IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Volume-6, Issue-2 E-ISSN: 2347-2693

DFID Time Complexity in Mobile Network

Justin Sophia. I[1*], N. Rama[2]

[1*]Department of Computer Science, Loyola College, Chennai ,India
[2]Department of Computer Science, Presidency College, Chennai ,India

*Corresponding Author: justinsophia4@gmail.com

Available online at: www.ijcseonline.org

Received: 29/Jan//2018, Revised: 08/Feb2018, Accepted: 20/Feb/2018, Published: 28/Feb/2018

***Abstract*—** Hexagonal network structures the base for mobile communication networks. To increase the efficiency of mobile communication, we implement the DFID algorithm for transmission. Let DFID denote the time taken by DFS and BFS preorder traversing to find the vertex x in graph DFID(x,G). In this section we discuss DFS and BFS for G beign a hexagonal mesh of dimension n i.e HXn..Thereby computing the time complexity of DFID algorithm for mobile networking.

Keywords— DFID, BFS, DFS, Hexagonal mesh, Time complexity.

I. INTRODUCTION

Several research activities are introduced to reduce the time for exchanging the node in Mobile Ad Hoc Network (MANET). Each node in a MANET works as both a host and router for transmission and receiving a data and also to maintain the information in the time of exchanging the nodes. MANET is one of the wireless communications and it didn't need any help from other devices, thus it can operate without the existing infrastructure and other mobile user.

The idea behind the MANET is based on multihop wireless networking [1]. MANET individually support for data forwarding and topology discover. The most important operation in the network layer is data forwarding and routing the data in the network[2]. In MANET the network layer protocol forms connectivity to all nodes from hop to neighbors [3].The concept behind the data forwarding and routing are based on packets. The packets from one link to another link is maintained by data forwarding therefore in this situation which packets are send from the source to destination is maintained by routing concept [4]. Data forwarding through packets prevent the drawbacks in wireless sensor network and helps to know about the present data packet in the network. The dynamic changes in MANET topology, communication between mobile nodes may be slow due to the low bandwidth. Thus routing information of nodes has changed to adapt current node movement.

In this paper the data forwarding is done using alight weight Proactive Source Routing (PSR)protocol. The Iterative Deepening Depth-First Search(IDFS) combines depth-first search space-efficiency and breadth-first search completeness that uses much less memory. Here information is periodically exchanged among neighboring nodes for updated network topology information. By using adaptive Hello messaging scheme the neighbor discovery is effectively suppress the unnecessary Hello messages. This scheme dynamically adjusts Hello intervals, andthe risk is reduced for the sender while transmitting a packet through a broken link that is not detected by Hello messaging; Instead of using a constant Hello interval, this proposed scheme uses a constant risk level. As the event interval increases, the Hello interval can also increase without increasing risk. If the event interval is extremely large, the Hello messaging interval is also correspondingly large; that is Hello messaging is practically suppressed. When anode receives or sends a packet, the Hello messaging interval is reset to a default value so that up-to- date information is kept in a neighbor table for active communication.

Thus this proposed system with DFID and adaptive Hello interval helps to reduce battery drain through practical suppression of an unnecessary Hello messaging. Based on the event interval of a node, the Hello interval can be enlarged without reduced detectability of a broken link, which decreases network overhead and hidden energy consumption.

Hence it allows supporting of both source routing and conventional IP forwarding. When doing this, the routing overhead of PSR is reduced.

II. RELATED WORK

A. A cellular network modeled by a triangular system.

Cellular communications have experienced an explosive growth recently. Cellular networks are commonly designed as triangular systems, where vertices serve as base stations (BSs) to which mobile users must connect to make or receive phone calls. Mobile users are normally connected to the nearest BSs and, thus, BSs divide the area such that each BS serves all users that are located inside a hexagon (a cell) centered at BS (see Figure 1). Mobile users with cellular phones have to register frequently to facilitate their location when phoning them. They move from cell to cell, but do not always contact their new cell to update their position since too many messages may be required and the system may be blocked for regular calls

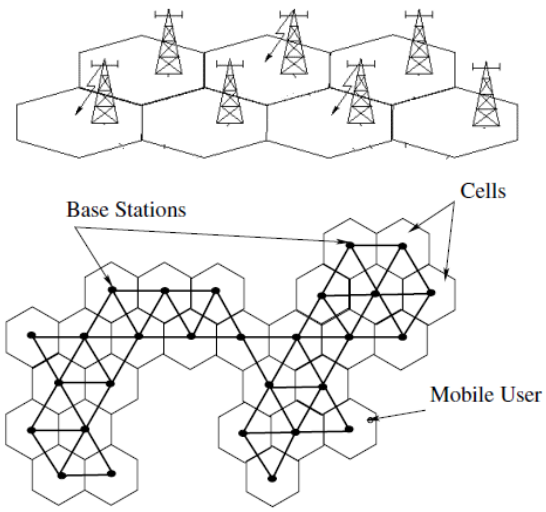

Figure 1. A cellular network modeled by a triangular system.

B. Honeycomb Network and Hexagonal Mesh Network

HM_n is made by the following method. HM_1 is in form of a hexagon. HM_2 is made by attaching each hexagon to the outside of the six edges of HM_1. HM_3 is made by attaching each hexagon to the outside of the edge of HM_2. In the same way, HM_n is made by attaching each hexagon to the outside of the edge of HM_{n-1}. Figure 2 shows HM_3. The honeycomb mesh is a bipartite graph. All nodes can be subdivided into two groups, which will be called black and white nodes, such that any edge joins a black and a white node. Vertex and edge symmetric honeycomb torus is obtained by adding wraparound edges to the honeycomb mesh.

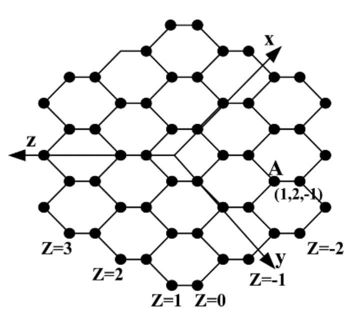

Figure 2 Honeycomb Mesh

The honeycomb mesh HM_n consists of $6n^2$ nodes and $9n^2 - 3n$ edges and is indicated in $HM_n = (V_{hm}, E_{hm})$. A node set V_{hm} and an edge set E_{hm} are defined as follows:

$$V_{hm} = \{(u, v, w) \mid -n + 1 \le u, v, w \le n, \\ 1 \le u + v + w \le 2\}$$

$$E_{hm} = \{((u, v, w), (u', v', w')) \mid |u - u'| + |v - v'| \\ + |w - w'| = 1\}$$

The node address of HM_n is indicated in (u, v, w). Address assignment method of HM is as follows: As seen from Figure 2, the crossing of the x, y and z axes is deemed as start of each axis. Node address is indicated in (u, v, w), u value of the node which is first met in the x axis from the start point is 1, and $u = u + 1$ at the address value of the previous node per node which is met in movement. In a reverse direction, u value of the node which is first met in movement is 0, and $u = u - 1$ at the address value of the previous node per node which is met in movement. In the $y(v)$ axis and the $z(w)$ axis, the address is assigned in the same manner as the x axis. Node A is indicated as an example in Figure 2. For all nodes placed in a zigzag form which meet with the z axis at a right angle, the address value of w is same. When $u + v + w = 1$, nodes adjacent to the node (u, v, w) are $(u + 1, v, w), (u, v + 1, w)$, and $(u, v, w + 1)$, and when $u + v + w = 2$, nodes adjacent to the node (u, v, w) are $(u - 1, v, w), (u, v - 1, w)$, and $(u, v, w - 1)$.

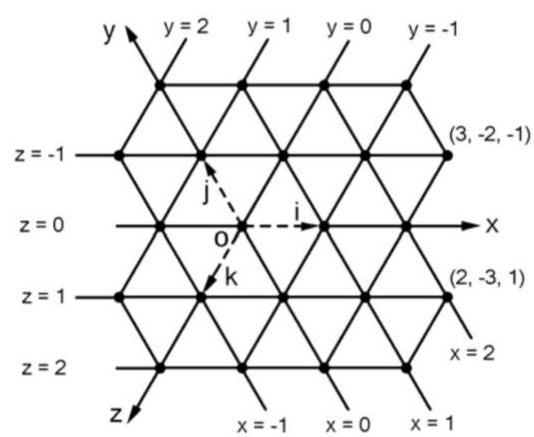

Figure 3 Hex Mesh HM

Given a hexagonal mesh network (Hex mesh) , choose any node in the network as the origin and assign (0, 0, 0) as its coordinates. For any other node p in the mesh, the coordinates of p are the ordering numbers of the mesh lines that have angles of 120 degrees in counter-clockwise direction with each axes, denoted as (x_p, y_p, z_p) .

When the origin node is assigned, every ordering number of each mesh line is unique for any given node, as shown in Figure 3, so the coordinates of any node in a Hex mesh is unique. This is one of the advantages over the other published schemas. In a Hex mesh, a node p with coordinates (x_p, y_p, z_p) has six neighbors that can be reached from p in a single genetic step. The neighboring nodes will be p+i, p+j, p+k, p-i, p-j and p-k, where operator +, - is vector addition and vector subtraction. For example, from node p through step i to node q can be expressed as $p + i = (x_p, y_p, z_p) + (1, -1, 0) = (x_{p+1}, y_{p-1}, z_p) = (x_q, y_q, z_q) = q$.

III. METHODOLOGY

To ease the computing the DFID algorithm for mobile networking, in this section we define a novel addressing scheme for hexagonal mesh network. All the vertices of Hex mesh except the center is partitioned into 6 sections as shown in Figure 4.

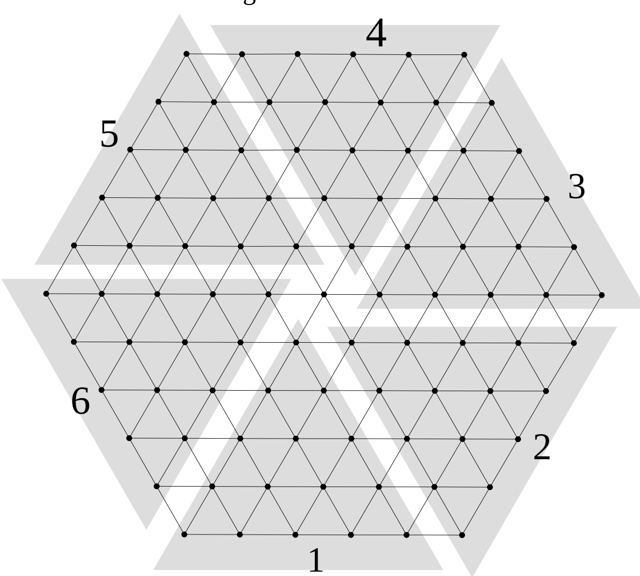

Figure 4 .DFID-Hexagonal meshes

Now each section forms up a triangular mesh. For each triangular mesh we provide a labelling as in figure 5.

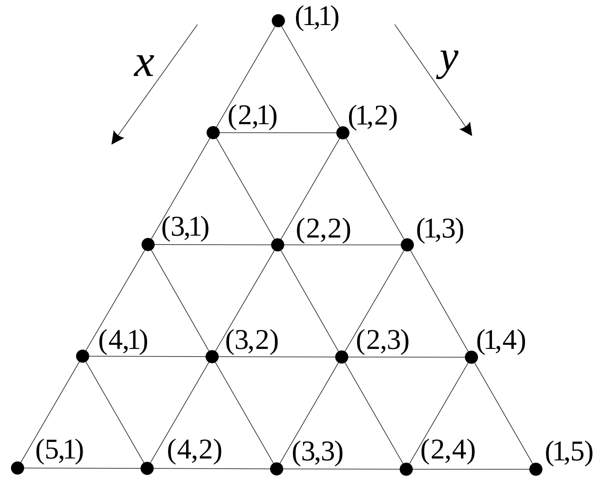

Figure 5 DFID-Triangular representation network

Finally we obtain a unique addressing scheme for hex mesh, with each vertex labelled as (u, v, w) where u refers to which section does it belongs to $(1 \leq u \leq 6)$. And v, w represents the addressing label of a triangular mesh $(1 \leq v, w \leq n-1)$. The center most vertex is referred as origin and labelled $(0,0,0)$ see figure 6.

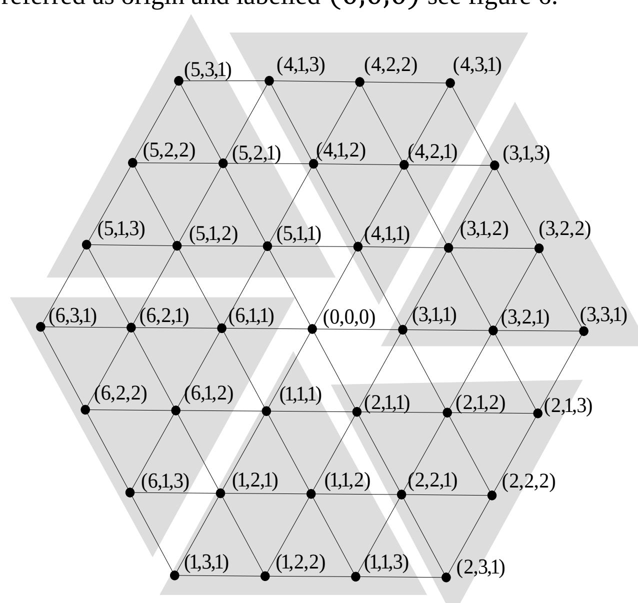

Figure 6. DFID for mobile adhoc network

IV. DFID ALGORITHM FOR MOBILE NETWORK

In this section we discuss the time complexity for DFID algorithm for mobile network. Since the base structure of the cellular network is a honeycomb. Where base station as fixed in each hexagon. Thereby the interconnection within the base stations forms a hexagonal mesh, which is obtained as a dual network of honeycomb, see figure 7.

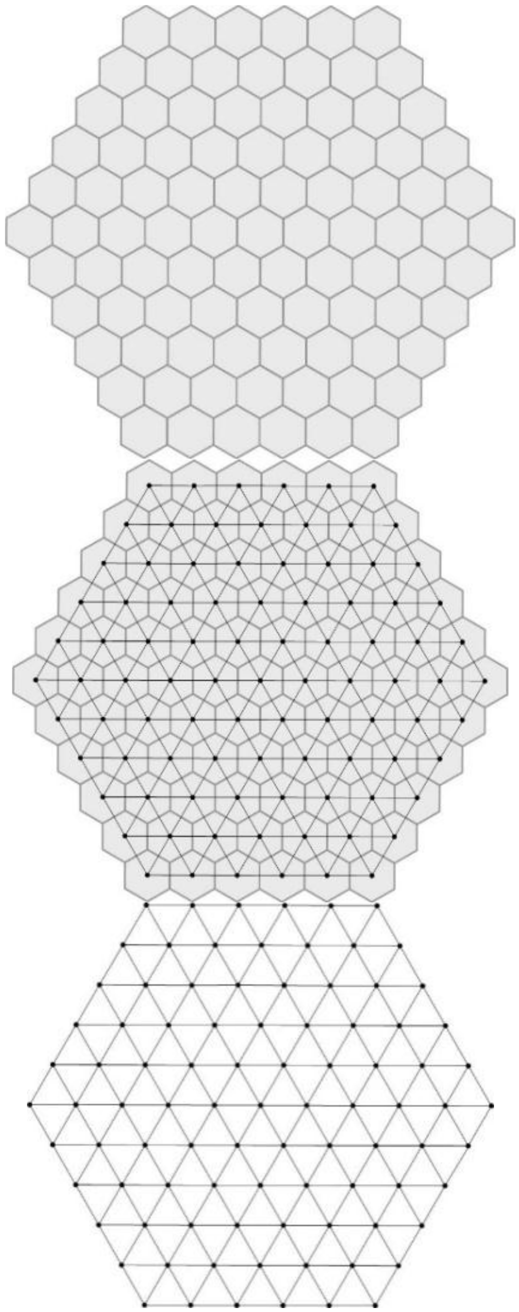

Figure7. Hexagonal mesh network

With confinement, lets consider the source of our problem as the origin vertex i.e. the vertex $(0,0,0)$ labelled as discussed in section 3. In order to apply the DFID on the network we consider a spanning tree as in figure 8. The spanning tree is redrawn in figure 9 with origin taken as root vertex.

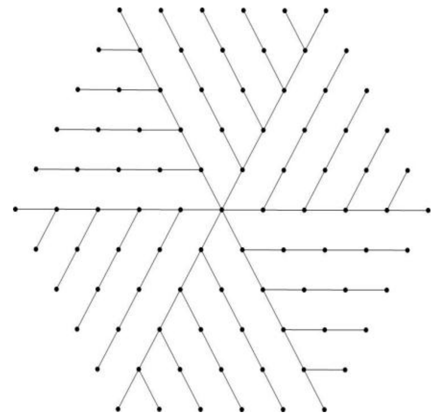

Figure 8. DFID spanning tree

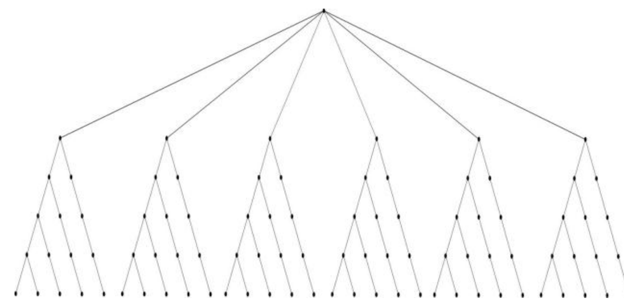

Figure 9. DFID binary trees

Let $DFS(x, G)$ denote the time taken by DFS by preorder traversing to find the vertex x in the graph G. In this section we discuss the $DFS(x, G)$ for G being a spanning tree of hex mesh of dimension n i.e. HX_n.

V. RESULTS AND DISCUSSION

Theorem 1: Let $(x, y, z) \in V(HX_n)$. Then,

$$DFS\big((x, y, z), HX_n\big) \leq \frac{1}{2}(n^2x - nx - 2n - 2ny + y^2 + 3y + 2z - 2)$$

Proof. Given that $(x, y, z) \in V(HX_n)$. Clearly the vertex belongs to x^{th} section $(1 \leq x \leq 6)$. HX_n has $3n^2 - 3n + 1$ vertices and each section in HX_n has $\frac{n(n-1)}{2}$ vertices. There are exactly

$$y + z - 2$$

vertices lying on the shortest path between $(0,0,0)$ and (x, y, z). These $y + z - 2$ remains visited by preorder

DFS. The remaining number of visited vertices within the x^{th} section being

$$\frac{(n-y-1)(n-y)}{2}$$

Also the vertices in 1[st]upto$(x-1)^{th}$ sections are visited whose sum is given as

$$\frac{n(n-1)(x-1)}{2}$$

The source vertex also remains visited.Thus,

$$DFS\big((x,y,z),HX_n\big) \leq \frac{n(n-1)(x-1)}{2} + \frac{(n-y-1)(n-y)}{2} + (y+z-2)+1$$

$$=\frac{1}{2}(n^2x-nx-2n-2ny+y^2+3y+2z-2)$$

Hence the theorem.

■

Let $BFS(x,G)$ denote the time taken by BFS traversing to find the vertex x in the graph G. In this section we discuss the $DFS(x,G)$ for G being a spanning tree of hex mesh of dimension n i.e. HX_n.

Theorem 2: Let $(x,y,z)\in V(HX_n)$. Then,

$$BFS\big((x,y,z),HX_n\big) \leq \frac{1}{2}[(2x+y+z-4)(y+z-1)+2z]$$

Proof. Given that $(x,y,z)\in V(HX_n)$. Consider the spanning tree in figure 9. Clearly the vertex belongs to x^{th} section $(1\leq x\leq 6)$. HX_n has $3n^2-3n+1$ vertices and each section in HX_n has $\frac{n(n-1)}{2}$ vertices. Moreover (x,y,z) belongs to $(y+z-1)^{th}$ row in section x. Thus

$$\frac{(y+z-1)(y+z-2)}{2}$$

vertices are visited in each sections.
Thus

$$\frac{(y+z-1)(y+z-2)}{2}\times n$$

vertices are visited. In addition $(y+z-1)$ number of vertices are visited in previous 1[st]upto$(x-1)^{th}$ sections, making a sum of,

$$(y+z-1)(x-1)$$

Also in $(y+z-1)^{th}$ row in section x, $(z-1)$ vertices are visited. And the source vertex is also visited. Thus,

$$BFS\big((x,y,z),HX_n\big) \leq \frac{(y+z-1)(y+z-2)n}{2} + (y+z-1)(x-1)+(z-1)+1$$

$$=\frac{1}{2}[(2x+y+z-4)(y+z-1)+2z]$$

Hence the theorem.

■

Having computed the depth first search and breath first search for HX_n we obtain the following theorem.

Theorem 3: Let $(x,y,z)\in V(HX_n)$. Then,

$$DFID\big((x,y,z),HX_n\big) \leq min\big[BFS\big((x,y,z),HX_n\big),DFS\big((x,y,z),HX_n\big)\big]$$

where,

$$BFS\big((x,y,z),HX_n\big) \leq \frac{1}{2}[(2x+y+z-4)(y+z-1)+2z]$$

and

$$DFS\big((x,y,z),HX_n\big) \leq \frac{1}{2}(n^2x-nx-2n-2ny+y^2+3y+2z-2)$$

VI. CONCLUSION AND FUTURE SCOPE

In this paper we discuss the time taken to search a vertex byIDFS algorithm for hexagonal mesh network of dimension n. Also, the time complexity for inorder and post order for hexagonal mesh are under investigation. Similar computation can be extended to mobile networking with constrains being taken in consideration.

REFERENCES

[1] T . H . Cormen, C. E . Leiserson, R. L. Rivest and C. Stein "*Introduction to Algorithms*" , Cambridge, MA: MIT Press , September 2009

[2] N. Gelfand , M . T. Goodrich and R. Tammasia,”*teaching Da t a Structure Design Patterns*,” Proceedings of the 29 SIGCSE Technical Symposium on Computer Science Education, vol. 30 (pp. 331-335) , Providence, RI, USA, 1998 .

[3] R.Bayer, “*Binary B-Trees for Virtual Memory* ,” Proceedings of ACM SIGFIDET Workshop on Data Description, Access and Control , pp. 219-235, San Diego , CA, USA, 1971

[4] R. Tarjan, “*Depth First Search and Linear Graph Algorithms*,” Proceedings of the 12th Annual IEEE Symposium on Switching and Automata Theory, pp. 141-121, East Lansing, MI, USA, 1971.

[5] Everitt, Tom, and Marcus Hutter. "*Analytical Results on the BFS vs. DFS Algorithm Selection Problem. Part I: Tree Search.*" Australasian Joint Conference on Artificial Intelligence. Springer International Publishing, 2015

[6] Everitt, Tom, and Marcus Hutter. *"Analytical Results on the BFS vs. DFS Algorithm Selection Problem: Part II: Graph Search."* Australasian Joint Conference on Artificial Intelligence. Springer International Publishing, 2015.

[7] J.A.Bondy, U.S.R. Murty, *Graph Theory with Applications*, North Holland, New York.Amsterdam.Oxford, 1976.

[8] F. Harray. Graph Thoery. Addison. Wesley, Reading. Mass (1969).

[9] Deo, Narsingh. *Graph theory with applications to engineering and computer science.* Courier Dover Publications, 2016.

[10] A.Bagchi, S.Hakimi, *Data Transfers in Broadcast Networks*, IEEE TransactionsComput. 41, (1992).

[11] R.Balakrishnan, K.Ranganathan, A Text Book of Graph Theory, Springer,(1999)

[12] C.Berge, *The Theory of Graphs*, Methuen, (1962).

[13] N.L.Biggs, E.K.Loyd, R.J.Wilson, Graph Theory, Oxford UniversityPress,(1976), 1736 { 1936 .R.Diestel, Graph Theory, Springer, (1991).

[14] J.L.Gross, J.Yellen, Graph Theory and its Applications, CRC Press, (1998).

Authors Profile

Mrs. Justin Sophia.I pursed Bachelor of Computer Science from St.Mary College, Tuticorin in 2004 and Master of Science from St.Josephs College, Trichirapalli in year 2006. He is currently pursuing Ph.D. at Presidency College, and currently working as Assistant Professor in Department of Computer Science, Loyola College since 2007. She has published more than 10 research papers in reputed international journals including Scopus, Thomson Reuters ,UGC approved journals and conferences it’s also available online. Her main research work focuses on Network Security, Cloud Security , Modelling and Simultion and Mobile Adhoc Networks. She has 10 years of teaching experience.