ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

AN EFFICIENT SELF-MAPPING ALGORITHM FOR HEXAGON-BASED GRIDDING IN AD HOC NETWORKS

[1]MOHAMMAD M. QABAJEH, [2]LIANA K. QABAJEH

[1]Faculty of Engineering and Technology, Palestine Technical University Kadoorie, Palestine

[2]Faculty of Information Technology and Computer Engineering, Palestine Polytechnic University, Palestine

E-mail: [1]mohammad.qabajeh@ptuk.edu.ps, [2]liana_tamimi@ppu.edu

ABSTRACT

Mobile Ad hoc NETwork (MANET) is a multi-hop autonomous network formed exclusively among a collection of mobile nodes without any centralized infrastructure. MANETs are generally unstable; the alternative for this non-infrastructure topology is to construct a virtual infrastructure. Using virtual clustering helps in creating an infrastructure for MANET to simplify routing and network management. Network terrain may be portioned into numerous shapes to support scalable routing. Hexagon-based gridding outperforms other gridding shapes; due to its geometric features. However, cell coordinate assignment and mapping node physical location into grid map is an important aspect. In this paper, we introduce a self-mapping algorithm to enable each mobile node to be aware of the precise cell it belongs to during the network lifetime without the need to communicate with other nodes. This algorithm is a core part of the position-based unicast and multicast routing protocols that rely on virtual hexagon infrastructure. Our algorithm has been developed to simplify routing discovery in large-scale MANETs and to ensure that the overhead is as low as possible.

Keywords: *MANETs, Position-based, Routing, GPS, Self-mapping*, *Multicast Routing, Unicast Routing*

1. INTRODUCTION

Mobile Ad hoc Networks (MANET) are collection of mobile nodes connected in a multi-hop manner by shared wireless links, forming a temporary network without infrastructure or central administration. The most important features of MANETs include self-organizing, self-configuring, self-administering, mobility and limited resources [1]. These characteristics coupled with the characteristics of the wireless medium make routing process one of the major issues to consider in MANETs.

Recently, there is increasing demand for large-scale wireless networks due the emergence of applications in large-area networks with large number of mobile nodes. This claim is realistic, due to the rapid advancements in mobile devices along with the evolution made in wireless communication technologies. As a consequence, scalable routing in MANETs has received significant attention over the recent years [1, 2]. However, routing in large-scale networks faces several difficulties and challenges that need to be addressed. These challenges include dynamic MANET topology, multicast packet forwarding and shared wireless medium.

MANETs have no physical infrastructure or backbone, an alternative for a physical backbone is the construction of a virtual backbone or infrastructure. One way to create infrastructure in MANETs is to perform node clustering. In our approach, we assume that mobile nodes can find their own locations, using Global Positioning System (GPS) receivers. This requirement is quite reasonable today since these location receivers are available, small in size, inexpensive, low-power and can support acceptable precision indoor and outdoor [2, 3].

The idea of using cluster routing in MANET is not new. Previous efforts in this issue appeared in [4-8]. Using clustering improves the performance in dense networks and reduces the number of nodes involved in routing management, thereby, reduces the exchange of routing information. This makes clustering more suitable for communication among nodes in large scale networks. Our approach is based on the same general idea. However, we optimize building such architecture to create a stable and simple network structure. Using such topology helps in network management and provides efficient routing, which will effectively reduce the routing overhead.

In hierarchical architecture models that rely on virtual partitions, the network topology is arbitrary divided into a number of non-overlapping virtual cells of any shape (square, triangle and hexagon). Using repetitive regular cell structures, the geometric properties for that shape can be utilized to enhance routing protocols. Typically, in hierarchical routing, network construction and handling nodes movement require notable control overhead, which affects the routing protocol performance. Additionally, MANET structure changes frequently due to node mobility and the node needs to know precise information about the cell it currently belongs to. However, research has shown that using hierarchical routing improves the performance in large scale networks [9].

Hexagonal gridding is the most efficient structure for network partitioning and it has been used in several previous works such as [10-12]. However, establishing a cell coordinate system and node location management for hexagonal gridding is an important issue that needs to be solved. As of this writing and to the best of our knowledge, there is no appropriate published self-mapping algorithm and cell coordinate assignment system for hexagonal gridding in MANETs. Thus, the contributions of this paper include:

a) Creating a cell coordinate system for the hexagonal grids to assign a unique address for each cell (Cell_ID).

b) Proposing a self-mapping algorithm that enables each mobile node to know its position, i.e., to compute exactly the cell it belongs to.

This rest of this paper is organized as follows. Section 2 briefly describes some previous efforts. In section 3, we provide an overview about our Position-Based QoS Multicast Routing Protocol (PBQMRP) that has been proposed in [13], which used this hexagonal algorithm. In this section, we also discuss the reasons behind selecting the hexagon grid shape, our proposed coordinates system and the self-mapping algorithm. Brief summary and concluding remarks are given in sections 4 and 5 respectively.

2. RELATED WORKS

Recently, location-based multicast routing protocols have attracted the attention of many researchers because these protocols scale quite well in large wireless networks in addition to the commercial proliferation of GPS devices. However, location-based unicast routing and location-based multicast routing face many challenges. An important problem to be solved is mapping the gridding to manage the routing mechanism, especially when hexagonal gridding is used. In hexagonal gridding, it is not an easy task to determine the Cell_ID based on the coordinates of a node.

Several position-based routing protocol [5][6][7] are using square gridding. However, hexagonal grids have several advantages over the square models. Many researchers use hexagonal grids to design their routing protocols. In this type of gridding, it is essential to develop efficient coordinate system and then solve the node location problem based on the proposed coordinate system.

In [14] a Fuzzy Location Service (FLS) algorithm has been proposed. In FLS the network plane is divided into hexagon cells, thus, the entire network can be covered with fewer hexagon cells. The division takes advantage of the node transmission radius since it is more close to a circle than a square or triangular grid. The center location of a cell is regarded as the fuzzy location of nodes in the cell. Assuming that a node's physical position is (x, y), FLS can map this position into the center location of the cell to which it belongs.

In [15], a two-dimensional logical coordinate system for hexagonal grids has been proposed, this approach maps the hexagonal grids to traditional rectangular grids based on the proposed coordinate system. Given the two-dimensional coordinate of any node in MANETs, the hexagonal grid where the node resides can be computed by the proposed approach. The location of the hexagon center can be computed, which is significant for location-based routing protocols in MANETs.

In [16], a coordinate system has been proposed. We can obtain the hexagon ID of another hexagonal grid when the hexagon ID of one hexagonal grid is given. But, given the coordinate of one node, it is difficult to know the coordinate of the center of the hexagon where the node resides. Also, given the physical location of one node, we cannot compute the Cell_ID of the hexagonal grid where the node resides.

3. THE PROPOSED ALGORITHM

In the subsequent section, a brief introduction about the multicast routing protocol that used this self-mapping algorithm is introduced. In subsection 2, the reasons behind selecting

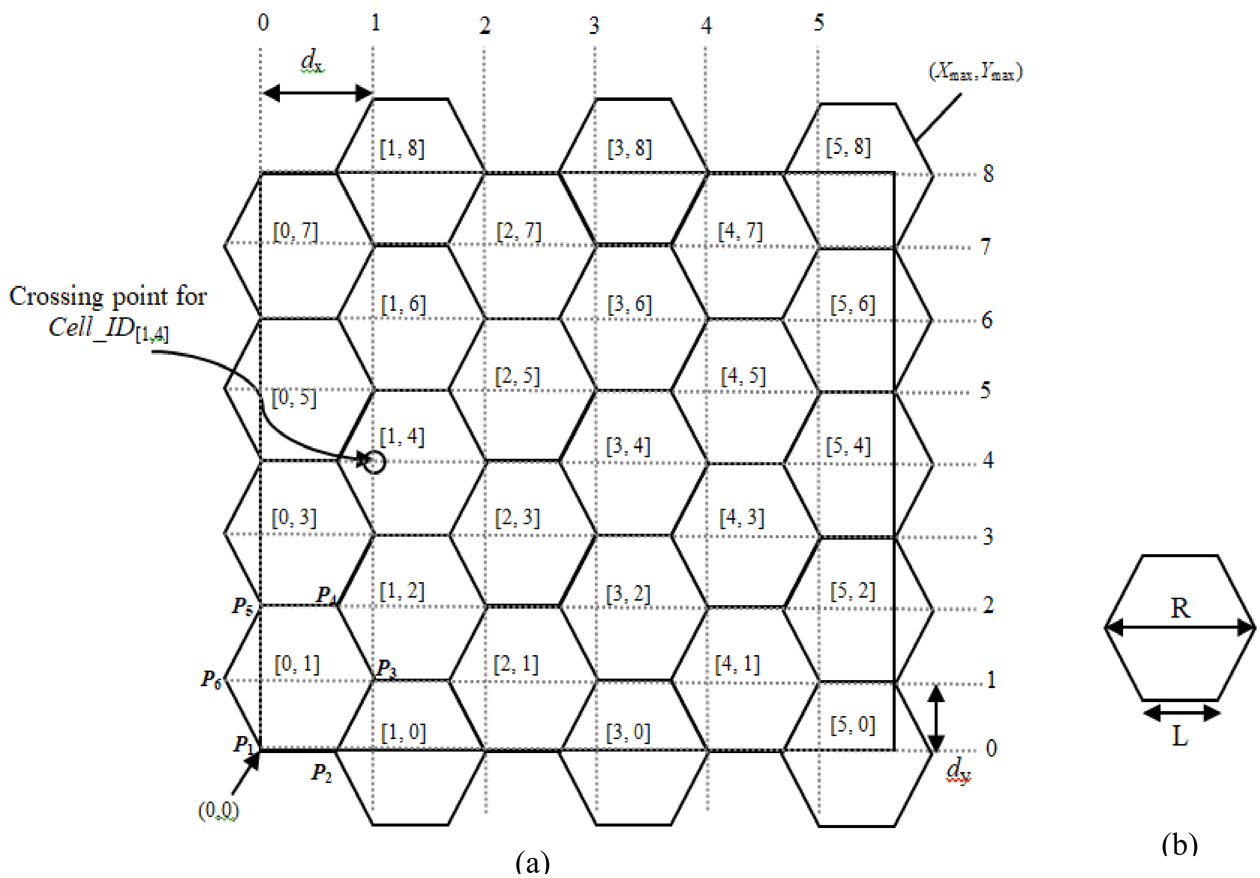
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

hexagonal gridding shape is provided. The proposed coordinates system of the hexagon grids is presented in subsection 3. In subsection 4, we discuss the details of the self-mapping algorithm.

3.1 Introduction to PBQMRP

In PBQMRP, the entire network area is partitioned into an arbitrary number of virtual cells with hexagonal shape. The cell size is chosen to enable 1-hop communication among all the nodes inside a particular cell.

In this protocol, after dividing the entire network topology into a number of cells (refer to Fig. 1(a)), an election algorithm is executed concurrently in all cells. This election is executed to elect the most eligible nodes (CL nodes) in order to take role as leaders and survive the longest possible time. This will keep the network construction as stable as possible. Each CL node would maintain information about all the nodes in its cell (the current location, the multicast groups they are interested to join) until they join a new cell. Other responsibilities of the CL node include the assistance in performing the location service algorithm and managing the joining process of new members to the multicast session.



(a) The entire network topology and cells' identities *(b) The relationship between L and R*

Fig. 1: MANET partitioning into hexagonal cells

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

When a source node wishes to send data packets to a particular multicast group, location service algorithm is initiated by utilizing the cluster structure to gather information about the subscribing nodes and provide the source node with this information. The packet is forwarded between non-reachable leaders using Restricted Directional Flooding (RDF). In RDF [17], the node resends the packet to its neighbors only if it is closer to the destination than its previous hop.

After execution of the location service algorithm, the source divides the multicast group members into manageable subgroups. In each of these subgroups, one of the group members is selected to be a coordinator. Then, route discovery and maintenance phase is initiated. In this phase, the source and the coordinators co-operate to search for routes to all destinations. Once the route discovery procedure is completed, data transmission phase takes place by sending data packets from the source to the intended destinations.

The performance of PBQMRP protocol has been assessed through both simulation and analytical approaches. Part of the results can be found in [13]. In the analytical approach, the upper bound of the control packets generated in each phase of the protocol is estimated. The analysis shows that using the hexagonal shape effectively reduces the control overhead of the cluster formation compared to other gridding shapes. The analysis also shows the efficiency of the proposed location service algorithm in reducing the control overhead and eliminating flooding the network. In the simulation approach, the GloMoSim [18] simulator has been used to implement and evaluate the performance of the proposed protocol. The simulation results show clearly that PBQMRP achieves a comparable normalized packet overhead and increased throughput and packet delivery ratio compared to the standard ODMRP.

3.2 Reasons behind Considering Hexagonal Gridding

The most popular gridding shapes are triangular, square and hexagon shapes since they cover the entire network without overlapping. We have considered hexagonal gridding in our protocol. The reasons behind this selection are summarized as follows:

- Considering the hexagonal, square and triangle cell shapes, and assuming the transmission range (R) as the maximum distance among the cell nodes, the relation between the side length (L) and (R) that satisfy this assumption is declared in Fig. 2. From this figure, 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 ($0.5\times R^2$). So, the hexagonal cell shape covers more area in a single transmission, which will increase the number of nodes that are affiliated with the leader of each cell. Also, larger cell area means less number of cell leaders; i.e., less cluster overhead maintenance. This would reduce the communication overhead and improve the propagation delay in performing location discovery.
- The hexagonal cell offers six directions of transmission with the same distance between the centers of the neighbor cells. Thus, a packet sent from cell leader node is propagated, roughly at the same speed in all directions. On the other hand, square and triangle shapes have larger number of neighbors (8 for square shape and 12 for triangle shape), but the distances between the centers of the neighbor cells are different (as shown in Fig. 1).

From Fig.2, it can be seen that the hexagon cell consists of 6 triangles.

Area of each triangle =

$$\frac{1}{2}\times L\times H = \frac{1}{2}\times L\times\sqrt{(L^2-(\frac{1}{2}L)^2)} = \frac{1}{2}\times L\times\frac{\sqrt{3}}{2}\times L = \frac{\sqrt{3}}{4}\times L$$

Hexagon coverage area =

$$\frac{\sqrt{3}}{4}\times L^2\times 6 = \frac{3\times\sqrt{3}}{2}\times L^2 = \frac{3\times\sqrt{3}}{2}\times\frac{R^2}{4} = 0.6495\times R^2$$

Triangle coverage area =

$$\frac{1}{2}\times L\times H = \frac{1}{2}\times L\times\sqrt{(L^2-(\frac{1}{2}L)^2)} = \frac{1}{2}\times L\times\frac{\sqrt{3}}{2}\times L$$

$$= \frac{\sqrt{3}}{4}\times L^2 = 0.433\times R^2$$

Square coverage area =

$$L^2 = \frac{R^2}{2} = 0.5\times R^2$$

where *H* is the height each triangle in the hexagonal cell.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

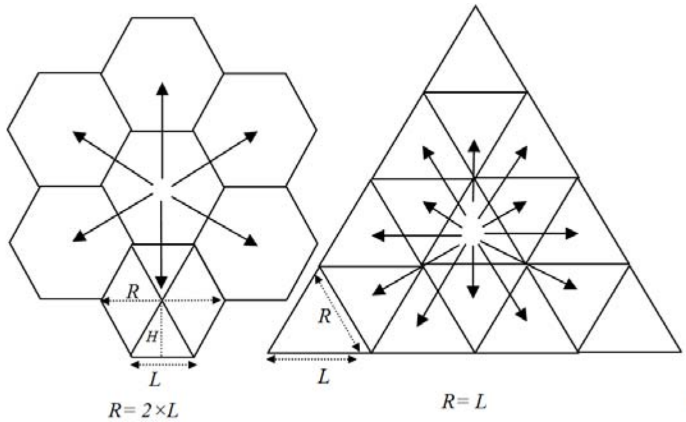


Figure 2: Comparison among square, hexagon and triangle cell shapes.

3.3 Determine Cell Identity and Cell Coordinates

In this algorithm, we denote R and L as the effective transmission range of each mobile node and the side length of hexagonal cell respectively. As a starting point, 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 (refer to Fig. 1(b)). The reason behind this assumption is to compromise between the overhead of network maintenance (overhead inside each cell) and the location service overhead (overhead between cells). This is essential when the network is large and the number of nodes is high. In general, 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 cell leaders during the 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. In case of high mobility, the location update packets will increase and probability of the boundary crossing will increase as well. So, using large cell size reduces the overhead of triggering this kind of packets. The following paragraphs discuss the cell identity and cell coordinate determining strategy.

Initially, the network area is defined by the coordinates (0, 0) and (X_{max}, Y_{max}). The partitioning is carried out starting from the bottom-left point of the routing area (0, 0) and spreading outward until completely covering the entire network. Fig. 1(a) shows the network entire topology and illustrates the identity of each cell. We assume that the network coordinates are obtained by the mobile nodes during network startup.

Each hexagonal cell has six coordinates denoted as P1, P2, P3, P4, P5 and P6. Coordinate P1 of the first cell is placed on the most bottom-left point of the routing area in order to minimize the number of cells (thus reducing the number of cells' leaders and the communication overhead associated with them). Each hexagonal cell inside the network has a unique identifier (*Cell_ID*). Since the cells are not arranged in a uniform pattern, the doted vertical and horizontal lines are introduced to the plan in Fig. 1(a) to help in explaining how each cell obtains its identifier.

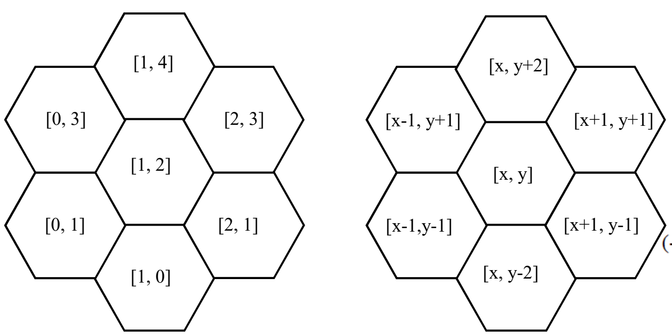
The distance between two successive points in the X-axis is set to d_x; for the Y-axis it is set to d_y. The values of d_x and d_y are defined in equation (1).

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$d_x = \frac{3 \times R}{4}$$

$$d_y = \sqrt{\left(\frac{R}{2}\right)^2 - \left(\frac{R}{4}\right)^2} = \frac{\sqrt{3 \times R^2}}{4} = \frac{\sqrt{3} \times R}{4} \qquad (1)$$

Therefore, each cell takes its *Cell_ID* from the two virtual perpendicular lines crossing each other inside that cell. Having the *Cell_ID* of a particular cell known, the identifiers of the six neighbor cells will be easily determined. The relation between the cell and its neighbor cells is shown in Fig. 3. The *Cell_ID* of the six neighbor cells of *Cell_ID*[1,2] are shown in Fig. 3(a) as an example, while Fig. 3(b) shows the general equations for any cell with *Cell_ID*[x,y]. This relation is utilized in the communications between the *CL* of a cell and its six neighbor cells.



(a) Cell_ID of the neighboring cells for Cell_ID[1,2].

(b) Relationship between Cell_IDs of the neighboring cells.

Figure 3: Illustration of the relationship between a cell and its neighbors.

Now, there is a need to specify the coordinates of the six corners of each cell in the entire network. This is important to detect the case when a node crosses the cell border and enters a neighboring cell. As shown in Fig. 4, a numerical example is given to explain the algorithm used to compute the six coners coordinates for a particular cell. In this example, the coordinates for the six corners of *Cell_ID*[0,1] are calculated, and accordingly a general equation for the other cells is concluded. For simplicity, let the transmission range be set to (R = 240*m*), then the side length is equal to (L = 120m). Therefor, the values of d_x and d_y are:

d_x= 120 + 60 =180

$$d_y = \sqrt{(120)^2 - (60)^2} = 103.9 \cong 104$$

Hence, the coordinates of the first cell are defined as:

$P_{1[0,1]}$= (0, 0)

$P_{2[0,1]}$= (120, 0)

$P_{3[0,1]}$= (180, 104)

$P_{4[0,1]}$= (120, 208)

$P_{5[0,1]}$= (0, 208)

$P_{6[0,1]}$= (-60, 104)

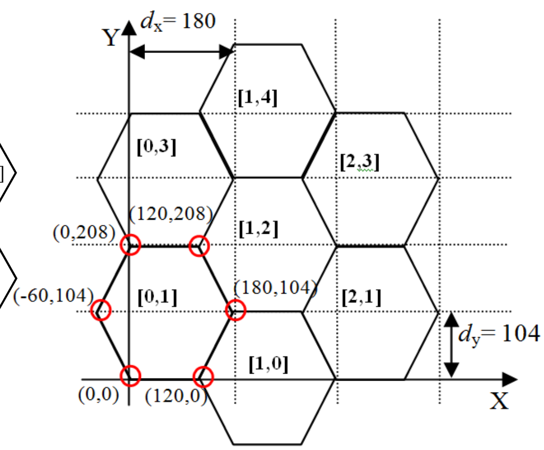


Figure 4: Illustration of corners coordinates of each cell.

According to our example, the coordinates for the six corners of *Cell_ID*[*i*,*j*] based on the value of R = 240*m* are formulated as:

$P_{1[i,j]} = (180\times i,\ 104\times (j - 1))$

$P_{2[i,j]} = (120 + (180\times i),\ 104\times (j - 1))$

$P_{3[i,j]} = (180\times (i + 1),\ 104\times j)$

$P_{4[i,j]} = (120 + (180\times i),\ 104\times (j + 1))$

$P_{5[i,j]} = (180\times i,\ 104\times (j + 1))$

$P_{6[i,j]} = (-60 + (180\times i),\ 104 \times j)$

In general, given the coordinate of a cell as *Cell_ID*[*i*,*j*] and the transmission range R, the coordinates for the six corners ($P_{1[i,j]}$ to $P_{6[i,j]}$) of that cell can be directly computed using the equations (2 to 7) below.

$P_{1[i,j]} = ((d_x \times i, d_y \times (j - 1))$ (2)

$P_{2[i,j]} = (\frac{R}{2}) + d_x \times i, d_y \times (j - 1))$ (3)

$P_{3[i,j]} = (d_x \times (i + 1), \; d_y \times j)$ (4)

$P_{4[i,j]} = ((\frac{R}{2}) + d_x \times i, d_y \times (j + 1))$ (5)

$P_{5[i,j]} = (d_x \times i, d_y \times (j + 1))$ (6)

$P_{6[i,j]} = (-\frac{R}{4} + d_x \times i, d_y \times j)$ (7)

3.4 Nodes Self-Mapping

The borders of the routing area are assumed to be known (i.e., nodes move around in a fixed region). This is applicable for many applications in MANETs. Such applications include soldiers in military battlefields, disaster relief scenarios, conferences and public events. Thus, for a given cell size A, the coordinates of each cell is also assumed to be fixed. As a result, given the location of a node, it is possible map the node location to the virtual cell it belongs to.

In this algorithm, node self-mapping is based on determining the node current position according to the slope of the line between (P2 and P3) or (P3 and P4) for the current cell of the node. The following two examples explain the different cases that can occur when the node computes the Cell_ID of the cell it belongs to. Then a general formula is derived for performing the node self-mapping.

Given the position of a particular node N1 as (xn, yn) = (181, 100) as shown in Fig. 5. Certainly this node is residing in the rectangle with corners [i, j], [i, j+1], [i+1, j], [i+1, j+1] (shown in bold lines), where i and j repesent the sequance number of the virtual doted vertical and horizontal lines and their values are computed based on equation 8 and 9 respectively. So, the Cell_ID of N1 would be [1,0] or [2,1]. Here, the temporary value of (i) and (j) are computed as:

$$i = \mathrm{int}(\frac{x_n}{d_x}) \quad (8)$$

$$j = \mathrm{int}(\frac{y_n}{d_y}) \quad (9)$$

$$i = \mathrm{int}(\frac{x_n}{\frac{3 \times R}{4}}) = \mathrm{int}(\frac{181}{180}) = 1 \text{ and}$$

$$j = \mathrm{int}(\frac{y_n}{\frac{\sqrt{3} \times R}{4}}) = \mathrm{int}(\frac{100}{104}) = 0$$

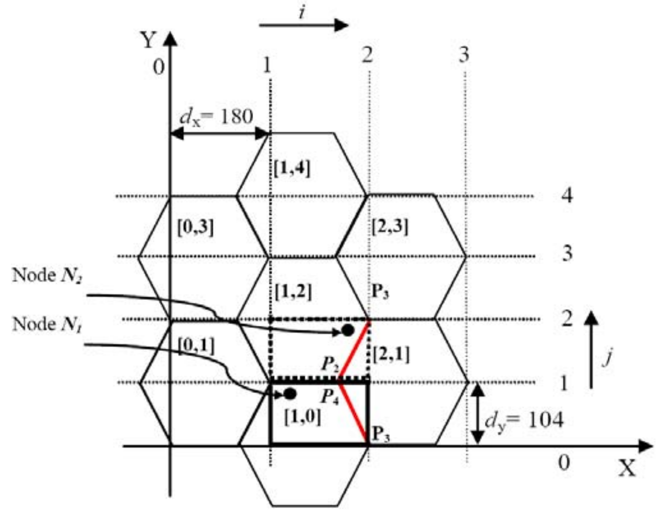


Figure 5: Illustration of the node self-mapping algorithm.

Therefore, to determine the exact *Cell_ID* of node N_1, the node position with respect to the slope of the line between P_3 and P_4 (shown in Fig. 5) has to be determined, which is calculated according to previously mentioned equations (4) and (5) as:

$$P_{3[i,j]} = P_{3[1,0]} = (x_3, y_3) = (360, 0) \text{ and}$$

$$P_{4[i,j]} = P_{4[1,0]} = (x_4, y_4) = (300, 104)$$

The slope (m) and the coefficient (b) for that line are computed as follows:

$$line(x) = m \times x + b$$

$$m = \frac{y_4 - y_3}{x_4 - x_3} = \frac{104 - 0}{300 - 360} = -1.73$$

$$b = y_4 - m \times x_4 = 104 - (-1.73 \times 300) = 623$$

Thus, $line(x) = -1.73 \times x + 623$

Now, in the obtained equation, x coordinate is substituted with x_n of node N_1 to determine whether the node is above or below the line.

$$line(x_n) = m \times x_n + b = -1.73 \times 81 + 623 = 309.87$$

If $line(x_n) > y_n$, then the current cell of node N_1 is (1, 0). Otherwise, the current cell of node N_1 is [1 + 1, 0 + 1]. In this example 309.87 > 100,

Journal of Theoretical and Applied Information Technology
15th September 2020. Vol.98. No 17

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

hence the current cell of node N_1 is [1, 0]. It could be noticed that in this example (i) is odd and (j) is even. When (i) is even and (j) is odd, the same mechanism is used to compute the *Cell_ID* of the cell that the node belongs to.

Considering another example, where it is supposed that the position of the node N_2 is (x_n, y_n) = (350, 200) as shown in Fig. 5. Certainly this node is residing in the rectangle (shown in dashed lines) with corners $[i, j]$, $[i, j+1]$, $[i+1, j]$, $[i+1, j+1]$ where:

$$i = \text{int}(\frac{x_n}{d_x}) = \text{int}(\frac{x_n}{\frac{3\times R}{4}}) = \text{int}(\frac{350}{180}) = 1$$

$$j = \text{int}(\frac{y_n}{d_y}) = \text{int}(\frac{y_n}{\frac{\sqrt{3}\times R}{4}}) = \text{int}(\frac{200}{104}) = 1$$

Since there is no cell with identifier [1, 1], so let j=j+1= 2. Then, to find the exact *Cell_ID* of node N_2, it is necessary to determine the node position with respect to the slope of the line between P_2 and P_3 (as shown in Fig. 5) which are calculated based on equations (3) and (4) as:

$$P_{2[i,j]} = P_{2[1,2]} = (x_2, y_2) = (300, 104) \text{ and}$$

$$P_{3[i,j]} = P_{3[1,2]} = (x_3, y_3) = (360, 208)$$

It is required to determine whether the node is above or below the line passing through $P_{2[1,2]}$ and $P_{3[1,2]}$. The slope (m) and the coefficient (b) are computed to be substituted in the general line equation $line(x) = m\times x + b$:

$$m = \frac{y_3 - y_2}{x_3 - x_2} = \frac{208-104}{360-300} = 1.73$$

$$b = y_3 - m\times x_3 = 208 - 1.73\times 360 = -414.8$$

Thus, $line(x) = 1.73\times x - 414.8$

Then, x coordinate is substituted with x_n of node N_2 to determine whether the node is above or below the line.

$$line(x_n) = m\times x_n + b = 1.73\times 350 - 414.8 = 190.7$$

If $line(x_n) < y_n$, then the current cell of node N_2 is [1, 2]. Otherwise, the current cell of node N_2 is [1 + 1, 2 – 1]. In this example 190.7 < 200, hence the current cell of node N_2 is [1, 2]. It is noticed that the original values of both i and j are odd. The same technique is used when the original values of i and j are both even. Considering the previously discussed two examples, the general algorithm that can be used to determine the current cell of a node N_x at position (x_n, y_n) is shown in Fig. 6.

5. MULTICAST ROUTING OVER THE HEXAGONAL STRUCTURE

We believe that the proposed virtual structure can be utilized to deliver the position query packets with very low overhead and non-duplicate packets. Position-based routing is proposed in MANETs to alleviate flooding. In this type of routing, the source needs to obtain the destination's current position through some location service. As the forwarding decisions are based on the local topology, this makes geographic routing exhibits higher scalability and robustness in a dynamic environment [11].

In our protocol (PBQMRP) that has been proposed in [13], after dividing the entire topology into a number of cells (refer to figure 2), an election algorithm is executed concurrently in all cells. This election is executed to elect the most eligible nodes (CL nodes) in order to take role as leaders and survive the longest possible time, which will keep the network construction as stable as possible. Each CL node would maintain information about all the nodes in its cell (the current location, the multicast groups they are interested to join) till they join new cell. Other responsibilities of the CL node include the assistance in performing the location service algorithm and managing the joining process of new members to the multicast session.

When a source node wishes to initiate a multicast session, it sends a query packet to its local CL node. This packet needs only one-hop unicast operation, because the location of the leader node is known for all the nodes inside the cell. If the source node is the leader of the cell, it does not need to initiate such packet. When the CL node of the same cell receives such a packet, it forwards the request to the CL nodes of the six neighbor cells. When the neighbor CLs are within the transmission range of the sending CL, they can communicate directly and the packet will reach them in only one-hop. Otherwise, the packet is forwarded using RDF towards the border of the neighbor cells. In RDF, the node resends the packet to its neighbors only if it is closer to the destination than its previous hop.

After contacting the neighbor cells, the CL of the original cell searches its Member_Table to see if there are nodes inside its cell that are interested in joining this multicast group. If as so, the CL node sends reply packet to the source

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

node directly. This reply packet contains the position and ID of the destination nodes found inside the cell.

Each neighbor CL, upon receiving such invitation packet for the first time, it checks it's Member_Table for possible participating nodes. If there is any, the CL node sends a reply packet using the reverse path until it reaches the CL node that issue the invitation request. The query packet is forwarded till it reaches all the network cells.

It is clear from the analysis that utilizing our technique results in reducing the overhead. Without using our technique, each node will broadcast the invitation request to all its neighbors, and each interested node will reply with a separate reply packet. Increasing number of packets sent will definitely results in increasing delays in processing the packets and dropping some of them. On the other hand, upon using our technique, packets are sent either using 1-hop unicast or RDF. Moreover, only the CL nodes bear the responsibility of forwarding the packets to their neighbors.

Self mapping Algorithm

Begin

Assume node position= (x_n, y_n)

Define $L = \frac{R}{2}$

Calculate $d_x = \frac{3R}{4}$ *and* $d_y = \frac{\sqrt{2}R}{4}$

Calculate $i = int(\frac{x_n}{d_x})$ *and* $j = int(\frac{y_n}{d_y})$

If ((i is odd) && (j is even)) || ((i is even) && (j is odd))

Six coordinates definition (Cell_ID[i,j])

*/*Consider the line passing through* $P_{3(i,j)} = (x_3, y_3)$ *and* $P_{4(i,j)} = (x_4, y_4)$**/*

Compute the slope m= , b= $y_4 - m \times x_4$ *, and* $line(x) = m \times x + b$

Substitute x_n *in the general line equation* $line(x_n) = m \times x_n + b$

If $line(x_n) > y_n$*, then*

the current cell of node N is [i, j]

else

the current cell of node N is [i+1, j+1]

If ((i is odd) && (j is odd)) || ((i is even) && (j is even))

Let j=j+1

Six coordinates definition (Cell_ID[i,j])

/ Consider the line passing through* $P_{2[i,j]} = (x_2, y_2)$ *and* $P_{3[i,j]} = (x_3, y_3)$**/*

Compute the slope $m = \frac{y3-y2}{x3-x2}$ *,* $b = y_3 - m \times x_3$ *and* $line(x) = m \times x + b$

Substitute x_n *in the general line equation* $line(x_n) = m \times x_n + b$

If $line(x_n) < y_n$*, then*

the current cell of node N is [i,j]

else

the current cell of node N is [i+1, j-1]

Six coordinates definition of Cell_ID[i,j]

Begin

$P_{1[i,j]} = ((d_X \times i, d_y \times (j-1))$

$P_{2[i,j]} = ((\frac{R}{2}) + d_X \times i, d_y \times (j-1))$

$P_{3[i,j]} = (d_X \times (i+1), d_y \times j)$

$P_{4[i,j]} = ((\frac{R}{2}) + d_X \times i, d_y \times (j+1))$

$P_{5[i,j]} = (d_X \times i, d_y \times (j+1))$

$P_{6[i,j]} = (-(\frac{R}{4}) + d_X \times i, d_y \times j)$

End

Figure 6: Self mapping algorithm pseudocode.

6. SUMMARY

As the need for group-oriented applications increases, multicast routing is getting attention as one of the critical issues. Meanwhile, there is a pressing need for large-scale Ad hoc networks in order to support applications with large number of nodes in large network area. This claim is reasonable, due to the evolution in mobile communications. As a consequence, scalable routing in MANETs has received significant attention over the recent years.

We believe that the proposed virtual structure can be utilized to enhance multicast routing over MANETs. In our research, the problem of scalability of multicast routing protocols over MANETs is investigated. In particular, a new position-based multicast routing protocol has been developed. The main objective of this protocol is to design a lightweight scalable QoS multicast routing scheme irrespective of the number of multicast members and network size.

The virtual architecture provides efficient management of dynamic movement of mobile nodes. Also, it is utilized to deliver the location service discovery packets between neighbor cells without duplicate packets. Details about these operations can be found in [13]. Hence, it is an important issue to propose a cell coordinate assignment and to enable nodes to map their physical locations into the exact identifier of the cell they belong to. The proposed self-mapping algorithm is executed on each node separately without causing extra overhead.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

7. CONCLUSIONS

In MANETs, the virtual infrastructure is used to facilitate network operations and to improve the efficiency and scalability of routing protocols. Hexagon-based gridding is presented as one of the gridding shapes since it outperforms other cell shapes. Using this structure, utilizes the network resources efficiently and ensures that the overhead is as low as possible. This helps the research community to propose new routing protocols for wireless mobile networks. However, the problem of location management for this type of topology was not solved in existing works. This paper has presented a coordinate system for the hexagonal grids. In addition, we proposed a self-mapping algorithm to enable mobile nodes to estimate the virtual cell they belong to without extra overhead.

ACKNOWLEDGMENT

The authors wish to thank Palestine Technical University Kadoorie, Palestine for their cooperation and support to publish this research

REFERENCES:

[1] F. D. Rosa, H. Leppakoski, A. Ghalib, L. Ghazanfari, O. Garcia, et al., "Ad Hoc Networks for Cooperative Mobile Positioning," in Mobile Ad-Hoc Networks: Applications, X. Wang, Ed., ed: InTech Publisher, 2011, pp. 289-304.

[2] I. Stojmenovic, D. Liu, and X. Jia, "A scalable quorum-based location service in ad hoc and sensor networks," International Journal of Communication Networks and Distributed Systems, vol. 1, pp. 71-94, 2008.

[3] H. Cheng, J. Cao, and X. Fan, "GMZRP: Geography-aided multicast zone routing protocol in mobile ad hoc networks," Mobile Networks and Applications, vol. 14, pp. 165-177, 2009.

[4] J. N. Al-Karaki and A. E. Kamal, "Efficient virtual-backbone routing in mobile ad hoc networks," Computer Networks, vol. 52, pp. 327-350, 2008.

[5] X. Xiang, X. Wang, and Y. Yang, "Supporting efficient and scalable multicasting over mobile ad hoc networks," IEEE Transactions on Mobile Computing, 2010.

[6] 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, pp. 1270-1279, 2008.

[7] W. H. Liao, J. P. Sheu, and Y. C. Tseng, "GRID: A fully location-aware routing protocol for mobile ad hoc networks," Telecommunication Systems, vol. 18, pp. 37-60, 2001.

[8] L. K. Qabajeh, M. L. M. Kiah, and M. M. Qabajeh, "A scalable and secure position-based routing protocol for ad-hoc networks," Malaysian J. Comput. Sci, vol. 22, pp. 99-120, 2009.

[9] K. Chen and K. Nahrstedt, "Effective location-guided overlay multicast in mobile ad hoc networks," International Journal of Pervasive Computing and Communications, vol. 5, pp. 388 - 410, 2009.

[10] H. Frey and G. Daniel, "Geographical cluster-based routing in sensing-covered networks," IEEE Transactions on Parallel and Distributed Systems, pp. 899-911, 2006.

[11] C. Chang, C. Chang, and T. Hsieh, "Hierarchical cellular-based management for mobile hosts in wireless ad hoc networks," Computer communications, vol. 24, pp. 1554-1567, 2001.

[12] I. S. Hwang, C. K. Chien, and C. Y. Wang, "A Novel GPS-Based Quorum Hybrid Routing Algorithm (GPS-QHRA) for Cellular-Based Ad Hoc Wireless Networks," Journal of information science and engineering, vol. 21, pp. 1-21, 2005.

[13] M. Qabajeh, A. Hashim, O. Khalifa, and L. Qabajeh, "A Cluster-based QoS Multicast Routing Protocol for Scalable MANETs," KSII Tansactions on Internet and Information Systems, vol. 5, pp. 741-762, Apr 2010.

[14] Y. B. Ko and N. H. Vaidya, "Location-aided routing (LAR) in mobile ad hoc networks," Wireless Networks, vol. 6, pp. 307-321, 2000.

[15] UCLA Parallel Computing Laboratory. (2011). Network Simulator (GloMoSim). Available: http://pcl.cs.ucla.edu/projects/glomosim/

[16] J. Li, A. Kubota, and H. Kameda, "Location management for PCS networks with consideration of mobility patterns," in Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Communications Societies., 2005, vol. 2, pp. 1207-1216: IEEE.

[17] Y. B. Ko and N. H. Vaidya, "Location-aided routing (LAR) in mobile ad hoc networks," Wireless Networks, vol. 6, no. 4, pp. 307-321, 2000.

[18] UCLA Parallel Computing Laboratory. (2011). Network Simulator (GloMoSim). Available: http://pcl.cs.ucla.edu/projects/glomosim/