai hoon, Performance Comparision of AODV, DSR and DSDV under Various Network Conditions: A Survey, Proceedings of of the IEEE International Conference on Ubiquitous Computing and Multimedia Applications (UCMA) Gwalior India, 3-7 (2011).

[3] Y. Xue and K. Nahrstedt, Providing fault-tolerant ad hoc routing service in adversarial environments, Journal of Wireless Personal Communications, **29**, 367-388 (2004).

[4] M. Obaidat, A novel multipath routing protocol for MANETs, Proceedings of the 7th IEEE International Conference on Wireless Communications Networking and Mobile Computing (WiCOM) New York USA, 1-6 (2011).

[5] NAli, A survey on on-demand multipath routing protocol in MANETs, Proceedings of the IEEE International Conference on Electronic Design(ICED) Kangar Malaysia, 1-4, (2008).

[6] P. Hung Le, An interference-aware multi-path routing protocol for mobile ad hoc network, Proceedings of the IEEE International Conference on Networking Sensing and Control (ICNSC) Trang Nguyen, **503**, 507 (2011).

[7] H. Labiod and M. Farikha, Load Balanced Multipath Routing in Mobile Ad hoc Network, Proceedings of the 10 th IEEE International Conference on Communication systems(ICCS) Paris, 1-5,(2006).

[8] X. Li and L. Cuthbert, On-demand node-disjoint multipath routing in wireless Ad Hoc networks, Proceedings of the 29th IEEE International Conference on Local Computer Networks (LCN 2004) Tampa, United States, **1**, 419-420 (2004).

[9] A. Colvin, CSMA with Collision Avoidance, Journal of Computer Communication, **6**, 227-235 (1983).

[10] W.Zemin,A Survey on Directional Antenna Networking, Proceedings of the 7 th IEEE International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM) Nanjing China, 1-4 (2011).

[11] N. Javan, ZD-AOMDV: A new routing algorithm for mobile Ad-Hoc networks, Proceedings of the 8th IEEE International Conference on Computer and Information Science ICIS Tehran Iran, 852-857, (2009).

[12] M. Marina and S. Das, On-demand Multipath Distance Vector Routing in AdHoc Networks, Proceedings ofthe 9th IEEE International Conference on Network Protocols (ICNP) Riverside California, **2**, 14-23 (2001).

[13] M.Marina, On-demand multipath distance vector routing in ad hoc networks, Proceedings of the 9th IEEE International Conference on Network Protocols,OH USA, 14-23 (2001).

[14] S. Bandyopadhyay, S. Roy, T. Ueda and k. hasuike, Multipath routing in Ad hoc wireless networks with directional antenna, Journal of Personal Wireless Communication, **234**, 45-52 (2001).

[15] T. Issariyakul and E. Hossain, Introduction to network simulator NS2, Technical Report 2008928147 Printed on acid-free paper, Springer Science+Business Media LLC, (2009)

H. Rafiee Pour received her M.Sc. in Software Computer Engineering at Science and Research Branch, Islamic Azad University, Iran, in 2013. Her main research interests include Wireless networks, Routing in wireless networks, Network security, Distributed data base and Object detection.

M. Kuchaki Rafsanjani received her Ph.D. in Computer Engineering, Iran in 2009. She is currently assistant professor at the department of Computer Science in Shahid Bahonar University of Kerman, Iran. She published about 100 research papers in international journals and conference proceedings. Her main areas of interest are Computer Networks (Wireless Networks, Mobile Ad hoc Networks (MANETs), Wireless Sensor Networks (WSNs)), Electronic Commerce, Artificial Intelligence, Grid & Cloud Computing.

H. Saadat received his Ph.D. in Mathematics, Iran in 2004. He is currently assistant professor at the department of Mathematics in Islamic Azad University, Iran. His Main Areas of interest are BCK-algebras, Fuzzy Mathematics and Computer Science.