ASEE AMERICAN SOCIETY FOR ENGINEERING EDUCATION

Journal of Engineering Technology
ISSN: 0747-9964
Volume 5, Issue 1, January, 2016, pp.72-92

ORIGINAL ARTICLE

Leader Election Algorithm in the Honeycomb Torus Networks

Mohammed Refai [1], Khalid Alkheder [2] and Ibrahim Aloqaily[3]

1. Chairman of Department of Computer Science, Faculty of Information Technology, Zarqa University, Jordan
2. Department of Computer Science, Faculty of Information Technology, Zarqa University, Jordan
3. Department of computer science, Prince Hussein Bin Abdullah II for Information Technology, Hashemite University, Jordan

Abstract: Leader election algorithm (LEA) is a fundamental issue in communication networks and distributed systems; it allows electing a new leader (coordinator) for the system or the network in order to achieve the stability in the network when the leader failed .In this paper, we propose a new leader election algorithm in honeycomb torus networks. The algorithm aims to elect one node to be a new leader when the leader fails. The proposed algorithm uses nodes positions and Coordinates of honeycomb network to direct the election messages. Also it uses sequences of links for one path or more to identify message source and destination. Synchronization and concurrency are considered in the proposed algorithm to identify start and end of each phase in the algorithm.

We evaluated the performance of leader election algorithm in honeycomb network by calculating the number of messages and time steps. We found that the number of messages is less than or equal O ($n^{1.5}$) within O ($\sqrt{N}$) time steps.

Keywords: Leader election algorithm, Honeycomb torus network.

1. Introduction

Distributed systems are multiple autonomous computer systems that appear to the users of the system as a single computer, these systems are working together to solve a problem, the problem is divided into many computers or processors to execute different tasks. the leader node (coordinator) is responsible for managing the distributed system where all nodes are communicating together via different network topologies like mesh , ring , bus ,hypercube ,honeycomb..Etc. These topologies may be either hardware processors, or software processes embedded over other hardware topology ([1] and [2]).

The main goal of distributed system is to enhance the performance of problem solving where the tasks can be executed concurrently by distributing the load to reduce the response time unless any failure event may occur to the leader node or the link between nodes. The distributed systems need one node to act as a controller (leader) in centralized control [3]. If the leader fails for any reason, the LEA which is distributed over all nodes solves this problem by electing a new leader for the networks to manage the use of a shared resource in the distributed systems [4].

The leader introduces the perfect solution to keep consistency for distributed systems through the synchronization of messages thus it is the optimal method to break the symmetry in distributed systems. So, by electing a node as the leader, it is possible to act as the centralized controller for decentralized system [5]

ASEE AMERICAN SOCIETY FOR ENGINEERING EDUCATION
Journal of Engineering Technology
ISSN: 0747-9964
Volume 5, Issue 1, January, 2016, pp.72-92
ORIGINAL ARTICLE

LEA is the process of selecting a new coordinator (leader) after the previous leader has failed, so the algorithm starts when one node at least detects the failure event and completes its mission by electing the leader with the largest identifier. The election algorithm starts when the leader failure is detected by one node and terminates when the new leader is elected and all other nodes become aware of the new leader ([6] and [7]).
The honeycomb architecture has been used in different applications such as in wireless networks where the base stations arranged in a honeycomb network[8], the representation of benzenoid hydrocarbons in chemistry, computer graphics and image processing [9], The honeycomb torus is obtained by adding wrap around edges to the honeycomb mesh to achieve edge and vertex symmetry, the honeycomb mesh and torus provide an approximate 40% reduction of network cost where the network cost is an indication of both network performance and implementation cost [10].
This paper is organized as follows. Section 2 presents related work. Section 3 describes the honeycomb torus model structure and properties. Section 4 presents the proposed algorithm. Section 5 presents the evaluation performance of the algorithm. In section 6 we conclude the paper.

Related work.

Several researchers have suggested several leader election algorithms in different network topologies. However, to the best of our knowledge, there is no algorithm to solve the leader failure in honeycomb torus networks.
Leader algorithms have been widely studied. Proposed algorithms depend on the type of the topology to solve the leader failure in the network .The researchers presented many algorithms to solve the leader failure in different topologies ([11], [12], [6],[14], [15], [2], [16], [17], [18], [7], [19], [20], [3], [21] and [22]). In distributed systems, a major problem is the leader failure and the relevant leader election algorithm. The leader election problem has been studied extensively in various models in distributed computing, including:
- Static and dynamic networks ([22], [23], [13], and [6]).
- Node Identity (ID) (unique identity vs. anonymous ID) (Distinguished vs. not distinguished) [21].
- Type of the Topology (ring, tree, complete graph, meshes, torus, hypercube ...etc) ([18] and [22]).
- Communication mechanism used (synchronous vs. asynchronous) [16].
- Transmission media (wired vs. wireless or radio) ([26], [17], [20])

The leader election algorithms started at the end of the seventies and eighties in previous century when the researchers studied the leader failure in the ring and complete topology, they presented algorithms to solve the leader failure ([11] and [26]). After that they studied meshes, hypercube and tree topologies. To date, these topologies and wireless networks are still being studied.
(Refai and etl) proposed leader election algorithms in torus and hypercube. They solved the presence of one link failure ([27], [19] and [20]). Refai in [24] proposed leader election algorithm in 3D torus network with many links failure.

Bully algorithm was proposed by Molina G [4]. It solved the leader failure for a complete network. When a process detects the failure event it sends an election message with the higher number to all processes. If one node has a higher number than the received number it becomes the leader else the process that detects the failure event becomes the leader.
The authors presented a comparative analysis of various leader election algorithms and proposed a new leader election algorithm in MANET in an analytical way which considered different factors such as node's position, time complexity, message complexity, battery life and security [25].
The researchers introduced two algorithms to solve deterministic and probabilistic leader election in strong collision detection systems ([5] and [26]).

Journal of Engineering Technology
ISSN: 0747-9964
Volume 5, Issue 1, January, 2016, pp.72-92

ORIGINAL ARTICLE

In [28] they presented an algorithm that works effectively in the ad hoc distributed systems. The goal of the algorithm is to save energy in the ad hoc network by minimizing the number of exchanged messages between nodes.
In [29] they presented leader election algorithm for a group of nodes connected by a tree where the algorithm based on heap structure.

3. Model description and properties

The honeycomb torus network is similar to honeycomb mesh, except, in the connection, of adding wrap around edges. Figure (1) shows a three dimensional honeycomb torus network. In honeycomb torus network, interconnection topology is a torus graph with $N = 6D^2$ nodes, where D is the

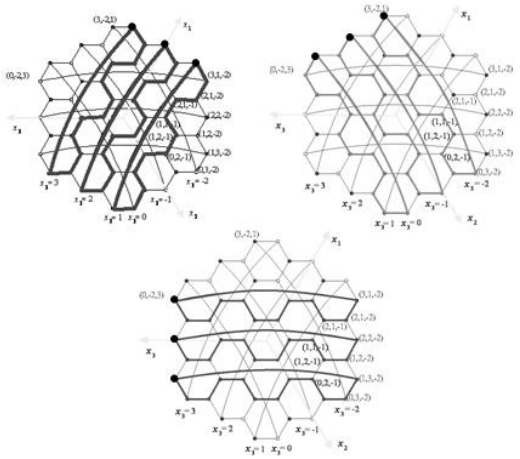


Fig. 1 Three level Honeycomb Torus Network.

Dimension of the honeycomb torus network. The number of links between all nodes is $9D^2$. The honeycomb torus is obtained by adding wrap around edges to the honeycomb mesh to achieve edge and vertex symmetry [8].

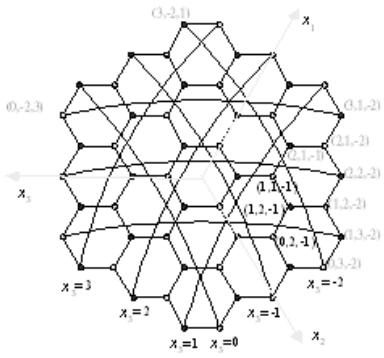


Fig. 2 3D honeycomb torus contains three rings parallel to each axis.

ASEE AMERICAN SOCIETY FOR ENGINEERING EDUCATION
Journal of Engineering Technology
ISSN: 0747-9964
Volume 5, Issue 1, January, 2016, pp.72-92
ORIGINAL ARTICLE

The honeycomb torus network with level (D) contains D rings parallel for each axis, for example in three level honeycomb torus network there are three rings parallel to X1 and three rings parallel to X2 and three rings parallel to X3 as shown in figure (2) where the black nodes represent the end of ring for each axis.

A node can send or receive messages simultaneously and uses all-port communication model where the node can send the message to all its neighbors at the same time.

The network uses two different types of nodes according to the summation of coordinates X1, X2 and X3 as shown in figure (3) where the summation of white and black nodes is equal to 1 and 2 respectively.

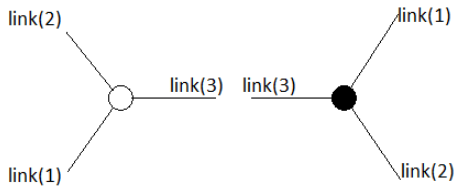


Fig. 3 Node links**.**

In this paper, we use the following model:

- The multicomputer consists of N nodes, labeled 0, 1, 2… N-1.
- The nodes physically form a three level honeycomb torus network.

This paper assumes the following:

- All communication links are bidirectional.
- Routers should work all the time even with faulty node because the fault is in leader properties.
- Leader node could fail due to different reasons which lead to lose of the leadership property. Other nodes can detect this failure, when the time out exceed without acknowledgement. Node that detects this failure starts the election algorithm.
- Each node calculates a weight that defines its relative importance. Then, compares it with the weight of other nodes that it has received and propagate the maximum weight. This weight is represented by distinguish identification (ID) for each node
- When the node compares the received ID with local ID it stores the higher ID in candidate ID variable.
- When the leader node crashes, its ID degrades to 0. So, it cannot win the election

Each node has the following variables:

- Candidate ID: a candidate node stores the higher ID which passed from it.
- Position: The label of its position.
- Leader ID: Leader position.
- Phase (The current phase during the algorithm execution) and step (The step number in this phase).
- State: leader, or normal, or candidate.

ASEE AMERICAN SOCIETY FOR ENGINEERING EDUCATION
Journal of Engineering Technology
ISSN: 0747-9964
Volume 5, Issue 1, January, 2016, pp.72-92
ORIGINAL ARTICLE

4 Proposed solution

This section describes and presents the proposed algorithm in phases. Each phase composed of many steps. Before describing the algorithm, the definition of the following variables will assist in understanding the algorithm:

1) Node State: during the execution of the algorithm each node will be in one of the following states:

- Normal: the network is normal and no leader failure is detected by this node.
- Candidate: there is a failure and the election process is in progress inside this node.
- Leader: only one node must have this state in a stable network, while this state is absent after leader failure [25].

The algorithm uses X1, X2 and X3 to represent the axes and x1, x2 and x3 to represent node position.

2) Phases: the proposed algorithm is composed of three phases, as follows:

a) Phase One: the node that detects leader failure (initiator) checks its position and sends an election messages toward link1 if it is located on one of the rings of X1-axis, link2 if it is located on one of the rings of X2-axis and link3 if it is located on one of the rings of X3-axis and the node becomes in candidate state.

The election message contains different paths according to the coordinates of initiator. The node that receives the election message reads the paths of the message and compares the received ID with local ID and selects the higher ID, after that the node becomes in candidate state and checks its position, the node sends an election messages to other rings depends on its position except the link it received the election message from it.

If the node is already in candidate state and the received ID higher than candidate ID it passes the election message according to its path without checking its position. This process will continue until the nodes that are located on the end of rings receives two election messages for the same node or receive one election message where the number of steps in the message is 4D. The results will be in specific nodes that will be explained next in this section.

b) Phase Two: when the node (1-D,0,D) finish phase one it sends election message through link 2 and compares between the candidate ID of the nodes that are located on the end of rings and have been finished phase one.

c) Phase Three: when the election message arrives at the last node in phase two the new leader becomes known to the last node in phase two, after that it broadcasts the information about the new leader to all nodes by sending a leader message.

The election message is composed of (message type, phase, steps with respect to X1,X2 and X3,paths of the message , greater ID, and position of the greater ID, message initiator).where there are three types of steps: step1 towards the X1 axis, step2 towards the X2 axis and step3 towards the X3 axis.

3) path : represents the way of the message where it follows a sequence of links . figure (4) shows an example that explains the path of the election message. The sequence (3132) means that the message will go through link 3 to the next node, the reieved node resends the message towards link1 to the next node and so on.

Journal of Engineering Technology
ISSN: 0747-9964
Volume 5, Issue 1, January, 2016, pp.72-92

ORIGINAL ARTICLE

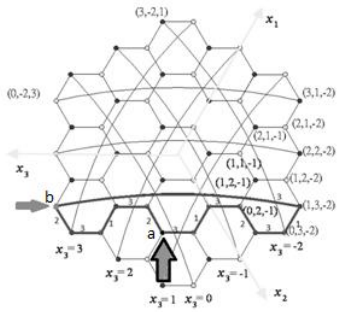


Fig. 4 An example explains the path of the election message**.**

Figure (4) shows that the node (a) detects the failure,then it sends two election messages in two direction to complete the ring along the X3 axis; message one follows the path toward link 3 according to the sequence (3132) and message two follows the path toward link 2 according to the sequence (2313). The messages will meet at the node (b) which represnts the end of the ring .

Now, let us explain the events in each phase of the proposed algorithm:

1) Phase One: the algorithm starts when a node (initiator) detects leader failure.The initiator changes it's state to candidate and make checks in its position as follow:

a) Check X1 : if X1=1 the initiator is located on one of the rings that parallel to X1, then it prepares the election message via determining the paths of election message with reference to X1 .for example if the value of x1 is odd the algorithm will add the path (1-1) to message 1 and path(2-1) to message 2.

b) Check X2: if X2=1 the initiator is located on one of the rings that are parallel to X2, then it prepares the election message via determining the paths of election message with reference to X2 . for example if the value of x2 is even the algorithm will add the path (1-3) to message 1 and path(2-3) to message 2.

c) Check X3: if X3=1 the initiator is located on one of the rings that parallel to X3, then it prepares the election message via determining the paths of election message with reference to X3. for example if the value of x3 is odd the algorithm will add the path (3-3) to message 3 and path(2-4) to message 2.

According to check X1,X2 and X3 ,the initiator prepares three election messages: message 1,message 2 and message 3 and update the number of steps in each message , after that it sends message 1 towards link1(X1 axis), message 2 towards link2 (X2 axis), and message 3 towards link 3 (X3 axis).

Each node receives election message do the following:

- If a candidate node receives another election message where the candidate ID is higher than the received ID, then the election message will be neglected. Otherwise it will read the election message.
- It extracts the paths from the message and reads the sequence of path and its reference. The election message contains one path or more where the path represents a sequence of links.
- There are three possible paths related to the reference:

1) The path with reference to X1: this means that the message follows this path to complete the ring which is parallel to X1.

Journal of Engineering Technology
ISSN: 0747-9964
Volume 5, Issue 1, January, 2016, pp.72-92

ORIGINAL ARTICLE

- If a node received a path with reference to X1 it first checks if this node is the end of ring through verifying the following condition as shown in figure (5) :

$$x1 = (D) \,\&\, \sum(x1, x2, x3) = 1 \qquad (1)$$

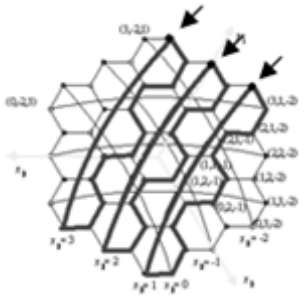


Fig. 5 The position of nodes that satisfy the conditions in relation (1).

- If the condition is true the election message will be stopped, otherwise the message will continue its path toward X1.
- The sequence will be shifted by one, for example if the sequence of path is (1234) the new sequence is (2341). Also the number of step1 will be incremented.
- The algorithm compares the received ID with local ID and candidate ID and selects the greater ID.
- The algorithm will select the appropriate message to add the new sequence to it, so if the first link is 1 it selects message1, if the first link is 2 it selects message 2 and if the first link is 3 it selects message 3, for the previous example the new sequence is (2341).so it will select message 2 to add the sequence (2341) to it.
- The number of step1 is updated.

2) The path with reference to X2: this means that the message follows this path to complete the ring which is parallel to X2.

- If a node received a path with reference to X2 it first checks if this node is the end of the ring through verifying the following condition as shown in figure (6) :

$$x2 = (1\text{-}D) \,\&\, \sum(x1, x2, x3) = 2 \qquad (2)$$

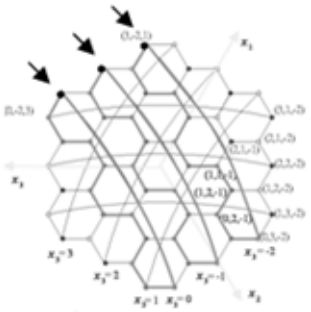


Fig. 6 The position of nodes that satisfy the conditions in relation (2).

ASEE AMERICAN SOCIETY FOR ENGINEERING EDUCATION
Journal of Engineering Technology
ISSN: 0747-9964
Volume 5, Issue 1, January, 2016, pp.72-92
ORIGINAL ARTICLE

- If the condition is true the election message will be stopped, otherwise the message will continue its path toward X2.
- The sequence will be shifted by one and the number of step2 will be incremented.
- The algorithm compares the received ID with local ID and candidate ID and selects the greater ID.
- The algorithm will select the appropriate message to add the new sequence to it, so if the first link is 1 it selects message1, if the first link is 2 it selects message 2 and if the first link is 3 it selects message 3.
- The number of step2 is updated.

3) The path with reference to X3: this means the message follows this path to complete the ring which is parallel to X3.

- If a node received a path with reference to X3 it first checks if this node is the end of the ring through verifying the following condition as shown in figure (7) :

$$x3 = (D) \,\&\, \sum(x1, x2, x3) = 1 \qquad (3)$$

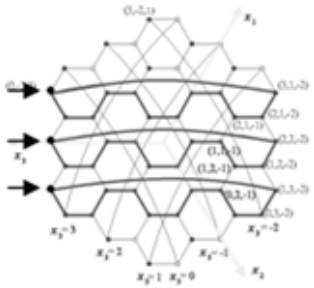


Fig. 7 The position of nodes that satisfy the conditions in relation (3).

If the state of the node is normal it will also do the following steps, it changes its state to candidate state and makes the following checks:

- It first verifies if the current node is located on the end of ring through verifying the last nodes conditions condition and finish phase one if true. The algorithm will verify from the unchecked axes in the message. For example if the election message contains only one path with reference to X1 it will check X1 and Check X2 to verify if the current node is located on one of the rings that parallel to X1 and X2, respectively.
- If X1=1 this means that the node is located on one of the rings that parallel to X1, then it prepares the election message via determining the paths of election message with reference to X1 and updates the number of step1.
- If X2=1 this means that the node is located on one of the rings that are parallel to X2, then it prepares the election message via determining the paths of election message with reference to X2 and updates the number of step2.
- If X3=1 this means that the node is located on one of the rings that parallel to X3, then it prepares the election message via determining the paths of election message with reference to X3 and updates the number of step3.

Eventually the node prepares the election messages and sends message 1 towards link1(X1 axis), message 2 towards link2 (X2 axis) and message 3 towards link 3 (X3 axis).

ASEE AMERICAN SOCIETY FOR ENGINEERING EDUCATION
Journal of Engineering Technology
ISSN: 0747-9964
Volume 5, Issue 1, January, 2016, pp.72-92
ORIGINAL ARTICLE

Phase one ends when the nodes that satisfy any of the relations (1,2and3) receive two election messages for the same node or receive one election message where the number of steps in the message is 4D. After that the node compares between the elections messages to select the greater ID.

2) Phase two: When the node (1-D, 0, D) completes phase one, it starts phase two by sending election message on link 1 according to the sequence (12) to compare the candidate ID of nodes that the summation of its coordinates (x1,x2 and x3) is 1 and finished phase one.

When the election message reach the node (0,1-D,D) it sends election message on link 1 according to the sequence (13) to compare the candidate ID of nodes that the summation of its coordinates (x1,x2 and x3) is 2 and finished phase one.

The election message continues its path until reach the node (D,1-D,1) then it sends election message on link 3 according to the sequence (32) to compare the candidate ID of nodes that the summation of its coordinates (x1,x2 and x3) is 1 and finished phase one.

When the election message reach the node (D,0,1-D) it stops the election message and compares the received ID with the ID's numbers of the election messages from links (1 and 2) and select the greater ID. So the coordinates of greatest ID number for the node that aware to be a new leader is in this node. Figure (8) summarize the paths of phase two

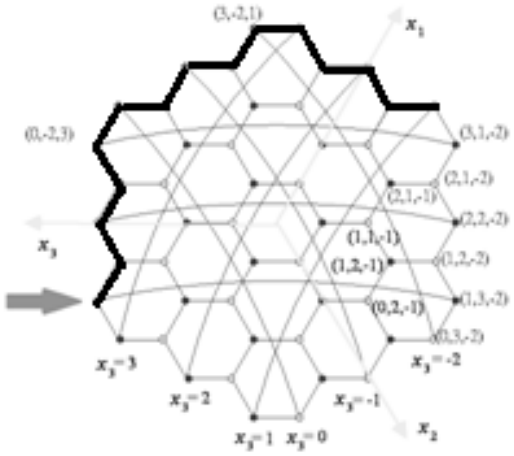


Fig.8 The paths election message in phase two

3) phase three: after the node (D,0,1-D)completes phase two, phase three will start by broadcasting the information about the new leader node to all nodes in the honeycomb torus network by sending leader message through links 1, 2 and 3.

When the candidate nodes receive the leader message it changes its state to normal state and broadcast the leader message through its links except the link it received from it and the normal nodes ignore the leader messages when receives it.

The leader election algorithm is:

Part one (functions)

Check X1:

Read the coordinates of node

ASEE AMERICAN SOCIETY FOR ENGINEERING EDUCATION
Journal of Engineering Technology
ISSN: 0747-9964
Volume 5, Issue 1, January, 2016, pp.72-92
ORIGINAL ARTICLE

If the values of x2 and x3 don't satisfy one of the following relations:
((x3≥1 & x2≤x3– D) OR (x3≤ -1 & x2> x3+D))
{
X1=1
}
Check X2:
Read the coordinates of node
If t he values of x1 and x3 don't satisfy one of the following relations:
((x1≥1 & x3≤x1– D) OR (x1≤ -1 & x3>x1+D))
{
X2=1
}
Check X3
Read the coordinates of node
If the values of x1 and x2 don't satisfy one of the following relations:
(x2≥2 & x1<x2– D) OR (x2≤ 0 & x1≥x2+D)
{
X3=1
}
Message 1: the election message toward link1
Path 1-1: the election message with the reference to x1 accord to the sequence 1312
Path 1-2: the election message with the reference to x1 accord to the sequence 1213
Path 1-3: the election message with the reference to x2 accord to the sequence 1232
Path 1-4: the election message with the reference to x3 accord to the sequence 1323
Message 2: the election message toward link2
Path 2-1: the election message with the reference to x1 accord to the sequence 2131
Path 2-2: the election message with the reference to x2 accord to the sequence 2123
Path 2-3: the election message with the reference to x2 accord to the sequence 2321
Path 2-4: the election message with the reference to x3 accord to the sequence 2313

Message 3: the election message toward link3
Path 3-1: the election message with the reference to x1 accord to the sequence 3121
Path 3-2: the election message with the reference to x2 accord to the sequence 3212
Path 3-3: the election message with the reference to x3 accord to the sequence 3132
Path 3-4: the election message with the reference to x3 accord to the sequence 3231)
Part two (algorithm starts here)
If any node (the initiator) detects the leader failure, then
Phase 1(starts when a node detects the failure)
Step1=0
Step2=0
Step3=0
State = candidate
Check X1

ASEE AMERICAN SOCIETY FOR ENGINEERING EDUCATION
Journal of Engineering Technology
ISSN: 0747-9964
Volume 5, Issue 1, January, 2016, pp.72-92
ORIGINAL ARTICLE

If X1 is 1
{Prepare the election message
Step1 = Step1 +1
If x1 is odd then
{Add Path 1-1 to message 1 and path 2-1 to message 2
Else
Add path 1-2 to message 1 and path 3-1 to message 3
If X2=1
{Prepare the election message
Step2=step2+1
}
Update step1
}
Check X2
If x2 is odd then
{Add path 2-2 to message 2 and path 3-2 to message 3
Else
Add path 1-3 to message 1 and path 2-3 to message 2
}
Update step2
}
Check X3
If X3=1
{Prepare the election message
Step3=Step3+1
If x3 is odd then
{Add Path 3-3 to message 3 and Path 2-4 to message 2
Else
Add Path 1-4 to message 1 and Path 3-4 to message 3
}
Update step 3
}
If message1 has one path at least send it on link1
If message2 has one path at least send it on link2
If message3 has one path at least send it on link3

If the node received election message, then
{
If the state of node is candidate and received an election message where the candidate ID higher than the received ID, then neglect the election message
Else
Read the message
Extract the paths from the message

ASEE AMERICAN SOCIETY FOR ENGINEERING EDUCATION
Journal of Engineering Technology
ISSN: 0747-9964
Volume 5, Issue 1, January, 2016, pp.72-92

ORIGINAL ARTICLE

Read the sequence of path and its reference
If the path with reference X1
{If the coordinates of node satisfy the following relation
(x1=D and the summation of x1, x2 and x3 is 1)
{Stop the election message
Else
Put the first link in the sequence at last
Step1=step1+1
Compare received ID with local ID and candidate ID and select the winnerID
}
Select the appropriate message:
If the first link is 1, select message 1
If the first link is 2, select message 2
If the first link is 3, select message 3
Add the new sequence to the appropriate message
Update step1
}
If the path with reference X2
{If the coordinates of node satisfy the following relation
(x2= (1-D) and the summation of x1, x2 and x3 is 2)
{Stop the election message toward X2
Else
Put the first link in the sequence at last
Step2=step2+1
Compare received ID with local ID and candidate ID and select the winnerID
}
Select the appropriate message:
if the first link is 1, select message 1
if the first link is 2, select message 2
if the first link is 3, select message 3
Add the new sequence to the appropriate message
Update step2
}
If the path with reference X3
{If the coordinates of node satisfy the following relation
(x3=D and the summation of x1, x2 and x3 is 1)
{Stop the election message toward X3
Else
Put the first link in the sequence at last
Step3=step3+1
Compare received ID with local ID and candidate ID and select the winnerID
}
Select the appropriate message:

Journal of Engineering Technology
ISSN: 0747-9964
Volume 5, Issue 1, January, 2016, pp.72-92

ORIGINAL ARTICLE

If the first link is 1, select message 1
If the first link is 2, select message 2
If the first link is 3, select message 3
Add the new sequence to the appropriate message
Update step3
}
If the state of node is normal
{State =candidate
If the reference X 1is not found in the message then, Check X1
{If X1 is 1
{
Step1 = 1
If x1 is odd then
{Add Path 1-1 to message 1 and path 2-1 to message 2
Else
Add path 1-2 to message 1 and path 3-1 to message 3
}
Update step1
}
}
If reference X2 is not found in the message, then Check X2
{
If X2=1
{Step2=1
If x2 is odd then
{Add path 2-2 to message 2 and path 3-2 to message 3
Else
Add path 1-3 to message 1 and path 2-3 to message 2
}
Update step2
}
}
If the reference X3 is not found in the message, then Check X3
{If X3=1
{Step3=1
If x3 is odd then
{Add Path 3-3 to message 3 and Path 2-4 to message 2
Else
Add Path 1-4 to message 1 and Path 3-4 to message 3
}
Update step 3
}
}

ASEE AMERICAN SOCIETY FOR ENGINEERING EDUCATION
Journal of Engineering Technology
ISSN: 0747-9964
Volume 5, Issue 1, January, 2016, pp.72-92
ORIGINAL ARTICLE

If message1 has one path at least send it on link1
If message2 has one path at least send it on link2
If message3 has one path at least send it on link3
}
Part Three (Phase Two starts here)
If the node (x1= (1-D) and x2=0) receives the election message from links (2 and 3) that belong to the same node
Phase 2 start
Compare between the two messages and select the higher ID
Step=1
Prepare the election message
Send election message on link 1 accord to the sequence 12.
If the node receives the election message with the sequence 12, then
{
If the summation of x1, x2 and x3 is equal 1 and the two election messages received from links (2 and 3) belongs to the same node, then
{Step=step+1
Compare received ID with the ID's numbers of the election messages received from links (2 and 3) that belong to the same node and pass the higher ID.
If the election message reaches the node which has the coordinates x2= (1-D) and
x3=D, then
Send election message on link 1 accord to the sequence 13
}
}
If the node receives the election message with the sequence 13, then
{
If the summation of x1, x2and x3 is equal 2 and the two election messages received from links (1 and 2) that belong to the same node, then
{Step=step+1
Compare received ID with the ID's numbers of the election messages from links (1 and 2) and pass the higher ID.
Step=step+1
If the election message reaches the node that has the coordinates x1=D and x2= (1-D),
then
Send election message on link 3 accord to the sequence 32.
}
}
If the node receives the election message with the sequence 32, then
{
If the summation of x1, x2and x3 is equal 1 and the two election messages received from links (1 and 2) that belong to the same node, then
{Step=step+1
Compare received ID with the ID's numbers of the election messages from links (1 and 2) and pass the higher ID.
If the election message reaches the node that has the coordinates x1=D and x3= (1-D)
{Stop the election message

ASEE AMERICAN SOCIETY FOR ENGINEERING EDUCATION
Journal of Engineering Technology
ISSN: 0747-9964
Volume 5, Issue 1, January, 2016, pp.72-92
ORIGINAL ARTICLE

Compare received ID with the ID's numbers of the election messages received from links (1 and 2) and select the greater ID.

}

}

}

Part Four (Broadcasting the leader node, phase three starts here)

If the node that has the coordinates x1=D and x3= (1-D) receives the election message, then

Phase 3

{Select the highest ID number.

Prepare the leader message.

Send the leader message on link 1, 2 and 3.

If a node received leader message from link 1

{ State =normal

If the node received two leader messages, then neglect the second message.

Send leader message on link 2 and 3

}

If a node received leader message from link 2

{ State =normal

If the node received two leader messages, then neglect the second message.

Send leader message on link 1 and 3

}

If a node received leader message from link 3

{ State =normal

If the node received two leader messages, then neglect the second message.

Send leader message on link 1 and 2

}

}

5 Performance evaluation

In this section we evaluate the algorithm by calculating the number of time steps and messages in simple case when one node only detects the leader failure or worst case when all nodes detect the leader failure.

5.1 Simple case

1) Number of messages:

Phase one: the node that detects the failure sends at most three messages if the initiator is located on three rings and the nodes that receive the election message send at most two messages because it does not send election message toward the link it received from it. So, the number of messages needed for this phase is given in relation (4) where N is the number of nodes in the honeycomb torus network.

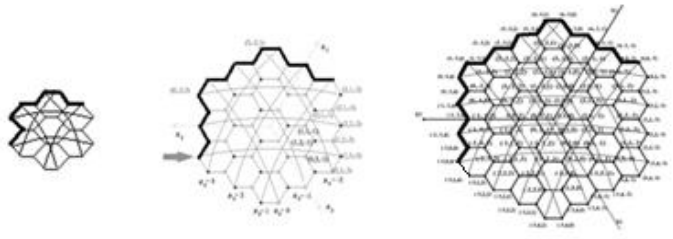
$$2N+1 \qquad (4)$$

Journal of Engineering Technology
ISSN: 0747-9964
Volume 5, Issue 1, January, 2016, pp.72-92

ORIGINAL ARTICLE

Phase two: Figure (9) shows the path of election message in phase two for different honeycomb torus networks (2D, 3D and4D), the number of election messages needed to complete phase two in level two is (8) messages where the number of nodes in the network is (24) nodes, while in level three it needs (14) messages where the number of nodes in the network is (54) nodes, Also the number of election messages in level four is (20) messages where the number of nodes in the network is (96) nodes. So, for any honeycomb torus network the number of messages required for phase two is calculated by the following relation:

$6\sqrt{N/6} - 4$ (5)



(a) (b) (c)

Fig.9 Phase two for 2D, 3D and 4D honeycomb torus network

Phase three: The first node in phase three sends three leader messages toward link 1, 2 and 3.when the other nodes receives the leader message from one link, it resends two leader messages toward the other two links .this means that all nodes send will 2N messages except the first node in phase three where it will send three messages. So, the total number of messages needed to complete phase three is:

$2N + 1$ (6)

The total number of messages in all phases is:

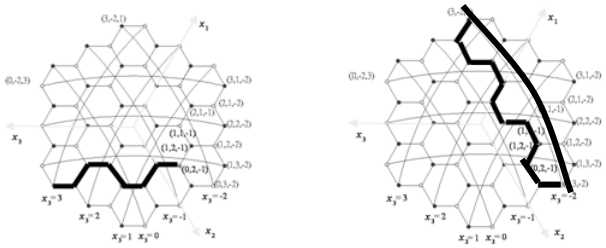
$(2N+1)+ (6\sqrt{N/6} - 4) + (2N+1) = O(N)$ messages (7)

2) Number of time steps:

Phase one: the number of time steps depends on the position of initiator that starts the election process, if the initiator locates on three rings it needs only $4\sqrt{N/6}$ time steps to accomplish phase one because all nodes execute the LEA in parallel, but if the imitator locates on one or two rings the number of time steps will change. Figure(10) shows the maximum number of time steps needed to complete phase one when the imitator locates on one ring, when a node detects the failure, it needs $(4\sqrt{N/6} - 5)$ time steps to send the election message to the nearest node in the last ring (0 ,2,-1) as shown in figure (10-a) .the node (0,2,-1) needs at maximum $(4\sqrt{N/6})$ time steps to complete phase one as shown in figure (10-b).

Journal of Engineering Technology
ISSN: 0747-9964
Volume 5, Issue 1, January, 2016, pp.72-92

ORIGINAL ARTICLE



(a) (b)

Fig.10 Maximum number of time steps in phase one

The total number of time steps needed in phase one is

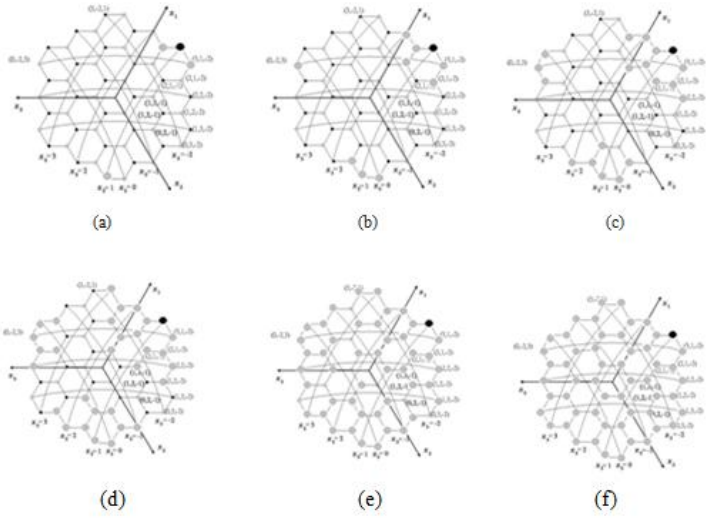
$$(4\sqrt{N/6}-5) + (4\sqrt{N/6}) \quad (8)$$

Phase two: The number of time steps is equal to the number of messages in phase two .so the number of time steps needed to complete phase two is:

$$6\sqrt{N/6} - 4 \quad (9)$$

Phase three: The number of time steps is equal to the number of time steps in the ring divided by 2 because of the nature of the topology where there are wraps around the edges, for example in 3D honeycomb torus the last node in phase two needs (6) time steps to broadcast the leader message to all nodes as shown in figure (11).so the number of time steps needed to complete phase three in the honeycomb torus network is:

$$2\sqrt{N/6} \quad (10)$$



(a) (b) (c) (d) (e) (f)

Fig.11 Phase three in 3D honeycomb torus network where it needs (6) time steps for broadcasting leader message.

The total number of time steps in all phases is:

$$((4\sqrt{\frac{N}{6}} - 5) + (4\sqrt{\frac{N}{6}})) + (6\sqrt{\frac{N}{6}} - 4) + 2\sqrt{\frac{N}{6}} + 5 = O(\sqrt{N}) \quad (11)$$

Journal of Engineering Technology
ISSN: 0747-9964
Volume 5, Issue 1, January, 2016, pp.72-92

ORIGINAL ARTICLE

5.2 Worst case:

1) Number of messages

Phase one: the number of messages depends on the ID numbers for the nodes when the message faces a node with higher ID, the node will pass it to the next nodes but if message faces a node with lower ID, the message will be stopped. To calculate the number of messages in the worst case we will assume the identifiers of nodes is arranged in ascending order as shown in figure (12).

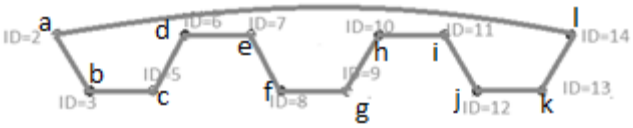


Fig.12 Arranging the identifiers in the ring

In this case when all nodes detects the leader failure ,it send election message in two directions, when the node receives election message it compares the received ID with its own ID . If the received ID higher than local ID, the node will pass the message to other nodes, otherwise the message will be neglected .in figure (12) the node (a) sends message to node (b) ,but node (b) will neglect the message because it has lower ID than the local ID for node (b) .Also during this time node (b) will send a message to node (c) and node (c) will neglect the message for the same reason .This process will be the same for the nodes (d, e, f, g, h, I, j and k) .node (l) will send a message to node (a) which represents the end of ring and it will stop . The algorithm needs (4D) to do these procedures.

Moreover, in each ring when the node sends election message toward another direction where each nodes has lower ID than the received ID, so the node will pass the message toward its neighbor. In step one node (b) sends message to node (a) , node (c) sends message to node(b) , node (d) sends message to node(c) , node (e) sends message to node(d) , node (f) sends message to node(e) , node (g) sends message to node(f) , node (h) sends message to node(g) , node (i) sends message to node(h) , node (j) sends message to node(i) , node (k) sends message to node(j) and node (l) sends message to node(k) . all messages will arrive to node (a) because each message will face a node which has lower ID .this means it requires one message from node (b) to node (a) ,two messages from node (c) to node (a),three messages from node (d) to node (a),four messages from node (e) to node (a) and so on. So, the total number of messages for each ring is given in relation (12):

$$1+2+3+4+\ldots+4D=\sum_{0}^{4D} i \qquad (12)$$

The number of rings in the honeycomb torus is 3D .so, the total number of messages for all rings in phase one is :

$$(4D + \sum_{0}^{4D} i) \times (3D) = O(D3) \qquad (13)$$

Phase two: the number of messages for phase two is the same as the number of messages in simple case which is calculated by the following relation:

$$6\sqrt{\frac{N}{6}} - 4 \qquad (14)$$

ASEE AMERICAN SOCIETY FOR ENGINEERING EDUCATION
Journal of Engineering Technology
ISSN: 0747-9964
Volume 5, Issue 1, January, 2016, pp.72-92
ORIGINAL ARTICLE

Phase three: the number of messages for phase two is same as the number of messages in simple case which is calculated by the following relation:

$2N+1$ (15)

The total number of messages in all phases is:

$((4D + \sum_{0}^{4D} i) \times (3D) + (6\sqrt{\frac{N}{6}} - 4) + (2N+1)$ (16)

After substituting $D=\sqrt{N/6}$ in equation (16) the total number of messages in phase one is O (n1.5).

2) Number of time steps:

Phase one: when all nodes detect the leader failure ,each ring needs the same number of time steps to complete this phase which is equal the number of nodes in each ring .so, the number of time steps needed for phase one is calculated by the following relation:

$$4\sqrt{\frac{N}{6}} \quad (17)$$

Phase two: The number of time steps is equal to the number of time steps in phase two .so the number of time steps needed to complete phase two is:

$$6\sqrt{\frac{N}{6}} - 4 \quad (18)$$

Phase three : The number of time steps is equal to the number of time steps in phase two which is equal the number of steps in the ring divided by 2 because the nature of topology where there are wraps around the edges .so the number of time steps needed to complete phase three is:

$$2\sqrt{\frac{N}{6}} \quad (19)$$

The total number of time steps in all phases is:

$$(4\sqrt{\frac{N}{6}}) + (6\sqrt{\frac{N}{6}} - 4) + (2\sqrt{\frac{N}{6}}) + 5 = O(\sqrt{N}) \quad (20)$$

6. Conclusions.

In this work, we proposed leader election algorithm in Honeycomb torus network. This algorithm consists of three phases. Each phase has many steps. In phase one the node that detects leader failure change its state from normal to candidate and sends an election messages toward one of the rings depends on its position. The election message contains different paths according to the coordinates of initiator. The node that receives the election message continues the election process with higher ID. The results phase one will be in specific nodes. In phase two, election is making in nodes that finished phase one to conclude the result in one node. In phase three this node broadcasts the information about the new leader to all nodes by sending a leader message. Number of messages needed to complete the algorithm is less than or equal to O (n1.5) within O ($\sqrt{N}$) time steps.

Journal of Engineering Technology
ISSN: 0747-9964
Volume 5, Issue 1, January, 2016, pp.72-92

ORIGINAL ARTICLE

This work can improved to deal with intermittent or complete link failure in dynamic algorithm in the future and researchers think to study leader election algorithms in other topologies like hive and hyper mesh networks.

References

1. Coulouris G, Dollimore J, Kindberg T and Blair G (2012) DISTRIBUTED SYSTEMS Concepts and Design. Addison –Wesley.
2. Kumar V, Grama A, Gupta A and Karypis G (2003) Introduction to Parallel Computing. The Benjamin/Cumminy Publishing Company.
3. Tanenbaum A (2002) Distributed Systems. Prentice-Hall International.
4. Gawali P D (2012) Leader Election Problem in Distributed Algorithm. International Journal of Computer Science and Telecommunications, Vol. 3, Issue 1
5. Ghaffari M and Haeupler B (2013) Near Optimal Leader Election in Multi-Hop Radio Networks. Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA'13).
6. Devillers M, Griffioen D, Romijn J and Vaandrager F (2004) Verification of Leader Election Protocol. Formal Method Applied to IEEE 1394, Springer International journal on Software Tools for Tecknology Transfer (STTT).
7. Radeva T (2013) Properties of Link Reversal Algorithms for Routing and Leader Election. Master's thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology.
8. Stojmenovic I (1997) Honeycomb Networks: Topological Properties and Communication Algorithms. IEEE Transactions on Parallel and Distributed Systems, vol. 8, no.10.
9. Liu K Y, Zhang B H and Li H L (2010) Fundamental Algorithms of Computer Graphics and the Needs for New Display Materials. International Journal of the Society of Materials Engineering for Resources, Vol.17, No.2.
10. Yin W A, Xu T C, Liljeberg P and Tenhunen H (2009) Explorations of Honeycomb Topologies for Network-on-Chip. Sixth IFIP International Conference on Network and Parallel Computing.
11. Akbar B and Effatparvar M (2006) Bully Election Algorithm Improvement with New Methods and Fault Tolerant Mechanism. Symposium Proceedings Volume II Computer Science & Engineering and Electrical & Electronics Engineering, PP 501-506.
12. Al Shayeji H M, Al-Azmi R A, AbdulAziz R, Al-Azmi R A and Samrajesh D M (2011) Analysis and Enhancements of Leader Elections algorithms in Mobile Ad Hoc Networks. Proc. of Int. Conf. on Advances in Information and Communication Technologies.
13. Ingram R, Radeva T, Shields P, Viqar S, Walter E J and Welch L J (2013) A Leader Election Algorithm for Dynamic Networks with Causal Clocks. Distributed Computing, 26(2):75-97.
14. Katwala H and Shah S (2003) Study on Election Algorithm in Distributed System. IOSR Journal of Computer Engineering, Volume 7, Issue 6, Dec. 2012.
15. Levitin A V(2002) Introduction to The Design and Analysis of Algorithms . Addison Wesley Company.
16. Lynch N, Radeva T and Sastry S (2012) Asynchronous Leader Election and MIS Using Abstract MAC Layer. Proceedings of FOMC 2012 (8th ACM International Workshop on the Foundations of Mobile Computing.
17. Marckert J F (2005) Quasi-Optimal Leader Election Algorithms in Radio Network with Log-Logarithmic Awake Time Slots.INRIA, pp.97-100.
18. Refai M, Oqily I, Alhamori A (2012) Leader Election Algorithm in 3D Torus Networks with the Presence of One Link Failure. World of Computer Science and Information Technology Journal (WCSIT), ISSN: 2221-0741, Vol. 2, No. 3, 90-97.
19. Refai M, Shari'ah A, Alshammari F (2010) Leader Election Algorithm in 2D Torus with the Presence of One Link Failure. IAJIT, Vol. 7, No. 2.

Journal of Engineering Technology
ISSN: 0747-9964
Volume 5, Issue 1, January, 2016, pp.72-92

ORIGINAL ARTICLE

20. Vasudevan S, DeCleene B, Immerman N, Kurose J and Towsley D(2003) Leader Election Algorithms for Wireless Ad Hoc Networks. In Proc. Of IEEE DISCEX III.
21. Yamshita M and Kammeda T (1999) Leader Election Problem on Networks in which Processor Identity Numbers are not Distinct. IEEE Transactions on Parallel and Distributed Systems, VOL.10, No.9.
22. Chang C C and Tsai J (2012) An Efficient Leader Election Algorithm of Performance-Related Characteristics for Dynamic Networks. International Conference on Smart Grid Systems (ICSGS), vol.45.
23. Flocchini P and Mans B (1996) Optimal Elections in Labeled Hypercube. Journal of Parallel and Distributed Computing 33, Article No. 0026, pp. 76-83.
24. Refai M (2012) Dynamic Solutions for Leader Failure in 3D Torus Networks.IAJIT, Vol.4, No.3, 31-37.
25. Bhoir S and Vidhate A (2013) A Modified Leader Election Algorithm for MANET. International Journal on Computer Science and Engineering (IJCSE), Vol. 5 No. 02.
26. Ghaffari M, Lynch N, and Sastry S (2012) Leader Election Using Loneliness Detection. Distributed Computing, 25(6): 427-450.
27. Refai M and Ajlouni N (2006) A new leader Election Algorithm in Hypercube Networks, Symposium Proceedings Volume II Computer Science & Engineering and Electrical & Electronics Engineering, European University of Lefke, North Cyprus, PP 497-501.
28. Al slaity N A and Alwidian A S (2012) A K-Neighbor-based, Energy Aware Leader Election Algorithm (KELEA) for Mobile Ad hoc Networks. International Journal of Computer Applications, Volume 59– No.19.
29. Sepehri M and Goodarzi M (2008) Leader Election Algorithm Using Heap Structure, 12th WSEAS International Conference on Computers.