KSII TRANSACTIONS ON INTERNET AND INFORMATION SYSTEMS VOL. 5, NO. 4, April 2011

A Cluster-based QoS Multicast Routing Protocol for Scalable MANETs

Mohammad M. Qabajeh[1], Aisha H. Adballa[1], Othman O. Khalifa[1] and Liana K. Qabajeh[2]
[1]Department of Electrical and Computer Engineering, International Islamic University Malaysia, Malaysia
[e-mail: m_qabajeh@yahoo.com, {aisha, khalifa }@iiu.edu.my]
[2] Faculty of Computer Science and Information Technology, University of Malaya, Malaysia
[e-mail: liana_tamimi@ppu.edu]
*Corresponding author: Mohammad M. Qabajeh

Received January 3, 2011; revised February 25, 2011; accepted April 18, 2011;
published April 29, 2011

Abstract

Recently, multicast routing protocols become increasingly important aspect in Mobile Ad hoc Networks (MANETs), as they effectively manage group communications. Meanwhile, multimedia and real-time applications are becoming essential need for users of MANET. Thus it is necessary to design efficient and effective Quality of Service (QoS) multicast routing strategies. In this paper, we address the scalability problem of multicast routing protocols to support QoS over MANETs. In particular, we introduce a Position-Based QoS Multicast Routing Protocol (PBQMRP). Basically, the protocol based on dividing the network area into virtual hexagonal cells. Then, the location information is exploited to perform efficient and scalable route discovery. In comparison with other existing QoS multicast routing protocols, PBQMRP incurs less control packets by eliminating network flooding behavior. Through simulation, the efficiency and scalability of PBQMRP are evaluated and compared with the well-known On-Demand Multicast Routing Protocol (ODMRP). Simulation results justify that our protocol has better performance, less control overhead and higher scalability.

Keywords: MANETs, multicast, routing, position-based, QoS

A preliminary version of this paper appeared in Journal of Engineering Letters, vol. 18, no. 3, pp. 212-225 August 2010. This version includes detailed description of the proposed protocol, a concrete analysis as well as simulation of new parameters. This research was supported by the Research Management Center at the International Islamic University Malaysia.

DOI: 10.3837/tiis.2011.04.007

1. Introduction

MANET is composed of collection of mobile nodes that communicate with each other over wireless links. The most important features of these networks include being self-organizing, self-configuring, self-administering, as well as dynamic topology and multi-hop routing [1]. In MANETs, each node acts as a host or a router at any time. Nodes intercommunicate through single-hop and multi-hop paths to forward packets to each other, which require cooperation between nodes to relay packets to their targets. MANETs face several challenges that have to be addressed including dynamic topology changes, lack of infrastructure, bandwidth constraint and limited resources. These features make working in MANET more complex than their wired counterpart [2].

MANETs are considered for many applications that involve point-to-multipoint or multipoint-to-multipoint communication patterns, where robustness and reliability are important aspects. Disaster recovery, search and rescue efforts, military battlefields, temporary offices and multi-party gaming are common examples of these applications [3]. As a consequence, multicast communication is emerged to support and facilitate effective and collaborative communication among groups of users with the same interest. Multicasting is defined as a scheme for delivering the same data from a source to a group of destinations. This is efficient in saving the bandwidth and improving the scalability, which are essential issues in MANETs [4][5]. Also, multicasting reduces transmission overhead both on the source as well as on the network nodes and speeds up information delivery at the receiving nodes.

Similar to unicast routing protocols, multicast routing protocols could be classified as either *topological-based* routing or *geographical-based* routing. Topological-based protocols are further classified into two main groups according to path distribution among multicast session members; namely, *mesh-based* and *tree-based*. While tree-based protocols construct multicast tree over the multicast members and provide efficient data forwarding, mesh-based protocols provide redundant paths to recover broken routes and they have better robustness against topology changes.

Geographic-based routing is recently attracting more and more researchers [6]. This is due to the availability of small-size, inexpensive and low-power Global Positioning System (GPS) receivers. Using GPS receivers, each node knows its own physical location; i.e., its precise geographic coordinates. The motivation behind using this kind of routing is that it eliminates the need to maintain routing information, which improves robustness and scalability in MANETs routing [7].

Nowadays, QoS multicast routing over MANETs has received more attention. However, this is a challenging task because supporting QoS for multicast protocols has to be designed in a different way to that for unicast protocols. The difference is that, in unicast QoS protocols the resource reservation is done between a source and a destination. While multicast QoS routing protocols should provide suitable QoS paths to all destinations of the multicast group. Also, the heterogeneous nature of paths to the destinations adds extra challenges to the design of QoS multicasting protocols. Moreover, the centralized design of the Medium Access Layer (MAC) and limited resources make it more difficult to guarantee QoS in this type of networks.

For the aforementioned reasons, it is not a trivial task to design a scalable QoS multicast routing protocol for MANETs. In this paper, we exploit the location information of the mobile nodes to create stable network structure. Then, this structure is utilized to perform efficient QoS routing. Basically, the whole network is partitioned into several hexagonal cells and a cell leader node and a backup node are elected in each cell. To further reduce the number of transmissions, we divide the multicast members into several sub-groups based on their geographic position.

The novelty of the work lies in the combination of QoS multicast and position-based routing. This will lead to scalable performance, effective routing, and reasonable control overhead. This is achieved by applying the following strategies: (a) Adaptive cluster management with response to topology updates; (b) Efficient and non-duplicate forwarding of packets between clusters; (c) Hierarchical construction algorithm for multicast members to improve scalability; and (d) Position-based routing to discover QoS routes between the source and multicast group members.

The effectiveness of the proposed approach has been examined through extensive simulations. A wide range of scenarios have been studied by varying the nodes density, multicast group size and bandwidth requirements.

The rest of the paper is organized as follows: In the consequent section, we discuss some related works. Section 3, presents detailed description of our proposed protocol. Section 4 shows our simulation and provides discussion of the observed results. Finally, conclusions and future research directions are drawn in section 5.

2. Related Work

Over the past few years, several QoS multicast routing protocols have been proposed for Ad hoc networks such as [3][8]. Recent surveys that study the up-to-date QoS multicast routing protocols are presented in [9][10]. In recent years, position-based routing over MANETs gains increasing attention from the research community. This focus from researchers is to achieve stability, reliability and reduced communication cost as well as improve data delivery ratio. To the best of our knowledge, few works have been proposed in QoS position-based multicasting such as [4][11]. However, our protocol outperforms the existing schemes in terms of scalability (network size and multicast members) as well as control packets reduction. In this section we will discuss some of the schemes that are directly related to our work.

In [11], a cluster-based QoS multicast routing protocol has been proposed. This protocol partitions the network into square clusters and the nearest node to cluster center is elected as a cluster-head. A gateway node is selected between the adjacent clusters to rely the packets when the adjacent cluster-head nodes are out of the effective transmission range. The source node starts the multicast session by sending *PROPE* packet to the cluster-head. The gateway forwards this packet to the proper neighbor cluster until the destination or intermediate node with valid route to the destination is reached. The destination or the intermediate node selects the optimal route using best predecessor replacement strategy [12], where each node chooses the next best predecessor that satisfies the QoS constrains (delay, cost). When the source receives the ACK reply packet, it starts data transmission. This protocol only uses cluster-head, source, gateway and destination nodes in routing. However, only the gateway is responsible for packet forwarding. Thus, the gateway selection becomes the key point of this protocol. Also, the paper did not discuss the network structure and maintenance, which perhaps produce significant traffic. Another drawback appears when the network is sparse, in this case the gateway nodes may fail to reach the neighbor cluster-head and then the route cannot be established. Additionally, using *PROPE* packets in multicast routing requires maintaining the local state at each node, which increases the contention at the wireless medium.

In [4], a Hypercube-based Virtual Dynamic Back-bone (HVDB) model for QoS-aware multicast communication is proposed. The clusters are formed using mobility prediction and location-based technique used in [13]. The structure is abstracted into three tiers: mobile node (MN), hypercube tier (HT) and mesh tier (MT). The network area is partitioned into overlapped circular shape and a cluster-head (CH) is elected for each circle. The CH is mapped to a hypercube node at the HT tier. Each hypercube is mapped as one mesh node at the mesh tier. The nodes periodically send the local memberships to its CH. Each CH periodically sends the group memberships to all CHs within the hypercube and one of the CHs periodically broadcasts the membership to all clusters in the network.

When a node has data to send data to group members, it sends the data to its CH. Then, the CH checks the summarized membership to determine the hypercubes that maintains the members of this group. The logical locations of these hypercubes are used to compute a multicast tree and the information about the multicast tree is encapsulated into the messages. A location-based unicast protocol is used to send the packets between hypercubes. When the packet enters a hypercube, it is forwarded to those hypercube nodes that contain the group members. HVDB protocol provides fault tolerance property and scalability. However, it suffers from high communication overhead due to the periodic messages in the three tiers. Also, the overlapping circles bring extra overhead for the cluster-heads. Another drawback is the mapping between the tiers and selection of border and inner cluster-heads which increases the overhead.

Recent research has shown that geographical routing significantly improves the performance of Ad hoc routing protocols [14]. Based on this view, a new routing protocol has been designed to exploit the location information to eliminate flooding and simplify the routing strategy. The proposed protocol tries to overcome some of the problems of the existing schemes along with enhancing the scalability and reducing the control overhead. The details of this protocol are presented in the following section.

3. Proposed Work

This section discusses the proposed Position-Based QoS Multicast Routing Protocol (PBQMRP) in detail. We first give an overview of the proposed protocol. Then, the network clustering strategy and network maintenance are described. Next, we explain the location service and multicast tree formation. Finally, our position-based route discovery mechanism, route reply, data delivery and the route maintenance mechanism are introduced.

3.1 Protocol overview

PBQMRP is a source-tree multicast routing protocol proposed to provide scalable QoS multicast routing over MANETs. PBQMRP aims to be implemented in large networks with large number of multicast members. To achieve this, virtual clustering strategy has been introduced. This strategy based on partitioning the network into hexagonal cells. Each cell has a Cell Leader (*CL*) node elected to maintain information about all the nodes in its cell till they join a new cell. Also each cell has Cell Leader Backup (*CLB*) node to replace the *CL* node when it fails or leaves the cell.

When a source node wants to send data packets to a particular multicast group, the cluster structure is efficiently used to gather information about the subscribing nodes and provide the source node with this information. After that, the source partitions the group members into manageable sub-groups. In each of these sub-groups, one of the group members is selected to be a coordinator. Later, the source and the coordinators co-operate to search for QoS routes to all destinations. PBQMRP reduces the number of nodes participating in forwarding route discovery packets through using Restricted Directional Flooding (RDF) based on nodes' position information. Using this mechanism eliminates broadcast storm and efficiently utilizes the network resources.

The protocol operation is divided into multiple phases. These phases include network construction, network maintenance, location service, multicast group partitioning, routing discovery and maintenance as well as data transmission. For convenient presentation, **Table 1** presents a summary of different phases of PBQMRP.

Table 1. Definition for the working phases of PBQMRP.

Phase name		Description	Start	End
Network construction (section 3.2)		The network terrain is partitioned into several cells and a leader node is elected to serve each cell.	Upon starting the network lifetime.	When the network structure is known and each node is aware of the operations that are assigned to it.
Network maintenance (section 3.3)		Provides efficient solutions to nodes mobility and failure to obtain stable structure.	After finishing the network construction phase.	By the end of the network lifetime.
Location service (section 3.4)		Providing the source node with the identities and positions of the multicast members.	When a source node wants to transmit data packets to a multicast group.	When the identities and positions of all the multicast members are known to the source node.
Formation of local groups (section 3.5)		Dividing the multicast members into manageable-size sub-groups and choosing coordinator node for each sub-group.	When the location service phase is finished.	When the multicast members are arranged into sub-groups.
Route discovery and maintenance (sections 3.6 and 3.7)	Route request	Searching for route paths that satisfy certain bandwidth and delay using RDF.	After finishing multicast group division. Initiated by sending *QoS_RREQ* packet from the source.	When the request time is elapsed or the number of route retries is finished.
	Route reply	Choosing the route that satisfies the required constraints and returning it back to source node.	Upon receiving the first *QoS_RREQ* packet by the destination.	When the suitable routes are chosen and received by the source node.
	Route recovery	Handling broken links during the life time of the network.	When a broken link is encountered in the data transmission procedure at.	When the broken link is reconstructed.
Data transmission (section 3.8)		Generating of data traffic from the source node to all the intended destinations.	When the route reply phase is finished.	When the source node finishes sending the data.

3.2 Network construction

This section presents the network construction phase in detail. The main objective of building this virtual backbone is to maintain stable network structure on which our protocol can perform efficient routing and can cope with an increasing number of nodes; having a more scalable routing protocol.

3.2.1 Virtual architecture formation

Since MANETs must operate in a physical geometric space, they naturally need to exploit location information. In our approach, we assume that the mobile nodes in the network can find their own locations using a GPS receiver or any GPS-free techniques. We also assume that the routing area is a two-dimensional plane and each node previously knows the borders of the routing area.

In our structure, the mobile nodes are distributed randomly in the network area and the network is defined by the coordinates (0, 0) to (X_{max}, Y_{max}). The clustering strategy starts by dividing the network area into disjoint, adjacent, fixed-size and regular hexagonal cells, which results in a fixed and scalable structure. There are two main reasons behind considering hexagonal gridding in our scheme. First, considering the hexagonal, square and triangle cell shapes, and assuming the transmission range (*R*) as the maximum distance among the cell's nodes, the area covered by the hexagonal ($0.6495*R^2$) is larger than that covered by the triangle ($0.433*R^2$) and the square ($0.5*R^2$). Larger cell area means less number of *CL* nodes; i.e., less overhead due to *CL* movement, failure and electing new *CLs*. Moreover, larger area means that the hexagonal cell shape covers more area in a single transmission, which increases the number of nodes that are affiliated with the leader of each cell. This would reduce the communication overhead and improve the propagation delay in performing location discovery. Second, the hexagonal cell offers six directions of transmission with the same distance between the centers of the neighbor cells. On the other hand, square and triangle shapes have larger number of neighbors (8 for square shape and 12 for triangle shape), but the distance between the centers of the neighbor cells are different. This may result in differences in packet propagation to different neighbor cells.

Let us denote *L* as the side length of hexagonal cell. The value of *L* is chosen as $L = R/2$ to guarantee that each two nodes located anywhere in the same cell can communicate with each other directly. The reason behind this assumption is to compromise between the overhead of network maintenance (overhead between cells) and the location service overhead (overhead inside each cell), especially when the network is large and the number of nodes is high. Considering other cell sizes will be one of the issues that should be studied in our future work. We believe that using large number of small cells would reduce overhead inside the cells (nodes will be reachable within 1-hop of each other), however small cells would make the broadcast go through the neighbor cells and will increase the delay to reach different *CLs* during location service.

On the other hand, using small number of large cells results in increasing the overhead of communication inside each cell. This is since nodes will not be reachable within 1-hop; i.e., each node is requested to rebroadcast the packets it receives. However, reducing the number of cells means less number of election processes and hence the number of election control packets will be reduced. Moreover, smaller number of cells may results in reducing the communication overhead and propagation delay during location service. Also, in case of high mobility, the location update packets will increase and probability of the boundary crossing will increase as well. Using large cell size effectively reduces the overhead of triggering this kind of packets.

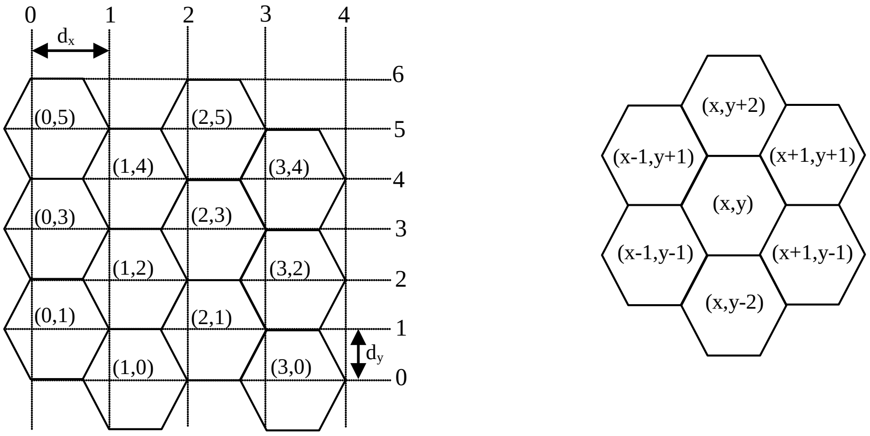
Fig. 1-(a) shows the network entire topology and illustrates the identity of each cell. Each

hexagonal cell inside the network has a unique identity (or *Cell_ID*). Since the cells are not arranged in a uniform pattern, let us introduce the doted vertical and horizontal lines to the plan to help in explaining how each cell obtains its identifier. The distance between two successive points in the X-axis is set to d_x, and that for the Y-axis is set to d_y. The values of d_x and d_y are defined as:

$$d_x = \frac{3R}{4} \ , \ d_y = \frac{\sqrt{3}R}{4}$$

Therefore, each cell takes its *Cell_ID* from the two virtual perpendicular lines crossing each other inside that cell. From this, the identity of the cell is known and the identifiers of the 6-neighboring cells will be known accordingly. This enables the *CL* node to communicate with its 6-neighboring cells. The relation between the cell and its neighboring cells are shown in **Fig. 1-(b)**.

In PBQMRP, for a given cell size, the position of each cell is fixed. Therefore, each mobile node is a member of only one cell at any time. Given the current location coordinates of a node, there is a way for mapping the node location to the cell in which it is located currently. We have developed a self mapping algorithm to manage the node movement. The detail of this algorithm will be published in a separate paper.



(a) Partition the physical network area and cells identities

(b) Illustration of the relationship between a cell and its neighbors

Fig. 1. Virtual clustering approach

3.2.2 Cluster leader selection

In PBQMRP, the mobile nodes are selected to play different roles in the network according to a certain criteria. Some of the most-valued nodes among all nodes in the network are selected to form the virtual backbone in order to perform management functions to keep the network structure stable as possible. Also, these nodes maintain information about each node under their responsibility until it joins a new cell. These nodes are called *CL* nodes. Beside the selected *CL* nodes, *CLB* nodes are selected in each cell to avoid single point of failure and avoid executing frequent election operations.

Since the elected leader should be the most-valued node among all nodes in a specific cell [15], the following metrics are taken into consideration upon leader selection: the distance of the node from the cell center, residual energy, computational power, available memory and mobility speed. Each of these metrics is assigned a weighting factor. We believe that considering these metrics may aid in performing fair election. Details about the election process are found in [16].

Each elected *CL* declares its leadership to the nodes inside the cell and to the *CL* nodes of the 6-neighboring cells rather than flooding it to all the *CLs* in the network. This will reduce the number of control packets and the overhead produced from maintaining information about the global network. Then, nodes inside the cell reply by sending their current location and the multicast groups they are interested to join. We assume that all nodes are aware of the existing multicast groups. Since capabilities of the mobile nodes can change over time, the *CL* periodically declares its leadership to nodes inside its cell.

By the end of the network construction phase, each node maintains only the identity of the cell where it resides along with the identity of both the *CL* and *CLB* nodes of that cell. However, each *CL* maintains information about the identities and positions of the nodes residing in its cell, the membership of these nodes in different multicast groups in addition to information about the 6-neighbor cells (including *Cell_ID* and *CL_ID*).

3.3 Nodes communication and network maintenance

In this section, the required communications among nodes are described to ensure the maintenance of the network structure. The proposed protocol provides efficient solutions to nodes' mobility and failure to maintain a stable structure.

3.3.1 Communication inside the cell

The communication between all nodes inside a particular cell is done within only 1-hop communication (cell broadcast). This is because the side length of the cell is chosen to enable every two nodes inside the cell to communicate directly. Any node in the neighbor cells upon receiving packets destined to nodes inside another cell will drop it without any further processing. This mechanism significantly reduces the traffic of control packets.

3.3.2 Communication between neighboring cells

By using the information stored in the *CL* node about nodes in its cell and the location information about the neighbor *CLs*, the *CL* can communicate with the neighbor cells within at most 3-hop communication as follows. If the neighbor *CL* is within the transmission range of the sending *CL*, they can communicate directly and the packet will reach its destination in only 1-hop. Otherwise, the sending *CL* will send the packet to its 1-hop neighbors. If the destination *CL* is reachable directly, then the communication needs only 2-hops from the original *CL*. Otherwise, the packet is forwarded another hop to reach the destination using RDF; i.e., only 3-hops from the original *CL*, which is the worst case. In RDF, the node resends the packet only if it is closer to the destination than its previous hop. Using RDF eliminates flooding the network and restricts packet forwarding to the nodes in the way to the anticipated destination.

3.3.3 Ordinary nodes movement within its cell

To handle node movement, ordinary nodes in each cell send a packet to inform the *CL* about their current location only when their distance from the last known position is larger than or equal to a predefined threshold distance (*D_th*). Since these packets are not sent periodically, the processing and packet overheads are significantly reduced.

3.3.4 Ordinary node movement between cells

When an ordinary node crosses the boundary of its previous cell, then it's known that the node joins new cell. In this case, the moving node sends a location update packet to the *CL* node of that cell. When the *CL* node receives this update packet, it will remove the information related to this node and reply by sending a packet containing information about the new cell. The moving node uses this information to inform the new *CL* node about its presence and the multicast groups it's interested to join. If the new cell is empty, the moving node is considered as the *CL* for that cell.

3.3.5 Communication between the CL and the CLB

Due to node movement and failure, the *CL* node should perform periodic backup with the *CLB* node inside the same cell. This periodic backup is performed using 1-hop communication. Other nodes drop the packets related to backup operation. This periodic backup time need to be selected wisely to insure availability of backup data while not producing extra overhead. When this periodic time elapsed, the *CL* sends to *CLB* a *CL_Backup* packet, which contains the modifications, occurred in its tables since the last backup operation. This backup information will be available when the *CL* suffers a fault, when it decides to leave its current cell or when it suffers battery power drain. If *CLB* does not receive the periodic election packet from the *CL* node, it will discover that the *CL* is not working. In this case, it will work as a *CL* of the cell and initiate a new election process.

3.3.6 CL movement and failure maintenance

Since the *CL* has to maintain the recent identity of the 6-neighbor *CLs*, a notification packet has to be sent to the 6-neighbor *CLs* when the distance that the *CL* has moved exceeds the predefined distance (*D_th*) from its last known position (while still being within the bounders of its cell). This notification packet aids the *CL* to be aware of the status of the neighbor cells. When the *CL* node moves away from its current cell or its residual energy is degraded significantly, it must contact the *CLB* node in order to act as a leader and destined the data forwarded to it to the *CLB* node.

When the *CL* moves out its original cell, if the *CL* was the only node in that cell it will inform the 6-neighbor *CLs* that the cell becomes empty. Otherwise, it contacts the *CLB* node to work as *CL*, forwards the data sent to it to the *CLB* and clears unneeded information related to the departed cell. It also contacts the *CL* node of the new cell (to be considered as a new member in that cell) and changes its state as ordinary node.

3.3.7 CLB movement and failure maintenance

When the *CLB* node moves inside the cell and the movement distance exceeds the predefined distance (*D_th*), the *CLB* informs the *CL* about its new position. While, when the *CLB* crosses the boundary of its previous cell, it informs the *CL* of that cell. The *CL* accordingly initiates a new election process. When the leaving *CLB* receives this packet, it removes the information related to the previous cell and contacts the new cell's *CL* to join the new cell as ordinary node.

3.3.8 Handling empty cells

A particular cell may become empty when all the nodes inside it move away due to frequent mobility. In our protocol, the last node leaving the cell is certainly the *CL* node. So, if it decided to leave the cell, it should send an *EMPTY_CELL* packet to its neighbor *CLs* to inform them that the cell will be empty. Also, when a node leaves its original cell to an empty cell, the *CL* of the old cell inform the leaving node that the destined cell is empty in order for that node to be the *CL* of this cell.

3.4 Location service

The location service algorithm enables the source to map the geographical positions of the destinations. Based on our previous specification that the *CL* knows all the nodes that want to be members in a given multicast group, the location service is performed as follows. When a source node wishes to initiate a multicast session, a query packet (*INCELL_INV_REQ*) is first directed to its local *CL* node to ask for nodes that are interested in the multicast group. This packet needs only 1-hop communication operation. If the source node is the *CL* of the cell, there is no need to initiate such a packet. The *INCELL_INV_REQ* packet contains the fields *Source_ID*, *G_ID*, *Cell_ID* and *invs_seq_id*. The field *G_ID* represents the *ID* of the multicast group. Each node in the network has *invs_seq_id* which is increased monotonically with each invitation packet. *Source_ID*, *invs_seq_id* and *G_ID* fields are used to uniquely identify each invitation packet. When the *CL* node of the same cell receives the *INCELL_INV_REQ* packet, it sends *OUTCELL_INV_REQ* to the 6-neighbor *CL*s. After that, the *CL* searches for possible participating nodes in the held multicast group. If so, the *CL* sends a reply packet to the source node directly. This reply packet contains the positions and *ID*s of the interested nodes inside the cell. The *OUTCELL_INV_REQ* packet is propagated until it reaches all the network cells using the forwarding mechanism discussed below.

3.4.1 Location query packets forwarding

Our protocol utilizes the network division to deliver the invitation packet to discover the subscribed group members with very low overhead and to prevent sending duplicate packets.

As shown in **Fig. 2**, firstly, the *OUTCELL_INV_REQ* packet is forwarded towards the border of the 6-neighbor cells as a first forwarding zone, based on our cell *ID* assignment algorithm. If the neighbor *CL* is reached directly, only 1-hop unicast operation is needed to forward this packet to neighbor *CL* node. Otherwise, the packet is forwarded using RDF mechanism towards the border of the neighbor cells. If the intermediate node is closer to the destination than the previous node, it stores the previous hop node to be used in the reverse path and forwards the packet to the next hop node. Otherwise, the packet is dropped.

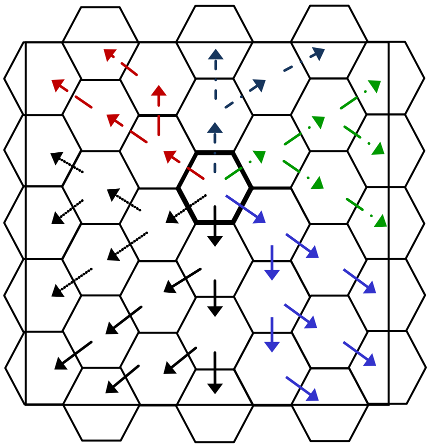


Fig. 2. Forwarding of *OUTCELL_INV_REQ* packets from the *CL* node to other *CLs*

Each neighbor *CL*, upon receiving such invitation packet for the first time continues forwards the invitation packet only to a specific set of neighbor *CLs* (as shown in **Fig. 2**). The proposed forwarding method enable the *CL* of each cell to take part in delivering the packet to at most two neighbor cells based on the border number that the packet comes from and the coordinates of the sending cell. This method insures that the *OUTCELL_INV_REQ* packet is propagated through the network with no duplicates and all the network cells are visited only once.

3.4.2 Location query reply

When the *CL* node, of the same cell where the source node resides, receives the *INCELL_INV_REQ* packet, it checks its *Member_Table* to see if there are nodes inside its cell that are interested in joining this multicast group. If so, the *CL* replies by sending an *INCELL_INV_REP* packet directly to the source node. This reply packet contains the positions and *ID*s of the destination nodes found inside the cell. Also, the *CLs* of neighbor cells that have multicast members reply by sending an *OUTCELL_INV_REP* packet using the reverse path until it reaches the *CL* node that issued the invitation request. This reply packet contains information about the current cell that is sending the reply, information about the cell that sent the multicast request and the positions and IDs of the destination nodes.

When the initiating *CL* node receives *OUTCELL_INV_REP* packets from the *CLs* of the network cells, it forwards these packets to the source node. The source node waits for a predefined time to aggregate the reply packets from the *CL* nodes to determine the nodes that want to participate in the group.

Consider the example shown in **Fig. 3** to explain the execution of the location service algorithm. First, an invitation packet is directed from the source node (*S*) to its *CL* node. The *CL* replies by sending information about the subscribing nodes in its cell if any. At the same time, the *CL* initiates an invitation request to neighboring cells. The Figure shows the forwarding process from *Cell_ID(1,2)* to *Cell_ID(2,3)*. It's clear from the **Fig. 3** that only node *A* and *B* forward this packet while other nodes drop it. This is because only node *A* and *B* are closer to the intended destination than their previous hop. Also, nodes in other cells drop this packet because the *Cell_ID* in the packet does not match their own *Cell_ID*. The forwarding process is continued until the destination (*CL* of *Cell_ID(2,3)*) is reached. When the destination receives the first request packet, it checks if there are destination nodes inside *Cell_ID(2,3)*. The reverse path of the request packet is used to send a reply packet to the sending cell (*Cell_ID(1,2)*).

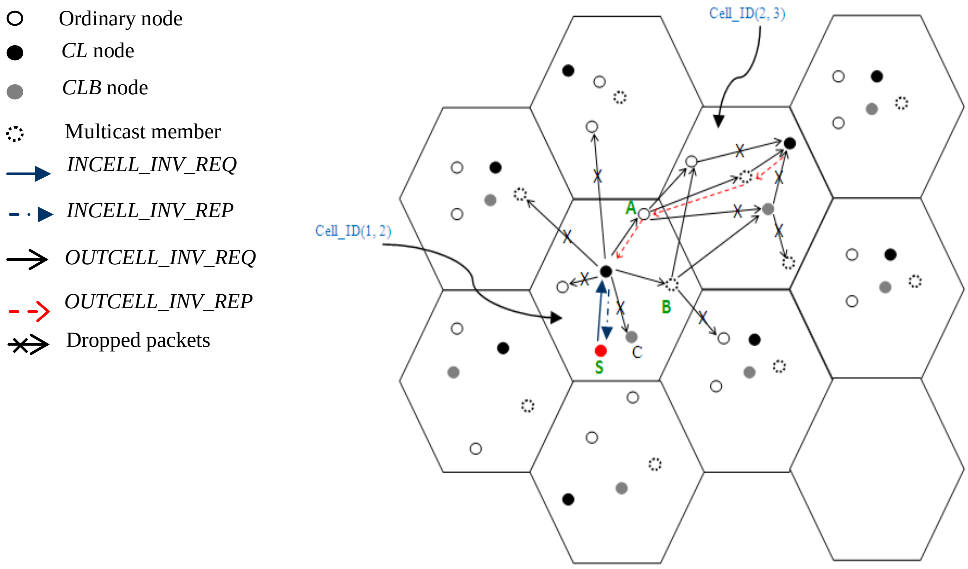


Fig. 3. Example of executing the location service algorithm

3.5 Formation of local groups

Upon executing the location service algorithm, the source node maintains the identities and positions of all the multicast members. After that, this list of members is arranged into several sub-groups. The groups' division process is based on the direct neighbor. In other words, the multicast members that concentrate in a local area (within transmission ranges of each other) are considered as a single sub-group. The division process is performed as follows. Initially, the first destination in the destination list is considered as a sub-group. The second destination node is considered as a member in this sub-group if it lies within the transmission range of the first node. Otherwise, it is considered as a new sub-group. The third destination node is checked whether it belongs to any of the previous sub-groups (if it lies within the transmission range of any node in these groups). If this is true, it will join the corresponding sub-group. Otherwise, it forms a new sub-group. If the new node is at a distance less than the transmission range from two nodes in two different sub-groups, then these two sub-groups are joined together forming one sub-group and this node will be a member in this sub-group.

This process is continued until all the nodes in the destinations list are arranged into disjoint sub-groups. If there is only one node at a sub-group, it will be the coordinator and unicast communication is performed between this node and the source node. To improve scalability, the number of members in each sub-group is limited to an application-dependent constant (*max_n*). For example, let node A and B be members of a sub-group SG_1, then the node C can be considered to be a member in SG_1 if:

$$((\mathrm{Dist}(C, A) < \mathrm{Trs\_rng\_}C) \ \text{OR} \ (\mathrm{Dist}(C, B) < \mathrm{Trs\_rng\_}C)) \ \text{AND} \ (\text{group-nodes} < max\_n)$$

As a following step, the source selects the closest node to himself from each sub-group to be the coordinator for the corresponding sub-group. The role of the coordinator is to manage the work of the sub-group and to forward the data packets to the members of the sub-group which is under its responsibility. With the hierarchical formation of the multicast members, each sub-group is recognized as separate group and represented by the coordinator. This approach would reduce the size of the route discovery packets, reduce the resulting overhead, efficiently improve the rate of the delivered data packets and minimize the bandwidth and resource usage.

3.6 QoS Multicast Route Discovery

In this section, the detail of route discovery process is discussed. Here, a QoS path which satisfies a requested bandwidth and delay requirements has to be established from the source to each destination in the destinations list. The requested bandwidth is included within the request packet and the upper limit of the delay value from the source node to any destination is represented as the number of hops.

In order to efficiently utilize the network bandwidth and reduce the communication overhead, the route discovery process starts by finding a route between the source and the coordinators; and later between the coordinators and other destinations in the same sub-group using the same mechanism. Furthermore, to distribute load among nodes, each coordinator will bear the responsibility of choosing the route from the source to itself, and each destination will choose the best route from the coordinator to itself.

3.6.1 Admission control and bandwidth estimation

Bandwidth estimation in wireless environment is more difficult than that in wired networks. This is due to the shared nature of the wireless medium and the transmission of neighboring nodes must be taken into consideration. Typically, bandwidth estimation is done at MAC layer by tracking the idle and busy time of the radio channel [17].

In our approach, the available bandwidth is estimated from the IEEE802.11 MAC and the Network layer performs routing based on the information coming from the MAC layer. The available bandwidth is estimated based on the “Listen” method proposed in [18]. In this method, each node listens to the radio channel for a period of time and tracks the length of time periods that the radio spends at the idle and busy time states. In MAC layer, the channel is considered as busy when the node is transmitting, receiving and when it senses the carrier channel. Then, such information is used to estimate the available bandwidth. The ratio of the idle time is considered as the ratio of the idle time to the overall time.

3.6.2 Route Discovery

After performing the sub-group division, the route discovery process starts by finding a route between the source and the coordinators; and later between the coordinators and the other destinations in the same sub-group using the same mechanism. The source starts by sending a Route Request packet (*QoS_RREQ*) to each coordinator individually using RDF mechanism. Using RDF gives high probability of having a path satisfying the needed number of hops in addition to giving opportunity of finding multi-segment paths satisfying the required bandwidth (while controlling the overhead).

The format of *QoS_RREQ* packet is shown in **Fig. 4** where *Req_Seq_No* is a sequence number increasing uniformly (for each source) and is used with the S_ID to uniquely distinguish the *QoS_RREQ* packets. The fields *prev_av_bw* and *Link_bw* represent the predecessor's available bandwidth and the available bandwidth on the link between the two successive nodes respectively. The parameter dist represents the distance between the sending and the destination nodes. The Route field is used to store the route that the packet goes on to discover its target. The *BW_Required* field contains the required bandwidth. The *DL_bound* field is initialized with the maximum number of hops between the source and each destination. This field is reduced by 1 at each forwarding node to ensure that the route that overcomes this delay bound will not be considered as a feasible route. The fields *Co_ID* and *Co_Pos* represent the *ID* and position of the coordinator node. The *Sub_group_list* field represents the *ID*s and positions of the destinations under the responsibility of the destined coordinator.

S_ID	*Req_Seq_No*	*Co_ID*	*Co_Pos*	*Sub_group_list*	*BW_Required*
DL_bound	*prev_av_bw*	*Link_bw*	*dist*	*Route*	

Fig. 4. *QoS_RREQ* packet format

When an intermediate node receives *QoS_RREQ* packet, it calculates the available bandwidth and calculates the link bandwidth between itself and the sending node. The intermediate node forwards the packet only if it has enough bandwidth, it is closer to the intended destination than the sending mode and the requested number of hops does not reach zero. The *QoS_RREQ* packet continues to be sent restrictedly until the destination node is reached or the maximum number of hops reaches zero. Each intermediate node upon receiving a *QoS_RREQ* packet that has been processed before will not process it again to prevent duplicate resource reservation; instead the route till this node will be considered as a segment path and sent to the coordinator to be used in route setup. The route discovery process is continued by searching for QoS paths between the coordinator and each sub-group member using the same strategy used in finding a QoS path between the source and the coordinators.

3.7 Route reply and setup

By the end of route discovery phase, different routes are discovered between the source node and the coordinator of each sub-group and between the coordinators and the rest of the destinations in their sub-groups. The request packets that reach the coordinators and the destinations come from the paths that satisfy the delay bound, so the route that has the needed end-to-end bandwidth has to be selected. To distribute load among nodes, each coordinator bears the responsibility of choosing the route from the source to itself, and each destination chooses the route from the coordinator to itself using the same mechanism.

When the coordinator receives the first route, it checks if the arrived route satisfies the required bandwidth at all the path nodes, then the coordinator select this route to be the optimal route, and sends this route back to the source via *QoS_RREP* packet. Otherwise, the coordinator will search for a segment that is parallel to the link that does not satisfy the bandwidth in the previous route. If a parallel segment is found, then the required amount of the bandwidth will be utilized and data on that branch node will be split into two parallel paths. This process is continued path by path until a best route is chosen. Using the same mechanism, the routes that satisfy the QoS constraints between the coordinators and each destination in each sub-group are selected. The use of multi-segment to satisfy the requested bandwidth can utilize the network resources and improve the performance over the uni-path routing [19].

When the route reply traverses back from the coordinators to the source (or from the destinations to the coordinator), each node along the chosen paths realizes that it will be a forwarding node, then it reserves the amount of the bandwidth to be used in the route and relays the packet to the node that sent the *QoS_RREQ* packet to it in the route discovery phase.

The format of *QoS_RREP* is shown in **Fig. 5**. The fields *Req_Seq_No*, *Co_ID* and *S_ID* are the same as their corresponding fields in the *QoS_RREQ* packet. The field *Route* of *QoS_RREP* packet contains the nodes that are in the selected paths or sub-paths.

S_ID	*Req_Seq_No*	*Co_ID*	*Route*

Fig. 5. *QoS_RREP* packet format

3.8 Data transmission

When the source receives the selected routes to all the multicast members, it adds the routes to the routing table and starts data forwarding using the selected routes. Data forwarding is started from the source to the coordinators. Whenever a data packet arrives to the coordinators, they continue forwarding it to each member of the sub-group. Each intermediate node simply relays data packets, as is, to its successor in the route obtained during the route initiation process.

3.9 Route maintenance

Route failure occurred due to nodes failure or movement during data transmission. In PBQMRP, nodes keep track of established routes. When a link break is detected, the upstream node (toward the source) of the broken link informs the source about this failure. The intermediate upstream nodes delete the routing table entry related to the broken link, free the reserved bandwidth and forward the packet towards the source node. When the source receives the broken link notification packet, it deletes the related routing table entry and initiates new route discovery to replace the failed route.

4. Performance Evaluation

The effectiveness of the proposed protocol was evaluated using Global Mobile Simulation (GloMoSim) [20] library. Since there are no benchmarks protocols have been defined for evaluating the performance of QoS protocols, we chose ODMRP [21] as a multicast routing protocol for comparison.

ODMRP is widely used as a benchmark for MANET multicast routing protocols and it is one of the elite multicast protocols. It is a mesh on-demand best-effort routing protocol that uses forwarding node concept. When the source has packets to send, it broadcasts a

JOIN_QUERY control packet to the entire network. The source periodically floods this packet (e.g. 3 seconds) to refresh the membership and update the routes. When a non-member node receives this packet it stores the upstream *node_ID* and rebroadcasts the packet. While, when a multicast member receives this packet, it creates or updates the source entry in its *MEMBER_TABLE* and broadcasts a *JOIN_REPLY* packet to its neighbors. Nodes receiving this reply packet check if the next *node_ID* of one of the entries in *JOIN_REPLY_TABLE* matches its own *ID*. If yes, the node realizes that it is on the path to the source, becomes part of the forwarding group and broadcasts its own *JOIN_TABLE* built upon matched entries. The set of forwarding nodes construct or update the multicast mesh from sources to receivers.

4.1 Parameters and metrics

A network of 240 mobile hosts placed randomly in an area of 2000m × 2000m with a node density of 60 nodes/km^2 is suggested for 600 second of simulation time. A node transmission range of 250 meters and a maximum channel bandwidth of 2Mb/s have been used. The initial positions of the nodes were chosen randomly, after that all nodes are granted the full mobility; they are allowed to move any where inside the whole area. Node mobility is simulated according to the random waypoint mobility model [22], since it is considered as one of the most utilized models in the literature. We have used IEEE 802.11 Distributed Control Function (DCF) as MAC layer. A single multicast group with a single source and 48 receivers have been used, unless otherwise specified. In the simulations, Constant Bit Rate (CBR) data traffic flows are injected into the network from the multicast traffic sources. The data payload has a size of 512 bytes per packet and transmitted every 500ms time interval. The multicast members are chosen randomly and join the multicast group at the beginning of the simulation and remain as members throughout the simulation. The parameters of ODMRP are chosen based on [21]. The common bandwidth requirement is set to 0.2Mb/s, unless otherwise specified. Multiple runs with different seed values were conducted for each simulated scenario and collected results were averaged over those runs. The following metrics were used to evaluate the efficiency and scalability of the proposed protocol.

1) ***Packet Delivery Ratio (PDR):*** The ratio between the number of multicast data packets delivered to all multicast receivers and the number of multicast data packets supposed to be delivered to multicast receivers. This ratio represents the effectiveness of the multicast routing protocol.

2) ***Packet Routing Load (PRL*): The ratio of control packets transmitted to data packets delivered. This ratio investigates the efficiency of utilizing the control packets in delivering data packets. The counted routing packets include those sent during route instantiation and maintenance phases for both protocols. While, in PBQMRP, all packets sent during the location service phase were also included in calculating this metric. The transmission at each hop along the route also was counted in the calculation of this parameter.

3) ***Average Path Length (APL):*** The average length of the discovered paths. It is calculated by averaging the number of hops traveled by data packets that reached their destinations.

4) ***Average Route Acquisition Latency (ARAL):*** The average delay needed for discovering a route to a destination. It is computed as the average time interval for both multicast session initiation (position discovery of the multicast members) and route initiation (sending route discovery request and the reception of the first reply).

4.2 Simulation Results

In this simulated performance evaluation, different network situations have been considered by varying the following parameters: node density, group size and bandwidth requirements.

4.2.1 Effect of node density

In this experiment, we investigate the scalability of the proposed PBMQRP and ODMRP with various node densities in the network. In our simulations, we assumed that a single multicast group with 48 receivers exists among the network nodes. To simulate different node densities, the following number of nodes is placed in 2000*m* × 2000*m* network: 80, 160, 240 and 320 nodes. These scenarios represent 20 nodes/km^2, 40 nodes/km^2, 60 nodes/km^2 and 80 nodes/km^2 node densities respectively.

In **Fig. 6-(a)**, the comparison of PDR is shown for each node density scenario. It can be observed that, as other position-based routing protocols, PBQMRP is sensitive to node density. It is well-known about geographic routing protocols that when the node density is small, there is less chance for an intermediate node to find a neighbor closer to the destination; hence resulting in higher packet loss [23].

At low node density (20 nodes/km^2), the network is weakly connected and the mobile nodes are far away from each other. Therefore, it is difficult to establish routes between a particular source and destination. Therefore, PDR of PBQMRP is less compared with ODMRP. However, with the increased node density, the network connectivity will increase and the probability of establishing routes will increase as well.

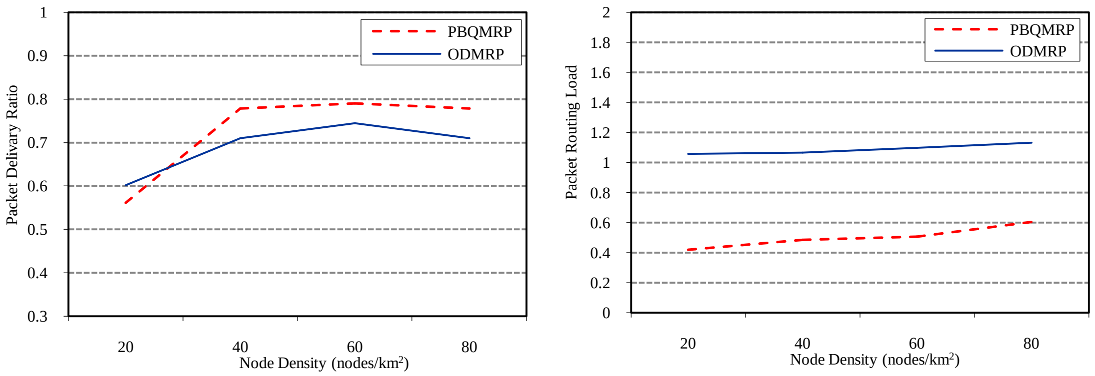
As expected, with increasing the node density to 40 nodes/km^2 and 60 nodes/km^2, PBQMRP performs better than ODMRP due to higher network connectivity and stable network structure. When the node density is higher than 60 nodes/km^2, PDR for both protocols gradually decreases. This decrease is a result of the large number of control and data packets, hence higher probability of collision and more dropped packets. However, the decrease in PDR for our protocol is slight compared to ODMRP.

It can be seen from **Fig. 6-(a)** that PBQMRP out performs ODMRP. PBQMRP is able to discover more links that satisfy QoS metrics; i.e., more reliable paths that deliver more data packets. While in ODMRP, the flooding behavior produces a lot of congestion in the network as the node density increases, which causes higher packet loss.

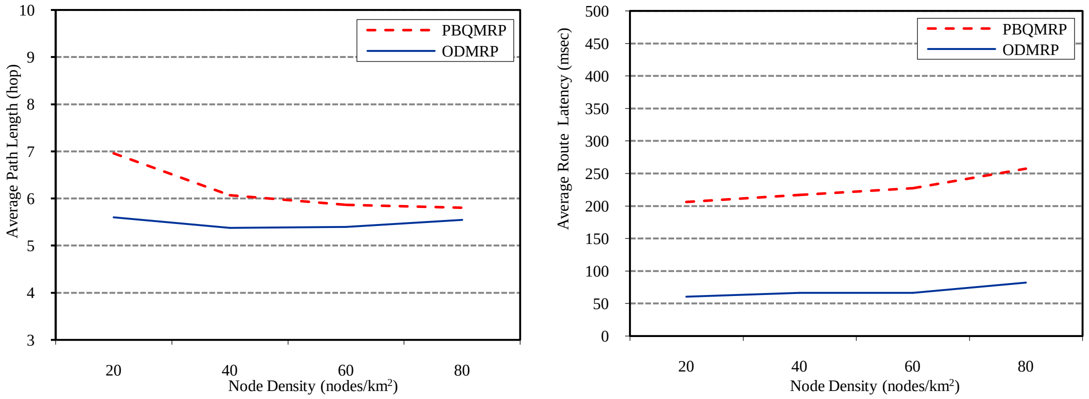
As shown in **Fig. 6-(b)** PRL for both protocols increases with increasing node density due to larger number of nodes participating in forwarding position and route discovery packets. PBQMRP has less PRL than ODMRP as a result of using restricted directional flooding in route discovery. Also, position discovery packets are sent only to *CL* nodes.

Fig. 6-(c) presents the APL for both protocols under different node densities. Results show that average number of hops decreases as the node density increases, because in dense network the probability of finding shorter paths increases. In ODMRP, paths have less number of hops due to the selection of the shortest path to forward the data packets. While in PBQMRP, the average path length is slightly longer due to two reasons. First, in PBQMRP the discovered paths have to satisfy QoS constraints, which may result in using longer paths. Second, the use of hierarchal construction of the destinations requires data packets to be delivered to the coordinators first and then forwarded to each destination independently. This strategy may result in longer paths.

Fig. 6-(d) shows that ARAL for both protocols increases with increasing node density due increasing number of participating nodes. Higher number of participating nodes causes congestion as well as delay in processing control packets. As the figure shows, PBQMRP has higher ARAL than ODMRP due to time required for position enquiry process.



(a) PDF vs. node density **(b)** PRL vs. node density



(c) APL vs. node density **(d)** ARAL vs. node density

Fig. 6. Effect of node density

4.2.2 Effect of the group size

The objective of this experiment is to study the effect of varying the group size from 20 members to 80 members. In this experiment, one source keeps sending CBR traffic to the multicast group.

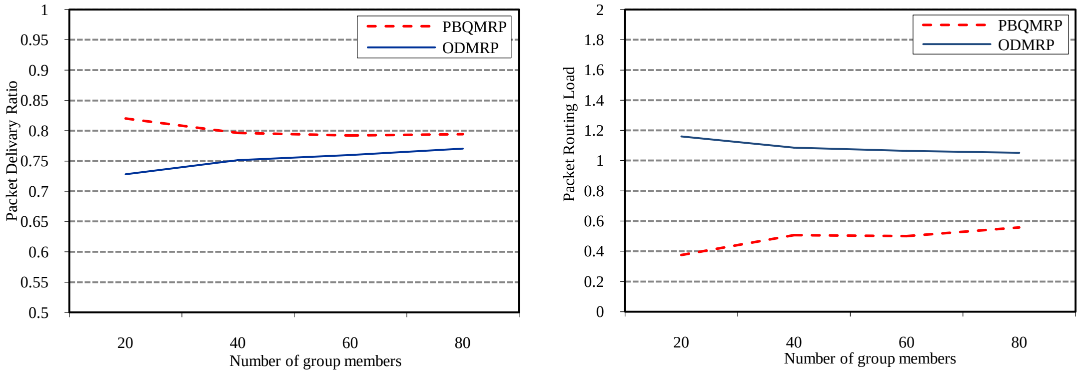
From **Fig. 7-(a)**, it is clear that PBQMRP delivers more fractions of data packets. The reason is that PBQMRP selects the routes that meet QoS parameters; so the selected routes are more reliable. Also, the hierarchal organizing of the multicast members previously discussed in section 3.6 improves the efficiency of forwarding the data packets. Compared to ODMRP, PBQMRP shows stable performance with a slight decreasing trend as the group size increases. This slight decrease is due to the packet losses resulted from tree disconnection. In ODMRP, PDR increases with increasing the group size. This is because ODMRP builds a mesh of forwarding nodes which becomes more reliable as the number of receivers increases. However, our protocol still has higher PDR regardless the group size.

Fig. 7-(b) shows that PBQMRP protocol maintains approximately stable PRL with a slight increasing trend. Increasing the group size means increasing the probability of having links breaks in routes towards these destinations. This requires reinitiating route discovery process; i.e., higher PRL. However, PRL of our protocol is still less due to using RDF. On the other hand, PRL for ODMRP slightly decreased with increasing group size since increasing the group size means more forwarding node, which makes the mesh more reliable in forwarding data packets.

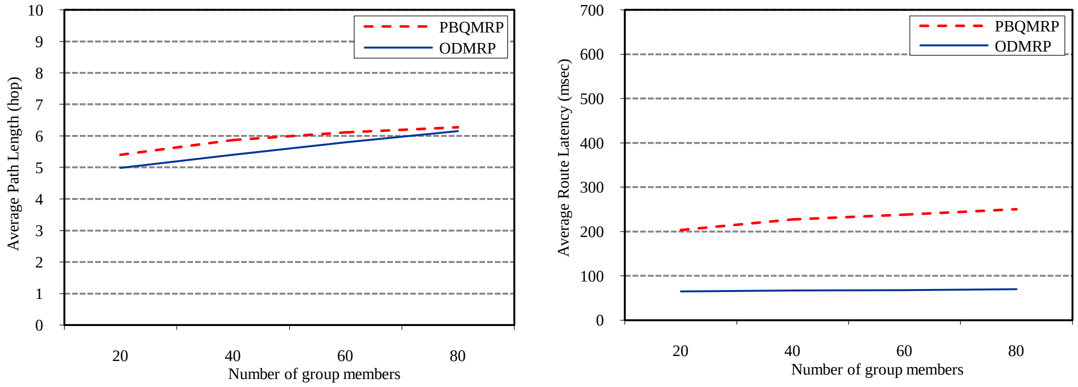
Fig. 7-(c) demonstrates the comparison results on average path length under different multicast group sizes. The figure shows that ODMRP outperforms PBQMRP in discovering

the shortest paths. This is expected because PBQMRP may select longer routes to satisfy the requested QoS parameters. However, the gap between the APL of both protocols is not significant. APL for both protocols increases as group size increases. This is may be justifiable for PBQMRP since it may choose longer paths satisfying the required bandwidth. However, we believe that the increase or decrease in APL depends on the positions of the new members added to the group.

In **Fig. 7-(d)**, the ARAL of PBQMRP is higher than that for ODMRP. The reason is that PBQMRP needs more time to get the positions of the destination nodes. Moreover, PBQMRP needs to check the available bandwidth on each node in the way to the intended destinations, which increases the time to select and return the suitable routes to the source node. For both protocols ARAL slightly increases with increasing group size due to the increase in APL shown in **Fig. 7-(c)**.



(a) PDF vs. group size **(b)** PRL vs. group size



(c) APL vs. group size **(d)** ARAL vs. group size

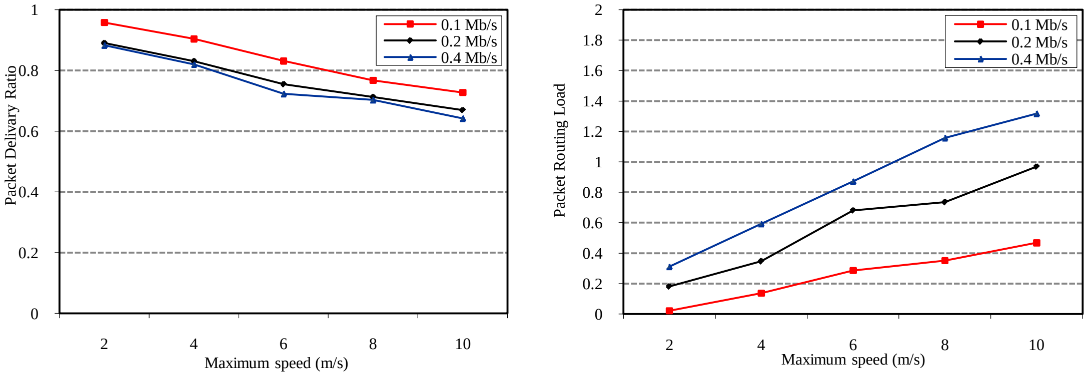
Fig. 7. Effect of group size

4.2.3 Effect of bandwidth requirement

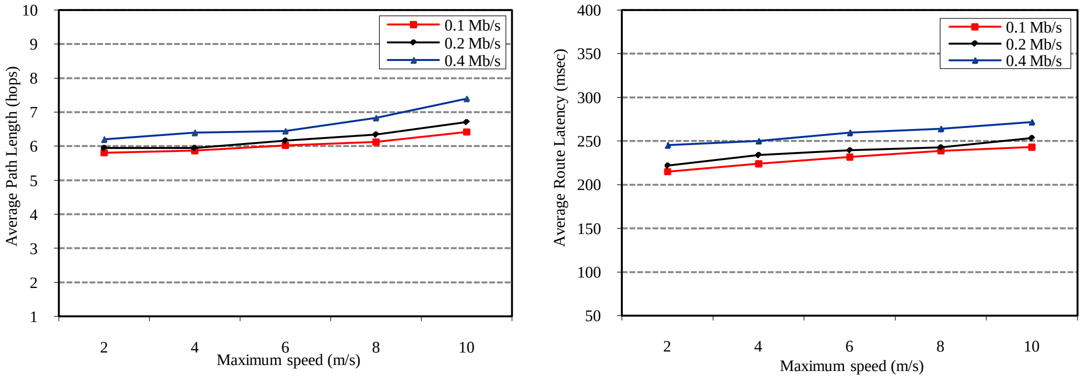
Here, we would like to see the effect of varying the bandwidth requirements for different mobility speeds. The common parameters are the number of mobile nodes (240 node), group size (48 member) and bandwidth requirements (0.1, 0.2, 0.4Mb/s). According to [24], for a network capacity of 2Mb/s, the maximum data rate will not exceed 0.5Mb/s. So in our simulations, we considered a range from low to high bandwidth requirements. For this parameter, we only simulate PBQMRP because ODMRP does not support QoS routing.

Fig. 8-(a) shows the PDR for different bandwidth requirements under different mobility speeds. We observe that at low bandwidth requirement (0.1Mb/s), the PDR gains higher

values (PDR=0.958) compared with (PDR=0.882) using the same mobility for high bandwidth requirement (0.4Mb/s). This is because using low bandwidth requirements increases the chance of finding paths that have enough available bandwidth, which reduces the rate of packet drop. These results indicate that in PBQMRP, the PDR is still good even if the requested bandwidth is high. This is due to using multi-path routing. In PBQMRP, if the destination does not find one uni-route with sufficient bandwidth, it waits for other possible parallel paths to fulfills the requested bandwidth as explained earlier is subsection 3.7. As we see from **Fig. 8-(a)**, the PDR is very sensitive to mobility. Increasing node mobility increases the probability of having link breakage; which accordingly results in dropping data packets.



(a) PDF vs. bandwidth requirement **(b)** PRL vs. bandwidth requirement



(c) APL vs. bandwidth requirement **(d)** ARAL vs. bandwidth requirement

Fig. 8. Effect of bandwidth requirement

Fig. 8-(b) shows that the PRL increases as node mobility increases for different bandwidth requirements. Increasing the mobility results in more broken links, and thus increases the routing overhead. When the requested bandwidth is high (0.4 Mb/s), the PRL gains its higher value for different mobility scenarios. This is because intermediate nodes do not forward route discovery packet except if they satisfy the requested bandwidth. This reduces the probability of finding routes to all destinations, which results in sending more and more route discovery packets; i.e., higher PRL.

Fig. 8-(c) reveals that when the requested bandwidth is high the number of hops becomes slightly higher. This is because the probability of finding a shorter path with QoS support is reduced. Also, with increasing the mobility, the nodes may become far from each other which makes the discovered routes longer.

Fig. 8-(d) shows that the ARAL is affected with mobility and ARAL increases with increasing the bandwidth requirement. This is because frequent movement of the nodes may result in having nodes far away from each other; increase the ARAL. As expected, when the requested bandwidth is high (0.4Mb/s), ARAL will be higher. The reason is that the destination/coordinator has to wait until a route with QoS is found. However, PBQMRP reduces the ARAL by selecting parallel segments to meet the requested bandwidth.

5. Conclusion

In this paper, we have designed a Position-Based QoS Multicast Routing Protocol (PBQMRP) for MANETs. The main objective of PBQMRP is addressing the scalability issue of QoS multicast routing protocols. In PBQMRP, the location information of the mobile nodes is used to develop a novel and scalable virtual architecture. This structure is utilized to perform efficient propagation of location service packets and manage nodes movement. After location discovery, efficient position-based route discovery for QoS paths is carried out.

Simulation results show that PBQMRP has decent performance in terms of packet delivery ratio and packet routing load. Also, PBQMRP shows real scalability for large area networks with large number of multicast members. The cost of our protocol is the time needed to inquiry about the multicast group members’ positions.

In the future, our aim is to develop a framework to support multimedia applications using multicast communication. Additionally, work can be done on examining the effect of the size of the cell and determining the optimal cell size that achieves good routing performance. Furthermore, we are looking for performing a simulation study to compare hexagonal gridding with square and triangle gridding using the same routing methodology. Last but not least, we are looking for studying the performance of PBQMRP using different sub-group sizes.

References

[1] Imrich Chlamtac, Marco Conti and Jennifer Liu, “Mobile ad hoc networking: imperatives and challenges,” *Elsevier, Ad Hoc Networks*, vol. 1, no. 1, p. 13–64, July 2003.Article (CrossRef Link)

[2] Prasant Mohaptra and Srikanth V. Krishnamurthy, “Eds., AD HOC NETWORKS: Technologies and Protocols,” *ISBN 0387226893. California, USA: Springer*, 2005.

[3] Kaan Bür and Cem Ersoy, “Ad Hoc Quality of Service Multicast Routing,” *Elsevier Science Computer Communications*, vol. 29, pp. 136-148, 2005. Article (CrossRef Link)

[4] Guojun Wang, Jiannong Cao, Lifan Zhang, Chan, K.C.C. and Jie Wu, “A Novel QoS Multicast Model in Mobile Ad Hoc Networks,” in *Proc. of 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS'05)*, vol. 4, no. 8, pp. 206b-206b, Apr. 2005. Article (CrossRef Link)

[5] Harald Tebbe, Andreas J. Kassler and Pedro M. Ruiz, “QoS-Aware Mesh Construction to Enhance Multicast Routing in Mobile Ad Hoc Networks,” in *Proc. of First International Conference on Integrated Internet Ad Hoc and Sensor Networks (INTERSENSE 2006)*, Nice, France, Jun. 2006. Article (CrossRef Link)

[6] Hui Cheng, Jiannong Cao and Xiaopeng Fan, “GMZRP: Geography-aided Multicast Zone Routing Protocol in Mobile Ad Hoc Networks,” *Mobile Networks and Applications, (ACM/Springer)*, vol. 14, no. 2, pp. 165-177, 2009. Article (CrossRef Link)

[7] Ivan Stojmenovic, “Position-Based Routing in Ad Hoc Networks,” *IEEE Communications Magazine*, vol. 7, pp. 128-134, Jul. 2002. Article (CrossRef Link)

[8] Harald Tebbe and Andreas J. Kassler, "QAMNet: Providing Quality of Service to Ad hoc Multicast Enabled Networks," in *First International Symposium on Wireless Pervasive Computing (ISWPC)*, Phuket, Thailand, 2006. Article (CrossRef Link)

[9] Mina Masoudifar, "A review and performance comparison of QoS multicast routing protocols for MANETs," *Elsevier, Ad Hoc Networks*, vol. 7, no. 6, pp. 1150-1155, Aug. 2009. Article (CrossRef Link)

[10] Aisha Hashim, Mohammad Qabajeh, Othman Khalifa and Liana Qabajeh, "Review of Multicast QoS Routing Protocols for Mobile Ad Hoc Networks," *IJCSNS International Journal of Computer Science and Network Security*, vol. 8, pp. 108-117, Dec. 2008.

[11] Tzay-Farn Shih, Chao-Cheng Shih and Chin-Ling Chen, "Location-Based Multicast Routing Protocol for Mobile Ad Hoc Networks," *WSEAS Transactions on Computers*, vol. 7, no. 8, pp. 1270-1279, Aug. 2008.

[12] Chao-Cheng Shih and Tzay-Fam Shih, "Cluster-Based Multicast Routing Protocol for MANET," *WSEAS Transactions on Computers*, vol. 6, no. 3, pp. 566-572, Mar. 2007.

[13] S. Sivavakeesar, G. Pavlou and A. Liotta, "Stable clustering through mobility prediction for large-scale multihop iintelligent ad hoc networks," in *Proc.of IEEE 2004 Wireless Communications and Networking Conference (WCNC 2004)*, vol. 3, pp. 1488-1493, Mar. 2004. Article (CrossRef Link)

[14] Asad Amir Pirzada and Chris McDonald, "Reliable Routing in Ad-hoc Networks using Direct Trust Mechanisms," in *Advances in Ad Hoc and Sensor Networks*, Maggie Xiaoyan Cheng and Deying Li, Eds.: Springer, ch. 6, pp. 133-159, 2008. Article (CrossRef Link)

[15] Muhammad Rahman, M. Abdullah-Al-Wadud and Oksam Chae, "Performance analysis of Leader Election Algorithms in Mobile Ad hoc Networks," *IJCSNS International Journal of Computer Science and Network Security*, vol. 8, no. 2, pp. 257-263, Feb. 2008.

[16] Mohammd Qabajeh, Aisha Hashim, Othman Khalifa and Liana Qabajeh, "A Novel QoS Multicast Routing Protocol in MANETs," in *Proc. of Annual International Conference on Network Technologies & Communications (NTC2010)*, Phuket,Thailand, 2010, pp. N1-N7, 2010.

[17] Kaixin Xu, Ken Tang, Rajive Bagrodia, Mario Gerla and M. Bereschinsky, "Adaptive Bandwidth Management and QoS Provisioning in Large Scale Ad Hoc Networks," in *Proc. of Military Comm. Conf. (MILCOM '03)*, Oct. 2003. Article (CrossRef Link)

[18] Lei Chen and Wendi B. Heinzelman, "QoS-Aware Routing Based on Bandwidth Estimation for Mobile Ad Hoc Networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 3, pp. 561-572, Mar. 2005. Article (CrossRef Link)

[19] Shafqat Ur Rehman1, Wang-Cheol Song and Gyung-Leen Park, "Associativity-Based On-Demand Multi-Path Routing In Mobile Ad Hoc Networks," *KSII Transactions on Internet and Information Systems*, vol. 3, no. 5, pp. 475-491, Oct. 2009. Article (CrossRef Link)

[20] UCLA Parallel Computing Laboratory, Glomosim, http://pcl.cs.ucla.edu/projects/glomosim.

[21] Sung Ju Lee, William Su and Mario Gerla, "On-Demand Multicast Routing Protocol in Multihop Wireless Mobile Networks," *Mobile Networks and Applications*, vol. 7, no. 6, pp. 441-453, Dec. 2002. Article (CrossRef Link)

[22] Jungkeun Yoon, Mingyan Liu and Brian Noble, "Random Waypoint Considered Harmful," in *Proc. of IEEE INFOCOM03, 2(4)*, San Francisco, p. 1312–1321, Apr. 2003. Article (CrossRef Link)

[23] Xiang Xiaojing, Wang Xin and Yang Yuanyuan, "Stateless Multicasting in Mobile Ad Hoc Networks," *IEEE Transactions on Computers*, vol. 59, no. 8, p. 1076–1090, Aug. 2010. Article (CrossRef Link)

[24] Jinyang Li, Charles Blake, Douglas S.J. De Couto, Hu Imm Lee and Robert Morris, "Capacity of Ad hoc Wireless Networks," in *Proc. 7th ACM Int. Conf. Mobile Comput. Networks* , Rom, Italy, pp. 61–69, 2001. Article (CrossRef Link)

Mohammad M. Qabajeh received his B.Sc. in Computer Engineering from Palestine Polytechnic University, Palestine in 2000 and M.Sc. in Computer Engineering from Jordan University of Science and Technology, Jordan in 2006. He is now working towards his PhD in Computer Engineering at International Islamic University Malaysia, Malaysia. His current research interests are Distributed Systems and Ad hoc Networks.

Aisha Hassan Abdalla Hashim received her B.Sc. and M.Sc. in Computer Engineering from University of Gezira, Sudan and her PhD in Computer Engineering from International Islamic University Malaysia (IIUM) in 2007. She was appointed as a lecturer in 1997. Currently she is Associate Professor at IIUM. Her current research interests include Data Communication and Computer Networking, ASIC Design, Computer Architecture and Grid Computing, Open Sources & Operating Systems. She published more than 100 papers in international journals and conferences. She is IEEE Senior Member, IEEE Women in Engineering Member and she works as a reviewer for many ISI journals.

Othman O. Khalifa received his B.Sc. in Electronic Engineering from the Garyounis University, Libya in 1986 and M.Sc. in Electronics Science Engineering and PhD in Digital Image Processing from Newcastle University, UK in 1996 and 2000 respectively. He worked in industrial for eight years and he is currently Professor and Head of the department of Electrical and Computer Engineering, IIUM. His area of research interest is Communication Systems, Digital image / video processing, Wavelets, Fractal and Pattern Recognition. He published more than 130 papers in international journals and Conferences. He is SIEEE member, IEEE computer, Image processing and Communication Society member.

Liana K. Qabajeh received her B.Sc. in Computer Engineering from Palestine Polytechnic University, Palestine in 2000 and M.Sc. in Computer Engineering from Jordan University of Science and Technology, Jordan in 2005. She is now working towards her Ph.D in Computer Science at University of Malaya, Malaysia. Her current research interests are Distributed Systems and Ad-Hoc Networks.