

Research Article

Developing a Fuzzy Logic Based on Demand Multicast Routing Protocol

Alireza Shams Shafigh, Kamran Abdollahi, and Marjan Kouchaki

Department of Computer Engineering, Islamic Azad University, Damavand Branch, Tehran, Iran

Correspondence should be addressed to Alireza Shams Shafigh, ali.shams@itrc.ac.ir

Received 14 March 2011; Revised 24 July 2011; Accepted 10 October 2011

Academic Editor: Adnan Kavak

Copyright © 2012 Alireza Shams Shafigh et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Multicast routing is an efficient method to lead data packets from one source group to several nodes as destination group. Although multicast routing algorithms could be efficient in many situations but their routing mechanism like as route request flooding packets likely results in poor performance in comparison to unicast routing algorithms. In this research, two efficient methods are proposed to improve the performance of On Demand Multicast Routing Protocol (ODMRP). The main proposed method tries to establish a small, efficient, and high-quality forwarding group. This is achieved by augmenting the Join Query packets with additional information such as speed, power level of node, and link bandwidths. Besides, the control overhead is further reduced by restricting the domain of control packet flooding (by restricting the domain of control packet flooding). The performance evaluation shows that the proposed scheme increases the packet delivery rate by up to 40%, while reducing average end-to-end delay and consumed power by about 35% and 45%, respectively.

1. Introduction

A mobile ad hoc network (MANET) [1–4] is formed by wireless mobile nodes which dynamically set up themselves to initiate and maintain a wireless network without any predefined infrastructure. For relaying the packets towards the destination, each node needs to implement routing functionality. Multicasting is the transmission of data packets to a receiver group which identified by a single multicast IP address. Multicasting efficiently supports a variety of applications. It is an efficient method to deliver packets from the source group to any number of receiver nodes.

Multicast routing algorithms [5–9] are one of the most important research fields in wireless ad hoc networks due to their ability to deliver data flows to a large set of nodes. Multicast protocols allow nodes to form a multicast delivery structure which in normal cases performs better than using multiple unicast routing paths. This is more critical in ad hoc environments, where bandwidth, power resources, and processing abilities are very valuable. The major problem, however, is that nodes are able to move randomly and make frequent and unpredictable topology changes. Although

multicast routing algorithms are desirable in many applications, they need to improve their forwarding mechanism due to their network resource consumption.

Multicast routing algorithms include three main groups: tree-based, mesh-based, and hybrid approaches with regard to their delivery structure. Tree-based algorithms [10, 11] establish a tree structure between source and receiver groups. Therefore, there is only one path between each source-receiver pairs. Due to their lower overhead, they are more efficient than mesh-based approaches. Mesh-based multicast routing algorithms establish a mesh to forward data packets to receiver group. They could be more efficient than tree-based algorithms due to redundant routes in mesh topology. Whenever the network topology changes frequently, mesh-based protocols overcome tree-based protocols due to the route redundancy. Many tree- and mesh-based protocols have been proposed [12–14] to improve performance of multicast routing protocols.

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 introduced by multicast protocols to network in order to provide a reliable packet transmission. For establishing the delivery structure, a standard approach is to use network sending of control packets in throughout of network. However, this method results in 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.

In this paper, two new methods have been proposed to improve the performance of the multicast routing protocols in MANETs. The first method exploits fuzzy logic control to deal with imprecise and partial information during the construction phase of a forwarding structure. The experimental and simulation results show that such enhancement provides more stable routes in terms of lower end-to-end delay and power consumption, higher packet delivery ratio.

Firstly, the fuzzy control has implemented in mesh-based ODMRP routing protocol by 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 *Join Query* packets at which the queries received from nodes with "better quality" gain the higher probability to be forwarded.

The second method supposes that during the route refreshment the new forwarding group has many nodes in common with the old one. This assumption enables restricting the *Join Query* flooding in the network thus reducing the overhead.

The rest of paper is organized as follows. Section 2 describes related work. Section 3 briefly reviews the operation of the basic ODMRP protocol. Section 4 presents modifications to current ODMRP to design the fuzzy-based FBODMRP protocol. In Section 5, an evaluation of the proposed approach is presented using simulation over a wide range of simulation scenarios, where FBODMRP is compared with original ODMRP through a number of network conditions. Finally, the paper concludes and presents suggestions for future works in Section 5.

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. As mentioned earlier, multicast routing protocols could be divided into tree-, mesh-based and hybrid approaches depending upon the delivery structure. Examples of tree-based protocols are MAODV [15, 16] based on the unicast routing protocol AODV [17], MOLSR [18] which is an extension to OLSR

unicast routing protocol [19], and Adaptive Demand-driven Multicast Routing protocol (ADMR) [20] that is an extension of AMRoute [21]. Some of important mesh-based routing protocols are On Demand Multicast Routing Protocol (ODMRP) [22–24] and Core-Assisted Mesh Protocol (CAMP) [25].

Many different routing protocols have been proposed for MANETs trying to solve the limitations of MANETs by improving of delivery structures and taking into account parameters such as reliability [26, 27], power consumption [28, 29], mobility, and bandwidth. The Bandwidth-Efficient Multicast Routing (BEMR) finds the nearest forwarding group member for nodes broadcasting Join Queries [30]. Association Based Ad hoc Multicast (ABAM) builds a source-based multicast tree [31]. 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 [32]. A forwarding group is created by a source at which requests are broadcasted. With the aim of maintaining 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 [33, 34]. The Independent-tree ad hoc multicast routing (ITAMAR) computes a set of independent multicast trees by several heuristics [35]. The Lantern-tree-based QoS multicast (LTM) is a bandwidth routing protocol with an improved success rate by means of multipath routing [36, 37]. The QoS-MAODV extends existing Multicast Ad hoc On Demand Distance Vector Routing protocol, by using admission control and bandwidth reservation in each node [38]. In QoS-MAODV, a cross-layer framework has been proposed to support admission control using information on available bandwidth, which was estimated without extra control overhead. Authors in [39] introduced a weight-based clustering technique for multicast routing protocol. Resilient on Demand Multicast Routing Protocol (RODMRP) is an ODMRP-based wireless multicast protocol that offers more reliable forwarding paths in face of node and network failures [40].

The main problem of existing approaches could be found in 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 enough such as composed of nodes which have high available capacity and good links in order to increase packet delivery further. In this paper, two novel methods are proposed which not only reduces overhead significantly but also allows creating robust delivery structures using fuzzy based control.

2.1. Overview of ODMRP. One of the mesh-based multicast routing protocols in MANETs is ODMRP. In ODMRP [41], group membership and multicast routes are established and updated by the multicast source on demand. ODMRP routing mechanism comprises a request and reply phase like as unicast routing protocol. While a multicast source has

packets to send, it floods a *Join Query* packet. This *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 nonduplicate *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.

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 rebroadcast the packet, otherwise it is discarded. Once the source has no more packets to send, it just stops sending and the forwarding state times out in the forwarding nodes.

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 increase 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 (e.g., 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.

2.2. Overview of AAMRP. AAMRP [42] protocol tries to develop broadcasting rather than multicasting when the group members are dense and the speed is high while uses multicasting when the group members are sparse and also the speed is low. As a common feature of ad hoc networks wherein nodes are highly dynamic, either of two strategies can be developed at various scenarios. This protocol diminishes the control overhead via reducing the number of group members participating in the construction procedure of the multicast tree and providing robustness to mobility by performing broadcasts in densely clustered local regions. Moreover, AAMRP achieves packet delivery statistics that are comparable to those with ODMRP but with significantly lower overheads (refer to Figures 22 and 23).

The process of group construction is as follows: each group member can be in 3 different states. It can be in a temporary mode wherein it is JOINING a session. It can be a cluster *LEADER*, or it can be a typical *MEMBER* of a cluster leader. Each node maintains a Group Member Table (*GMTable*) which contains the information of the joining group members. The information maintained in *GMTable* can be achieved using *ADVERTISE* and also the *LEADER* messages. Each cluster leader maintains a Cluster Member Table (*CMTable*) including information of the entire cluster members that are associated with the cluster leader. The information maintained in this table is obtained by reception of *MEMBER* messages that are sent out by each cluster member. Wherein each cluster leader periodically broadcasts a *LEADER* message to its cluster, to notify its presence to the cluster members. Also, this message can be assumed as an invitation for new members desiring to associate with the cluster. Each cluster member estimates the distance between itself and the leader and periodically sends this distance information to the cluster leader through a unicast *MEMBER* message. The frequency of these unicast updates from a member depends on this distance of the member from the leader (i.e., if the cluster member is closer to its leader, it is less possible that it is out of range). When the cluster leader receives a data packet from the multicast source, the packet is broadcasted to the group members within its cluster. At this stage, some type of adaptive broadcast is performed wherein the maximum broadcast range is depending on the furthest child of the cluster leader, that is, the broadcast range can be reduced as per the distance of the furthest child, stored in the *CMTable*. Owing to this adaptive broadcast, redundant and unwanted transmissions of data can be reduced.

3. FBODMRP (Fuzzy Based ODMRP)

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

3.1. Handling the 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. In proposed approach, a node is classified as “strong” when it has certain desirable properties which increase the probability of successful data delivery when such node will participate in the forwarding group. In this paper, such properties are defined as 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 is less likely to be included in the forwarding process. For example, if a node has a low power level, the probability that the mode powers down in the near future will be high leading to packet loss due to node failures and rerouting.

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. However, in many cases, it is impossible to establish forwarding group in the network only by using of strong nodes. For example, some nodes might have weak links since mobility and link quality might change over time. Also, it might be possible that no *strong* group exists. A *strong* forwarding group will be smaller than a *weak* forwarding group, because such strong group removes weak nodes from forwarding group. In the standard ODMRP, any node which receives a new *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 processed by a node with low power level. One simple approach is to wait until all copies of a *Join Query* packets are received and select the best *Join Query* using extra information inserted into such packets. However, this approach delays arrival time of *Join Query* packets at receivers.

When classifying nodes into *strong* or *weak* ones, it is proposed 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. Aiming at applying the fuzzy logic method, precise information is necessarily required facilitating classification of nodes into *weak* or *strong* groups. Such information will then be exploited in the delivery structure creation in order to include strong nodes in the mesh. Therefore, several fields are attached to the *Join Query* packet which carries extra information on bandwidth availability, loss rate experienced, moving speed, or power level to allow the nodes to perform a better route selection in the route establishment process. Relying on such information, next nodes will be able to compute the probability of caching and forwarding the received *Join Query* message.

In the proposed method, higher quality routes (i.e., routes composed of nodes and links having high bandwidth, high power level, and low mobility) have a high probability to be included in forwarding group. Meanwhile, *weak* routes (low bandwidth, low power level, and high mobility) have a lower probability. As a result, a small and strong forwarding group is established 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 rerouting. Once a node receives a *Join Query*, it needs to process the additional parameters in a way in order to achieve a better forwarding structure. This paper exploits fuzzy logic to cope with network dynamics, uncertainty, and imprecise information. Therefore, when processing the *Join Query*, the node applies fuzzy rules on bandwidth, speed of node, and power level of previous node saved in *Join Query*. As a result, the node computes the probability of caching and forwarding of the request. If a *Join Query* has

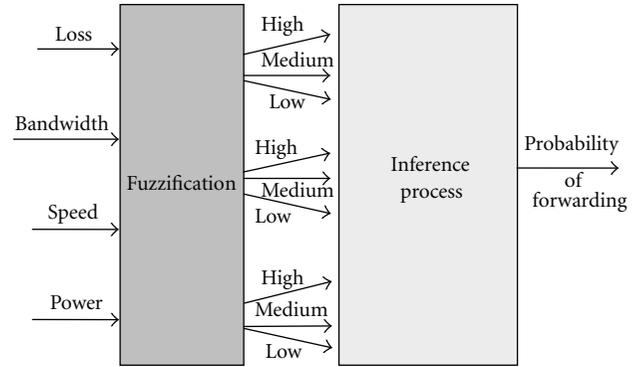


FIGURE 1: Computing probability of caching or forwarding.

arrived through a *strong* node, the packet is cached and forwarded with a high probability. To fuzzify parameters, simple membership functions are used. 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, the node replaces the parameters in the *Join Query* packet before it forwards the packet with its own information.

Our proposed fuzzy method introduces no extra overhead to exchange the required information about bandwidth, loss, and power. It only reuses the control packets (i.e., *Join Query* packets) to carry the average values (loss, power, and bandwidth) to its neighbors and only some bytes are added. The initial probability of *Join Query* forwarding is set to 100%. If it is assumed 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 of proposed method is evaluated in Section 4.5. 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, only information inside *Join Query* packets and some local updated average values for loss, bandwidth, and power are used. These average values are used in designing fuzzy rules. They determine how much parameters (loss, bandwidth, and power) are good. These average values would be more accurate with receiving large number of *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. The input to this process is the previous node’s operating parameters (such as bandwidth, speed, and power) where the probability of caching and forwarding of the packet is the output of the process (Algorithm 1).

3.2. *Partial Flooding in FBODMRP*. In standard ODMRP, periodic sending of *Join Query* packets is a critical problem,

```

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

```

ALGORITHM 1

because *Join Query* packets propagate in the entire network. As a result, such wide network 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, it is proposed to restrict the domain of the *Join Query* packet flooding.

The source ID and the time at which the last *Join Query* are received, it is recorded for each multicast group member. If no *Join Query* is received from a source within the refresh period the entry is removed from the *Member Table*. In original ODMRP when a node has data to transfer, a *Join Query* packet is flooded to the entire network. Hence, a source node with a continuous traffic flow will refresh the previous forwarding group before it is 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. The *Join Query* packets which intended to be forwarded to a region far away from the previous forwarding group will be dropped.

To implement this idea, a new field is added 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 does not see any previous FG nodes, it is assumed that this *Join Query* packet is useful, so it is discarded. Thus, when a node receives a new *Join Query* packet, it extracts NOPFG (number of previous forwarding group) and Hop count fields from the incoming packet. The Hop count field is the number of hops traversed from the sender to such node. When a node receives a *Join Query* packet with a hop count greater than the minimum value, it decides whether to forward the packet or discard based on a random value. This random value is based on the forwarding probability which is calculated by fuzzy model (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.

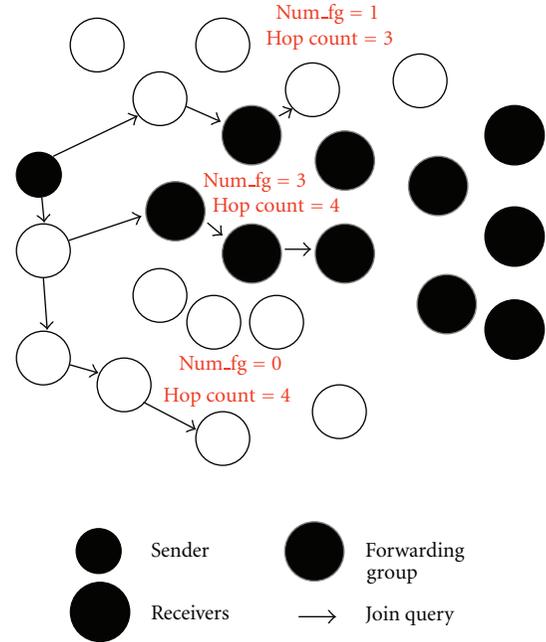


FIGURE 2: Intersection of new forwarding group with previous forwarding group.

Figure 2 shows an example for NOPFG field. Node 1 has received a *Join Query* with NOPFG and hop count equals to 1 and 3, respectively.

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 such confining technique when it knows there is not the previous forwarding group. It informs this with setting the NOPFG to a minus value.

To that extent, in particular when the density of the network is very low, the second method may lead to failure in finding route to destinations. Thus, in the initial stage of a network, when there is no previous forwarding group it will be an optional policy. This policy decides whether the second method must be applied or not. Each source node before sending a *Join Query* restores time of the previous *Join Query* packet. If such period is very low, this implies the link broken, long periods show that source node has originated some data packets through the network. The source node makes a decision based on the previous *Join Query* about running the second method by intermediate nodes. It sets the field of type to a special value (e.g., 2). Thus, the second method is not applied when there is not any forwarding group. It means that the time period, between two successive sending of join queries, is very low (smaller than 3 seconds) or very high (greater than 4 seconds). Concisely, reception of data is not guaranteed at all receivers. The second proposed method only is used in high-dense networks, and implementation of this method in low-dense network is highlighted as the area of future research.

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

FIGURE 3: Structure of Join Query packet.

```

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
}

```

ALGORITHM 2

3.3. *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. Figure 3 shows the new structure of the *Join Query* packet. The grey fields indicate extra fields added in the proposed method. The first field defines type of our policy. The policy specifies that the method must be run by members of the network. This field is set to Zero to inform other nodes to run both of the proposed methods, value of one results in running of the first method, and the value of two causes running of the second method. When there are some problems in finding route to destination, this field is adjusted to three to inform members to run the original ODMRP without any changes. The second field is used by the second method, which confines flooding range. The third to the sixth fields are quality parameters used by the first method, which try to add strong link or node to forwarding group.

The processing of *Join Query* messages is described in the following. The packet forwarding process of FBODMRP is identical to the original ODMRP. In the route establishment process of FBODMRP, each node forwards packets with higher probability, if the previous node is a strong node. The pseudo code in Algorithm 2 presents all actions taken while receiving a *Join Query* packet.

Four policies might be taken in various situations by source node. The standard model of ODMRP is implemented with type of four. Three policies are defined by 0, 1, and 2. The pseudocode in Algorithm 3 shows functions of *Dl.jq* and fuzzy method.

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.

```

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;
}

```

ALGORITHM 3

The *jq_packet* denotes a *Join Query* packet and *num_fg* counts the number of previous forwarding group members which have been visited by such *Join Query* packet. Forwarding of the query is accomplished based on the calculation value of the probability p . A *Join Query* packet is sent with probability of p and discarded with probability of $(1 - p)$.

A *Join Query* packet is declared as lost if it has traversed several hops but has not visited any node from the previous forwarding group. In the simulation part, the refresh time of forwarding group is set to three seconds while memberships of forwarding group nodes are set to four seconds. Therefore, if the *Join Query* packet is not processed by any node from the last forwarding group within the refresh interval, it is considered as lost packet or the route as wrong.

4. Result and Discussion

To evaluate the performance of our approach, several scenarios are simulated using GloMosim [45]. This simulation study modeled a network of mobile nodes placed randomly within an area of 3000×3000 (area simulation is 3000 meter \times 3000 meter.). Each simulation is executed for 600 seconds of simulation time. The mobility model is Random-Way point where nodes randomly selected the moving direction, and when they reached the simulation terrain boundary they bounced back and continued to move.

Each node has different physical properties, and all nodes have a random bit rate between 2 Mbps to 10 Mbps. The assumption is made in order to present performance of the proposed method. Such bit rate is used as bandwidth in *Join Query* packets. Also, it is assumed that nodes are equipped with GPS to get information on their position and speed.

The default values for the fuzzy membership functions are the following: low and medium values for speed are 6 m/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.

The number of data sessions was set to 8, and the multicast group size was set constant at 8. Each member node

sends constant bit rate traffic at the rate of 5 packets (1024 bytes) per second to all other member nodes. The following three methods have been simulated and compared along with standard ODMRP.

Fuzzy Method. This method uses fuzzy control to select high-quality nodes.

DL-jq Method. This method deletes the lost *Join Query* packet using number of previous forwarding group. This method is tested 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, and some destinations possibly do not receive any packets.

Fuzzy and 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.

- (i) *End-to-end delay:* average end-to-end delay of packets which have been received by receivers during the simulation period.
- (ii) *Number of received packets:* overall number of packets which have been received during the simulation period. This allows calculating the packet loss rate.
- (iii) *Overall overhead:* sum of control and data packets (in bits) per data packet delivered.
- (iv) *Life time:* it is the minimum time from start time until the first node uses up its power.
- (v) *Consumed power:* sum of consumed power used in the entire network per data packet. The standard GloMosim power calculations have been used.

Besides, 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, the speed, and power membership functions is analyzed in this paper.

4.1. Impact of NOPFG and Minimum Hop Count. In case of continuous traffic flow, new forwarding group is refreshed before expiry of previous forwarding group. 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 experiments analyzes the impact of NOPFG and minimum hop count on delay and packet loss. The mobility speed of nodes was fixed to 20 m/s, and *dL-jq* was simulated with 400 nodes. Figures 4 and 5 show end-to-end delay and number of received packets under increasing NOPFG and variation of hop count.

It is observed that a high minimum hop count leads to fewer lost packets and thus increases delivery probability. However, it also leads to higher average delay. The Minimum hop count had a higher impact compared to NOPFG since

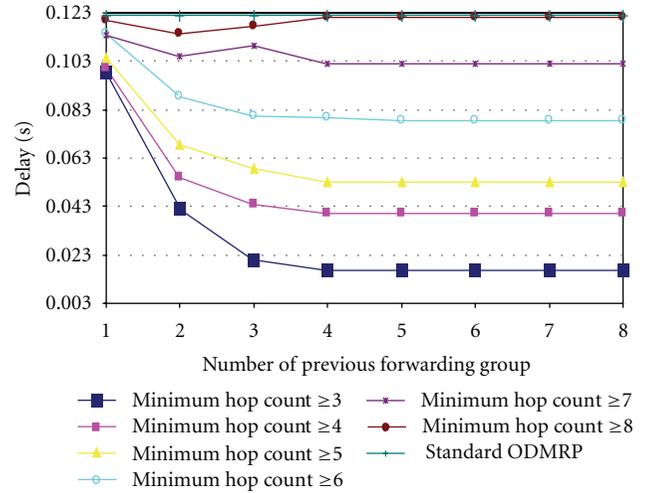


FIGURE 4: End-to-end delay.

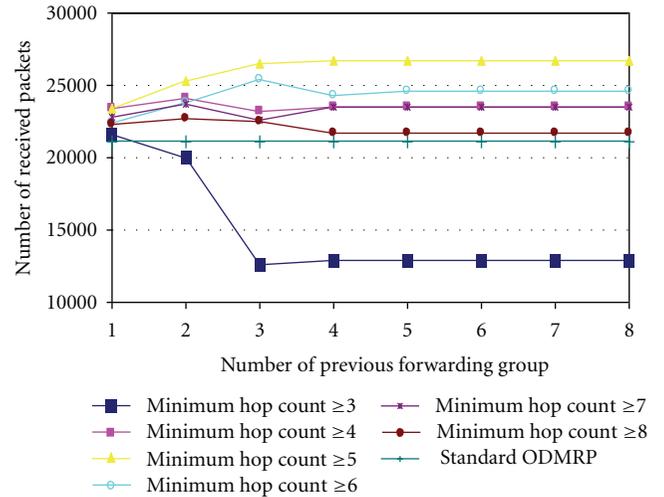


FIGURE 5: Number of received packets.

increasing the NOPFG beyond some threshold had minor impact on performance.

For a low minimum hop_count value, fewer packets have been received compared to ODMRP. When minimum hop count is low, a large number of *Join Query* packets are declared lost. This results in a weak 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. Figure 6 shows the overall overhead per data packet delivered as function of number of members in previous forwarding group. The *dL-jq* outperforms ODMRP, when the number of previous forwarding group increases.

A high overhead is translated into higher delay and power consumption and low throughput. While in standard ODMRP periodic flooding of *Join Query* packet results in high overall overhead, in *dL-jq* overhead is significantly reduced due to many lost *Join Query* packets. A similar result

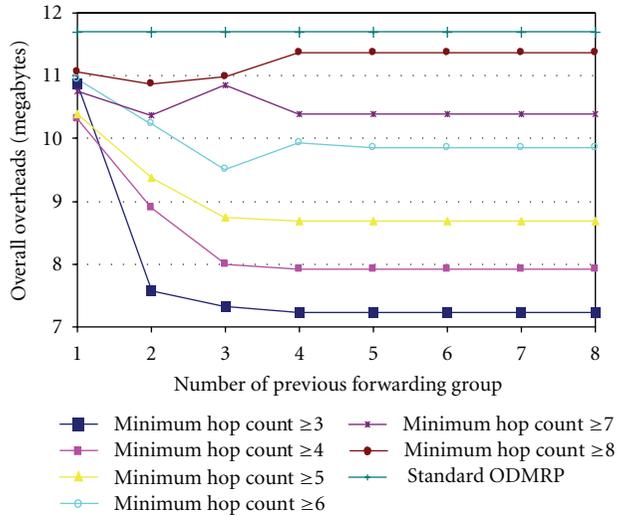


FIGURE 6: Overall overhead (megabytes) per a data packet.

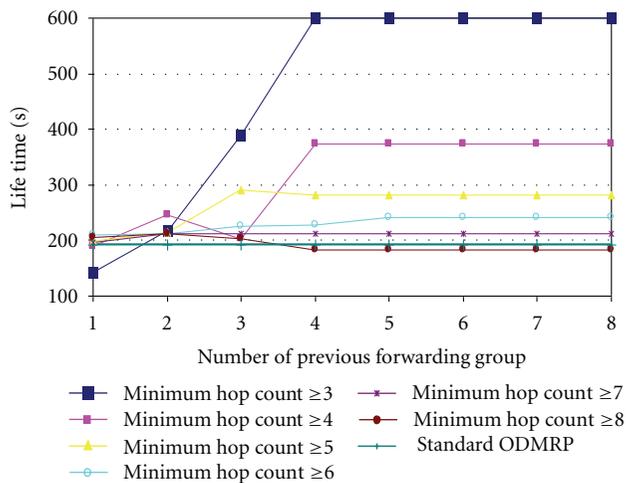


FIGURE 7: Life time.

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 compared to ODMRP increased (see Figure 7). The *dL.jq* with minimum hop count = 3 shows the highest life time because of the smallest overhead.

According to experimental results, it is concluded that values for NOPFG = 4 and minimum hop count = 5 result in an acceptable compromise between overhead, delay, and delivery probability. Therefore, these values are used in the following scenarios.

4.2. Impact of Speed. Node mobility leads to topology change which is a challenge for multicast routing protocols. In this section, the performance of FBODMRP is analyzed under increasing node speed. Figure 8 compares end-to-end delay of the proposed methods with standard ODMRP under

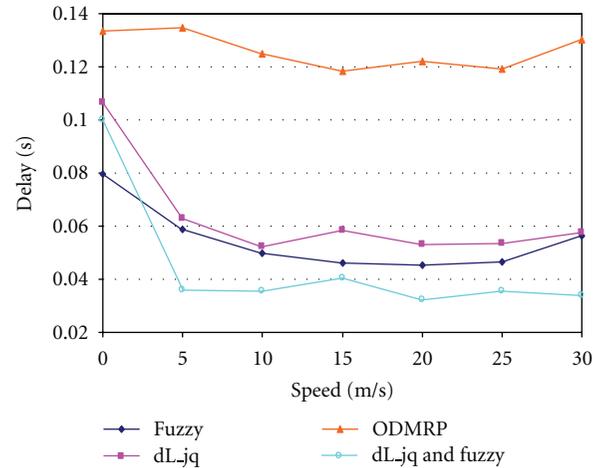


FIGURE 8: End-to-end delay with increasing speed of nodes.

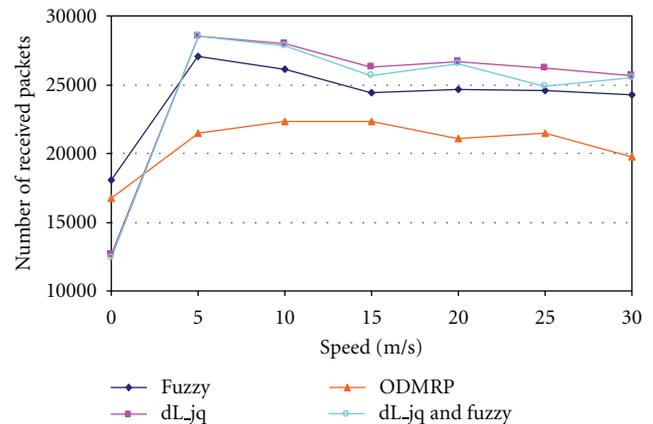


FIGURE 9: Number of received packets with increasing speeds of nodes.

different node mobility speeds. The original ODMRP has high delay compared to our proposed methods. ODMRP periodically sends *Join Query* messages to refresh forwarding group. This makes it tolerant to mobility of nodes but also leads to increased waiting time in queues. In contrast, all of the proposed methods perform restricted flooding. In the combined method (i.e., *dL.jq and fuzzy*), number of *Join Query* packet discarding is high which results in low end-to-end delay.

It would appear from Figure 9 that, as the mobility speed increases, the highest number of received packets is achieved in *dL.jq*. However, ODMRP presents the minor number of received packets in presence of mobility compared to other studied protocols. As it is shown, the largest increase in number of successfully delivered packet in four approaches occurred when speed is 5 km/hr. Since then, the graphs have followed a steady trend.

Undoubtedly, high delivery ratio in *dL.jq* result in the high control and data overhead (Figure 10). However, such overhead is still an order of magnitude smaller than the overhead introduced in ODMRP. The benefit of the fuzzy

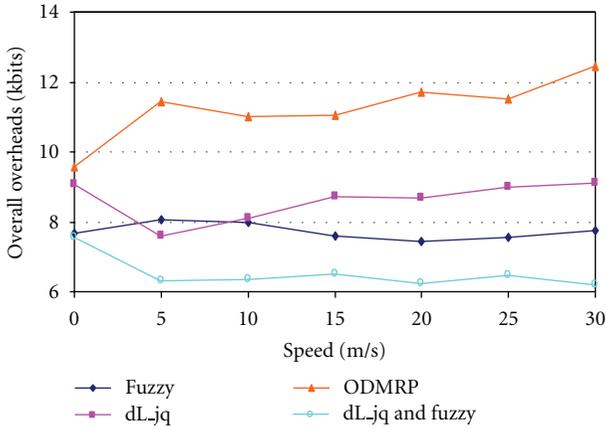


FIGURE 10: Overall overhead per a data packet.

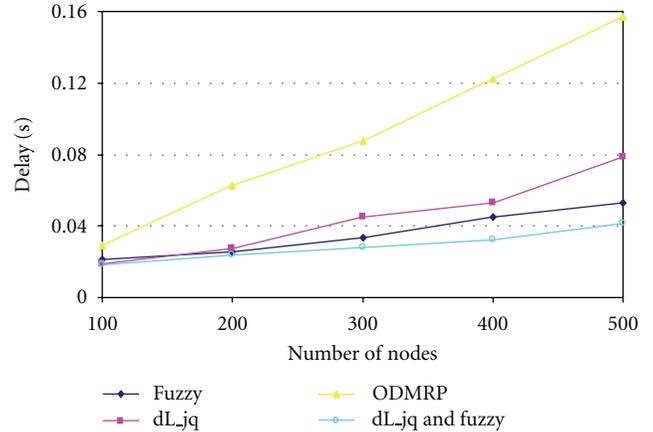


FIGURE 12: End to end delay with increasing number of nodes.

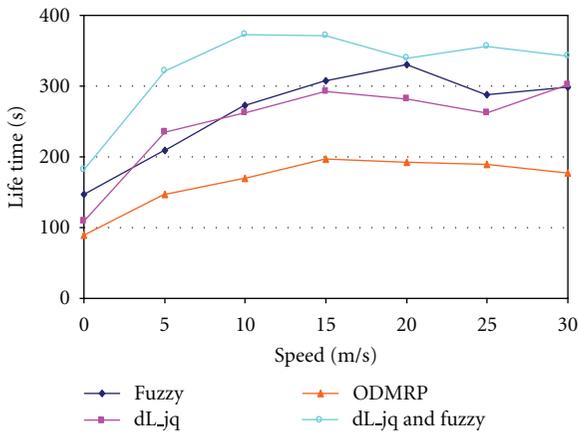


FIGURE 11: Life time with increasing speed of nodes.

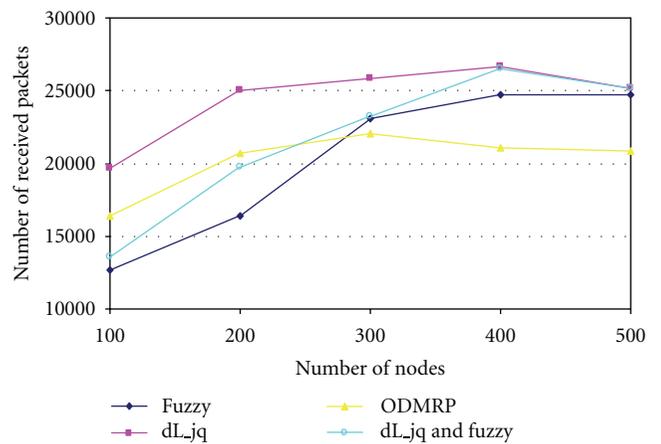


FIGURE 13: Number of received packets with increasing number of nodes.

method is to use a stronger set of forwarding nodes to configure new forwarding group.

While standard ODMRP is tolerant to increasing mobility speed, it suffers from high overhead especially at highly mobile environments. The proposed methods, especially *dL_jq* combined with fuzzy method, are completely tolerant to increasing mobility. This is attributed to mechanism of fuzzy method at which *Join Query* packets are discarded from nodes likely to show poor performance in terms of packet delivery. A high overhead is translated into high power consumption. Since *dL_jq* method has the smallest overhead among studied protocols, it could be translated into the smallest power consumption and the longest network life time. It would appear from Figure 11, that as the mobility speed increases, three proposed methods outperform ODMRP in terms of network longevity.

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* and fuzzy method shows the longest network life time because it discards a high number of *Join Query* and data packets.

4.3. Impact of Number of Nodes. Figure 12 presents end-to-end delay under increasing number of nodes. When the number of nodes in the network increases, overall overhead of ODMRP increases since the flooding domain is larger compared to when less number of nodes exist in the network. Obviously, in the network with a large number of nodes, forwarding groups include more forwarding members. Also upon increasing the network density, loss probability is more likely to increase. When the number of nodes is less than 300, original ODMRP shows good performance in terms of delivery ratio because it performs complete flooding in entire network (see Figure 13). Original ODMRP has good performance while network has sufficient capacity to transfer overall overhead.

Since then (number of nodes more than 300), number of nodes which are included in *Join Query* packet forwarding and forwarding group is increased; this problem is critical when number of nodes is very high. Fuzzy methods try to use reliable nodes and routes which lead to higher delivery probability, lower overhead, and delay, especially at dense networks.

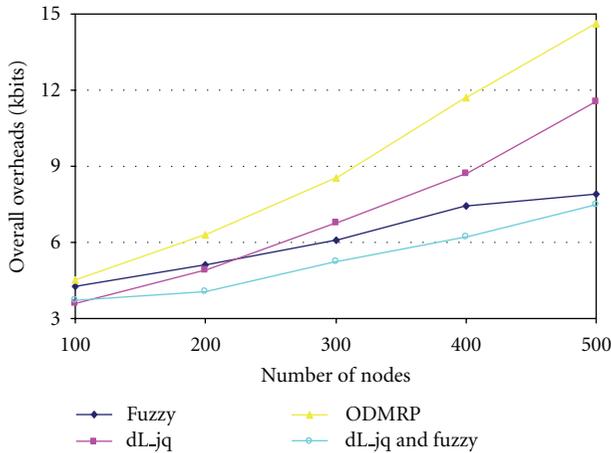


FIGURE 14: Overall overhead per a data packet.

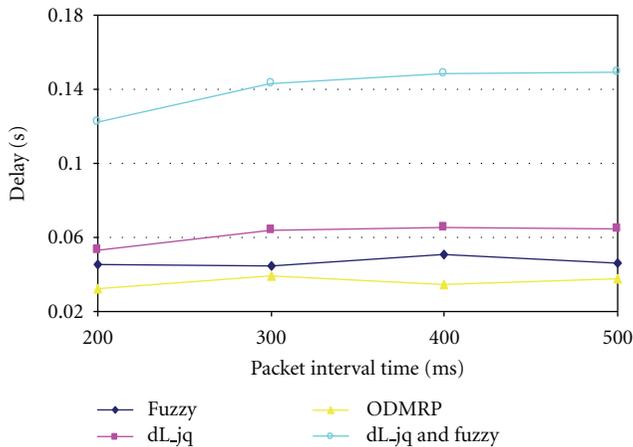


FIGURE 15: End to end delay with increasing packet interval time.

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

The fuzzy method has many choices to select a better route or node, which results in short delays. A combination of *fuzzy* approach with *dL_{jq}* shows the best performance due to the constraint flooding domain.

A higher overhead leads to higher power consumption and lower network life time. Again, *dL_{jq}* and *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.

4.4. Impact of Traffic Density. Figures 15–17 present protocols performance with varying traffic pattern. It is implicit that with increased packet interval time, it sends less packets. Simulations were performed with 8 nodes acting as senders of the session with randomly selected receiver nodes and data payload is considered 1024 bytes. If the interval time (i.e., time between two packets) increases, the number of data packets broadcasted in forwarding group increases.

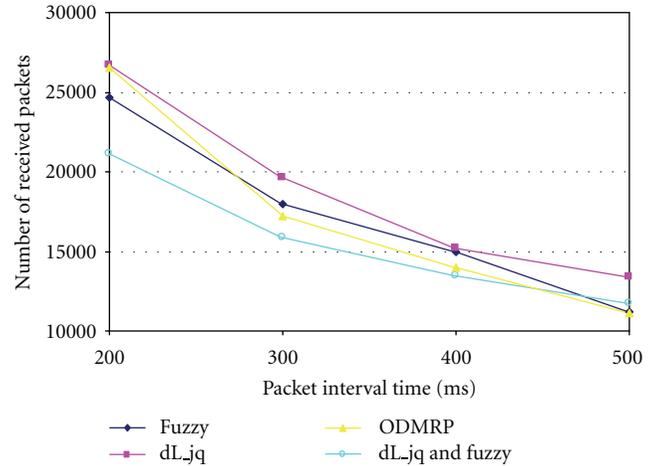


FIGURE 16: Number of received packet with increasing packet interval time.

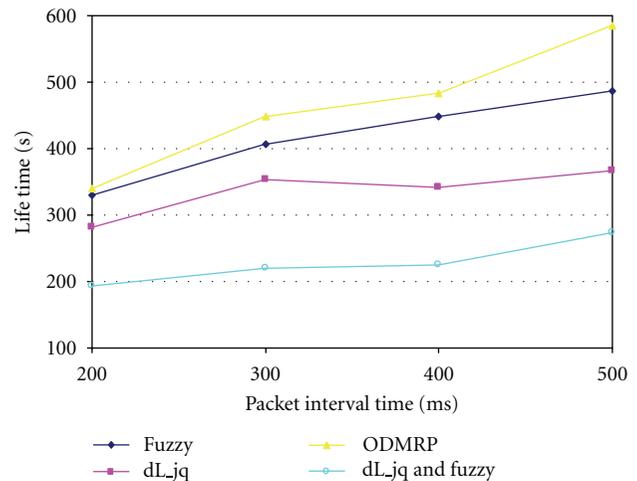


FIGURE 17: Life time with increasing packet interval time.

Figure 15 shows end-to-end delay for all simulated methods. The original ODMRP floods data packets in a great forwarding group. The proposed methods try to form a small and efficient forwarding group. A small forwarding group has very low overhead in comparison to the original ODMRP. Therefore the proposed methods have very short end-to-end delay in comparison to original ODMRP.

Figure 16 presents the number of received data packets by various methods. The fuzzy methods use high-quality nodes and routes to forward data packet. Therefore they deliver high number of data packets. Also, these methods form a small and efficient forwarding group instead of big and poor performance forwarding group.

High traffic loads (low interval time between packets) in such weak forwarding group result in increasing overall overhead and power consumption. Also, it ends up with wasting the life time of the network. The *dL_{jq}* and *fuzzy* method has satisfactory results in these figures because it tries to use reliable and strong node.

```

If (speed of previous node <  $\alpha$ ) then Speed of node is low
If (speed of previous node >  $\alpha$ ) 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
    
```

ALGORITHM 4

In this series of experiments, forming a strong forwarding group has impact on performance in comparison to elimination of the *lost Join Query* packets. While creation of weak and big forwarding groups is possible in *dL-jq* method, such mechanism is denied in fuzzy method. Thus, fuzzy method has high performance in comparison to *dL-jq* method.

Overall overhead of original includes data overhead and control overhead. Data overhead is achieved by data flooding in forwarding group. Therefore, increasing input traffic dramatically results in increase to data overhead. As a consequence of high overhead, MANET nodes use a great deal of their power; so the life time of network is reduced. Briefly original ODMRP is very sensitive to increasing input traffic while in the proposed method this sensitivity reduces due to reduction of *Join Query* packets and formation of strong forwarding groups. (Because they delete the lost *Join Query* packets or form strong and small forwarding group.)

4.5. 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 rule in Algorithm 4 shows the implementation of the speed membership function.

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. Each *Join Query* packet carries extra data regarding speed, power, and bandwidth of nodes. The α and β parameters have high impact on the multicast routing process. 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 constructing 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, 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 2 m/s to 10 m/s. High value of α increases lower border and most nodes in the network are included as low-speed node. Also, a high value of α results in high end-to-end delays because most of the nodes are included in the forwarding group.

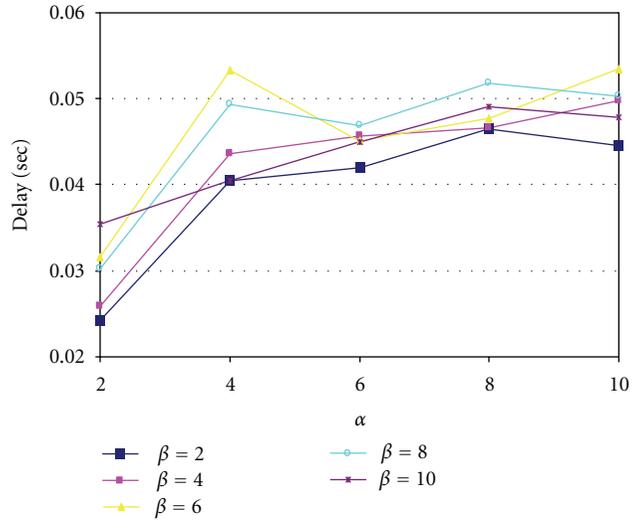


FIGURE 18: End to end delay.

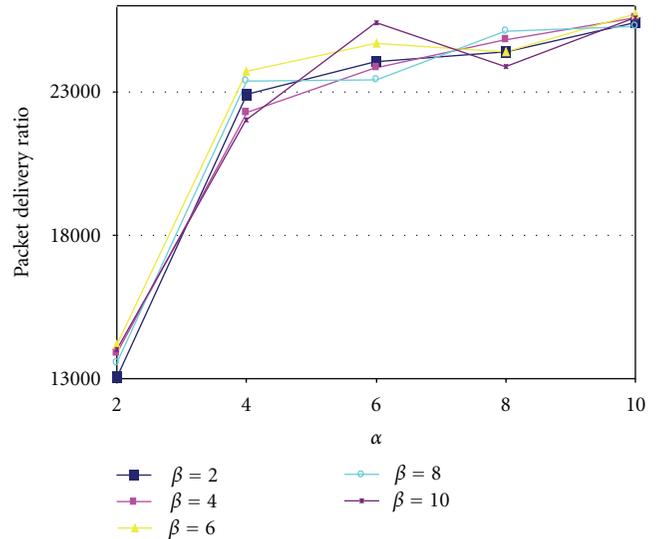


FIGURE 19: Number of received packets.

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.

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

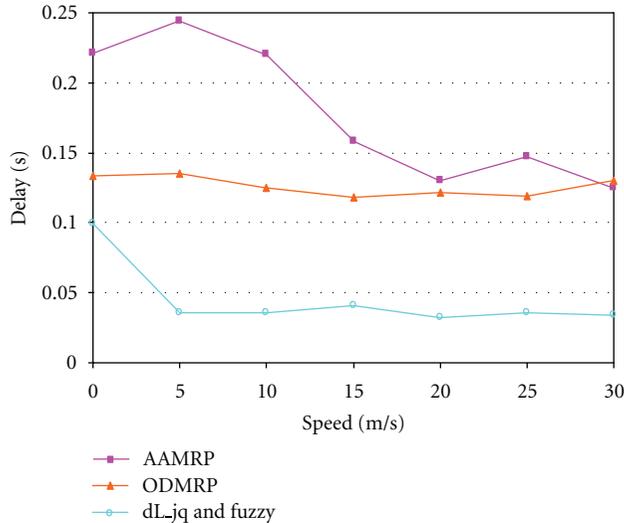


FIGURE 20: End-to-end delay.

forwarding group. If the value of β is high it is possible that a high number of *Join Query* packets discarded.

4.6. Performance Comparison with AAMRP. So far, performance of the proposed algorithm was compared with original ODMRP. In addition to ODMRP, in this point, the performance of the proposed work compares favorably with AAMRP. The end-to-end delay of three investigated protocols is shown in Figure 20. The proposed scheme (i.e., *dL_jq* and *g*) outperforms the other two protocols in terms of end-to-end delay at the starting point. Although the end-to-end delay introduced by AAMRP decreases dramatically as the speed increases, it still maintains the longest delays in contrast to other two protocols. Not surprisingly, *dL_jq* and *fuzzy* experiences the shortest delays in comparison with the other two protocols. Generally, end-to-end delays of ODMRP and *dL_jq* and *fuzzy* protocols show the same pattern as the speed increases to 5 m/s and onwards. This could be attributed to the same mesh structure maintained in both protocols.

Figure 21 shows the throughput of AAMRP, ODMRP, and *dL_jq* and *fuzzy* as a function of mobility speed. It is evident that AAMRP and ODMRP follow the same trend roughly as the mobility speed increases (crossing at 20 m/s) while maintaining lower position than that of proposed *dL_jq* and *fuzzy* method. In majority of cases, *dL_jq* and *fuzzy* retains the first position in throughput higher than AAMRP and ODMRP, respectively.

It can be observed that the results for the AAMRP and ODMRP protocols are approximately the same at the starting point of the simulation where there is no mobility.

As mentioned earlier, broadcasting method is used in AAMRP, as common feature of broadcasting redundancy is maintained at the tree structure of AAMRP. Such redundancy results in extra robustness in conditions of mobility and eventually high packet delivery rate. However, as the mobility speed increases, AAMRP diverges and shows 35% less number of delivered packets at 5 m/s. Since then the

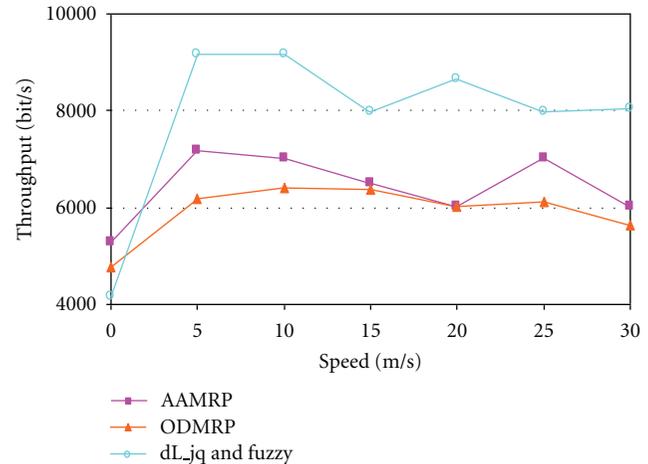


FIGURE 21: Throughput.

number of delivered packets by AAMRP decreases dramatically and experiences a crossing point with ODMRP at 25 m/s. In moderate to highly mobile environments (i.e., beyond 10 m/s), the proposed method delivers the greatest percentage of originated data packets compared to other two investigated protocols. On the other hand, in majority of cases, ODMRP delivers the lowest number of originated packets to receivers of the multicast groups.

Interestingly, *dL_jq* and *fuzzy* method introduces the least overhead to the network while maintaining the shortest end-to-end delay at the same mobility speed (i.e., 5 m/s). Although, in AAMRP, mobile nodes are easily adopted to network dynamics due to adaptive mechanism employed in this protocol, such nodes cannot recognize the routes with higher quality in terms of bandwidth, power, mobility and loss. It is evident in Figure 23 that the proposed scheme results in less overhead compared to other two investigated protocols. This is due to using of the fuzzy strategy utilized in this protocol and used in selection of quality routes based on parameters such as end-to-end delay and the number of received packets.

Moreover, in this protocol, overall overhead decreases slightly as the mobility speed increases. ODMRP maintains the largest amount of overhead compared to other two investigated protocols through all simulation scenario which can be attributed to sending the control packets in variable refresh intervals. As mentioned earlier, the forwarding nodes and the routes are refreshed frequently. By way of contrast, AAMRP achieves packet delivery statistics that are comparable to that with ODMRP (Figure 21) with significantly lower overheads (Figure 22). This is attributed to adaptive broadcast maintained in AAMRP wherein redundant and unwanted transmissions of data can be avoided. This is due to the cluster leader mechanism wherein it performs an adaptive localized broadcast within its cluster in order to disseminate multicast packets received from the source. Periodic invocation of control messages with high frequency.

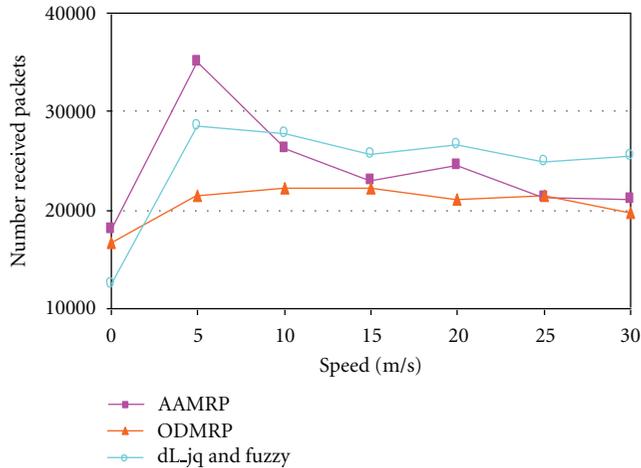


FIGURE 22: Number of received packets.

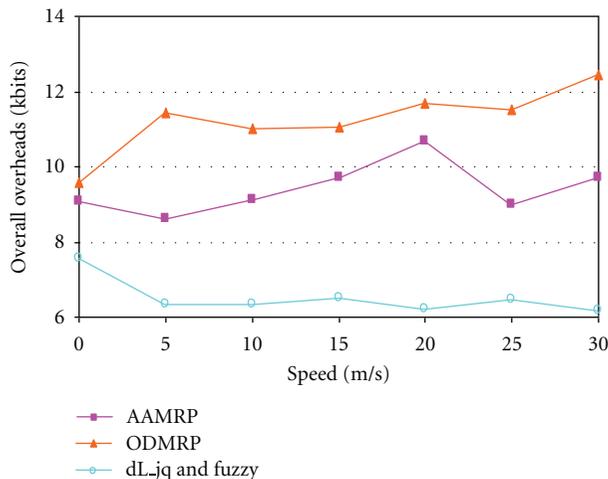


FIGURE 23: Overall overhead.

5. Conclusion

In this paper, two different novel methods are proposed to improve the performance of multicast routing algorithms in MANETs especially the ODMRP. The first method that uses fuzzy logic tries to construct a small and efficient forwarding group instead of great and inefficient one. The second method reduces the domain of control packet flooding, and it achieves improvement via reduction of control overhead. Simulation results show that the proposed method has many advantages compared to original ODMRP and AAMRP such as less control and data overhead, end-to-end delay, high packet delivery ratio, and very low power consumption. Thus, it could be used in many applications in ad hoc networks.

In the future work, we will consider tuning fuzzy rules using heuristics approaches. Besides, we try to evaluate our protocol in real world situations.

Acknowledgment

The authors would like to thank *Islamic Azad University (Damavand Branch)* of Iran for the research grant and for providing the research facilities in a collaborative environment.

References

- [1] M. Iiyas, *The Hand Book Of Ad Hoc Networks*, CRC Press, 2003.
- [2] C. Siva, R. Murthy, and 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/manet-charter.html>.
- [4] J. Schiller, *Mobile Communications*, Addison-Wesley, 2000.
- [5] K. Obraczka and G. Tsudik, "Multicast routing issues in ad Hoc networks," in *Proceedings of the 6th IEEE International Conference on Universal Personal Communications (ICUPC '98)*, Florence, Italy, October 1998.
- [6] E. Baburaj and 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.
- [7] C. M. Cordeiro, H. Gossain, and D. P. Agrawal, "Multicast over wireless mobile ad Hoc networks: present and future directions," *IEEE Network*, vol. 17, no. 1, pp. 52–59, 2003.
- [8] H. Gossain, C. M. Cordeiro, and D. P. Agrawa, "Multicast: wired to wireless," *IEEE Communications Magazine*, vol. 40, no. 6, pp. 116–123, 2002.
- [9] C. W. Wu, Y. C. Tay, and C. K. Toh, "Ad Hoc multicast routing protocol utilizing increasing id-numbers (AMRIS) functional specification," Internet-draft, work in progress, draft-ietf-manet-amris-spec-00.txt, November 1998.
- [10] K. Viswanath, K. Obraczka, and G. Tsudik, "Exploring mesh and tree-based multicast. Routing protocols for MANETs," *IEEE Transactions on Mobile Computing*, vol. 5, no. 1, pp. 28–42, 2006.
- [11] R. Bagrodia, M. Gerla, J. Hsu, W. Su, and S.-J. Lee, "A performance comparison study of ad Hoc wireless multicast protocols," in *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies*, pp. 565–574, March 2000.
- [12] D. Pompili and M. Vittiucci, "A probabilistic predictive multicast algorithm in ad Hoc networks (PPMA)," in *Proceedings of the Ad Hoc Networking Workshop (MEDHoc-NET '04)*, Mugla, Turkey, June 2004.
- [13] B. Rong and G. Amoussou, "Mobility prediction aided dynamic multicast routing in MANET," in *Proceedings of the IEEE/Sarnoff Symposium on Advances in Wired and Wireless Communication*, Princeton, NJ, USA, April 2005.
- [14] M. Saghir, T. Wan, and R. Budiarto, "Qos multicast routing based on bandwidth estimation in mobile ad Hoc networks," in *Proceedings of the International Conference on Computer and Communication Engineering (ICCC'06)*, vol. 1, Kuala Lumpur, Malaysia, May 2006.
- [15] E. Royer and C. Perkins, "Multicast Ad Hoc on-Demand Distance Vector (MAODV) Routing," Internet Draft: draft-ietf-manet-maodv-00.txt, 2000.
- [16] E. M. Royer and C. E. Perkins, "Multicast operation of the ad-Hoc on demand distance vector routing protocol," in *Proceedings of the ACM/IEEE International Conference on Mobile*

- Computing and Networking (MOBICOM '99)*, pp. 207–218, Seattle, Wash, USA, August 1999.
- [17] C. Perkins, E. Belding-Royer, and S. Das, “Ad Hoc On-Demand Distance Vector (AODV) routing,” RFC 3561, 2003.
- [18] A. Laouiti, P. Jacquet, P. Minet, L. Viennot, T. Clausen, and C. Adjih, “Multicast optimized link state routing,” INRIA Research Report RR-4721, 2003.
- [19] T. Clausen and P. Jacquet, Eds., *Optimized link state routing protocol (OLSR)*. RFC 3626, Network Working Group, October 2003.
- [20] J.-G. Jetcheva and D. B. Johnson, “Adaptive demand-driven multicast routing in multi-hop wireless ad Hoc networks,” in *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '01)*, pp. 33–44, Long Beach, Calif, USA, October 2001.
- [21] E. Bommaiah, M. Liu, A. MacAuley, and R. Talpade, “AMRoute: ad Hoc multicast routing protocol,” Internet-draft, work in progress, drafttalpade-manet-amroute-00.txt, August 1998.
- [22] S. J. Lee, W. Su, and M. Gerla, “On-demand multicast routing protocol in multihop wireless mobile networks,” *Mobile Networks and Applications*, vol. 7, no. 6, pp. 441–452, 2002.
- [23] M. Gerla, G. Pei, S. Lee, and C. C. Chiang, “On-Demand Multicast Routing Protocol (ODMRP) for Ad-Hoc networks,” IETF MANET Working Group Internet-Draft, November 1998.
- [24] S. Lee, C. C. Chiang, and M. Gerla, “On-Demand Multicast Routing Protocol,” in *Proceedings of the IEEE Conference on Wireless Communication and Networking*, vol. 3, pp. 1298–1302, September 1999.
- [25] J. J. Garcia-Luna-Aceves and E. L. Madruga, “Core-assisted mesh protocol,” *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp. 1380–1394, 1999.
- [26] R. Chandra, V. Ramasubramanian, and K. P. Birman, “Anonymous gossip: Improving multicast reliability in mobile ad-Hoc networks,” in *Proceedings of the 21st IEEE International Conference on Distributed Computing Systems*, pp. 275–283, IEEE CS Press, April 2001.
- [27] J. Luo, P. T. Eugster, and J.-P. Hubaux, “Route driven gossip: probabilistic reliable multicast in ad Hoc networks,” in *Proceedings of the 22nd Annual Joint Conference on the IEEE Computer and Communications Societies (INFOCOM '03)*, pp. 2229–2239, IEEE Press, April 2003.
- [28] J. H. Ryu, S. Song, and D. H. Cho, “Energy-conserving clustering scheme for multicasting in two-tier mobile ad-Hoc networks,” *Electronics Letters*, vol. 37, no. 20, pp. 1253–1255, 2001.
- [29] C. Tang, C. S. Raghavendra, and V. Prasanna, “Energy efficient adaptation of multicast protocols in power controlled wireless ad Hoc networks,” in *Proceedings of the International Symposium on Parallel Architectures, Algorithms and Networks*, pp. 80–85, May 2002.
- [30] T. Ozaki, J. B. Kim, and T. Suda, “Bandwidth-efficient multicast routing for multihop, ad-Hoc wireless networks,” in *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '01)*, pp. 1182–1191, Anchorage, Alaska, USA, April 2001.
- [31] C.-K. Toh, G. Guichala, and S. Bunchua, “ABAM: on-demand associativity-based multicast routing for ad Hoc mobile networks,” in *Proceedings of the 51st IEEE Vehicular Technology Conference (VTC '00)*, pp. 987–993, Boston, Mass, USA, September 2000.
- [32] S. Lee and C. Kim, “A new wireless ad Hoc multicast routing protocol,” *Computer Networks*, vol. 38, no. 2, pp. 121–135, 2002.
- [33] L. Ji and M. S. Corson, “Differential destination multicast: A MANET multicast routing protocol for small groups,” in *Proceedings of the 22th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '01)*, pp. 1192–1202, 2001.
- [34] L. Ji and M. S. Corson, “Explicit multicasting for mobile ad Hoc networks,” *Mobile Networks and Applications*, vol. 8, no. 5, pp. 535–549, 2003.
- [35] Sajama and Z. J. Haas, “Independent-ree ad Hoc multicast routing (ITAMAR),” *Mobile Networks and Applications*, vol. 8, no. 5, pp. 551–566, 2003.
- [36] Chen Y. S, Y. W. Ko, and T. L. Lin, “A lantern-tree-based QoS multicast protocol for wireless ad-Hoc networks,” in *Proceedings of the 11th IEEE International Conference on Computer Communications and Networks (ICCCN '02)*, Miami, Fla, USA, October 2002.
- [37] Y. S. Chen and Y. W. Ko, “A lantern-tree-based QoS multicast protocol for wireless ad-Hoc networks,” *IEICE Transactions on Communications*, vol. E87-B, no. 3, pp. 717–726, 2004.
- [38] V. lashkari and M. Dehghan, “QoS-aware multicast ad Hoc on-demand distance vector routing,” in *Proceedings of the International Conference on Wireless Networks (ICWN '07)*, pp. 1506–1511, September 2007.
- [39] I. Hwang and W. Pang, “Energy efficient clustering technique for multicast routing protocol in wireless ad Hoc networks,” *International Journal of Computer Science and Information Security*, vol. 7, no. 8, 2007.
- [40] D. Pathirana, *Resilient on demand multicast routing protocol*, M.S. thesis, Rochester Institute of Technology, 2007.
- [41] H. Moustafa and H. Labiod, “A performance comparison of Multicast routing protocols in ad-Hoc networks,” in *Proceedings of the 14th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, pp. 497–501, 2003.
- [42] A. Sabari and K. Duraiswamy, “Ant-based Adaptive Multicast Routing protocol (AAMRP) for ad Hoc networks,” *International Journal of Computer Science and Network Security*, vol. 6, no. 2, 2009.
- [43] A. S. Shafigh, K. Abdollahi, and A. J. Kassler, “Improving performance of ODMRP by fuzzy logic control,” in *Proceedings of the IEEE International Conference on Wireless Communications, Networking and Information Security (WCNIS '10)*, Beijing, China, 2010.
- [44] K. Abdollahi, A. S. Shafigh, and A. J. Kassler, “Improving performance of on demand multicast routing by deleting lost join query packets,” in *Proceedings of the 6th Advanced International Conference on Telecommunications (AICT '10)*, pp. 316–322, Barcelona, Spain, May 2010.
- [45] X. Zeng, R. Bagrodia, and M. Gerla, “GloMoSim: a library for parallel simulation of large-scale wireless networks,” in *Proceedings of the 12th Workshop on Parallel and Distributed Simulation (PADS '98)*, pp. 154–161, May 1998.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

