Journal of Theoretical and Applied Information Technology
31st March 2022. Vol.100. No 6

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

A RING BASED LEADER ELECTION ALGORITHM FOR HIVE NETWORKS

[1]MOHAMMED AL-REFAI, [2]YOUSEF ALRABA'NAH

[1]Department of Software Engineering, Zarqa University, Zarqa, Jordan.

[2]Department of Software Engineering, Al-Ahliyya Amman University, Amman, Jordan.

E-mail: [1]refai@zu.edu.jo,[2]y.alrabanah@ammanu.edu.jo

ABSTRACT

Leader Election Algorithms (LEAs) play a vital role in distributed systems. The leader organizes and synchronizes processes in distributed systems and communication networks. The leader is prone to fail which in turn makes the network inconsistent. LEAs solve the leader failure problem by electing a new one. In this paper, a new LEA is proposed to solve the leader failure in hive networks. The proposed algorithm elects one node with the highest priority to be the new leader. The algorithm is evaluated theoretically by calculating number of messages and time steps required to elect a new leader. We show that the algorithm needs O(n) messages in $O(\sqrt[3]{n})$ time steps in the best case, while it needs $O(n^{1.3})$ messages in $O(\sqrt[3]{n})$ time steps in worst case.

Keywords: *Leader Election Algorithm, Hive Networks, Network Cost, Honeycomb Networks, Distributed Systems.*

1. INTRODUCTION

Distributed systems consist of multi-interconnected computers that communicate with each other to achieve common goals. Tasks are divided over computers in distributed systems to increase the computational speed. However, processes that are communicating through a distributed system need coordinating and controlling their activities, so one process can be chosen as a leader or coordinator to reduce the communication complexity [1]. Leader election algorithm (LEA) is a program that is embedded in each node of the distributed systems, any node detects the leader failure (notes the leader does not respond to its requests), starts the election procedure by initiating a certain run of a LEA. The election ends when a leader is chosen and all other nodes know who the new leader is and agree on that. An election can't be called more than once by the same node at a time, but N nodes have the ability to call N concurrent elections [2]. The leader election takes the system from an initial state (unstable state), where there is no leader, and all the nodes are in the same state (candidacy state), into a new state (stable state) where only an individual node is distinguished (leader state) and all other nodes are aware of this leader (normal state) [3].

If all nodes have the same characteristics, then it is difficult to choose one of them to be distinct, so that every node should has a unique identification (ID). However, it is common approach to elect one node with the highest priority to be the leader. For example, the node with lowest computational load, largest memory, or highest processor speed, among all nodes would be chosen as the leader [2]. In general, there are three states in which any process in the system can stay in, these states are: normal, candidate or leader. Each node in the system has a state variable indicates its state value, when the process is in ordinary operation and the leader is active in the system, then state variable contains the normal value. Candidate state denotes that the leader is failed and the system is in the leader election process, leader state is obtained by only one process when the election process terminates. Moreover, every node in the system have a local variable leader, which indicates the current leader of the system [4]. It is necessary that, all processes have the same LEA. As well, any LEA must satisfy the following properties: safety and liveness. Safety property indicates that, the nodes in the system never disagree on the elected leader. After the termination of the election process, at most one node will be elected as a leader, and all other nodes must have the same leader variable value. The second property is liveness, all involved nodes in the election process must eventually reach a situation where all of them are in normal state, except unique one which is in a leader state [5].

This paper proposes a new LEA to solve the leader failure problem in hive networks. The rest of the paper is organized as follows: Section 2 discusses the related works. In Section 3, we introduce the hive networks and its properties. Section 4 presents the proposed algorithm in details. Section 5 evaluates the performance of the proposed algorithm. In Section 6, we conclude the paper and give directions for future works.

2. RELATED WORKS

Leader election is fundamental problem in distributed systems and have been largely studied in the literature. Several algorithms were proposed to solve the leader failure in various networks topologies. The same algorithm is difficult to be applied on different systems since LEAs are varied based on the nature of the algorithm, network topology, transmission media, and whether the network size is (known) or not known [3].

Le Lann in [6] proposed a LEA for ring networks. The algorithm assumes that the processes are logically ordered and organized in a ring form. The election starts when a process detects a leader failure, the process sends election message to next process in the ring. The election message forwards among processes until returns back again to the process that started the election. All other processes put theirs IDs in the election message, so the started process can elect the process with highest ID as the new network leader. The message complexity is $O(n^2)$. However, many researchers improve the ring algorithm [7] [8].

Bully algorithm proposed in [9] to elect a leader in the fully connected network. In bully algorithm, all processes can communicate directly with each other's. When a process P detects a leader failure, it sends an election message to all processes with higher ID, if no process responds to the election message, process P announces itself as the new leader. However, when a process K receives the election message, it replies with an OK message to node P indicates that the process K will take over the election process. The election continues until only one process with highest ID is continue and all other processes give up. The bully algorithm requires $O(n^2)$ messages. The bully algorithm studied and improved in [10] [11] and [12]. A LEA is proposed in [13] for tree networks. The algorithm based on heap structured. The algorithm starts by building a tree of processes and ends when one process with the highest ID is stored at the root of the tree. The root process declares itself as the tree leader by sending leader messages to others. The message complexity of the tree algorithm is O(n).

Refai et al. [14] proposed a LEA for 2D honeycomb torus networks. The algorithm consists of three phases and requires O(n) messages to complete. Another LEA is proposed in [15] for hypercube networks, the algorithm requires O(n) messages.

Refai et al. [16] studied honeycomb mesh networks and proposed a LEA for it. The algorithm consists of four phases. The algorithm assumes that the network composed of rings, each ring nominates a leader, and finally the nominates leaders are compared together to elect one of them as a leader for the network. The algorithm requires O(n) messages.

Various network topologies have been used in interconnected networks, such as ring, hypercube, mesh, tree and honeycomb. Networks topologies are differentiated using several criteria; one of the most used criteria is the network cost. Network cost referred to the network performance and its implementation cost [17]. It is computed by multiplying network's diameter with its degree, the network's diameter refers to the message transmission time, while the degree refers to the hardware cost. However, in this paper, we study hive networks and proposes a new LEA for it. Hive networks consist of a set of honeycomb mesh layers. Hive have network cost less than three-dimensional honeycomb mesh network by about 20% [18]. The leader in hive networks is prone to fail, so to keep the network stable a LEA is required.

3. HIVE NETWORKS

A Hive network is a special form of 3D Honeycomb Mesh networks 3DHM. The 3DHM networks are composed of multiple interconnected HM networks (which called layers) that are identical and connected by vertical links. However, the 3DHM network contains a large number of vertical links, these links are expensive, area consuming and lead to high implementation cost [19]. Another variation was proposed in [18] called hive network, which reducing the large number of vertical links about to half, see figure 1.

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

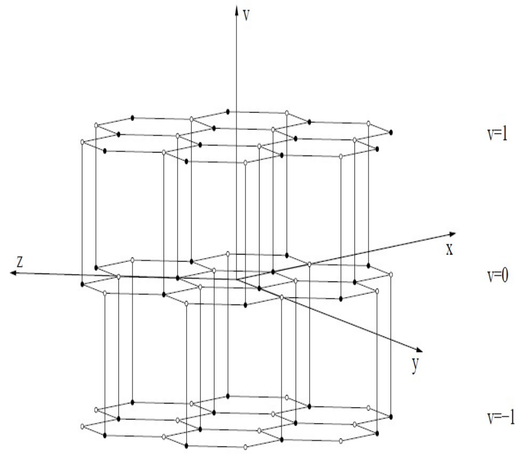

Figure 1: Hive Network Of Size 2

Szczesniak in [18] suggested that the number of HM layers that constructed the hive network is determined by its size. A HM layer consists of a number of hexagons, one hexagon forms a HM of size one, which is denoted as HM1. The HM of size two (HM2) is obtained by surrounding the boundary edges of HM1 with 6 hexagons. In such manner, HM of size t (HMt) is constructed by attaching one hexagon to each edge on the outside boundary of HMt-1 as shown in figure 2. For the hive network, the hive network of size 1 consists of one layer of HM1, and the hive network of size 2 consists of three HM2 layers, as shown in figure1. Thus, the hive network of size t consists of 2t-1 layers of HMt. Note that, the HM and the hive networks are identical exactly when the size is 1.

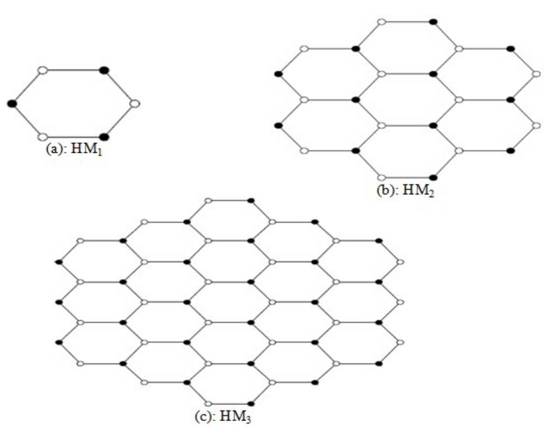

Figure 2: Honeycomb Mesh Topology With Different Sizes

Each node in the hive network is addressed using four integer coordinates (x,y,z,v), such that $-t+1 \leq$ x, y, $z \leq t$ and $-t+1 \leq v \leq t-1$. The V axis is used to determine the layer of the processing node. Therefore, the coordinate system for the hive uses four coordination axes, X, Y, Z, and V, as shown in figure 3. The V axis starts from the middle layer, which has v=0, the first above layer has v=1, the next above layer has v=v+1. While the first down layer has v=-1, the next down layers has v=v-1. The rest layers are addresses in the same way [18][20].

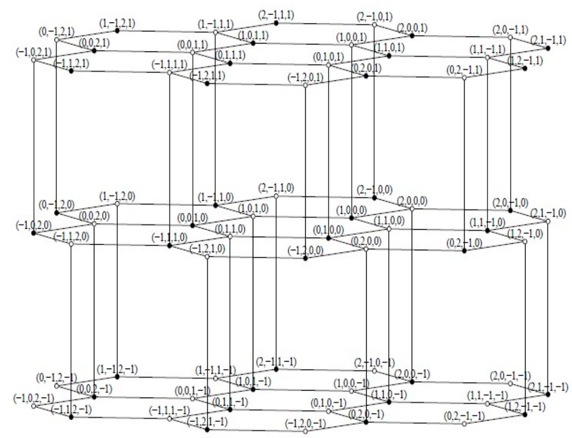

Figure 3: Coordinate System Of Hive Network Of Size 2

As in HM, each node in the hive has either white color or black color. The summation of coordinates (x,y,z) equals 2 for each white node and 1 for each black node. The adjacent of a white node (x,y,z,v) are (x-1,y,z,v), (x,y-1,z,v), (x,y,z-1,v), (x,y,z,v-1), and (x,y,z,v+1). While the adjacent of a black node (x,y,z,v) are (x+1,y,z,v), (x,y+1,z,v), (x,y,z+1,v), (x,y,z,v+1), and (x,y,z,v-1). The node color also indicates the availability of vertical links, where the white node has a vertical link to the lower layer, and the black node has a vertical link to the upper layer. An exception is made for white nodes in last down layer, and for black nodes in the first above layer, which have not vertical links; figure 4 shows the links in hive topology [18].

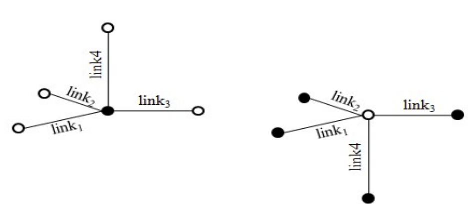

Figure 4: Links Numbers In Hive Network

Two nodes in the network are connected by an edge if the distance between them is 1, the distance in the hive between any pair of nodes p and p' is computed as |x-x'| + |y-y'| + |z-z'| + |v-v'|. The overall number of nodes n in hive network of size t is obtained by counting the nodes in each layer, this achieved by multiply the number of layers with the number of nodes in any layer [21]. In HMt the number of nodes is $6t^2$ [17], and thus the overall number of nodes in give networks is $6t^2(2t-1)$. Also, the overall number of edges in hive is $24t^3 - 21t^2 + 3t$. The diameter represents the longest path from the set of network's shortest paths between any pair of nodes [22], the hive diameter is $6t - 3$. The node degree is 4, so the network cost of the hive is $4(6t-3)$. The hive network cost is approximately 20% better than 3DHM networks.

4. THE PROPOSED ALGORITHM

The hive network is built by queuing a number of identical HM layers. As the hive network is considered as an extended form of the HM networks, algorithms such as routing and broadcasting that were proposed for HM networks can be also extended to adapt the hive networks. Our proposed algorithm is an extension for our LEA that have been proposed in [16] for honeycomb networks. The idea here is to do an election process in each layer in the hive to get one leader in each layer, known as the nominee leader. An extra election process is required among the nominee leaders to elect one of them as the leader for whole network.

4.1 Definitions

- Node State: each node in the network can be in one of three states: normal, candidate or leader. Normal state indicates that the node does not participate in any election process and there is no leader failure. Candidate state implies that this node is now participating in current election process and therefore there is a leader failure. The last is the leader state, only one node in the network owns this state. The leader state is lost when the leader fails, which will make the network instable.
- Position: position of the node in the network according to its coordinates (x, y, z, v).
- Node Ring: each node in the network can belong only to one ring. We assume that, each layer in the network is composed of t overlapped rings, where t is the network size. The rings numbered as (t, t-1, t-2, t-3, …, 1). Figure 5 shows the rings for a layer in hive network of size 4.
- Path: the path represents the route a message can follow. This path consists of two links number such as 23 or 12.
- Ignore Node: this node ignores the message it receives, and does not forwards it to the next node. The message is ignored by either a candidate node, or by a node that belongs to the message initiator's ring which have the same election path.

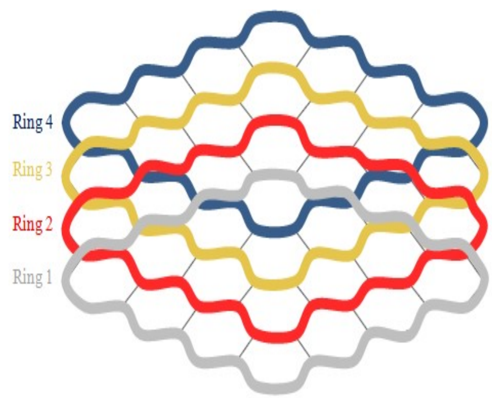

Figure 5: Rings In A Layer Of Hive Network Of Size 4

- Alteration Nodes: it is a set of nodes that changes the election path, the path is changed to ensure that the message will follow its route within its ring and visit all nodes. The alteration nodes are lies on the border of the rings where z coordinate values are t, 1, 0, and t-1, see Fig. 6 which shows alteration nodes in a layer of Hive 3.

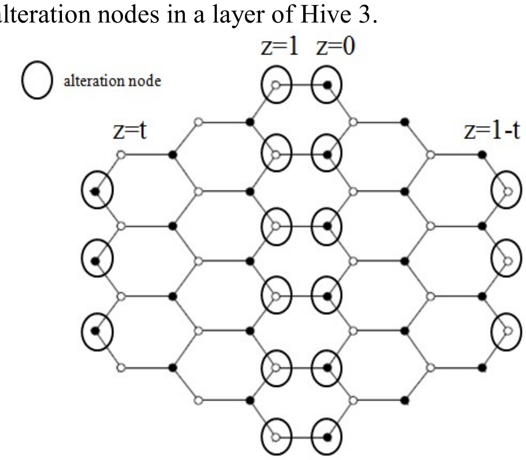

Figure 6: Alteration Nodes In A Layer Of Hive Network Of Size 3

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

- Types of Message: during the election process a set of message is initiated and exchanged to elect a new leader, these message are:
  - Inform message: this message is initiated by the node that detects the leader failure in order to inform others to start election process. The inform message consists of: initiator position, phase, step, ignore_node, path_1, path_2.
  - Election message: this message consists of: initiator position, phase, path, ring_number, greater ID, position of greater ID. The election message is exchanged among nodes within the message initiator ring and follows a specific path.
  - Ring leader message: similar to the election message, which compares all rings leaders of a layer in the network to elect one of them as the layer leader. this message is initiated in phase three.
  - Layer leader message: this type of messages used in phase four. The layer leader message is initiated to gain information about each layer leader in the network and elect one of them as leader for the whole network.
  - Leader message: when the election process terminates, one node is elected as a leader. A leader message is initiated to announce the new leader for all nodes within the network. The leader message consists of: leader id and leader position.
- Ring gather node: we assume that a node with position (ring number, -ring number + 1, 1, v) in each ring in the network is used as stop node of the election process in that ring and gets the election result.
- Layer gather node: in each layer, there is a node with positon (1, 0, 1, v) that selects a leader for that layer. The layer gather node gets the rings leaders in its corresponding layer and elects one of them as the layer leader.
- Network gather node: it gathers the result of the network election process. The node with position (1, 0, 1, 0) gets the all layers' leaders and elects one of them which have the highest ID as the new network leader. This node also broadcasts a leader message to all networks' nodes to announce the new leader.
- Each node in the networks have the following information: Node ID which is unique, node position, leader ID, leader position, election phase, node ring and node state.

4.2 The Algorithm Phases

We assume that, each hive's layer consists of t overlapping rings. Each ring selects one of its nodes with highest ID as nominate leader. Afterwards, only one node with the highest ID among t leaders in a layer will be selected as the layer leader. Finally, all leaders in t layers are compared together to select one of them as the networks' leader. Figure 7 shows the overlapping rings in hive network of size 2.

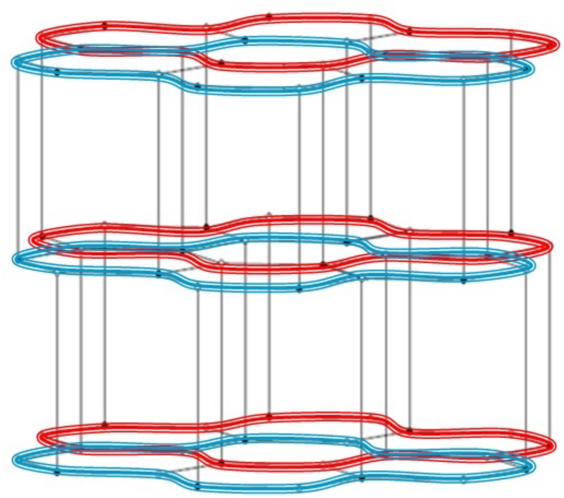

Figure 7: Rings in hive network of size 2

The hive algorithm consists of five phases, which can be briefly presented as following:

Phase one: The node that detects the leader failure, sets its state to candidate, and informs other nodes with the leader failure to start the election process in theirs' rings. The informed node has the same summation of the message initiator x, and y coordinates, and the same z coordinate value. However, comparing with message initiator v coordinate value, the informed node within the message initiator layer has the same v value, while in other layers has different value.

Phase two: Each candidate node in phase one starts the election by sending election messages to all nodes within its ring in its layer, the message is sent in two different directions, each node received the election message compares the received ID with its ID and passes the higher one, the election result of each ring in each layer is gathered by a certain node in that ring which is called a ring gather node.

Phase three: The rings gather nodes that have the results of phase two make an additional election to get the election result of its layer in one node, which is called a layer gather node. This node is familiar with the nominee leader of its layer.

Phase four: The layers gather nodes in phase three, choose one leader with highest ID from the set of nominees' leaders to be the new leader. The result is gathered the network gather node. This node is aware of the new leader.

Phase five: The network gather node is aware of the new leader; it broadcasts leader messages to all nodes in the network to announce the new leader. Figure 8 summarizes the algorithm phases.

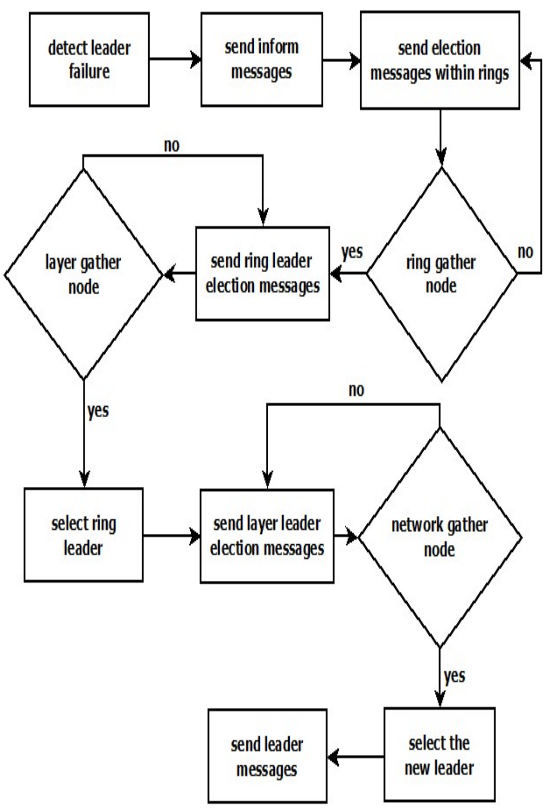

Figure 8: Algorithm Phases

Next, we will discuss the hive algorithm in more details, consider the five phases of the algorithm, which started at the moment of leader failure and finished when all nodes know the new leader, these phases are:

Phase One: The algorithm starts by a node that detects leader failure. This node changes its state to candidate, determines the election paths and inform message ignore condition based on its position, and sends inform messages to t-1 nodes across links 1, 2, and 4, the inform message guaranties that, all informed nodes will have the same election paths. Each node received the inform message checks its x, and y coordinate summation with the summation of its corresponding message initiators' x, and y coordinate. If the summations are same then the node changes its state to candidate, otherwise the node state remains normal. The node also passes the inform message to other nodes across links 1, 2, and 4 except the link in which it received the message from. Each candidate node starts phase two by extracting the paths from the inform message, chooses its ID as higher ID and sends election messages to other nodes within its ring across the first link in each election path. Note that, if the state of the node is a candidate, the inform message will be immediately ignored.

Phase Two: Each candidate node that have been informed about the leader failure in phase one sends election messages to all other nodes within its ring. The messages are sent in two different directions based on the election paths attached within the inform message. Every node that received the election message compares the received ID with its ID, selects the higher ID and passes the election message to next node after changing the path. The ring gathers nodes which have positions (ring_number, - ring_number + 1, 1, v) gather the election result of each ring in the network; these nodes have the nominees' leaders of all rings in the layer. During phase two each node receives the election message checks its ring number and the alteration node condition. The nodes that have the same ring number of the election message initiator are authorized to compare the received ID with its local ID, the nodes that differ than the ring number of the message initiator will pass the election message without comparing the IDs. The alteration node condition is checked to decide whether the path numbers will be only swapped, replaced without swapping, or replaced and swapped. when the ring gather node which have the position (t, -t+1, 1, v) receives two election messages belong to the same initiator, it selects the higher ID and starts phase three by sending ring result message across a path with sequence 12 to compare its nominee leader with all nominees leaders in its layer.

Phase Three: The node with position (t, -t+1, 1, v) starts phase three by sending ring leader election message across path: 12. As in phase one, the message sent to t-1 nodes, each node that has the same summation of message initiators' x, and y coordinates, will compare its local ID with received ID and select higher one, then it passes the message to next node. If the phase of the receiving node does not equal to the phase of the incoming message, the message will suspend until the phase of the node becomes three. This process continues until the message reaches the layer gather node (1, 0, 1, v), this node selects the node with highest ID to be the nominee leader of the layer.

Phase Four: In this phase, the nominee leader in each layer is compared with the others, the layer gather node with position (1, 0, 1, t-1), in the first layer from above and the layer accumulator node with position (1, 0, 1, 1-t) in the last layer from down, start phase four by sending layer leader election message across path: 14 or 41 depending on the availability of the link 4 to other nodes that have the same (x,y,z) coordinates in all layers. If the message reach to the network gather node which has the same summation of x, y, and z but v=0, then the message is stopped. The network accumulator node has position (1, 0, 1, 0), when it receives two election messages from link1 and link 4, it selects the nominee leader with the highest ID as the new leader for the whole network, and starts phase five by sending leader message to other nodes through links 1, 2, 3 and 4.

Phase Five: The node (1, 0, 1, 0), is aware of the new leader ID and position. It broadcasts leader messages to other nodes on all available links 1,2,3, and 4. Each node receives this message changes its state to normal, updates the leader information depending on the leader message, and passes the message to other adjacent nodes.

5. PERFORMANCE EVALUATION

The complexity of the hive algorithm in terms of number of messages, and time steps is computed as in following sections.

5.1 Number of Messages

The algorithm requires O(n) messages to select a leader in best case, while it requires $O(n^{1.3})$ in worst case. The overall number of messages required by the algorithm is the summation of messages number in each phase, as following:

5.1.1 Best Cases

Number of messages in phase one: The number of horizontal messages in one layer is 2t [16], for 2t-1 layers we need 2t(2t-1), an extra message in each layer may be used depending on the node position, so number of messages is $2t(2t-1) = 4t^2-2t$. Each node receives the message sends it vertically on link 4 if the link 4 is available. Number of vertical links in the target z coordinate equals t(2t-2), so we need t(2t-2) vertical messages, also an extra message is needed in each layer depending on the node position, so number of messages is $2t^2-2t$. As a result the number of messages in this phase is $4t^2-2t+2t^2-2t= 6t^2- 4t$.

Number of messages in phase two: The number of messages in this phase is computed by multiply the number of layers with the number of messages in each layer. The number of messages in one layer equals t(8t-2) [16], where 8t-2 is the size of the ring, the number of layers is 2t-1. Consequently the number of messages in this phase in all layers is $(2t-1)(8t^2-2t)= 16t^3-12t^2+2t$.

Number of messages in phase three: Since HM algorithm needs 2t-2 messages in phase three [16], the hive algorithm requires $(2t-1)(2t-2) = 4t^2-4t+2$ messages in this phase.

Number of messages in phase four: When t is odd, the number of messages is 4(t-1). One message is required to get out of the layer v = t-1, in each layer between v=t-1 and v=0 two messages are sent, the layer v=0 needs one message. So the number of messages is 2(t-2)+1+1=2(t-2)+2. Two messages is needed to get out of layer=1-t, in each layer between v=1-t and v=0 two messages are sent, so the number of messages that are needed is 2(t-2)+2. Therefore the total number of messages that is required when t is odd equal 4(t-2)+4= 4(t-1) messages. When t is even, the number of needed messages is 4(t-1). Two messages are needed to get out of the layer v = t-1, in each layer between v=t-1 and v=0 two messages are sent, the layer v=0 needs one message. So the number of messages is 2(t-2)+2+1=2(t-2)+3. One messages is needed to get out of layer=1-t, in each layer between v=1-t and v=0 two messages are sent, so the number of messages that are needed is 2(t-2)+1. Therefore the total number of messages that is required when t is even equal 4(t-2)+4= 4(t-1) messages. So in both cases this phase requires 4(t-1) messages.

Number of messages in phase five: The node (1, 0, 1, 0) broadcasts leader message to all adjacent nodes, so it sends four messages (based on node degree), each node receives the message will send messages across all links except the link it received the message from, each node has degree four will send three messages, and each node with degree

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

three will send two messages, while each node with degree two will send one message. The number of nodes with degree two is 6t, and the number of nodes with degree three $6t^2+6t(2t-3)$, the number of nodes with degree four is $(6t^2-6t)+(2t-3)(6t^2-6t)$ the first node sends one extra message, so the number of messages that is required by this phase is $6t+2(18t^2-18t)+3(12t^3-24t^2+18t)+1 = 24t + 36t^3-36t^2 + 1$. However, The total number of messages in all phases is $52t^3-38t^2+22t-1$. By solving the equation of number of nodes n = 6t2(2t-1), we found t as follows:

$$t=\frac{1}{6}\left(\sqrt[3]{3\sqrt{9n^2+2n}+9n+1}+\frac{1}{\sqrt[3]{3\sqrt{9n^2+2n}+9n+1}}\right)+\frac{1}{6}$$

So, the number of required messages in best case is O(n) messages.

5.1.2 Worst Case

Number of messages in phase one: All nodes that detect the failure send inform message on all available links except link three, each node receives the message will send messages across all links except the link it received the message from, each node has degree three with respect to z coordinate will send three messages, and each node with degree two will send two messages, while each node with degree one will send one message. The number of nodes with degree one is 4t, and the number of nodes with degree two is 6t2+4t(2t-3), the number of nodes with degree three is $(6t^2-4t)+(2t-3)(6t^2-4t)$, so the number of messages that is required by this phase is $4t+2(6t^2+8t^2-12t)+3(6t^2-4t+12t^3-8t^2-18t^2+12t) =36t^3-32t^2+4t$.

Number of messages in phase two: The total number of rings in the entire network is t(2t-1). The worst case for the number of messages is computed for honeycomb mesh in [16], for 2t-1 layers we need to multiply 2t-1 with number of messages in [18], as (2t-1)(t((8t-2)(8t-1)/2+8t-3)).

Number of messages in phase three: As in best case, the algorithm needs $4t^2 - 4t + 2$.

Number of messages in phase four: This phase requires 4(t-1) messages.

Number of messages in phase five: In phase five, the algorithm requires $36t^3 - 56t^2 + 44t + 1$ messages to announce the new leader.

The total number of messages in worst case is the sum of messages in all phases, which is $O(n^{1.3})$

5.2 Number of Time Steps

The algorithm requires $O(\sqrt[3]{n})$ time steps to select a leader in best case, while it requires $O(\sqrt[3]{n})$ in worst case. The overall number of time steps required by the algorithm is the summation of time steps in each phase, as following:

5.2.1 Best Case

Number of time steps in phase one: In phase one, the node that detects the leader failure sends two inform messages to inform t-1 nodes with same summation of x, and y coordinates in its z coordinate, and sends one inform message to another layer depending on the link four availability. For one layer the algorithm needs 2t time steps, the node in the last layer from down or the first node from above needs at most additional 4t-4 time steps to get each other. So the total number of time steps in this phase is 2t+4t-4 = 6t-4.

Number of time steps in phase two: As in HM algorithm [16], this phase includes two cases: the first case is when the node at the middle of a ring detects the leader failure, it needs 4t-1 time steps. Whereas the second case is when the node (ring_number, -ring_number+1, 1, v) detects the failure, it needs 8t-2 time steps to complete the ring.

Number of time steps in phase three: The time steps in this phase is similar to the time steps in HM algorithm [16], which is 2t-2.

Number of time steps in phase four: When t is odd, the nodes (1, 0, 1, 1-t) and (1, 0, 1, t-1) need 2(t-2)+2 time steps. The message that are sent by node (1, 0, 1, t-1) requires one time step to get out of its layer. In next layer, it requires two time steps. Therefore the message in each layer between v=t-1 and v=0 requires two time steps to get out of that layer. When the message reaches layer v=0, it needs one time step to reach the node (1, 0, 1, 0). The message that are sent by node (1, 0, 1, 1-t) requires two time steps to get out of its layer, in next layer it needs 2 time steps until it reaches node (1, 0, 1, 0). The total number of messages for two nodes is 2(t-2)+2=2t-2. When t is even, the node (1, 0, 1, t-1) requires two time steps to get out of its layer, in each layer between v=t-1 and v=0, the message needs two time steps, when it reaches layer with v=0, it needs one time step to reach the node (1, 0, 1, 0). So the total number of time steps in this case is 2(t-2)+2+1=2t-1. The node (1, 0, 1, 1-t) requires one time steps, next layer need two time steps, each

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

layer between v=1-t and v=0, needs two time steps until it reaches (1, 0, 1, 0), the total number of time steps is 2(t-2)+1= 2t-3. We take the maximum needed time steps which is 2t-1.

Number of time steps in phase five: In each layer, we need 2t+1 time steps to deliver the leader message to all nodes in that layer, also we need 2t-2 time steps until the messages reach all layers in the network, so the overall number of time steps in this phase is 4t-1.

The total number of time steps needed by hive algorithm in best case is computed with regard to first case in phase two as 18t – 10. While in the second case in phase two, the total number of time steps is 22t – 11. By substitute t with n, both results are in O($\sqrt[3]{n}$) time steps.

5.2.2 Worst Case

Number of time steps in phase one: In phase one, the number of time steps that are required to complete this phase is 1, since all nodes detect the failure simultaneously and become candidate, the inform messages are ignored by candidate nodes.

Number of time steps in phase two: In the worst case, the nodes with positions (ring_number, -ring_number+1, 1, v) have the higher ID, they need in each ring 8t-2 time steps, which is equal to the ring size. The total number of time steps in this phase is 8t-2.

Number of time steps in phase three: As in best case, this phase requires 2t-2 time steps.

Number of time steps in phase four: The hive algorithm in this phase needs at most 2t-1 time steps.

Number of time steps in phase five: This phase requires 4t-1 time steps, as in best case.

The total number of time steps in worst case that are needed by hive algorithm to complete all phases and elect a new leader for the network is 16t – 5, which is in O($\sqrt[3]{n}$) time steps.

6. CONCLUSION

In this paper, we propose a new algorithm to solve failure of the leader in hive networks. To simplify the election process, the algorithm partitions the network layers into a set of overlapping rings. Each ring nominates its leader and then each network layer select one of its rings leaders as a layer leader. Afterwards, the algorithm selects one of the network layers leaders as the network leader. We observe that the algorithm requires O(n) messages in O($\sqrt[3]{n}$) time steps in best case, while O($n^{1.3}$) messages in O($\sqrt[3]{n}$) time steps in worst case. The result shows that the algorithm have a good performance.

The research paper introduced a LEA for hive networks for the first time. Some of research papers discusses leader election problems in different topologies, but to best of our knowledge, no one proposed a LEA for hive networks.

There are challenges in our algorithm that remains for more investigations, such as dealing with one or more links failures, in addition to the ability of using the overlapping rings in hive networks routing algorithm. As future work, it would be interesting to design a fault tolerance LEA to deal with links failures. The algorithm can also be improved to solve leader election problem in three-dimensional HM networks.

REFERENCES:

[1] A. Biswas, A. K. Tripathi and S. Aknine, "Lea-TN: leader Election Algorithm Considering Node And Link Failures In A Torus Network," *The Journal of Supercomputing*, vol. 77, pp. 13292-, 13329, 2021.

[2] S. Verma, D. Yadav, and G. Chandra, "Leader Election Algorithm in Fault Tolerant Distributed System," *in Proceedings of Second Doctoral Symposium on Computational Intelligence 2021,* pp. 471-480.

[3] M. Refai, A. Sharieh and F Alshammari, "Leader Election Algorithm in 2D Torus Networks with the Presence of One Link Failure," *International Arab Journal of Information Technology*, vol. 7, pp. 105-114, 2010.

[4] S. Kanwal, Z. Iqbal, A. Irtaza, R. Ali, and K. Siddique, "A genetic based leader election algorithm for IoT cloud data processing," *Comput. Mater. Contin*, vol. 68, pp 2469-2486, 2021

[5] A. Biswas, K. M. Ashish, A. Tripathi, and S. Aknine, "FRLLE: A Failure Rate And Load-Based Leader Election Algorithm For A Bidirectional Ring In Distributed Systems," *The Journal of Supercomputing*, vol. 77, pp. 751-779, 2021.

[6] G. Le Lann, "Distributed Systems-Towards a Formal Approach," in *IFIP congress* 1977, vol. 7, pp.155-160.

[7] D. S. Hirschberg, and J. B. Sinclair, "Decentralized Extrema-Finding In Circular

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

Configurations Of Processors," *Communications of the ACM,* vol. 23, pp. 627-628, 1980.

[8] E. Chang and R. Roberts, "An Improved Algorithm For Decentralized Extrema-Finding In Circular Configurations Of Processes," *Communications of the ACM,* vol. 22, pp. 281-283, 1979.

[9] H. Garcia-Molina, "Elections In A Distributed Computing System," *IEEE transactions on Computers,* vol. 100, pp. 48-59, 1982.

[10] D. Mitra, A. Cortesi, and N. Chaki, "ALEA: An Anonymous Leader Election Algorithm for Synchronous Distributed Systems," in *Progress in Image Processing, Pattern Recognition and Communication Systems*, 2021, pp. 46-58.

[11] V. K. Madisetti and S. Panda, "A Dynamic Leader Election Algorithm for Decentralized Networks," *Journal of Transportation Technologies*, vol.11, pp. 404-411, 2021.

[12] M. Abdullah, I. Al-Kohali, and M. Othman, "An Adaptive Bully Algorithm for Leader Elections in Distributed Systems," in *International Conference on Parallel Computing Technologies*, *Springer*, Cham, 2019, pp. 373-384.

[13] M. Seperhi and M. Godarzi, "Leader election algorithm using heap structure," in *12th WSEAS Int. l Conf. on Computers*, 2008, pp. 23-25.

[14] M. Refai, K.Alkheder, and I. Aloqaily, "Leader Election Algorithm In Honeycomb Torus Network," *Journal of Engineering Technology*, vol. 5, pp. 72-92, 2016.

[15] M. Refai, "A new leader Election Algorithm in Hypercube Networks," in *Symposium Proceedings Volume II Computer Science & Engineering and Electrical & Electronics Engineering, European University of Lefke, North Cyprus, PP*, 2006.

[16] M. Refai, Y. Alraba'nah, M. Alauthman, A. Almomani M. Al-kasassbeh and M. Alweshah, "A Novel Leader Election Algorithm For Honeycomb Mesh Networks," *Journal of Theoretical and Applied Information Technology*, vol 97, pp. 3783 -3795, 2019.

[17] I. Stojmenovic, "Honeycomb networks: Topological properties and communication algorithms," in *IEEE Transactions on parallel and distributed systems 8.10*, 1997, pp. 1036-1042.

[18] I. SZCZEsniak, "The Hive Network and its Routing Algorithm," *Archive of Theoretical and Applied Informatics*, vol. 16, pp. 171-179, 2004.

[19] C. Nicopoulos, N. Vijaykrishnan and R. D. Chita, "A Novel Dimensionally-Decomposed Router for On-Chip Communication in 3D Architectures," *Network-on-Chip Architectures. Springer*, vol. 45, pp. 171-197, 2009.

[20] W. Mikanik, "Three-dimensional Variants of Honeycomb Networks," *Technical Report, Silesian University of Technology*, Gliwice, Poland, 1999.

[21] J. Carle, J. Myoupo and I. Stojmenovic, "Higher Dimensional Honeycomb Networks," *Journal of Interconnection Networks*, vol. 2, pp. 391-420, 2001.

[22] A. Yin, A. Wei, T. C. Xu, P. Liljeberg and Hannu, "Explorations of honeycomb topologies for network-on-chip," in *Sixth IFIP International Conference on Network and Parallel Computing*, 2009, pp. 73-79.