Engineering Letters, 18:3, EL_18_3_02

Geographical Multicast Quality of Service Routing Protocol for Mobile Ad-Hoc Networks

Mohammad M. Qabajeh, Aisha H. Abdalla, Othman Khalifa, and Liana K. Qabajeh

Abstract—**Recently, the necessity of applications where many users have to interact in a close manner over mobile Ad-Hoc networks gains high popularity. Multicast communication is essential in this type of applications to reduce the overhead of group communication. For group-oriented multimedia applications Quality of Service (QoS) provision is a basic requirement, which makes an efficient QoS multicast routing protocol a very important issue. This paper proposes a location-based QoS multicast routing protocol via cooperation between Network and MAC layers. Along with this protocol, a location and group membership management scheme has been proposed. To further reduce the control overhead and bandwidth consumption, we apply clustering strategy by partitioning the network topology into hexagon cells. Thus, maintaining the network topology is limited to certain nodes. The performance of the proposed protocol is evaluated using GloMoSim simulation environment. Simulation results show that our approach provides high packet delivery ratio associated with low control overhead.**

***Index Terms*—Ad-Hoc Networks, Multicast Routing, Position-based, QoS**

I. INTRODUCTION

Mobile Ad-Hoc NETworks (MANETs) are collection of mobile nodes communicating in a multi-hop manner without any fixed infrastructure or central administration. In MANETs, all mobile nodes function as hosts and routers at the same time. Two nodes communicate directly if they are within the transmission range of each other. Otherwise, they reach each other via a multi-hop route.

Group communication becomes increasingly important in MANETs since a lot of applications relay on cooperation among team members. Video conferencing, interactive television, temporary offices and network gaming are common examples of these applications [1]. As a consequence, multicast routing has received significant attention recently. Multicast communication has emerged to support applications that facilitate effective and collaborative communication among groups of users with the same interest. In multicasting, packets are delivered from a source to a group of destinations identified by a single group address. Multicast routing protocols try to utilize the network resources by sharing some parts of the paths from the source to the destinations, which is essential in MANETs [2][3].

The increasing popularity of using multimedia and real time applications in different potential commercial scenarios in MANETs, make it logical step to support Quality of Service (QoS) over wireless networks. QoS support is tightly related to resource allocation and reservation to satisfy the application requirements. These requirements include bandwidth, delay, delay-jitter and probability of packet loss. It is a challenge to support QoS in MANETs since network topology changes as the nodes move; also network state information is generally imprecise. This requires extensive collaboration among nodes, to establish the route as well as to secure the necessary resources to provide QoS. Moreover, the MAC layer centralized design and limited resources of the nodes make it more difficult to guarantee QoS in Ad-Hoc networks. As a result, combining QoS with multicasting faces several challenges due to the difficulty in finding paths between the source and all destinations those satisfy certain QoS requirements.

Recently, the availability of small, inexpensive, low-power GPS receivers and techniques for calculating relative coordinates based on signal strengths realize the location-based routing for Ad-Hoc networks [4].

In this paper, we investigate the problem of QoS routing in MANETs using multicast communication. In view of the advantages of location-based routing, a novel location-based QoS multicast routing protocol for MANETs has been proposed. Along with this protocol, a location and group management schemes have been proposed. The physical area is partitioned into a number of equal-sized cells. In each cell, a selection algorithm is executed to determine a leader and a backup node. The cell leader should be powerful enough to take charge of its connecting nodes. This leader is responsible for maintaining positions of the nodes inside the cell as well as their group memberships.

Manuscript received in August 28, 2009. This work was supported by Research Management Center at the International Islamic University Malaysia.

Corresponding author: Mohammad M. Qabajeh is now working towards his Ph.D. from Electrical and Computer Engineering Department (ECE) ,International Islamic University Malaysia (IIUM), Malaysia. (phone:+60126468600, email: m_qabajeh@yahoo.com).

Aisha Hassan Abdalla is with Department of Electrical and Computer Engineering, IIUM, Malaysia (email: aisha@iiu.edu.my).

Othman Khalifa is currently the Head of Department of Electrical and Computer Engineering, IIUM, Malaysia (email: Khalifaaisha@iiu.edu.my).

Liana K. Qabajeh is now working towards her Ph.D. from Computer Science and Information Technology Faculty, University of Malaya, Malaysia (email: liana_tamimi@ppu.edu).

When a source node has data to be sent to a group of destinations, an efficient communication procedure is carried out between cell leaders to provide the source with all the nodes interested in this multicast session and their positions. Now the source will be able to divide the group members into manageable sub-groups and choose a coordinator for each sub-group to start the multicast session. The QoS requirements that have been taken into consideration in this protocol are bandwidth and delay.

The Network layer interacts with the MAC layer to estimate the available bandwidth based on channel status. Since, the bandwidth is shared among the network nodes, the activities of the neighboring nodes are taken into consideration, which makes our protocol more practical.

The proposed scheme exploits the residual bandwidth efficiently by using multi-segments paths if the bandwidth of a single path is not sufficient. Most of the communication is done using 1-hop or restricted directional flooding in order to reduce packet overhead and utilize the network bandwidth. The proposed protocol is supposed to be scalable for large area networks with large number of multicast group members. It is suitable to be used when the data files to be sent are large (which is the case in multimedia applications) and require setting up a route before starting data transmission. Also, it is suitable to be used in both dense and sparse networks.

Due to large number of nodes and large geographical area of Ad-Hoc networks, extensive simulations are carried out to evaluate the performance of the new protocol. We will study a wide range of scenarios by varying different performance metrics.

The remainder of this paper is structured as follows. We present some related works in section II. The proposed model is introduced in section III. Section IX presents the simulation model and the simulation results. Finally, section X concludes this paper.

II. RELATED WORK

Multicasting in MANETs is relatively unexplored research area, when it is compared with unicast routing [3]. Multiple QoS multicast routing protocols have been proposed for Ad-Hoc networks such as [5][6][7][8][9]. Also, many position-based multicasting protocols have been proposed including [10][11][12][13][14]. However, few works have been done in QoS position-based multicasting such as [15][16]. We divide the related work into three main groups: position-based multicast routing protocols (section A), QoS multicast routing protocols (section B) and QoS position-based multicast routing protocols (section C).

A. Position-Based Multicast Routing

A Location Guided overlay multicasting protocol is proposed in [13]. It is a stateless scheme based on packet encapsulation in a unicast envelops to be transmitted to group of nodes. It builds an overlay packet distribution on top of the underling unicast routing protocol based on the geometric locations of the group nodes only. In Location-Guided *k*-ary (*LGK*) scheme, the sender first selects the nearest *k* destinations as children nodes, then the rest of the nodes are grouped to its *k* children according to close geometric proximity. In Location-Guided Directional (*LGD)* tree, the sender partition the space into multiple cone areas centering about itself, the nearest node in each cone is selected as its child. In Location-Guided Steiner (*LGS*) tree, based on the geometric distance as a measurement of closeness, a Steiner tree is constructed by using the multicast group members as tree nodes.

A generalization of position-based unicast forwarding has been discussed in [14]. In this protocol, the sender includes the addresses of all the destinations in the header of the packet. Based on the nodes position information, each node determines the neighbors that it should forward the packet to. When the current node selects more than one next hope node, then the multicast packet is split. Also, when there is no direct neighbor to make progress toward one or more destination a repair strategy is used. Position-Based Multicast (PBM) is limited to groups with small number of nodes because the location and group membership information is included in the data packets.

In Dynamic Source Multicast (DSM) [12], each node floods the network with information about its own position, thus each node knows the positions of all other nodes in the network. The source node constructs a multicast tree from the position information of all receivers and encodes the paths in the header of the packet. In DSM, the periodic flooding of position information for all the nodes on the network reduces the scalability of the system and increase the processing overhead of the nodes.

B. QoS Multicast Routing

The Lantern-Tree-Based (LTB) in [6] is a bandwidth constrain QoS multicast routing protocol. A lantern is defined as one or more sub-paths with a total bandwidth between a pair of two neighboring nodes. A lantern path is a path with one or more lanterns between a source and a destination. The multicast tree contains at least one lantern path between any of its source-destination pairs. Lantern-tree protocol measures the bandwidth as the available amount of free slots based on CDMA-over-TDMA channel model at MAC layer, which needs distributed time synchronization. One drawback of LTB is the long time needed to find all the paths and to share and schedule the time slots. Another drawback is the use of high number of links, which increase the contention at the MAC layer.

On-demand QoS multicasting protocol is proposed in [9]. This protocol simultaneously use multiple paths or trees in parallel to meet the required bandwidth of a single QoS request within a delay bound between the source and the destination. The bandwidth is considered as the number of free slots using CDMA-over-TDMA channel model. They propose three multiple path construction strategies to enable the source node to aggregate the bandwidth over the links.

The source computes the optimal routes to the destinations and manages the group membership, which overload the source with extra processing overhead. Also, using flooding to discover the paths add extra processing overhead for non-member nodes and waste the network resources.

QoS Multicast Routing Protocol (QMR) [7] is an on-demand mesh protocol that uses forwarding mesh same as On-demand Multicast Routing Protocol [17]. The bandwidth is estimated at each node and reserved only when the QoS request is accepted. The bandwidth is divided into “shared” and “fix reserved”. The intermediate nodes forward the data packets if shared bandwidth is available. The forwarding nodes are updated when multiple sources sending to the multicast group simultaneously. This prevents congestion and performs load balancing in the network. However, the redundant flooding increases the congestion and the overhead, which affects the QoS flow.

C. QoS Position-Based Multicast Routing

A cluster-based QoS multicast routing protocol is proposed in [16]. In this protocol, the area is partitioned into equal-size square clusters and the nearest node to the center is elected as a cluster-head for each cluster. Next, a gateway node is selected to forward the packet when the headers of the adjacent clusters are out of the effective transmission range. The source node starts the multicast session by sending PROPE packet to the cluster-head. The gateway forward 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 [18], where the 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 doesn’t mention the network structure and maintenance, which perhaps produce significant traffic. Additionally, in sparse networks, the gateway nodes may fail to deliver packets between neighbor cluster heads. Therefore, the route cannot be established.

In [15], 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 [19]. 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 to 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 *CH*s within the hypercube and one of the *CH*s periodically broadcasts the membership to all the clusters in the network.

When a node wants to send data to group members, it sends it to its *CH*. Then, the *CH* check the summarize membership to determine the hypercubes that maintain the members of this group. The logical locations of these hypercubes 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’s forwarded to those hypercube nodes that contain the group members. *HVDB* protocol provides fault tolerance property and scalable. However, it produces a lot of 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.

III. PROPOSED PROTOCOL

The proposed protocol divides the whole network into several hexagonal cells. Each cell has a Cell Identity (Cell_ID). In each cell, a leader node is elected to maintain the current location of the nodes inside that cell, along with the multicast groups they are interested to join. Other responsibilities of the cell leader includes forwarding the location service packets, contacting leaders of the 6-neighboring cells and managing the joining process of new members to the multicast session. Also each cell has Cell Leader Backup (CLB) node, which is responsible for keeping a copy of the data stored on the CL in order not to be lost if the CL node became off or moved outside the cell. The cell size is chosen to enable 1-hop communication among all the nodes inside a given cell in order to reduce communication overhead.

When a source node *S* wants to send data packets to a given multicast group, an efficient communication procedure is done among cell leaders to provide the source with the members of that multicast group as well as their positions. 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. The selected paths should satisfy a particular bandwidth and delay requirements.

To present a complete idea about our model, the network setup phase is described first. Then the location service algorithm used to determine the location of the target nodes is explained. Finally, handling the QoS request is discussed.

A. Network Setup

The first step in our model is the network setup. This phase includes partitioning the network into virtual cells and specifying the responsibility of each node in this structure.

A.1 Area Partitioning

The network is partitioned into non-overlapping equal-sized hexagonal cells. Based on our assumption that all the nodes inside a specific cell use 1-hop communication; the maximum distance between any two nodes inside the cell should not exceed the effective transmission range (R). Considering the hexagonal, square and triangle cell shapes, and substituting R as the maximum distance among the cell nodes; it is obvious that the area covered by the hexagonal ($0.6495\times R^2$) is larger than that covered by the triangle ($0.433\times R^2$) and the square ($0.5\times R^2$). This enables 1-hop communication among higher number of neighbors than provided by the other shapes and reduces the number of cells leaders. This, in turn, results in reduction of the control overhead. Moreover, the advantage of the hexagonal cell is that it offers six directions of transmission, thus possibly fastening the processes requiring cell leaders to communicate with the neighboring cells. In the literature, many researchers have used the hexagonal gridding such as [20][21][22]. For all the aforementioned reasons it was decided to use the hexagonal shape.

For simplicity we assume the routing area is a two-dimensional plane. Also, we assume that each node has its unique ID and can obtain its geographic coordinate by using GPS receivers or some other ways. Thus, all nodes are able to do self-mapping of their physical locations onto the cell they reside in. Figure 1 shows the general overview of the network architecture.

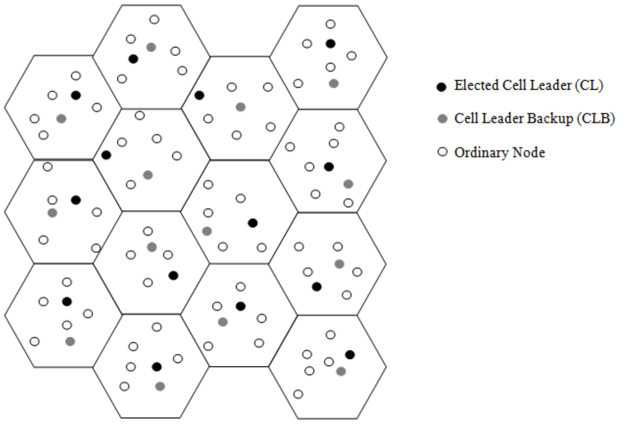

Figure 1: General overview of the network architecture

We denote the transmission range of a node as R and the side length of the cell as L. In our protocol, the value of L is set as $L = R/2$ to guarantee that each two nodes in the same cell are always within the effective transmission range of each other. Thus, the nodes inside a given cell can communicate with each other directly.

A.2 CLs and CLBs Election

An adaptive election algorithm is developed to elect the nodes that satisfy different metrics in order to take the role of the leaders and survive the longest possible time. Since the elected leader should be the most valued-node among all the nodes in a particular cell [23], the following metrics are taken into consideration upon the leader selection:

Distance from the cell center, residual energy, computing power, available memory and mobility speed.

Each of the above mentioned metrics is assigned a weighting factor. So, each node computes locally its capability to be a leader, and exchange it with other nodes found in the cell via a 1-hop transmission. For example, node *A* residing in cell *i* will send the following capability message (*CL-Cap*) to the nodes inside this cell:

$Node_A \longrightarrow Cell_i\_Nodes$: [*CL_Cap, Node*$_A$*_ID, Cell_ID, capability*]

Every node now is aware of the capability of all other nodes in its cell, so it can recognize that the node with the highest capability will be the CL node and the node with the second highest capability will play the role of the CLB node. Each CL node should announce its leadership by sending a *New_CL* packet only to the nodes inside the cell and the CL nodes of the 6-neighboring cells rather than flooding it to all CLs in the network. This will reduce the number of control packets and reduce the overhead resulted from maintaining information about the global network. The CLB information is contained in the message in order to enable the nodes to contact the CLB in case of sudden CL failure.
The *CL* of cell *i* for example will send:

$CL_i \longrightarrow Cell_i\_Nodes$: [*New_CL, CL_ID, Cell_ID, CLB_ID*]
$CL_i \longrightarrow Neighbor\_CL_i$: [*New_CL, CL_ID, Cell-ID_N6_List, CL_Pos, CLB_ID*]

Each node inside the cell, upon receiving the *New_CL* packet, replies to the CL by sending *Node_Pos* packet which contains its current location (*Node*$_i$*_Pos*) along with the multicast groups it is interested to join (*GID_list*). We assume that all nodes are aware of the existing multicast groups.

$Cell_i\ \_Nodes \longrightarrow CL_i$: [*Node_Pos, Cell_ID, CL_ID, Node*$_i$*_ID, Node*$_i$*_Pos, GID_list*]

Since capabilities of the mobile nodes can change over time, the CL periodically sends a *leader_refresh* packet to nodes inside the cell. Each node in that cell compares its capability with that of the current CL. If there is no node with higher capability than the CL, the leader information remains as is and no packets are generated. Otherwise, only nodes with higher capabilities than the CL will reply for the *leader_refresh* packet. The current CL chooses new CL and CLB nodes and notifies nodes inside the cell with this change. The format of the *leader_refresh* packet is:

$CL_i \longrightarrow Cell_i\ \_Nodes$: [*Leader_Refresh, Cell_ID, CL_ID, NC*$_{CL}$]

By the end of network setup phase, each node maintains only the identity of both the CL and CLB nodes of the cell where it resides, in addition to the identity of that cell. While, each CL keeps information about the identity and position of nodes in the cell it is responsible for, the membership of these nodes in different multicast groups as well as information about the 6-neighboring cells (including cell identity, identity and position of the CL and CLB nodes).

B. Nodes Communication

In this section the methods used for communication among different nodes in the network are explained briefly.

B.1 Communication inside the Cell

All nodes inside a particular cell (including CL and CLB*)* communicate via 1-hop unicast communication. This is since the side length of the cell is chosen to enable any two nodes inside the cell to communicate directly. Any node in the neighbor cells upon receiving of packets destined to nodes inside another cell will drop them immediately. This mechanism reduces the control packets traffic significantly.

B.2 Communication between the Neighboring Cells

By utilizing information stored in the CL node about nodes in the cell as well as the location information about the CLs of the neighbor cells, the CL can communicate with the neighbor cells within at most 3-hop communication. If the distance to the neighbor CLs is less than the transmission range, the CL can communicate deliver the packet to them in 1-hop. Otherwise, the CL will relay the packet towards a node on the border of the cell that its leader is not reached directly. The border node will resend the packet to its 1-hop neighbors. If the leader of the neighbor cell is within the transmission range of the border node, it will receive the packet directly (2-hops from the original CL). Otherwise, the nodes in the neighboring cell that have received the packet will send it to their CL (3-hop from the original CL).

C. Network and Group Membership Maintenance

This section describes the needed procedures to maintain the structure of the network.

C.1 Communication between the CL and Cell Nodes

Ordinary nodes in a specific cell sends a position update packet to inform the CL about their current locations only when the distance from the last known position is larger than or equal to a predefined threshold distance (*D_th*). Hence, packet processing overhead will be significantly reduced.

Suppose that node *A* in cell *i* has moved the predefined distance, it will unicast the following 1-hop packet:

Node$_A$ → *CL*$_i$: [*Pos_update, Node*$_A$*_ID, Cell_ID, CL_ID, Node*$_A$*_Pos*]

When any modification happened to the multicast groups that the nodes joining, the nodes will inform the CL with this modification through the *Group_update* packet. Suppose that node *A* in cell *i* wants to modify the multicast groups it is interested to join, then it uses the following (*Group_update)* packet to update the multicast group:

Node$_A$ → *CL*$_i$: [*Group_update: Node*$_A$*_ID, Cell_ID, CL_ID, GID_list*]

C.2. Communication between the CL and the CLB

The 1-hop communication is performed between the CL and CLB periodically in order to send a copy of the information stored in the CL to its CLB. The CL sends to CLB the changes happened in its tables since the last backup operation has been performed. The copy of the information stored at CLB is used when the CL suffers a fault, when the CL decides to leave its current cell or when the CL loses power dramatically.

CL → *CLB*: [*CL_Backup, CL_ID, CLB_ID Cell_ID, DataModification*].

C.3 Communication between CLs in Neighboring Cells

Each CL node sends a *still_alive_Nbr* packet to the 6-neighboring CLs only if it has moved *D_th* from its last known position to notify them of its presence. This *still_alive_Nbr* packet helps the CL of each cell to be aware of the status of the neighbor cells.

CL$_{\text{i}}$ → *CL*$_{\text{6-neighbor}}$ [*Still_Alive_Nbr: CL_ID, CL*$_{\text{6-neighbor}}$*_ID, CL_Pos*]

C.4 Node Movement between Cells

When a node moves outside its current cell, it will immediately inform the CL node of its old cell that it’s leaving the cell by sending a *cell-leave* packet. Accordingly, the CL will update its member table to ensure freshness. When the CL node receives this packet, it will reply by sending the *Cell_ID* and the identity of the CL node of 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.

Suppose that node *A* at cell *i* is moving to cell *j*, it will send and receive the following packets:

Node$_A$ → *CL*$_i$: [*Cell_Leave, Node*$_A$*_ID, CL_ID*]
CL$_i$ → *Node*$_A$: [*Leave_Reply, CL_ID, Node*$_A$*_ID, CL*$_j$*_ID*]
Node$_A$ → *CL*$_j$: [*Cell_Join, Node*$_A$*_ID, CL*$_j$*_ID, Node*$_A$*_Pos, GID_list*]

C.5 CL Movement between Cells (or CL failure)

When the CL node moves away from its current cell or its power is degraded significantly, it must contact the CLB node to act as a leader and destined the data forwarded to it to the CLB node. In case of CL sudden failure, the CLB will discover this failure through the CL-CLB periodic packet. If the predefined period ends without receiving the packet from the CL, then the CLB will discover the CL failure. In both cases, CL movement and sudden failure, the CLB node informs the nodes inside the cell that it becomes the new temporarily leader. In addition, the CLB sends this packet to the CL nodes of the neighbor cells to inform them about this change. Also, the moving CL node clears the tables related to the old cell. The following packets are used:

CLi → *CLB*$_i$: [*CL_ Retire, CL_ID, CLB_ID*]
CLB → *Cell*$_i$*_Nodes* [*New_CL, CL_ID, Cell_ID*]
CLB → *Neighbor_CL*$_i$: [*New_CL, CL_ID, Cell_ID_N6-List, Cell_ID, CLB_ID*]

After that, new election is performed to elect new CL and CLB nodes. Here, each node calculate its leadership capability as done previously in the network setup phase and send this capability to the temporary CL. The temporarily CL elects the new CL and CLB and sends the result of this election to all the nodes inside the cell and the CL of the neighboring cells.

C.6 Empty Cells

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 CLs of its neighboring cells. Also, when a node leaves its cell to an empty cell, the CL of the old cell will include "none" in the *CL_ID* field of the *Leave-Reply* message sent to the leaving node. This is to inform the leaving node that the destined cell is empty and it will be the CL of this cell.

IV. MULTICAST COMMUNICATION AND ADMISSION CONTROL

An efficient location service algorithm to determine the position information of the nodes in the intended multicast group will be discussed in section A.1. In section A.2, we will present an admission control scheme that will use the available bandwidth calculation to determine if the Route Request can be accepted or not.

A.1 Location Service Algorithm

When a source node wants to start a multicast session, it starts by sending an invitation packet (*InCell_Invitation_REQ*) to the CL node where it is located to ask for nodes that are interested with this multicast group. This packet needs only 1-hop unicast operation because the location of the CL node is known for all the nodes inside the cell. The *InCell_Invitation_REQ* packet contains *GID, Source_ID and Node_Seq_No*. The field *GID* represents the *ID* of the multicast group. Each node in the network has *Node_Seq_No* which is increased monastically with each invitation message. The tuple (*GID, Source_ID, Node_Seq_No*) is used to uniquely identify each invitation message. Suppose that a source node *S* at cell *i* is asking for the nodes inside the cell that want to join a multicast group *GID*, then the following packet is sent:

$S \rightarrow CL_i$: [*InCell_Invitation_REQ, Source_ID, CL_ID, GID, Node_Seq_No*]

When the CL node receives the *InCell_Invitation_REQ* packet, it searches in its member table about the nodes interested in joining this multicast group, and then it replies by sending an *InCell_Invitation_REP* packet directly to the source node. This reply contains the position and *ID* of the destination nodes inside the cell. Suppose that a source node *S* sends an invitation packet inside the cell and CL finds that there are 3 destinations (D_1, D_2 and D_3) want to join the group, and then the reply message will be:

$CL_i \rightarrow S$: [*InCell_Invitation_REP, CL_ID, Source_ID,* (D_1*_ID*, D_1*_Pos*), (D_2*_ID*, D_2*_Pos*), (D_3*_ID*, D_3*_Pos*)]

The search for additional destinations is continued by sending an invitation packet (*OutCell_Invitation_REQ*) to the CL of the 6-neighboring cells. The CL node forwards this packet towards the border nodes of the neighbor cells except the cell from where it received this packet. When the border node receives this packet, it stores the previous hope node to be used in the reverse path and forwards the packet to the next hop node. If the next hop node is ordinary node, it forwards the packet to the CL node. If the next node is the target CL node, then it stores the previous hop node. Then, the request packet is propagated cell-by-cell until it covers the entire network using the same mechanism.

The *OutCell_Invitation_REQ* packet contains *Cell_ID*, *CL_ID*, *Cell_Seq_No*, *GID* and *(CL_ID, position)* of the neighbor cell. Suppose that CL_i is the leader node at cell *i* and it wants to ask the neighbor cell *j* if there is nodes want to join the group using this message.

$CL_i \rightarrow CL_J$: [*OutCell_Invitation_REQ,* CL_i*_ID,Cell_ID, Cell_Seq_No, GID,* CL_j*_ID,* CL_j*_pos*]

The pair (*Cell_ID*, *Cell_Seq_No*) is used to uniquely identify each out cell invitation packet. The *Cell_Seq_No* is a sequence number for each cell which increasing monastically to check for duplicate packets. In the cell that has destination nodes, the CL node sends an *OutCell_Invitation_REP* packet using the reverse path until it reaches the CL node that initialize the invitation request. This packet includes {$Cell\_ID_s$, *Dest_List*, *GID*, $Cell\_ID_d$}. The field $Cell\_ID_s$ represents the *ID* of the cell sending this reply and $Cell\_ID_d$ represents the *ID* of the cell sending the multicast request. The field *Dest_List* contains the positions and *ID*s of the destination nodes. The pair ($Cell\_ID_d$, *SEQ_ID*) is used to uniquely identify the reply of the invitation packets.

The source node waits for a predefined time to aggregate the reply packets from the CL nodes in the network in order to determine the nodes that want to participate in the group.

The algorithm executed at the source node is illustrated in Appendix A1.

A.2 Admission Control

In our model we provide an on-demand multicasting protocol to satisfy a certain bandwidth and delay requirements from one source node to a group of destinations. This is because bandwidth and delay are critical requirements for real time applications.

The available bandwidth along the path *p* is represented as *BW(p).* Let $N_1, N_2, \ldots N_i$ be the nodes along the path *p* and the link bandwidths are represented as $BW(N_1, N_2)$, $BW(N_2, N_3)$, … $B(N_{i-1}, N_i)$. The path bandwidth *BW(p)* is the minimum bandwidth of a link along the path *p*.

$$BW(p) = \min \{ BW(N_1, N_2), BW(N_2, N_3), \ldots B(N_{i-1}, N_i) \}$$

The available bandwidth is estimated based on the "Listen" method proposed in [24][25]. In this method, each node listens to the radio channel and tracks the traffic of the neighboring nodes in order to determine the available bandwidth. In other words, each node listens to the channel and determines the idle duration for a period of time.

In MAC layer, the channel is considered as busy when the node is transmitting, receiving and when it senses the carrier channel. The amount of the available bandwidth is considered as the ratio of the idle time to the overall time. Assume that the monitoring period is M_T and the summation of idle times during this period is Idle_T, then the ratio of the idle time is:

$$\text{Ratio of Idle time } (R) = \frac{\text{Idle}_T}{M_T}$$

The available bandwidth at a node (*i*) is estimated as the channel bandwidth times the ratio of the idle time as follows:

$$ABW_i = C_i \times R$$

Where C_i is the raw channel capacity.

When the intermediate node receives the route request packet, it calculates the link bandwidth between itself and the previous node based on the information about the available bandwidth at this node and the previous node. The request packet is dropped if there is no available bandwidth at the node.

If the link available bandwidth is less than or equal the required bandwidth, the amount of the available bandwidth is allocated. Else, the extra bandwidth is considered as free bandwidth and can be allocated for a new route. This allocation for the bandwidth at the intermediate nodes continues for a period of time (*allocation-time*) until the QoS path is selected.

When the destination node selects the routes that satisfy the requested bandwidth (as will be discussed in section VI), the intermediate nodes those are part of the selected route will change their status from allocate to reserve. The reservation state remains until the data transfer is completed, and then the reserved bandwidth will be free. After a predefined time, the intermediate nodes which are not considered as part of the QoS path will free all the allocated bandwidth. When the call setup fails, it is necessary to free all the reserved bandwidth along all paths between the source and destinations. This is important because the bandwidth is limited in wireless Ad-Hoc networks.

V. ROUTE DISCOVERY

In this protocol a QoS path which satisfies a particular bandwidth and delay requirements has to be established from the source to each destination in the destinations list. The requested bandwidth on the link between two successive nodes is included within the request packet. 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 source node divides the list of destinations into a number of sub-groups. A node is selected in each sub-group (coordinator) to manage the work of this sub-group. To distribute the 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. After that, the source and the coordinators will check if there is a common links between paths, in order to send the packet once in the shared links.

A. Sub-groups Construction and Coordinator Selection

The list of destinations maintained at the source node is divided into sub-groups. 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 will be considered as a new sub-group. The third destination node is checked if it belongs to any of the previous sub-groups or not (if it lies within the transmission range of any node in these groups). If this is true, then 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 separated 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. In order to have manageable-sized sub-groups and achieve real scalability, the number of nodes in a given sub-group is limited to a predefined value (*max_n*). This will minimize the packet size, which reduce the overhead by limiting the length of some fields such as the *Route field* in *QoS-RREQ* packet. 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:

((Dist(C, A) < Trs_rng_C) OR (Dist(C, B)< Trs_rng_C)) AND (group-nodes < max_n)

As a following step, the source selects the node that is closer to him from each sub-group to be as a coordinator for that sub-group as shown in figure 2. The role of the coordinator is to minimize the bandwidth and energy usage and forward the data packets to the members of the sub-group those are under his responsibility. The data flow starts by forwarding a copy of the data packet from the source node to each coordinator then from each coordinator to its sub-group nodes as destinations. The number of replications of the original data packet from the source depends on the number of sub-groups.

By using this sub-grouping mechanism, the size of the packet header is reduced significantly which allows the system to be scalable to larger number of group members. Also, there is no need for the source to maintain information about the global network topology. Moreover, there is no communication overhead for this sub-groups construction since it is performed only at the source node.

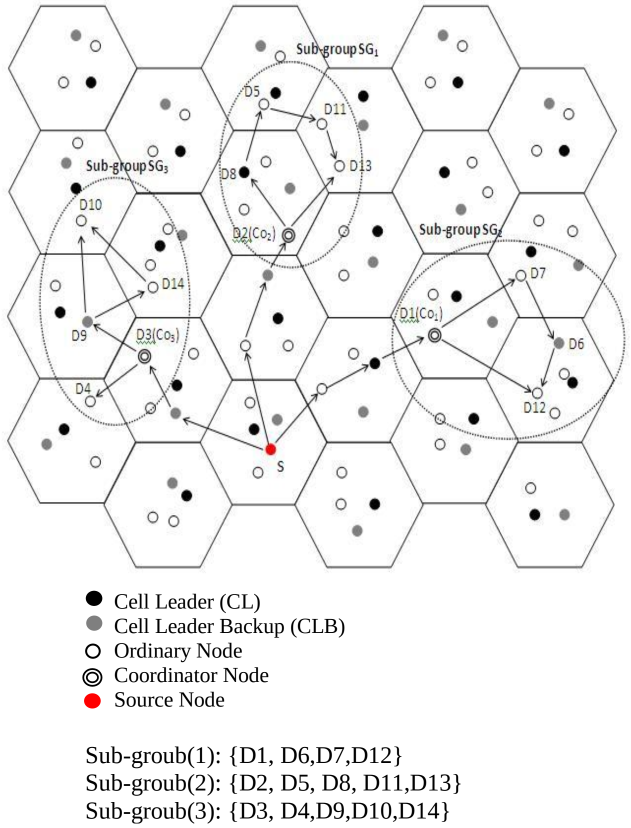

Figure 2: Sub-groups construction of the multicast group.

B. *Source-Coordinators QoS Route Discovery*

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. Hence, we will begin our discussion by explaining the selection of the *QoS* paths from the source to the coordinators.

After the sub-groups construction process, the list of the multicast group members *(G)* will have been divided into sub-groups SG_1, SG_2,…,SG_i and the coordinator nodes of the sub-groups are denoted as Co_1,Co_2,…,Co_i. Now, the source starts by sending a Route Request packet (*QoS-RREQ)* to each coordinator individually using Restricted Directional Flooding (RDF). In RDF, the node resends the packet only if it is closer to the destination than it's previous hop. Using RDF increases the 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. The algorithm that illustrates RDF is presented at Appendix A2.

The *QoS-RREQ* packet contains *Source_ID, Seq_ID, Co_ID, Co_Pos, Sub_group_list, BW_Required, Node_Available_BW, Link_BW, DL_bound* and *Route*. The "*Seq_ID"* is a sequence number increasing uniformly (for each source) and it is used with the "*Source_ID"* to uniquely distinguish the *QoS-RREQ* packets. Also, "*Node_Available_BW"* field is used to record the amount of the available bandwidth for each node along the QoS path, the "*Link_BW"* field represents the available bandwidth on the link between two successive nodes. The *"Route"* field is used to record the routing information.

The field "*DL_bound"* is set to the delay bound which restricts the scope of route search. "*Co_ID"* and "*Co_Pos"* fields represent the *ID* and the position of the coordinator node. The *"Sub_group_list"* field represents the *ID* and position of the destinations under the responsibility of the destined coordinator.

When an intermediate node receives *QoS-RREQ* packet, the packet is dropped if there is no available bandwidth. Otherwise, if the value of *DL_bound* is greater than zero and there is available bandwidth on the node, then the node determines the amount of link bandwidth available between itself and the previous node and stores this amount to the *Link_BW* field. Then it stores the amount of the remaining available bandwidth in the field *Node_Available_BW*. The value of *DL_bound* is reduced by 1, this is to ensure that the route that overcomes this delay bound will not be considered as a feasible route. The node then adds itself to the *Route* field. The *QoS-RREQ* packet continues to be sent until the coordinator node is found or *DL_bound* reaches zero. Suppose that S_i is a source node and Co_j is the coordinator for a sub-group, then the following *QoS-RREQ* packet is sent from S_i to Co_j as follows:

$S_i \longrightarrow Co_j$:[*QoS-RREQ, Source_ID, Co_ID, Seq_ID, Co_Pos, Sub_group_list, BW_Required, Node_Available_BW, Link_BW, DL_bound, Route*]

Each intermediate node upon receiving a *QoS-RREQ* packet with a (*Source_ID, Seq_ID*) pair that have been processed before, it will not processed 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 coordinator node will receive multiple *QoS-RREQ* packets. So, it will record information about each request and wait either for a pre-specified time out or pre-specified number of *QoS-RREQ* packets before responding to the request. This is in order to receive multiple routes and choose the best route among them as well as to prevent waiting for routes that take long time. The coordinator chooses the best route from the source to itself (as will be discussed in section VI) and sends a reply packet along the selected path, which reserves the corresponding resources (bandwidth required) on the corresponding nodes on their way back to the source. The algorithm executed to handle the QoS_RREQ packet at the coordinator node and the intermediate node is illustrated at Appendix A3 and A4 respectively.

$Co_j \longrightarrow S_i$: [*QoS-RREP, Co_ID, Source_ID, Selected-Route*]

The source node waits for route replies from the coordinator nodes and stores routes to each coordinator in order to be used for data transfer. If the wait time has been elapsed and the source did not receive any reply from a particular coordinator, this coordinator is considered as died node and the source choose new coordinator from the same sub-group and start a route discovery process to the new coordinator.

C. Coordinator-Destinations QoS Route Discovery

Since the locations of all members of the sub-group are known to the coordinator from the *QoS-RREQ* packet it has received from the source. Now, it's required to find a route between each coordinator and all the members in its sub-group.

The search for QoS paths between the coordinator and each sub-group member is performed using the same strategy used in finding a QoS path between the source and the coordinator. The coordinator broadcasts a *Sub-QoS-RREQ* packet using *RDF* to find a route to each destination from the *Sub_group_list*. This packet contains *Co_ID*, *Seq_ID, Dest_ID, Dest_pos, BW_Required*, *Node_Available_BW, Link_BW, New_DL_bound* and *Route*. "*New_DL_bound"* represent the remaining delay limit for the "*DL_bound"* after the packet reach the coordinator. When the value of the "*New_DL_bound"* field goes down zero, this means that the *Sub-QoS-RREQ* packet exceeded the acceptable delay for the intended application, and then the "*Sub-QoS-RREQ"* is dropped. Information (*Dest_ID, Dest_pos*) is used to represent the address of the target destination.

Each intermediate node upon the receipt of a packet with a (*Co_ID*, *Seq_ID*) pair that have been processed before will not process it again; instead the route till this node will be considered as a segment path and reply it back to the coordinator to be used in route setup. Suppose that Co_i is a coordinator node and wants to discover a route to destination *j* which is one member of Co_i sub-group, the *Sub-QoS-RREQ* packet sent is as follows:

$Co_i \longrightarrow D_j$: [*Sub-QoS-RREQ, Co_ID, Dest_ID, Seq_ID, BW_Required, Node_Available_BW, Link_BW, New_DL_bound, Route*]

When a node receive *Sub-QoS-RREQ* packet from the coordinator, it adds its amount of the available bandwidth to the field "*Node_Available_BW"*, it determines the available bandwidth on the link between itself and the previous node and adds it to the "*Link_BW"* field. Also, the value of "*New_DL_bound"* field is decrease by 1 and the packet is forwarded using RDF to the next node. If a destination node is reached and the value of "*New_DL_bound"* does not reach zero, it waits for a pre-specified time out or for a pre-specified number of "*Sub-QoS-RREQ"* packets before a destination chooses the best route that guarantees QoS requirements using the mechanism described in section VI, then it sends a reply packet along the selected path and reserve the amount of bandwidth on the corresponding nodes. The format of the *Sub-QoS-RREP* packet contains is:

$D_j \longrightarrow Co_i$: [*Sub-QoS-RREP, Dest_ID, Co_ID, Selected-Route*]

If the value of "*New_DL_bound"* reaches zero and the destination is not reached yet, this request packet is dropped. And if the coordinator does not receive a reply packet from a given destination, it recognizes that there is no route that satisfy QoS requirement to that destination and this destination is removed from the destination list.

VI. ROUTE SETUP

After the discovery of different routes between the source nodes and the coordinator of each sub-group as well as between the coordinator and the rest of the destinations in each sub-group. The request packets that reach the coordinator and the destinations comes from the paths that satisfy the delay bound. So, it is now required to select a route that satisfies the required end-to-end bandwidth between the source and all destinations.

To select a route, a coordinator node is responsible to select the most stable route and return the route reply to the source node. When it received the first route from the source, it selects this route for data transmission if it satisfies the requested bandwidth at all path nodes, since it is the route with less delay due to its early arrival. After that, it sends a route reply packet to the source. Otherwise, the coordinator waits for the second route and checks if that route satisfies the bandwidth requirement, if this is true, it will be selected. If not, the coordinator will search for a segment that is parallel to the link that does not satisfy the bandwidth in the previous route in order to satisfy the requested bandwidth. If a parallel segment is found, it will take the required amount of the bandwidth and splits the data on that branch node into two parallel paths. This process is continued path by path until a best route is selected.

When the route reply traverses back from the coordinator to the source (or from the destinations to the coordinator), each node along the chosen paths reserves the amount of the bandwidth that is considered to be used in the route and relies the packet to the node that send to it in route discovery. During constructing the routes between the coordinator and its destinations, the source node start delivering data over the QoS route to coordinator. When the coordinator receives the data packets from the source node, it delivers a copy of the data packet to each member of the sub-group. The same mechanism is executed recursively between the coordinator and each destination to select a route that satisfies the required end to-end bandwidth.

For example, assume that we have one source (*S)* and 7 destinations (*D1... D7*) and the destinations are considered as one sub-group as shown in Figure 3. Also, assume that the bandwidth requirement is 3 units and the delay bound is within 6 hops. The figure shows the links and the available bandwidth on each link.

In our model the nearest node to the source will be chosen as the coordinator of the sub-group (*D1* in our example). The source will communicate with the coordinator in order to transfer data to all the sub-group members. Here we will discuss how destination *D7* chooses the best route to receive data from the coordinator node. Suppose that the routes that have arrived to D7 are as follows:

1) D1 $\underline{\;3\;}$ D2 $\underline{\;4\;}$ D5 $\underline{\;2\;}$ D7
2) D1 $\underline{\;4\;}$ D3 $\underline{\;2\;}$ D5 (sub-route)
3) D1 $\underline{\;4\;}$ D2 $\underline{\;2\;}$ D5 $\underline{\;4\;}$ D6 $\underline{\;2\;}$ D7
4) D1 $\underline{\;4\;}$ D3 $\underline{\;1\;}$ D2 (sub-route)
5) D1 $\underline{\;3\;}$ D2 $\underline{\;5\;}$ D5 (sub-route)
6) D1 $\underline{\;4\;}$ D3 $\underline{\;6\;}$ D4

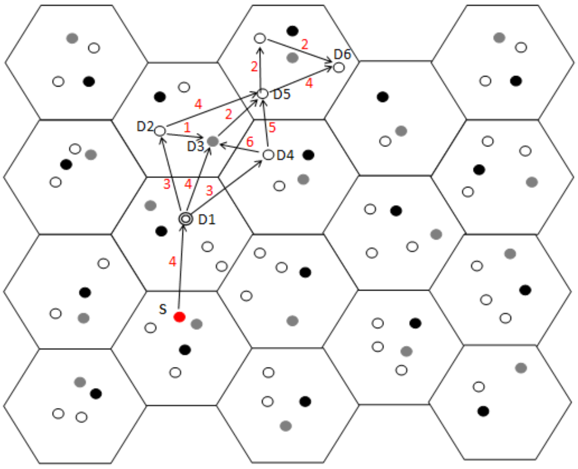

Figure 3: Example for route setup

In the first route, it is clear that the link *D5-D7* does not satisfy the bandwidth requirements, so *D7* will wait for the next route. The second route cannot be chosen as a whole optimal path simply because it's a sub-route (not a complete path from *D1* to *D7*). Also, it does not have a segment parallel to *D5*, *D7* that satisfy the required bandwidth.

In the third route, *D7* discovers that the link *D2-D5* does not satisfy the bandwidth requirement. So, the third route cannot be taken as a whole route, however it contains a parallel segment to link *D5-D7* that can be used for data transfer. The bandwidth in the link *D5-D7* in the first route is 2 units, so we need only 1 unit from the segment {*D5, D6*, *D7*}. The resulting route that will satisfy the requirement after using the parallel segment is shown in figure 4.

All the destinations in the sub-group use the same technique to find the route from the coordinator D1 to them.

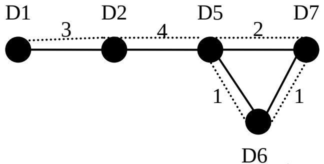

Figure 4: The selected elected route from the coordinator (D1) to destination (D7)

VII. ROUTE MAINTENANCE

After the route setup is done nodes can move freely in the network, join or leave the multicast groups, become corrupted or even destroyed, etc. Dealing with these issues is discussed in the following sub sections.

A. Node Leaving the Multicast Group

When any member of the sub-group decides to leave the multicast group and move away, it sends a *LeaveAck* packet to the coordinator through its upstream nodes in order to update the member list after this node leaves. When the coordinator node receives the *LeaveAck* packet, it checks if this node does not have downstream nodes, then it is removed from the destination list of the coordinator and is excluded from future forwarding computations. Otherwise, the coordinator should reconstruct new routes to the affected members of the sub-group by resending a *Sub-QoS-RREQ* packet. After that, the information associated with this multicast group member has to be removed from the coordinator and other members in the sub-group. Lastly, if the leaving node is the sole node in the sub-group, the source suspends sending data to this sub-group.

B. Node Joining the Multicast Group

The new node sends a join request packet *(JOIN_REQ)* to the *CL* of the cell where this node resides in order to add itself to the required multicast group. This request packet includes the *Source-ID*, *GID*, *node ID* and the amount of the available bandwidth at this node. The *CL* through using the information about the previously joining nodes, it inform the new joining node the address of the nearest node that can connect it with the multicast group. In case that there is no multicast members inside the same cell, communication is started between the *CL* and some neighboring *CL* (nearest to the new node) to select the node that can pass data to the new node. The new joining node waits for a period of time to collect the *JOIN-REPLY* packets, and then it choose the best route that connects it to the multicast group within the number of hops limitations. If this route satisfies the needed bandwidth, it will be considered as the final route to the new node. Otherwise, a new join packet is sent to the coordinator in order to begin a RDF to search for QoS-path to the new member.

C. Coordinator Failure and Movement

When the coordinator decides to leave the multicast group, it must inform the source node by sending a coordinator leave (*Co-Leave)* packet. When the source receives this packet, it will assign new coordinator to serve the sub-group.

Moreover, when the coordinator stops working suddenly, the whole sub-group which this coordinator is responsible for will no longer be able to receive data packets from the source. We thus need a maintenance procedure. The source node will piggy-back an *ACK* packet on to the data packet periodically, upon receiving this message, each coordinator needs to reply with an *ACK-Reply* packet. Thus, the source node can check each coordinator if the *ACK-Reply* packet has arrived within a predefined period of time. When a coordinator is recognized as not working, the source needs to choose another node in the sub-group and assign it the responsibilities of the sub-group coordinator.

VIII. BROKEN LINKS

During data transmission, some nodes may not receive data due to broken links caused by nodes failure or movement. This link breakage is detected if no packets are received from the upstream nodes after a pre-defined time interval. If there are no downstream nodes for the failed (or moving) node then its failure will not affect the other nodes. Otherwise, the affected nodes need to re-join the multicast group as discussed in section VII.B. The nodes that find a broken route to any multicast member inform the coordinator through this route about the failure by sending a link-Error message. When the coordinator node receives this message, it will delete the information related to this broken link and start searching for alternative path to the affected nodes.

IX. PERFORMANCE EVALUATION AND ANALYSIS

A. Simulation Environment

In this section, we examine the effectiveness of the proposed approach using simulation. The simulator used for this evaluation is the network simulator Global Mobile Simulation (GloMoSim). Since GloMoSim is a scalable simulation (up to thousand of nodes) environment for mobile wireless networks using parallel discrete-event simulation capability provided by PARSEC [26].

The simulation models a MANET network of 300 nodes moving over a space area of 3000 × 3000 m for 900 second of simulation time with a node density of 100 nodes per square kilometer is suggested. Each node is equipped with a radio transceiver whose transmission range is up to 250 m over a wireless channel. The MAC layer in all simulations was IEEE 802.11 with a maximum channel capacity of 2 Mb/s. The nodes in our simulation moves according to the random way-point mobility model provided by GloMoSim with mobility speed varies from 0-60 m/s and a pause time of 30s.

In the simulations, Constant Bit Rate (CBR) data traffic flows are injected into the network from the multicast traffic sources. The data payload had a size of 512 bytes per packet and transmitted by the multicast sources every 500ms time interval (2 packets per second). The simulation was run for a scenario with 3 multicast sources and 30 multicast destinations in all experiments. We assume that any destination can receive data from any source node and the bandwidth requested by the sources are the same). The QoS constraint we concerned in the simulation is the bandwidth and the delay with the bandwidth requirements are set to 0.1, 0.2, and 0.4Mb/s. Each run is simulated 15 times with different seed numbers and collected data is the average of those runs. The multicast members are selected randomly and join the multicast session at the start of the simulation and remain as members through the simulation. The simulation parameters are given in table 1.

Table 1. Simulation parameters

Parameter Description	Value
Area size	3000m × 3000m
Number of nodes	100/km^2
Simulation duration	900s
Wireless transmission range	250m
Mobility Model	Random Way Point
Traffic type	CBR
MAC	IEEE 802.11
Maximum Channel bandwidth	2 Mb/s
Bandwidth Requirement	0.1, 0.2, 0.4
Mobility speed	0-60m/h
Periodic election time	20s
Ack timeout	25 millisecond
Backup time	200s
Membership duration	900s
Distance from border	30m
Position update notification	10m

To assess the performance of the proposed protocol the following metrics are considered:

Packet Delivery Ratio (PDR): defined as the relation between the number of data packets delivered to the destinations over the number of data packets supposed to be delivered to the destinations.

Control Overhead Ratio (COR): defined as the ratio of number of control packets transmitted per number of data packet delivered.

B. Simulation Analysis

In the simulation, we have studied the packet delivery ratio under various mobility scenarios as can be seen in Figure 5. The figure shows that the packet delivery ratio is very sensitive to mobility and as the node mobility increases the delivery ratio decreases. This is because the fast movement of the network nodes increases the possibility of route failure, which leads to higher packets drop out. Also, the increase of participating nodes will increase the possibility of forwarding the data to the destinations instead of being dropped, which increase the PDR.

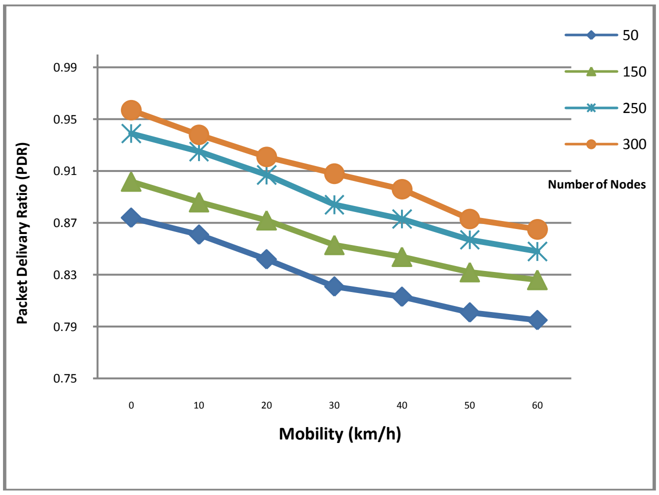

Figure 5: Packet Delivery Ration versus Mobility

Figure 6 shows the total number of control packets under different mobility scenarios for different network populations. As expected, the control overhead increases with increasing the node movement. This is due to the increase in the number of control packets needed to be transmitted to keep the multicast tree connected and preserve the construction of the network. Also, the management of nodes movement between the cells and handover with other neighbor cells will produce extra overhead. This overhead will be simple-minded in order to have large network scale with large number of multicast members. Although, the backup mechanism increases the control overhead, this overhead is expected to be worse in case of leader failure and losing the cell information.

While, it may appear that our protocol suffers from large control overhead. However, our protocol has smaller control packets in size and in a limited neighbor (maximum 3 hops). Furthermore, our protocol reduces the number of packets transmitted during the location service and route initiation phases to a minimum; this is since the search is only done towards the intended destination nodes and requests which do not satisfy the required QoS constraints are dropped. From the simulation results, we noted that the resulted overhead is acceptable compared with the amount of data packets transmitted. Finally, it’s clear from figure 6 that increasing the number of nodes found in the network will increase the number of nodes participating in sending and forwarding packets in different phases; hence increasing the control overhead.

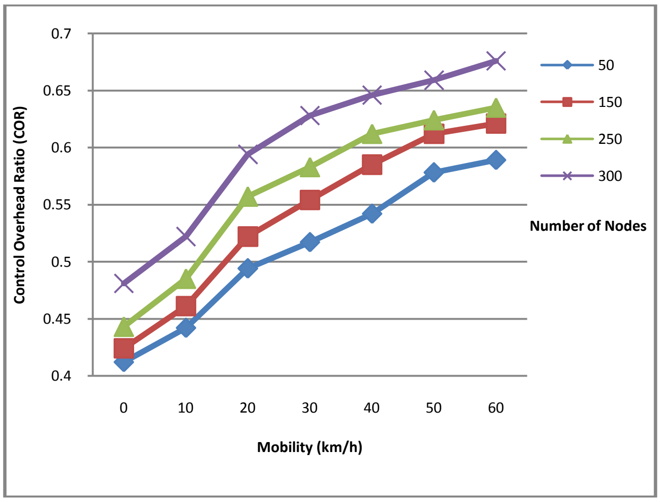

Figure 6: Control Overhead versus Mobility

X. CONCLUSION

This paper proposes a new position-based multicast routing protocol for MANETs with multiple QoS constraints. The used hierarchical scheme is optimized to utilize the limited network resources. Moreover, the distributed admission control mechanism exploits the residual bandwidth efficiently. This approach is efficient in providing QoS capability with significant reduction in control, storage and processing overhead. Also, it is scalable for large area networks with large number of multicast members. The simulation results reflect the efficiency of our protocol in providing QoS multicast routing with low control overhead.

Nevertheless, more simulation and analysis works are still required in the future to study the performance of the protocol for various system operating environment and different application requirements.

REFERENCES

[1] K. Bu¨r and C. Ersoy, "Ad hoc quality of service multicast routing," Elsevier Computer Communications, vol. 29, no. 1, pp. 136-148, December 2005.

[2] J. A. Sanchez, P. M. Ruiz, J. Liu, and I. Stojmenovic, "Bandwidth-Efficient Geographic Mutlicast Routing Protocol for Wireless Sensor Networks," IEEE Sensors Journal, vol. 7, no. 5, pp. 627-636, May 2007.

[3] Y. Ko and N. Vaidya, "Geocasting in Mobile Ad Hoc Networks: Location-Based Multicast Algorithms," in Proc. of IEEE WMCSA, 1999.

[4] I. Stojmenovic, "Position based routing in ad hoc networks," IEEE Commmunications Magazine, vol. 40, no. 7, pp. 128-134, July 2002.

[5] H. Tebbe and A. Kassler, "QAMNet: Providing Quality of Service to Ad hoc Multicast Enabled Networks," in 1st International Symposium on Wireless Pervasive Computing (ISWPC), Phuket, Thailand, 2006.

[6] Y.-S. Chen and Y.-W. Ko, "A Lantern-Tree Based QoS on Demand Multicast Protocol for A wireless Ad hoc Networks," IEICE Transaction on Communications, vol. E87-B, pp. 717-726, 2004.

[7] M. Saghir, T. C. Wan, and R. Budiarto, "Load Balancing QoS Multicast Routing Protocol in Mobile Ad Hoc Networks AINTEC, Bangkok, Thailand, Lecture Notes in Computer Science, Ed. K. Cho, P. Jacquet," Bangkok, Thailand, Lecture Notes in Computer Science, Ed. K. Cho, P. Jacquet, Springer-Verlag, vol. 3837, pp. 83-97, AINTEC 2005.

[8] D. Promkotwong and O. Sornil, "A Mesh-Based QoS Aware Multicast Routing Protocol," Springer – Verlag Berlin / Heidelberg, vol. 4658/, pp. 466-475, 2007.

[9] H. Wu and X. Jia, "QoS multicast routing by using multiple paths/trees in wireless ad hoc networks," Elsevier, Ad Hoc Networks , vol. 5, p. 600–612, 2007.

[10] A. Mizumoto, H. Yamaguchi, and K. Taniguchi, "Cost-Conscious Geographic Multicast on MANET," In proc. IEEE SECON, Santa Clara, CA, USA., pp. 44-53, 2004.

[11] H. Fuessler, J. Widmer, M. Mauve, and W. Effelsberg, "A hierarchical approach to position-based multicast for mobile ad-hoc networks ," Springer, Wireless Networks, vol. 13, no. 4, pp. 447-460, August 2007.

[12] S. Basagni, I. Chlamtac, and V. R. Syrotiuk, "Location aware, dependable multicast for mobile ad hoc networks," Elsevier Computer Networks, vol. 36, no. 5-6, pp. 659-670, August 2001.

[13] K. Chen and K. Nahrstedt, "Effective location-guided overlay multicast in mobile ad hoc networks ," International Journal of Wireless and Mobile Computing, vol. 3, no. Special Issue on Group Communications, 2005.

[14] M. Mauve, H. Fuessler, J. Widmer, and T. Lang., "“Position–based multicast routing for mobile ad–hoc networks," Department of Computer Science, University of Mannheim, Technical Report TR-03-004 2003.

[15] G. Wang, J. Cao, L. Zhang, K. Chan, and J. Wu, "A Novel QoS Multicast Model in Mobile Ad Hoc Networks," Proceedings. 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS'05), vol. 4, no. 8, pp. 206b-206b, April 2005.

[16] T.-F. Shih, C.-C. Shih, and C.-L. Chen, "Location-Based Multicast Routing Protocol for Mobile Ad Hoc Networks," WSEAS Transactions on Computers, vol. 7, no. 8, pp. 1270-1279, August 2008.

[17] S.-J. Le, W. Su, and M. Gerla, "On-Demand Multicast Routing Protocol in Multihop Wireless Mobile Networks," Mobile Networks and Applications, vol. 7, no. 6, pp. 441-453, December 2002.

[18] C. C. Shih and T. F. Shih, "Cluster-Based Multicast Routing Protocol for MANET," WSEAS Transactions on Computers, vol. 6, no. 3, pp. 566-572, March 2007.

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

[20] H. Frey and D. Go¨rgen, "Geographical Cluster-Based Routing in Sensing-Covered Networks," IEEE Transactions on Parallel and Distributed Systems, vol. 17, no. 8, September 2006.

[21] H. W. I-Shyan, C. Chih-Kang, and W. Chiung-Ying, "A Novel GPS-Based Quorum Hybrid Routing Algorithm," Journal of Information Science and Engineering, no. 21, pp. 1-21, 2005.

[22] C. Y. Chang and C. T. Chang, "Hierarchical cellular-based management for mobile hosts in ad hoc networks," Computer Communication Journal, vol. 24, pp. 1554-1567, 2001.

[23] M. M. Rahman, M. Abdullah-Al-Wadud, and O. 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, February 2008.

[24] L. Chen and W. 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, March 2005.

[25] Y. Yang and R. Kravets, "Contention-Aware Admission Control for Ad Hoc Networks," IEEE Transactions on Mobile Computing, vol. 4, no. 4, pp. 363-377, July/August 2005.

[26] R. Bagrodia et al., "“Parsec: A Parallel Simulation Environment for Complex Systems"," IEEE Computer, vol. 31, no. 10, pp. pp. 77-85, October 1998.

APPENDIX A

A.1 Code executed at source node

```
While (true)
   { /*start the location service process*/
        If (current node has data to be sent (source node)) && (no route exist)   then
         {  Mem_Needed= = true
            Initialize out_dest_agg_time
            If (current node is CL node)   then
               { Send Outcell_Inv_Req packet to 6-neighbor CLs
            Else
                Send Incell_Inv_Req packet to CL node of the current cell
               }
          }
       /*start the search for QoS path */
        If (out_dest_agg_time is elapsed) && (Mem_Needed = = true)    then
           {  Mem_Needed = = false
              Complete = false
              Group_partitioning()
              Restricted_Directional_Flooding_S_Co (QoS-RREQ)
              Route_Needed_S_C = true
              Initialize QRep_wait_time
            }
       /*when the source receive the reply for QoS request*/
       If (current node is Source)    then
           {    If (QRep_wait_time is elapsed) && (Route_Needed_S_C = = true)     then
                  {  Start data transmission to all coordinators that send the QoS-RREP reply packet
                     Route_Needed_S_C = = true
                     For all Coordinators that did not send QoS-RREP packet
                       {
                         Consider the coordinator as died and remove it from the set of coordinators
                         Choose new coordinator node for the same sub-group.
                         Restricted_Directional_Flooding_S_Co (QoS-RREQ)
                         Route_Needed_S_C = = true
                         Initialize QRep_wait_time
                       }
                   }
            }
```

A.2 Restricted directional flooding (RDF)

```
/*code executed to forward the packet restrictedly to neighbor cells*/
 {   Record distance between itself and destination (neighbor CL)(current_node_dest_distance)
     Broadcast the packet
     While (current node is not the destination)
          If (current_node_dest_distance < sender_node_dest_distance)     then
               Broadcast the packet
        Else
               Discard the packet }
```

A.3 Code Executed to Handle the QoS_RREQ Request

```
/*code executed to handle the QoS_RREQ packet sent from the source node*/
Handle_QoS_RREQ()
{  If(current node is coordinator)   then
       { If (packet not duplicate)    then
           {  Route_Needed_S_C = true
              While (Sub_group_list  is not empty)
                 {  Restricted_Directional_Flooding_Co_dest(Sub_QoS_RREQ)
                    Route_Needed_C_D = true   }
            }
         If (Route_Needed_S_C = = true)    then
              Check_Route_C()
         }
   Else
        Restricted_Directional_Flooding_S_C(QoS_RREQ)
 }
```

A. 4 Route discovery between source and coordinators

```
/*function to forward the QoS-RREQ packet from the source to the coordinator using RDF */
Restricted_Directional_Flooding_S_C(packet)
  {  If (complete = 0)     then
          Forward the packet to next hop until it reaches the coordinator;
     Bandwidth_calc(current_node)
     If (no Node_Av_BW )   or (delay_bound <=1)     then
           Discard QoS-RREQ packet
     Else
          {   DL_bound = DL_bound – 1;
              If  (DL_bound >0) && (current node is not the destined coordinator)
                 {   Calculate distance between current node and the destined coordinator (current_Co_dist)
                     If (distance between current_Co_dist  >  prev_hop_Co_distance)      then
                            Discard Packet
                     Else
                       {  Record Node_Av_BW to QoS-RREQ packet.
                          Let link_Av_BW = Min{ Node_Av _BW(prev_hop), Node_Av _BW (current node)}
                          If (link_Av_BW <= BW_Required )    then
                                 Let BW_Status = allocate
                          Else
                               Allocate the value of BW_Required and free the rest of the Node_Av _BW(Node_alloc_BW)
                          Add current node to the Route field in the QoS-RREQ packet
                          If (QoS-RREQ received before)    then          /*with a (Source_ID, ReqS_Seq_No) */
                               { complete = 0
                                 Send Route field to the destined coordinator to be used in setting up the route
                       } /*else*/
                 }/*if*/
          }/*else*/
     }
```