

Improving Performance of On Demand Multicast Routing Protocol by Fuzzy Logic

Alireza Shams Shafigh¹, Kamran Abdollahi¹, Marjan Kouchaki¹

¹ Department of Computer Engineering, Damavand Branch, Islamic Azad University, Tehran, Iran
E-mails: ali_shams@itrc.ac.ir, K_abdollahi@itrc.ac.ir, M_kouchaki@itrc.ac.ir
(Ali_shams,K_abdollahi,M_kouchaki@itrc.ac.ir)

Abstract: A Mobile ad hoc network (MANET) is a collection of wireless communication nodes that dynamically organize themselves to form a network without the need for any fixed infrastructure or centralized administration. The network topology dynamically changes frequently in an unpredictable manner since nodes are free to move. Multicasting can be considered as an efficient way to deliver information from source nodes to several client nodes. Although multicast routing algorithms are desirable, their forwarding structure and network resource consumption makes them significantly less efficient than unicast routing algorithms. In this research, a fuzzy-based policy is used to improve performance of On-Demand Multicast Routing Protocol. The main goal of FBODMRP is to establish a small, high quality and efficient forwarding group. This is achieved by augmenting the join query packets with additional information such as speed, power level of node and link bandwidths. By introducing fuzzy control in the route selection, we can cope with uncertain and imprecise information. By restricting the domain of control packet flooding, we can further reduce the overhead. An evaluation shows that our approach increases packet delivery rate by up to 40%, reduces average end to end delay and consumed power by about 35% and 45% respectively.

[Alireza Shams Shafigh, Kamran Abdollahi, Marjan Kouchaki in Islamic Azad University. **Improving Performance of On Demand Multicast Routing Protocol by Fuzzy Logic**. *N Y Sci J* 2016;9(1):71-85]. ISSN 1554-0200 (print); ISSN 2375-723X (online). <http://www.sciencepub.net/newyork>. 13. doi:[10.7537/marsnys09011613](https://doi.org/10.7537/marsnys09011613).

Keywords: Mobile Ad hoc Network (MANET), Multicast routing, Fuzzy logic

1. Introduction

A Mobile ad hoc network (MANET) [1,2,3,19] is a collection of wireless nodes which dynamically configure themselves to form a network without the need for fixed infrastructure. For relaying the packets towards the destination, each node needs to implement routing functionality. As a result, the connection between any two nodes forms a multi-hop path supported by other nodes. However, for the network to operate, each node must be willing to forward packets on behalf of other nodes.

Multicasting is the transmission of datagrams to a group of hosts identified by a single destination address and hence is intended for group-oriented computing. Multicasting can efficiently support a variety of applications that are characterized by collaborative efforts and data transmission. Multicasting techniques can be considered as an efficient way to deliver packets from the source to any number of client nodes.

Multicast routing algorithms [4,5,6,7,13] have become increasingly important in the field of wireless ad-hoc networks, because they enable the distribution of data to a potential large set of nodes. Nodes form a multicast delivery structure which in normal cases performs better than using multiple unicast routing paths. This is crucial in ad-hoc environments, where bandwidth and power resources are at a premium. The major impediment, however, is that nodes can move

randomly, possibly causing frequent and unpredictable topology changes. Multicast network algorithms, however, must deliver packets to several hosts simultaneously, which then must discern if their role is to receive or forward the packets towards the multicast sinks. Although multicast routing algorithms are desirable in many situations, it is crucial to optimize their forwarding mechanism and network resource consumption. Packet delivery ratio and end-to-end delay are the principal performance variables taken into account when considering QoS aware applications and network resource management issues.

We can classify multicast routing algorithms according to the delivery structure they build. Tree-based approaches have only one path between the source-receiver pair. Due to their lower overhead, they are more efficient than mesh-based approaches. In a mesh-based multicast routing algorithm, there may be more than one path between a source-receiver pair, thus making it more robust. However, tree-based on-demand protocols are not necessarily the best choice [14,15,18]. In a harsh environment, where the network topology changes frequently, mesh-based protocols seem to outperform tree-based protocols, due to the availability of alternative paths, which allow multicast datagram to be delivered to all or most multicast receivers even if links fail.

Many variants of tree or mesh based protocols have been developed which to improve quantitative

performance due to the harsh environments they work in. A major concern is the quality of links and performance of nodes participating in the forwarding process. For example, if remaining battery of most of the nodes in the mesh or tree is low, probability of route failures will be high. In a similar way, if most of the links are weak and suffer from high packet loss, reliability of the delivery structure will be low. However, it is not clear when a link can be classified as a weak link or when the battery of a node is classified as low because such kind of information is typically imprecise or not fully available. Clearly, there is a trade-off involved. Having more precise information available would allow making better decisions but would also result in more overhead for making such information available.

Another concern is the overhead that multicast routing protocols create. For establishing the delivery structure a standard approach is to use network wide flooding of control packets. However, this leads to high overhead. Minimizing such overhead is crucial for high performance as such control packets might also collide with data packets leading to increased packet loss.

The main contributions of this paper are the following. We design two methods to improve performance and reliability of multicast routing protocols in a MANET. The first method uses fuzzy logic control to deal with imprecise and partial information during the construction phase of a forwarding structure. This will allow constructing better and more stable routes. We have applied the fuzzy control to the mesh based ODMRP routing protocol by firstly augmenting the join query packet with additional information such as battery status and link quality. Secondly, a fuzzy logic control process has been applied to the processing and forwarding of such join query messages which allows forwarding such control messages which have arrived from nodes on "better quality" routes with higher probability. The second method supposes that during route refresh the new forwarding group has many nodes in common with the old group. As a result, we restrict join query flooding in the network thus reducing the overhead.

The rest of paper is organized as follows. Section 2 describes related work. In Section 3 we briefly review the operation of the basic ODMRP protocol. Section 4 presents our changes that lead to the design of the fuzzy based FBODMRP protocol. In Section 5 we present an evaluation of our approach using simulation, where we compare FBODMRP with standard ODMRP. Finally, the paper concludes and presents suggestions for future works in Section 6.

2. Related Works

Several multicast routing protocols have been developed to cope with characteristics of MANETs such as frequent topology change due to mobility of nodes. We can classify the protocols according to the delivery structure that they create and maintain into tree-based and mesh-based. Examples of tree-based protocols are MAODV [27,12] which is an extension to the unicast routing protocol AODV [28] based on a group shared tree, MOLSR [29] which is an extension to OLSR unicast routing protocol [30], based on a Dijkstra tree, and Adaptive Demand-driven Multicast Routing Protocol (ADMR) [8]. A protocol which is based on overlay trees is AMRoute [10]. The most important mesh-based routing protocols are On-Demand Multicast Routing Protocol (ODMRP) [9,25,26] and Core-Assisted Mesh Protocol (CAMP) [31].

Several protocol variants have been developed that try to optimize delivery structures taking into account parameters such as reliability [16, 17], power consumption [22, 23], mobility and bandwidth. Bandwidth-Efficient Multicast Routing (BEMR) finds the nearest forwarding group member for nodes broadcasting join queries [32]. Associatively-Based Ad hoc Multicast (ABAM) builds a source-based multicast tree [10]. Association stability helps the source select routes to members which will probably last longer and need fewer reconfigurations. Neighbor-Supporting Multicast Protocol (NSMP) utilizes node locality to reduce overhead [33]. A forwarding group is created by a source, which broadcasts a request. In order to maintain the connectivity of the forwarding group, the source periodically sends local requests.

Differential Destination Multicast (DDM) lets source nodes manage membership and stores the forwarding state encoded in packet headers to achieve stateless multicasting [24, 34]. Independent-tree ad hoc multicast routing (ITAMAR) provides several heuristics to compute a set of independent multicast trees, such that a tree is used until it fails and then replaced by one of its alternatives [35]. Lantern-tree-based QoS multicast (LTM) is a bandwidth routing protocol with an improved success rate by means of multipath routing [36, 37]. Probabilistic predictive multicast algorithm (PPMA) tracks relative node movements and statistically estimates future relative positions to maximize the multicast tree lifetime by exploiting more stable links [38].

Mobility Prediction Aided Dynamic Multicast Routing (MPADMR) [39] consists of two steps. In the first step, with the aid of mobility prediction, a link lifetime constrained minimum hop-count multicast tree is established at the beginning of a multicast session. In the second step, a dynamic multicast tree

maintenance procedure is employed to rearrange the existing multicast tree when a group member joins/leaves the multicast session.

The QoS-MAODV [40] extends existing Multicast Ad hoc On Demand Distance Vector Routing protocol, by using admission control and bandwidth reservation in each node. In [41] a cross-layer framework has been proposed to support admission control using information on available bandwidth, which was estimated without extra control overhead.

In [21], a weight based clustering technique for multicast routing protocol is proposed. Here, nodes use weighted cost function based on the transmission power level, residual power and node speed to form clusters and choose a cluster head in each cluster.

Resilient On Demand Multicast Routing Protocol (RODMRP) is a ODMRP-based wireless multicast protocol that offers more reliable forwarding paths in face of node and network failures. A subset of the nodes that are not on forwarding paths rebroadcast received packets to nodes in their neighborhoods to overcome perceived node failures [42].

The main problem of presented approaches is their high data and control overhead which might also lead to low packet delivery probability due to collisions. Also, multicast delivery structures should be robust, e.g. composed of nodes which have high available capacity and good links in order to increase packet delivery further. In this paper we propose a novel method which not only reduces overhead significantly but also allows creating robust delivery structures using fuzzy based control. This allows dealing with imprecise and incomplete information.

3. Overview on ODMRP

One of the most important and robust multicast routing protocols in MANETs is the mesh based ODMRP. In ODMRP [3,4,5], group membership and multicast routes are established and updated by the multicast source on demand. Similar to on-demand unicast routing protocols, a request and reply phase comprise the protocol.

While a multicast source has packets to send, it floods a member advertising packet with data payload piggybacked. This so called join query packet is periodically broadcasted to the entire network to refresh the membership information and update the routes. When an intermediate node receives a non-duplicate join query, it stores the upstream node ID into the routing table and rebroadcasts the packet. Once a join query packet reaches a multicast receiver, the receiver creates and broadcasts a join table to its neighbors. When a node receives a join table, it checks if the next node ID of one of the entries matches its own ID. If it does, the node realizes that it

is on the path to the source and thus is part of the forwarding group. It then sets the FG flag (Forwarding Group flag) and broadcasts its own join table built upon matched entries. The join table is thus propagated by each forwarding group member until it reaches the multicast source via the shortest path. This process establishes (or updates) the routes from sources to receivers and builds a mesh of nodes, the forwarding group.

Once the join table reaches the source, it multicasts packets with the help of the forwarder nodes through the forwarding mesh. Intermediate nodes just need to check if the FG-flag for the given group has been set through the reception of the proper join table. If it is set, they re-broadcast the packet, otherwise it is discarded. The periodic rebroadcasting of the join query packet maintains the soft states in the forwarding nodes. Once the source has no more packets to send, it just stops sending and the forwarding state times out in the forwarding nodes. This avoids the need to explicitly tear down the mesh structure. Network hosts running ODMRP are required to maintain the following data structures:

Member Table: Each multicast receiver stores the source information in the Member Table. For each multicast group the node is participating in, the source ID and the time when the last join query is received from the source is recorded. If no join query is received from a source within the refresh period that entry is removed from the Member Table.

Routing Table:

A Routing Table is created on demand and is maintained by each node. An entry is inserted or updated when a non-duplicate join query is received. The node stores the destination (i.e., the source of the join query) and the next hop to the destination (i.e., the last node that propagated the join query). The Routing Table provides the next hop information when transmitting Join Tables.

Forwarding Group Table:

When a node is a forwarding group node of the multicast group, it maintains the group information in the Forwarding Group Table. The multicast group ID and the time when the node was last refreshed are recorded. **Message Cache:** The Message Cache is maintained by each node to detect duplicates. When a node receives a new join query or data, it stores the source ID and the sequence number of the packet. Note that entries in the Message Cache need not be maintained permanently. Schemes such as LRU (Least Recently Used) or FIFO (First In First Out) can be employed to expire and remove old entries and prevent the size of the Message Cache to be extensive.

As a summary, ODMRP has many advantages but it suffers from high overhead. This overhead is attributed mainly due to the mesh delivery structure and the network wide broadcasting of join query packets. When there are many nodes or multicast sources in the network, data and control overhead increases significantly, especially for large networks. However, due to the redundant paths within the mesh delivery structure, reliability is increased leading to higher packet delivery ratio under mobility.

Therefore, one important point to consider is how to reduce the overhead for the mesh creation and maintenance. Another point which determines performance is which nodes and links participate in the mesh based delivery. As a principle, not all nodes have equal performance (for example battery might be low for some nodes or some nodes might have low CPU processing facilities) and not all links have high reliability or bandwidth availability. Therefore, it is important to consider link quality and resource availability in the mesh creation phase.

4. FBODMRP (Fuzzy-Based ODMRP)

In this section, we detail our two main approaches[43,44] for increasing the performance of multicast routing in a MANET. First, we use a fuzzy logic based approach to deal with imperfect knowledge about link and node characteristics. Second, we restrict the domain of control packet flooding to reduce the overhead. Finally, we demonstrate how these approaches can be integrated into mesh based multicast mechanisms, which then guides the design of our new mesh based multicast routing protocol denoted as Fuzzy-Based ODMRP (FBODMRP).

4.1 Dealing with imperfect information

The main idea of our approach is to establish a small and strong forwarding group which should lead to decreased resources consumption and higher stability of the delivery structure. Clearly, there is a trade-off involved. Having larger forwarding group increases resilience against node and link failure but also increases overhead. A node is classified as strong when it has certain desirable properties which increase the probability of data delivery when this node will participate in the forwarding group. Examples of such properties are high bandwidth availability, low loss rate, low moving speed, or high power level. If a node is not strong it will be classified as a weak node and it might be not optimal to include the node in the forwarding process. For example, if a node has a low power level, the probability that the node powers down in the near future will be high leading to packet loss due to node failures and re-routing.

A strong forwarding group is made out of a set of strong nodes in the network. A weak forwarding group is made out of weak nodes in network. But in many cases, it is impossible to establish forwarding group in the network only by using strong nodes. For example, some nodes might have weak links due to mobility and link quality might change over time. Also, it might be possible that no strong group will exist. A strong forwarding group will be smaller than a weak forwarding group, because it removes weak nodes from forwarding group networks.

In standard ODMRP, any node which receives a join query packet caches it and discards other copies of join query packets. The first copy of join query has low delay but it might be received from a node which has low power level. One simple approach is to wait until all copies of a join query packet are received and select the best join query using extra information inserted into join query packets. However, this approach delays arrival time of join query packets at receivers. Therefore, we use a different approach.

When classifying nodes into strong or weak ones, we have to deal with imperfect and incomplete information. Therefore, we propose to use fuzzy logic to distinguish between strong or weak nodes. Another benefit of the approach is that fuzzy logic is simple to implement and has low complexity. In order to apply fuzzy logic, we need to have information which allows classifying nodes into weak and strong ones. Such information will then be exploited in the delivery structure creation in order to include strong nodes in the mesh. Therefore, we add several fields to the join query packet which carry extra information on e.g. bandwidth availability, loss rate experienced, moving speed, or power level to allow the nodes to perform a better route selection in the route request process. Based on such information, the next nodes will be able to compute the probability of caching and forwarding the received join query message.

In the proposed method, better routes (e.g. composed of nodes and links having high bandwidth, high power level, low mobility) have a high probability to be saved and forwarded but weak routes (low bandwidth, low power level, high mobility) have a lower probability. As a result, we can establish a small and strong forwarding group instead of a larger and weaker one. A larger forwarding group suffers from high data and control overhead and consumes more power, bandwidth and wireless resources. Having weak nodes participating in the forwarding group increases the probability of packet loss and re-routing.

Once a node receives a join query, it needs to process the additional parameters in a way to arrive at a better forwarding structure. We propose to use fuzzy logic to cope with network dynamics, uncertainty and

imprecise information. Therefore, when processing the join query, the node fuzzifies bandwidth, speed of node and power level of previous node. As a result, the node computes a probability of caching and forwarding of the request. If a join query has arrived through a “strong” node, the packet is cached and forwarded with a high probability. To fuzzify parameters, we use a simple membership function. Its horizontal axis shows the value of node’s parameters and the vertical axis shows the membership probability. When a node receives a join query packet it extracts the value of previous node’s parameters, and uses these parameters to classify them as low, medium or high. Finally, before forwarding, the node replaces the parameters in the join query packet with its own information.

Our proposed fuzzy method presents no extra overhead to exchange the required information about bandwidth, loss and power. It only use join query packets to update the average values (loss, power and bandwidth) of its neighbors.

If ((bandwidth is high) **and** (power is high) **and** (speed is low)) **then**
Increase prob. of forwarding join query packet

If ((bandwidth is low) **and** (power is low) **and** (speed is high)) **then**
Decrease prob. of forwarding join query packet

If ((bandwidth is low) **and** (power is high) **and** (speed is medium)) **then** *Do not change prob. of forwarding join query*

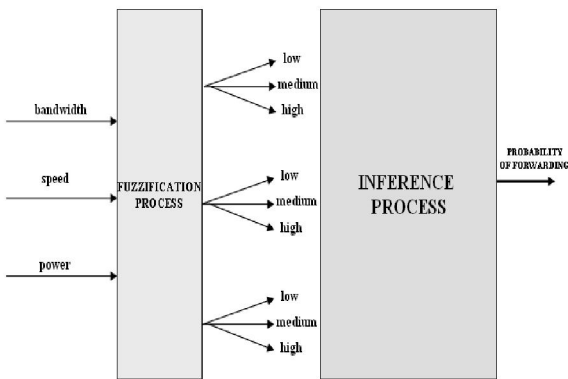


Figure 1. Computing probability of caching or forwarding

The initial probability of join query forwarding is set to 100%. If we assume that the fuzzy controller has n rules, the i th fuzzy rule decreases the overall probability of forwarding by b_i . The impact of

different parameters on the performance is evaluated in Section 5-5. When a node in the network receives a join query packet, it runs the fuzzification and inference process to derive the probability of caching and forwarding the join query to other nodes.

To decide about saving or discarding a received join query packet, we use only inside information of packets and some updated average values for loss, bandwidth and power. These average values would be more accurate with receiving number join query packets.

In the inference stage of the fuzzy process, inference laws are used to compute the probability of caching and forwarding based on the simple rules.

Figure 1 shows each node’s decision process based on the fuzzy logic. Input to this process is the previous node’s operating parameters (such as bandwidth, speed and power) where the probability of caching and forwarding is the output of the process. These parameters are described in section 5.

4.2 Reducing the overhead

In standard ODMRP, periodic sending of join query packets is a critical problem, because join query packets propagate in the entire network. As a result, such network wide flooding results in very high overhead. However, such periodic flooding allows ODMRP to create a robust delivery mesh structure. If the mobility of nodes is not too high, the topology should not have changed much between the last update of the mesh structure. Therefore, we propose to restrict the domain of the join query packet flooding.

For each multicast group the node is participating in, the source ID and the time when the last join query is received from the source is recorded. If no join query is received from a source within the refresh period that entry is removed from the Member Table. In standard ODMRP when a node has data to transfer, a join query packet is flooded to the entire network. Therefore a source node with a continuous traffic flow will refresh the previous forwarding group before it expired. If mobility of nodes is low, intersection of new forwarding groups and previous forwarding group will be high. In such scenarios it is beneficial to limit the propagation domain of join query packets to an area restricted to a few hops away of the members of the previous forwarding group. However, the current forwarding group of a network with high mobility has lower intersection with the previous forwarding group.

In the proposed method, the new forwarding group can be established from the current forwarding group. Join query packets which are going to be sent to a region far away from the previous forwarding group will be dropped. To implement this idea, we added a field to the join query packet (*Number of*

previous forwarding group), which counts the number of the previous forwarding group nodes which was visited by this join query packet. If a join query packet has visited many nodes but it doesn't see any previous FG nodes, we can assume that this join query packet will not be useful so it will be discarded. Therefore, when a node receives a new join query packet, it extracts NOPFG¹ and Hop count fields from the incoming packet. Hop count field is the number of hops to this node from the sender. When a node receives a join query packet with a hop count greater than the minimum value, it decides if the join query packet will be forwarded or discarded based on a random value. This random value is based on the forwarding probability which is calculated by fuzzy model (the figure 1). The minimum value of hop count allows a join query packet to traverse sufficient number of hops to prevent from discarding all copies of join query packets.

Figure 2 shows an example for NOPFG field. Node 1 has received a join query with NOPFG and hop count equals 1 and 3 respectively.

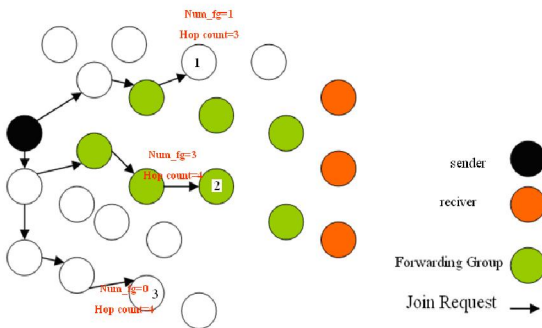


Figure 2. Intersection of new forwarding group with previous forwarding group

Node 1 will discard this join query packet (minimum hop count is 3). Node 2 has received a join query with NOPFG equal 2.

Source sends a join query packet without this confining technique when it knows there isn't the previous forwarding group. It informs this with setting the NOPFG to a minus value.

In some situations, especially when the density of network is very low; our second method leads to failures in finding route to destinations. Therefore, in the initial stage of a network, when there is no previous forwarding group we take an optional policy. This policy decides if the second method must be applied or not. Every source node before sending a join query, it restores time of the previous join query. This period is very low, when links are broken. The long period shows that source node has had data for

sending since a time. The source node makes a decision based on the previous join query about running the second method by intermediate nodes. It set the field of type to a special value (for example 2). Thus, we don't use the second method when the previous forwarding group doesn't exist. It means that the period, time between two successive join queries sending, is very low (smaller than 3 seconds) or very high (greater than 4 seconds). Concisely, there is no guarantee in receiving data to all receivers. We only use this method in high-dense networks and applying it in low-dense networks would be included in the future works.

4.3 Packet Header Format and Message Processing in FBODMRP

In FBODMRP, join query packets carry extra information such as power level, bandwidth and speed of a node. As a result, the forwarding algorithm can be optimized to create a forwarding group which has fewer members with more desirable properties. The figure 3 shows the new structure of the join query packet. The grey fields indicate extra fields in the proposed method. The first field defines type of our policy. Each policy specifies which methods must be run by member of networks. This field can be set to Zero to inform to other nodes to run both of the proposed methods. The value of one results only in running of the first method and the value of two causes only to running of the second method. When there are some problems in finding route to destination, we can adjust this field to three to inform members to run the basic and standard ODMRP without any changes. The second field is used by the second method, which confines flooding range. The third to the sixth are quality parameters used by the first method, which try to add strong link or node to forwarding group.

In what follows we describe the processing of join query messages. The packet forwarding process of FBODMRP is identical to the standard ODMRP. In the join query phase of FBODMRP, every node forwards packets with higher probability, if the previous node is a strong node.

Type	Number of previous forwarding group		
Bandwidth	Loss	Speed	Power
Query	Last address	Sequence number	Hop count

Figure 3. Structure of join query packet

The following pseudo code presents all actions that may be taken with receiving a join query packet.

¹ Number of previous forwarding group

```

Join_query_handle_function (Join_query_packet
jq_packet)
{
  If jq_packet isn't a new join query then exit;
  If (type==0) do both Dl_jq and fuzzy algorithms
  If (type==1) do only fuzzy algorithm
  If (type==2) do only Dl_jq algorithm
  If (type==3) do the basic ODMRP
}

```

There four policy that might be taken in some situation by source node. We implement the standard model of ODMRP with type of four. Three policies that defined by 0, 1 and 2. The following pseudo code shows functions of Dl_jq and fuzzy method.

```

Fuzzy_Dl_jq_handle_function (Join_query_packet
jq_packet)
{
  Get hop_count and num_fg from jq_packet fields;
  If jq_packet was lost then exit;
  Get parameters (loss, bandwidth, speed, power)
  from jq_packet fields;
  Fuzzify parameters;
  Compute probability of join query forwarding
  based on fuzzification results;
  Replace parameters of this node to jq_packet fields;
  Forward jq_packet based on the computed
  probability;
  If jq_packet was forwarded then cash jq_packet;
}

```

When a node decides to cache and forward the join query packet, it will discard other copies of the join query packet. The following function describes the FBODMRP routing process.

In the above function, jq_packet denotes a join query packet and num_fg counts the number of previous forwarding group members which have been visited by this join query packet. In the packet processing functions, forwarding is done based on calculated probability p . It means a join query packet is sent with probability of p and discarded with probability of $(1-p)$.

A join query packet has been lost if it has traveled several hops but not seen any node from the previous forwarding group. In the simulations we set refresh time of forwarding group to three seconds while memberships of forwarding group nodes are expired after four seconds. Therefore, if a join query packet within refresh time doesn't see any node from the previous forwarding group, it has been lost or its path was wrong.

5. Performance Evaluation

In order to evaluate the performance of our approach, we simulated several scenarios using glomosim [20]. Our simulation's duration is 600 second. Area simulation is 3000 meter *3000meter. All mobile node use random waypoint for their mobility model with pause time equal zero.

In these simulations every node has different physical properties. It's supposed that all nodes in simulation have a random bit rate between 2Mbps to 10 Mbps. this supposition is only to present performance of the proposed method. This bit rate is used as bandwidth in join query packets. Also it is assumed that nodes have GPS hardware to get information on its position and speed.

The default values for the fuzzy membership functions are the following: low and medium values for speed are 6m/s and 12 m/s; low and medium values for power are 0.3 and 0.5; low and medium values for bandwidth are 2 mbps and 4 mbps. In these memberships if input value is greater than medium value it's a high value.

We use 8 multicast groups, each one having 8 member nodes. One member node sends constant bit rate traffic with 5 packets (1024 bytes) per second to all other member nodes from start to end of the simulation. The following three methods have been simulated and compared with standard ODMRP:

Fuzzy method: this method uses fuzzy control to select high quality nodes.

Dl_jg method: this method deletes the lost join query packet using number of previous forwarding group. We try out this method in high-dense networks because the range of flooding will be confined by this method. So, this mechanism leads to failures in data forwarding process, some destinations possibly don't receive any packets.

Fuzzy & Dl_jq method: this method uses a combination of both methods.

To evaluate the performance of the proposed methods, the following parameters have been analyzed:

- End to end delay: average end to end delay of packets which have been received by receivers during the simulation period.

- Number of received packets: overall number of packets which have been received during the simulation period. This allows calculating packet loss rate.

- Overall overhead: sum of control and data packets (in bits) per a data packet.

- Life time: minimum time until the first node finishes its power.

- Consumed power: sum of consumed power used in the entire network per data packet. The standard glomosim power calculations have been used.

We were especially interested in analyzing the impact of the NOPFG value and minimum hop count, the speed of the nodes, the number of nodes in the simulation, the traffic intensity, and the speed and power membership functions.

5-1. Impact of NOPFG and minimum hop count

In case traffic flows is continuous, new forwarding group is refreshed before previous forwarding group has been expired. Also if nodes have low mobility, intersection of new and previous forwarding groups is high. Therefore it is possible to decide forwarding of join query packets based on previous forwarding group. If a join query packet has traversed more than minimum hop count, forwarding of join query packet depends on the NOPFG value. The first set of simulation analyzes the impact of NOPFG and minimum hop count on delay and packet loss. Speed of nodes in this scenario has been fixed to 20 m/s, dl_jq was simulated with 400 nodes. In the figures, results for standard ODMRP are also shown. Figure 4 and 5 show end to end delay and number of received packets under increasing NOPFG and under variation of hop count.

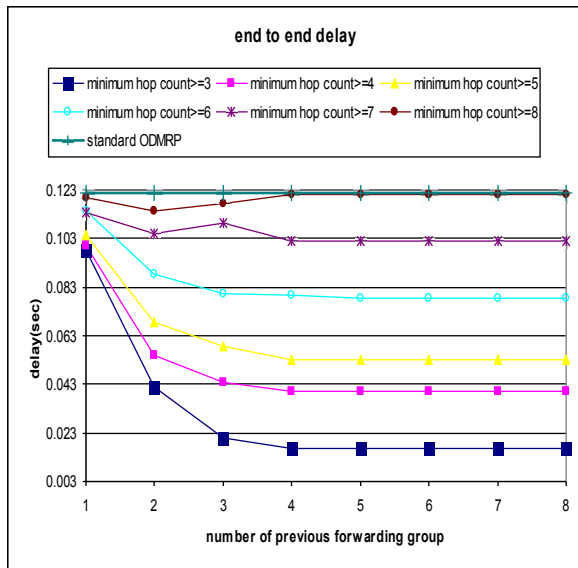


Figure 4. End to end delay

As you can see, a high minimum hop count leads to fewer lost packets and thus increases delivery probability. However, it also leads to higher average delay.

Minimum hop count had a higher impact compared to NOPFG as increasing the NOPFG

beyond some threshold had minor impact on performance.

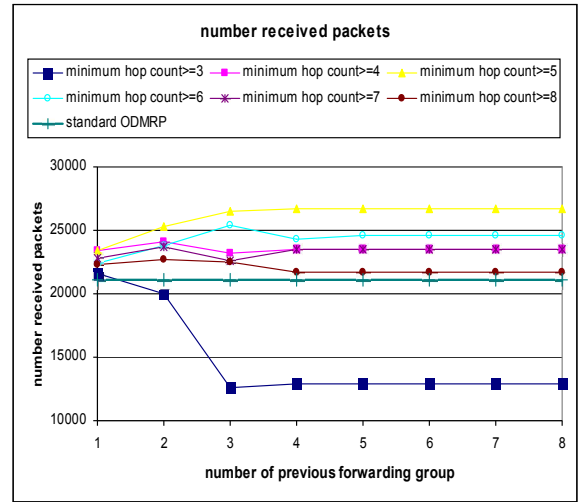


Figure 5. Number of received packets

For a low minimum hop_count value, fewer packets have been received than ODMRP. When minimum hop count is low, many join query packet are nominated to be lost. This results in a weaker delivery structure. On the other hand, a higher minimum hop count results in a more robust structure increasing delivery ratio but also increasing the average throughput. Regarding overhead in terms of data and control packets per every received data packets, Figure 6 shows that dl_jq outperforms standard ODMRP. The benefit is greater with lower hop count value.

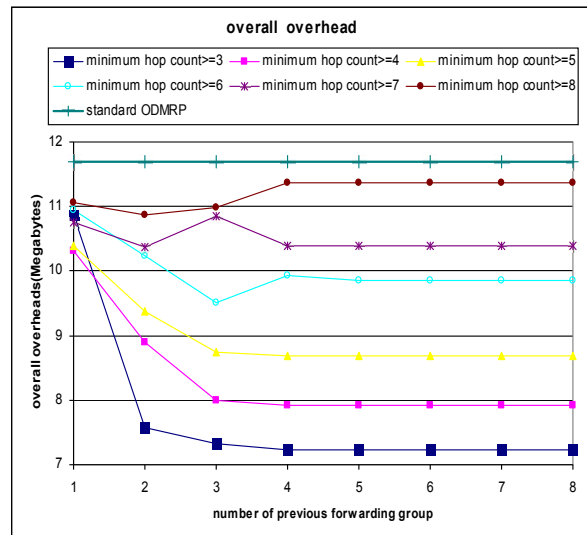


Figure 6. Overall overhead (megabytes) per a data packet

A high overhead translates in higher delay and power consumption and low throughput. While in standard ODMRP periodic join query packet flooding results in high overall overhead, in dl_jq overhead is significantly reduced due to many lost join query packets. A similar result can be observed in terms of power consumption, where dl_jq outperforms standard ODMRP due to the lower overhead which results in smaller power consumption.

As a consequence of the reduced overhead, network life time is significantly increased (see Figure 7) compared to ODMRP. The dl_jq with minimum hop count = 3 shows highest life time because of the smallest overhead.

Based on those results, we can conclude that values for NOPFG = 4 and minimum hop count = 5 result in a good compromise between overhead, delay and delivery probability. Therefore, we will use them in the following scenarios.

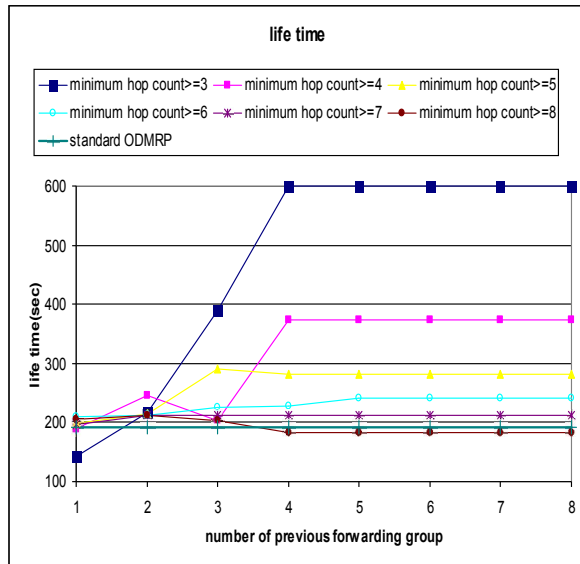


Figure 7. Life time

5-2 Impact of speed

Node mobility leads to topology change which is a challenge for multicast routing protocols. In this section, we analyze the performance of FBODMRP under increasing node speed. Figure 8 compares end-to-end delay of the proposed methods with standard ODMRP under different node speed. As you can see, the standard ODMRP has high delay compared to our methods. ODMRP periodically sends join query messages to refresh forwarding group. This makes it tolerant to increasing speed of nodes but also leads to increase of waiting time in queues. In contrast, all of the proposed methods perform restricted flooding. In the combined method (dl_jq&fuzzy), number of join

query packet discarding is high which results in low end-to-end delay.

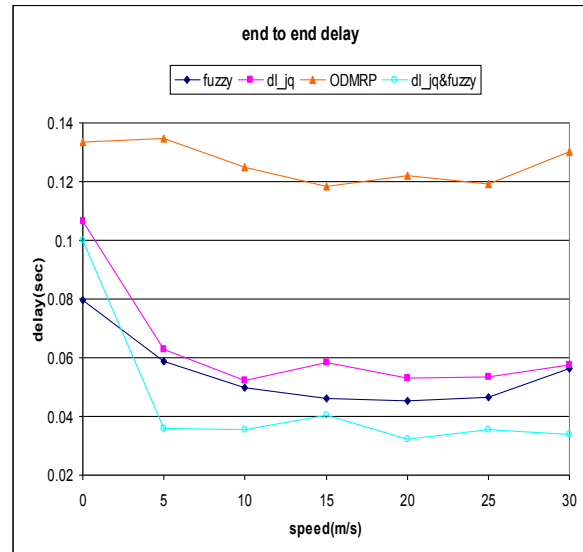


Figure 8. End to end delay with increasing speed of nodes

Figure 9 shows that number of received data packets is highest for dl_jq. In standard ODMRP, when speed of nodes increases, the number of link failures also increase.

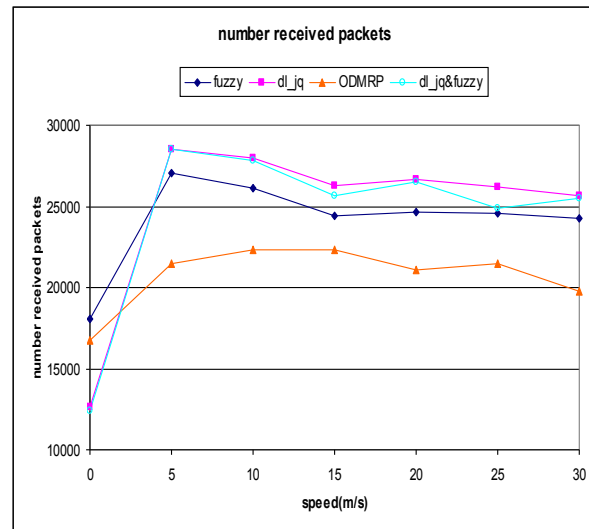


Figure 9. Nr. of received packets with increasing speed of nodes

While dl_jq shows high delivery ratio it has also higher overhead (Figure 10) compared to fuzzy and dl_jq combined with fuzzy. The benefit of the fuzzy method is to use a stronger set of forwarding nodes to configure new forwarding group.

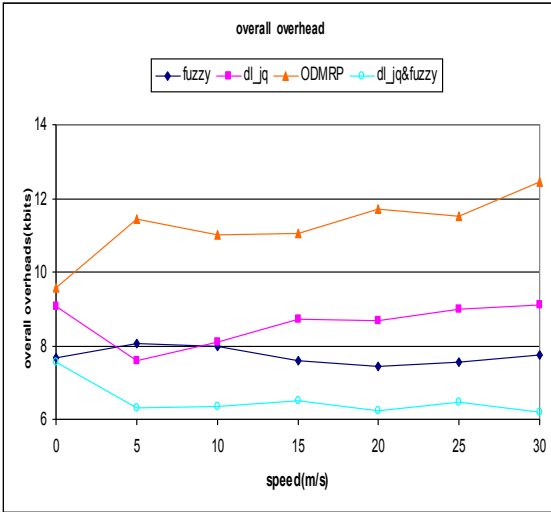


Figure 10. Overall overhead per a data packet

While standard ODMRP is tolerant to increasing speed of nodes, it suffers from high overhead especially at high node mobility. The proposed methods, especially dl_jq combined with fuzzy method is completely tolerant to increasing mobility. Because of the fuzzy method discards join query packets from nodes which are likely to show low delivery probability such as high mobility nodes.

A high overhead translates into high power consumption. Because dl_jq method has lowest overhead, it has lowest power consumption and highest network life time.

This can be seen from Figure 11, which also shows that under increasing speed of nodes all methods outperform ODMRP. In the fuzzy method, power consumption of any node interferes with join query packet discarding.

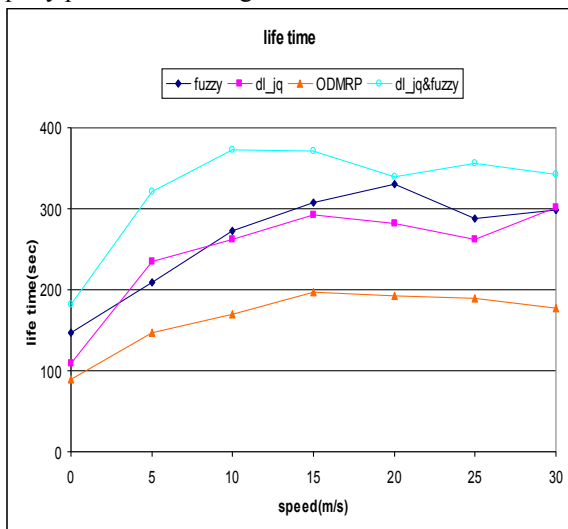


Figure 11. Life time with increasing speed of nodes

As a result, if a node has high power consumption value, it has low chance to be included in the forwarding group. With increasing speed of nodes combination of dl_jq fuzzy method shows highest network lifetime because it additionally discards a high number of join query and data packets.

5-3. Impact of number of nodes

Figure 12 presents end-to-end delay under increasing number of nodes. When number of nodes in network increases, overall overhead of ODMRP increases because the flooding domain is increased. In a network with more number of nodes, forwarding group has more members.

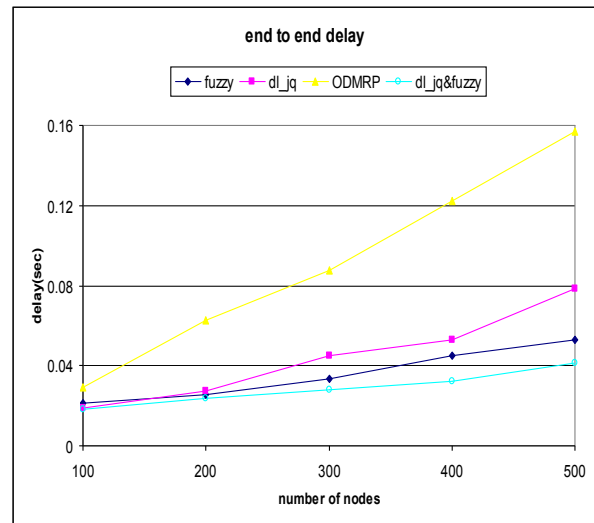


Figure 12. End to end delay with increasing number of nodes

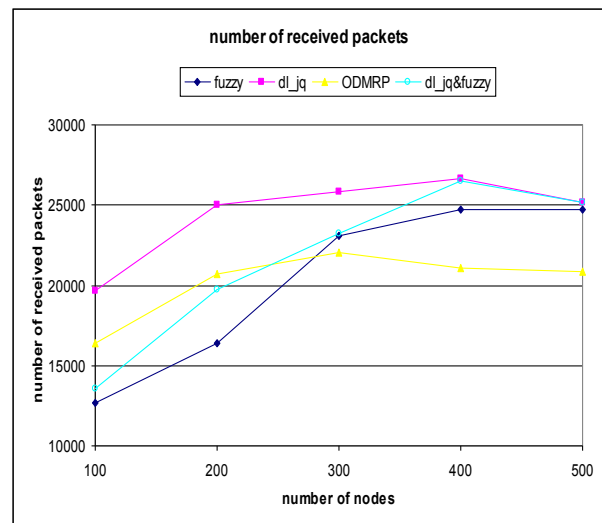


Figure 13. Nr. of received packets with increasing number of nodes

Also if network density is increased, loss rate is increased. If number of nodes is less than 300, standard ODMRP shows good performance in terms of delivery ratio because it performs complete flooding in entire network (see figure 13).

Standard ODMRP has good performance while network has sufficient capacity to transfer overall overhead.

When density of network is higher, number of nodes which are included in join query packet forwarding and forwarding group is increasing. This problem is critical when number of nodes is very high. Fuzzy methods try to use good nodes and routes which lead to higher delivery probability, lower overhead and delay, especially at denser networks.

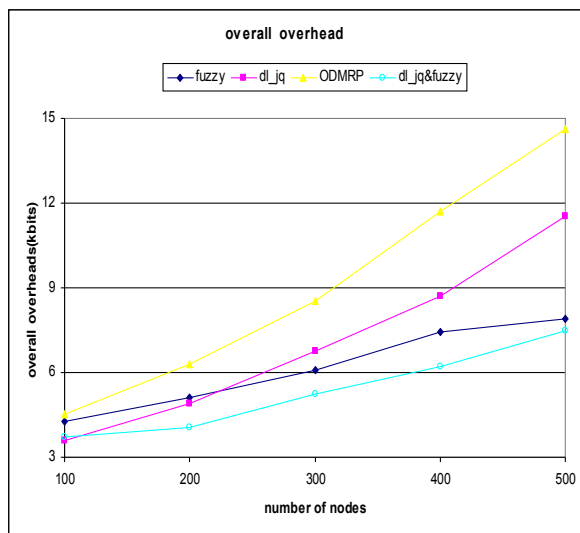


Figure 14. Overall overhead per a data packet

As a result, standard ODMRP delivers higher overhead (see Figure 14) to network resulting in higher delay and higher probability for collisions.

The fuzzy method has many choices to select a better route or node, which results in smaller delay. A combination of fuzzy approach with dl_jq shows the best performance due to the limitation of the flooding domain.

A higher overhead leads to higher power consumption and lower network life time. Again, dl_jq & fuzzy has lowest power consumption due to the lowest overhead. Details are not provided due to lack of space. While the standard ODMRP is very sensitive to increasing network density the proposed methods are more tolerant.

5-4 Impact of traffic density

In this scenario we varied the offered traffic load by changing the packet interval time. It's implicit if increase packet interval time, it send less packets. We

use 8 data senders which send contentiously packets with size of 1024 bytes. If interval time of packet (time between two packets) sending increase, the number of data packet flooding in forwarding group increase. Figure 15 shows end to end delay for all simulated methods. The standard ODMRP floods data packets in a great forwarding group. The proposed methods try to form a small and good forwarding group. A small forwarding group has very low overhead in comparison to standard ODMRP. Therefore the proposed methods have very low end to end delay in comparison to standard ODMRP.

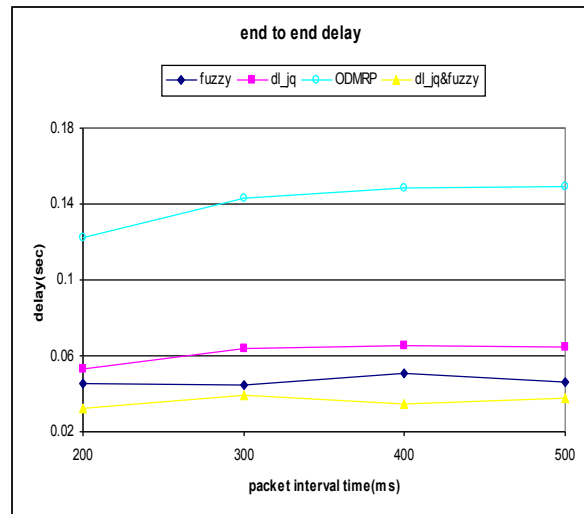


Figure 15. End to end delay with increasing packet interval time

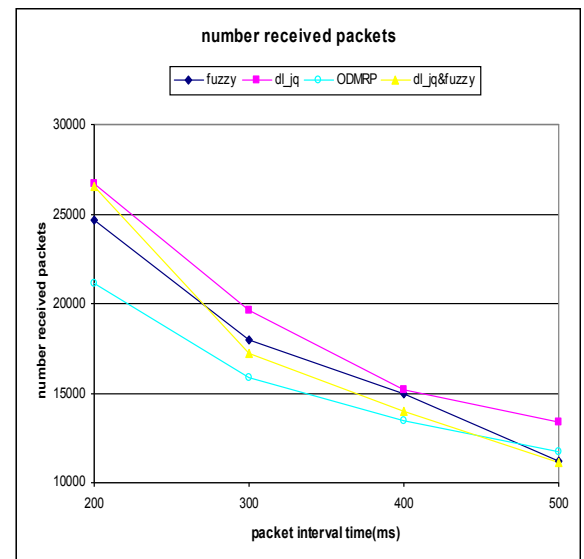


Figure 16. Number of received packet with increasing packet interval time

Also figure16 present number of received data packet by various methods. The fuzzy methods use good nodes and routes to forward data packet. Therefore they deliver very high number of data packets. Also these methods form a small and strong forwarding group instead great and weak forwarding group.

High rate data flooding (low interval time between packets) in a great and weak forwarding group makes increasing overall overhead and power consumption. Also it makes decreasing life time of network. The dl_jq & fuzzy method has very good results in these figures because it tries to use good and strong node. Also it withhold from forwarding of lost join query packets. In all of the figures for this scenario, the methods have same sequence. This sequence is derived from value of overall overhead.

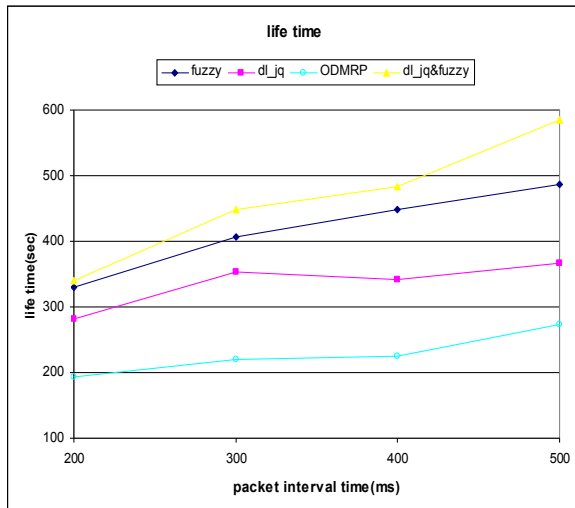


Figure 17. Life time with increasing packet interval time

In these figures forming strong forwarding group has high impact on performance in comparison to removing the lost join query packets. It is possible to configure a weak and great forwarding group in the dl_jq method. Therefore the fuzzy method has high performance in comparison to dl_jq method.

Overhead of standard ODMRP includes data overhead and control overhead. Data overhead is data flooding in forwarding group. Therefore increasing input traffic dramatically makes to increase data overhead. As result of high overhead, all sensor nodes use a great deal of their power; so the life time of network is reduced. Briefly standard ODMRP is very sensitive to increasing input traffic. But in the proposed method this sensitivity reduces because they delete the lost join query packets or form strong and small forwarding group.

5-5 Impact of speed membership functions

In this scenario, the impacts of different values for the speed membership function for the fuzzy method are analyzed. The following rule shows the implementation of the speed membership function:

If(speed of previous node < α)then
Speed of node is low
If(speed of previous node > α) and (speed of previous node < ($\alpha + \beta$))then
Speed of node is medium
If(speed of previous node > ($\alpha + \beta$))then
Speed of node is high

In order to form a strong and small forwarding group, it is required to consider many parameters which impact the quality such as speed, power and bandwidth. In the route discovery process, a node receives a join query packet and determines to discard or forward it. Any join query packet carries extra data about speed, power and bandwidth of nodes. The α and β parameters have high impact on the multicast routing process.

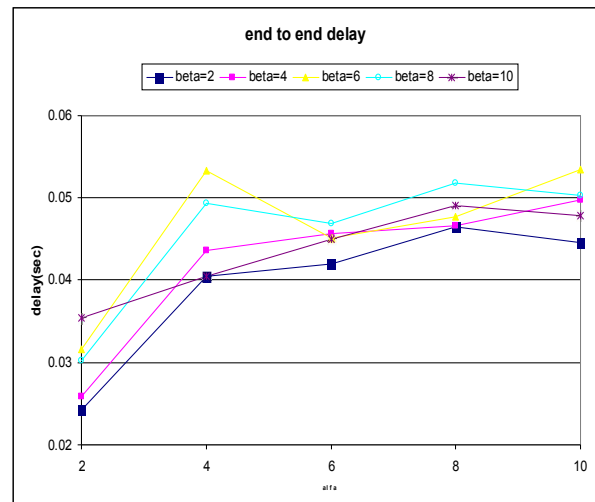


Figure 18. End to end delay

If a node has mobility value smaller than α then it is classified as a low speed node. Therefore it should be beneficial to include it in forwarding group leading to more stable delivery structure. If a node has mobility value greater than $\alpha + \beta$ then it is classified as a high speed node. Otherwise it is classified as a medium speed node. Therefore, the α value defines low border and the β value defines border between medium and high. High or medium speed nodes should be used with less probability as the delivery structures will become more fragile. Figure 18 shows end to end delay for various values of α and β , varying

between 2m/s to 10m/s. High value of α increases lower border and most nodes in the network are included as low speed node. Therefore, a high value of α results in high end to end delays because most of the nodes are included in the forwarding group.

The value of β has only minor effect because it defines the border between medium and high. In general, the effect of these parameters depends on the rules. Figure 19 shows the number of received data packets with different values for α and β . A high value of α increases the number of low speed nodes in network. As more nodes are classified as low speed there is a very high choice to select better routes. Also, a larger forwarding group is formed resulting in higher overhead. As a result, the fuzzy method will have high overall overhead and power consumption together with lower life time.

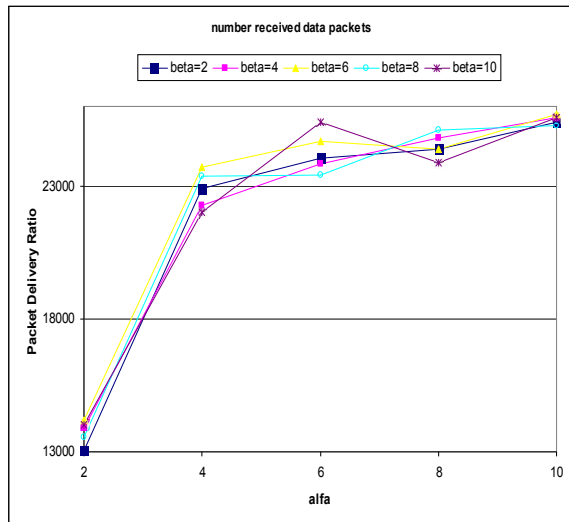


Figure 19. Number of received packets

The value of β changes the border between medium and high speed. In the rules for the fuzzy controller, the difference between low and high border has high impact on the forwarding probability. Therefore, if β is low, the number of nodes which are classified as high speed node is increased. As a consequence, a high value of α allows to form a good forwarding group. If the value of β is high it is possible that a high number of join query packets are deleted.

6. Conclusion and future works

In this paper we proposed two different methods to improve performance of multicast routing algorithms in MANETs specially the ODMRP. A fuzzy based approach is used to classify nodes into weak and strong ones. This is utilized when building a forwarding structure which optimally contains only strong nodes. As a result, only links and nodes which

are more robust or have more available power will participate in the forwarding mesh. The fuzzy based control allows us to deal with imprecise or incomplete information. By reducing the domain of control packet flooding, control overhead is further reduced. Based on those modifications, we implemented, simulated and evaluated a fuzzy based FBODMRP multicast routing mechanism for MANETs. The FBODMRP has many advantages to the standard ODMRP such as lower control and data overhead, low end to end delay and high packet delivery ratio in the high network and traffic density. Thus it could be used in many applications in ad hoc networks.

The fuzzy rule set has high impact on ODMRP performance. Ideally, the fuzzy rules should be selected based on network density, traffic intensity and other characteristics such as speed of nodes. As a result, in future work we will include tuning fuzzy rules using heuristics or using cognitive approaches based on learning observed behavior. In this paper manual tuning was used to adjust fuzzy controller. Also, based on result our method has high performance in noisily and high traffic environment, therefore it would be worth comparing FBODMRP with other multicast routing protocols such as MAODV. In this paper we simulate high-dense scenario. As the future works, we will develop the proposed policy to ensure that all the receivers in a multicast group can receive a packet.

Acknowledgements:

Authors are grateful to the Department of Computer Engineering, Damavand Branch, Islamic Azad University of Iran for financial support to carry out this work.

Corresponding Author:

Mr. Alireza Shams Shafigh
 Department of Computer Engineering
 Damavand Branch, Islamic Azad University
 Dmavand, Tehran, Iran
 E-mail: ali_shams@itrc.ac.ir

References

1. M. Iiyas, The Hand Book Of Ad Hoc Networks, CRC Press, 2003.
2. C. Siva, R. Murthy, B.S. Manoj, "Ad Hoc Wireless Networks Architectures and Protocols", PRENTICE HALL, 2004.
3. IETF Mobile Ad Hoc Networks (MANET) Working Group Charter, <http://www.ietf.org/html.charters/manetcharter.html>.
4. Obraczka, K, G. Tsudik, "Multicast Routing Issues in Ad Hoc Networks," Proceedings of IEEE ICUPC, Florence, Italy, October 1998.

5. E. Baburaj, V. Vasudevan, "Performance evaluation of multicast routing protocols in MANETs with Realistic Scenarios", *IETECH Journal of Advanced Computations*, Vol. 2, No. 1, pp. 15-20, 2008.
6. C.M. Cordeiro, H. Gossain, D.P. Agrawal, "Multicast over wireless mobile ad hoc networks: present and future directions", *IEEE Network*, Vol. 17, Issue 1, Jan.-Feb. 2003, pp. 52-59.
7. H. Gossain, C.M. Cordeiro, D.P. Agrawal, "Multicast: wired to wireless", *IEEE Communications Magazine*, Vol.40, Issue 6, June 2002, pp. 116-123.
8. J-G. Jetcheva, D.B. Johnson, "Adaptive Demand-Driven Multicast Routing in Multi-Hop Wireless Ad Hoc Networks," *Proceedings of ACM MobiHoc'01*, Long Beach, CA, October, 2001, pp. 33-44.
9. S.-J. Lee, W. Su, M. Gerla, "On-demand multicast routing protocol in multihop wireless mobile networks", *ACM/Kluwer Mobile Networks and Applications*, vol. 7, no. 6, pp. 441-452, December 2002.
10. E. Bommaiah, M. Liu, A. MacAuley, R. Talpade, "AMRoute: Ad hoc Multicast Routing Protocol," *Internet-draft*, work in progress, drafttalpade-manet-amroute-00.txt, August 1998.
11. C.-K. Toh, G. Guichala, S. Bunchua, "ABAM: On-Demand Associativity-Based Multicast Routing for Mobile Ad hoc Networks," *Proceedings of IEEE VTC-2000*, Boston, MA, September 2000, pp.987-993.
12. E.M. Royer, C.E. Perkins, "Multicast operation of the ad-hoc on demand distance vector routing protocol," *Proceedings of ACM/IEEE MOBICOM'99*, Seattle, WA, August 1999, pp. 207-218.
13. C.-W. Wu, Y.-C. Tay, C.-K. Toh, "Ad Hoc Multicast Routing Protocol Utilizing Increasing id-number S (AMRIS) Functional Specification," *Internet-draft*, work in progress, draft-ietf-manet-amris-spec-00.txt, November 1998.
14. Hasnaa Moustafa, Houda Labiod, "A performance comparison of Multicast routing protocols in ad-hoc networks", *The 14th IEEE 2003 International Symposium on personal, indoor and mobile radio communication proceedings*, p. 497-501.
15. Kumar Viswanath, Katia Obraczka, Gene Tsodik, "Exploring Mesh and Tree-Based Multicast Routing Protocols for MANETs", *IEEE Transactions on mobile computing*, vol. 5, No. 1, p.28-42, Jan. 2006.
16. R. Chandra, V. Ramasubramanian, K. Birman, "Anonymous Gossip: Improving Multicast Reliability in Mobile Ad-Hoc Networks," *Proc. IEEE Int'l Conf. Distributed Computing Systems*, IEEE CS Press, 2001, pp. 275-283.
17. J. Luo, P.T. Eugster, J-P. Hubaux, "Route Driven Gossip: Probabilistic Reliable Multicast in Ad Hoc Networks," *Proc. IEEE Infocom 2003*, IEEE Press, 2003, pp. 2229-2239.
18. R. Bagrodia, M. Gerla, J. Hsu, W. Su, S.-J. Lee. "A performance comparison study of ad hoc wireless multicast protocols", *Proc. of the 19th Annual Joint Conf. of the IEEE Computer and Communications Societies*, March 2000, pages 565-574.
19. J. Schiller. "Mobile Communications" Addison-Wesley, 2000.
20. X. Zeng, R. Bagrodia, M. Gerla, "Glomosim: A library for parallel simulation of large-scale wireless networks," in *Proc. of PADS Workshop*, May 1998.
21. I.Hwang, W.Pang," Energy Efficient Clustering Technique for Multicast Routing Protocol in Wireless Ad Hoc Networks", *IJCSNS International Journal of Computer Science and Network Security*, VOL.7 No.8, August 2007
22. J.H. Ryu, S. Song, D.H. Cho, "Energy-conserving clustering scheme for multicasting in two-tier mobile ad hoc networks", *IEEE Electronics Letters*, Vol. 37, Issue 20, Sept. 2001. pp. 1253-1255.
23. C. Tang, C.S. Raghavendra, V. Prasanna, "Energy efficient adaptation of multicast protocols in power controlled wireless ad hoc networks", *International Symposium on Parallel Architectures, Algorithms and Networks*, May 2002. pp. 80-85.
24. L. Ji, M. S. Corson, "Differential Destination Multicast: A MANAT Multicast Routing Protocol of Small Groups", *Proceedings of IEEE Infocomm*, pp. 1192-1202, 2001.
25. M. Gerla, G. Pei, S.Lee, C.C. Chiang," On-Demand Multicast Routing Protocol (ODMRP) for Ad-Hoc Networks", *IETF MANET Working Group*, INTERNET-DRAFT. November 1998.
26. S. Lee, C. C. Chiang, M. Gerla, "On-Demand Multicast Routing Protocol", *IEEE Conference on wireless communication and networking*, vol.3, September 1999, pp1298-1302.
27. E. Royer, C. Perkins. Multicast Ad hoc On-Demand Distance Vector (MAODV) Routing, *IETF*, *Internet Draft: draft-ietf-manet-maodv-00.txt*, 2000.
28. C. Perkins, E. Belding-Royer, S. Das. Ad hoc on-demand distance vector (AODV) routing, *RFC 3561*, 2003.
29. A. Laouiti, P. Jacquet, P. Minet, L. Viennot, T. Clausen, C. Adjih. Multicast Optimized Link

- State Routing, INRIA research report RR-4721, 2003.
30. T. Clausen P. Jacquet (editors). Optimized link state routing protocol (OLSR). RFC 3626, October 2003. Network Working Group.
 31. J. Garcia-Luna-Aceves, E. L. Madruga. The core-assisted mesh protocol. *Journal on Selected Areas in Communications*, 17(8):1380-1394, August 1999.
 32. Ozaki, T, J.B. Kim, T. Suda, "Bandwidth-Efficient Multicast Routing for Multihop, Ad Hoc Wireless Networks," *Proceedings of IEEE INFOCOM*, Alaska, USA, April 2001.
 33. Lee, S, C. Kim, "A new wireless ad hoc multicast routing protocol," *Elsevier Science Computer Networks*, Vol. 38, pp. 121-135, February 2002.
 34. Ji, L, M.S. Corson, "Explicit Multicasting for Mobile Ad Hoc Networks," *Mobile Networks and Applications (MONET)*, Vol. 8, No. 5, pp. 535-549, October 2003.
 35. Sajama, S, Z.J. Haas, "Independent-Tree Ad Hoc Multicast Routing (ITAMAR)," *ACM Mobile Networks and Applications*, Vol. 8, No. 5, pp. 551-566, October 2003.
 36. Chen, Y.S, Y.W. Ko, T.L. Lin, "A Lantern-Tree-Based QoS Multicast Protocol for Wireless Ad Hoc Networks," *Proceedings of IEEE ICCCN*, Miami, USA, October 2002.
 37. Chen, Y.S, Y.W. Ko, "A Lantern-Tree-Based QoS On-Demand Multicast Protocol for a Wireless Mobile Ad Hoc Network," *IEICE Transactions on Communications*, Vol. E87-B, No. 3, March 2004.
 38. Pompili, D, M. Vittucci, "A Probabilistic Predictive Multicast Algorithm in Ad Hoc Networks (PPMA)," *Proceedings of MEDHOC-NET*, Mugla, Turkey, June 2004.
 39. B. Rong, G. Amoussou, "Mobility Prediction Aided Dynamic Multicast Routing in MANET", *Sarnoff* 2005.
 40. V.lashkari,M.dehghan,"QoS-aware Multicast Ad hoc On demand Distance Vector Routing", pp1506-1511, *ICWN2007*.
 41. M. Saghir, T. Wan, R. Budiarto, "QoS Multicast Routing Based on Bandwidth Estimation in Mobile Ad Hoc Networks", *Proceedings of the Int. Conf. on Computer and Communication Engineering, ICCCE'06 Vol. I*, 9-11 May 2006, Kuala Lumpur, Malaysia.
 42. D. Pathirana, "Resilient On Demand Multicast Routing Protocol", *Master of Science thesis, ROCHESTER INSTITUTE OF TECHNOLOGY*, February 2007.
 43. A. Shams Shafigh K. Abdollahi, A. J Kassler, "Improving performance of ODMRP by Fuzzy Logic Control", *WCNIS2010, china*.
 44. K. Abdollahi, A. Shams Shafigh, A. J Kassler, "Improving performance of ODMRP by Deleting Lost Join Query Packets", *ACIT2010, Spain*.