Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 6, June 2014, pg.374 – 379

RESEARCH ARTICLE

Leader Election Algorithms in Distributed Systems

Seema Balhara[1], Kavita Khanna[2]

Student, CSE Department, PDM College of Engineering for Women[1]

Vice Principal, PDM College of Engineering for Women[2]

Balhara.nikki86@gmail.com, kvita.khanna@gmail.com

Abstract:* A Distributed system is an application which executes a collection of protocols to coordinate the actions of multiple processes on a network, such that all components cooperate together in order to perform a single or small set of related tasks. It is very difficult for processes to cooperate with each other because of failures of any of the process during communication. The leader election is critical problem in distributed system as data is distributed among different node which is geographically separated. To maintain co-ordination between the node, leader node have to be selected. This paper contains the information about the various existing leader election mechanisms which is used for selecting the leader in different problems.

Keywords: coordination, distributed system, election algorithm

I. INTRODUCTION

The leader election is important problem in distributed system as data is distributed among different node which is geographically separated. Designating a single node as an organizer in distributed systems is a challenging issue that calls for suitable election algorithms [1]. In distributed systems, nodes communicate with each other using shared memory or via message passing. To execute any distributed task effectively. The key requirement for nodes is coordination. In a pure distributed system, there does not exist any central controlling node that arbitrates decisions and therefore, every node has to communicate with the rest of the nodes in the network to make a proper decision. Often during the decision process, not all nodes make the same decision, thus the communication between nodes time-consuming and decision-making process. Coordination among nodes becomes very difficult when consistency is needed among all nodes. Centralized controlling nodes can be selected from the group of available nodes to reduce the complexity of decision making [2]. Many distributed algorithms require one node to act as coordinator, initiator, or otherwise perform some special role. Leader election is a technique that can be used to break the symmetry of distributed systems. In order to determine a central controlling node in a distributed system, a node is usually elected from the group of nodes as the leader to serve as the centralized controller for that decentralized system. The purpose of leader election is to choose a node that will coordinate activities of the system. In any election algorithm, a leader is chosen based on some criterion such as choosing the node with the largest identifier. Once the leader is elected, the nodes reach a particular state known as terminated state. In leader election algorithms, these states are partitioned into elected states and non-elected states. When a node enters either state, it always remains in that state [3]. Every leader election algorithm must be satisfied by the safety and liveness condition for an execution to be admissible. The liveness condition states that every node will eventually enter an elected state or a non-elected state. The safety condition for leader election requires that only a single node can enter the elected state and eventually, become the leader of the distributed system [4]. Information is exchanged between nodes by transmitting messages to one another until an agreement is reached. Once a decision is made, a node is elected as the leader and all the other nodes will acknowledge the role of that node as the leader [5].

II. ELECTION ALGORITHMS

Many distributed election algorithms have been proposed to resolve the problem of leader election. Among all the existing algorithms, the most prominent algorithms are as:

a. Bully Algorithm presented by Gracia-Molina in 1982.
b. Improved Bully Election Algorithm in Distributed System presented by A.Arghavani in 2011.
c. Modified Bully Election Algorithm in Distributed Systems presented by M.S.Kordafshari and group.
d. Ring Algorithm
e. Modified Ring Algorithm

III. BULLY ALGORITHM

Bully Algorithm is one of the most promising election algorithms which were presented by Gracia Molina in 1982.

```
Algorithm:
If (Node n realizes(leader crash))
sends an ELECTION message to all node with higher number
If(response from higher node)
select as LEADER.
&send ok message back to the sender to indicate that he is alive and will take over.
else node n is considered as LEADER
```

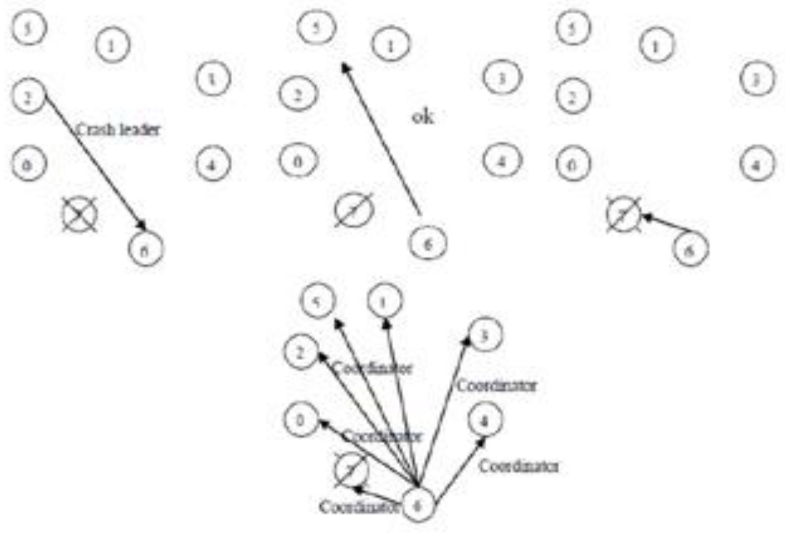

Fig1: Bully algo.

Disadvantages:

Bully algorithm has following disadvantages.

- It required that every process should know the identity of every other process in the system so it takes very large space in the system.
- It has high number of message passing during communication which increases heavy traffic .the message passing has order o (n2).

IV. IMPROVED BULLY ELECTION ALGORITHM

This algorithm is presented by A.arghavani, E.ahmadi, A.T.haghighat in 2011. It also overcomes the disadvantages of the original bully. The main concept of this algorithm is that the algorithm declares the new coordinator before actual or current coordinator is crashed.

Algorithm:

This algorithm selects the coordinator before current coordinator is crashed. So it takes extra stages. In this algorithm before the coordinator is failed, the current coordinator tries to gather information about processes in the system and through the current coordinator, declares the next possible coordinator to the processes. With increasing knowledge and get the id of all other process, a process with the bigger id attempts to execute the bully algorithm.
If the coordinator is failed, each process that notices this failure compares its id with the id which it has received via the coordinator. And select the new coordinator

Disadvantages

- It has complex structure.
- Every time process updates its database.
- Large database required to maintain the information of each process in database of every process.

V. MODIFIED ELECTION ALGORITHM

The Modified bully Election Algorithm is presented by M.S. Kordafshari, M.gholipour, M.jahanshahi, A.T.haghighat in 2005.the algorithm resolve the disadvantages of the bully algorithm.

Algorithm

1. When any process p notices that coordinator is not responding, it initiates an election and send election message to all process with higher priority number.
2. If no process responds, process P wins the election and becomes new coordinator.
3. Process with the higher priority sends ok message with its priority number to process P.
4. When process p receive all the response it select the new coordinator with the highest priority number process and sends the grant message to it.
5. Now the coordinator process will broadcast a new coordinator message to all other process and informs itself as a coordinator.

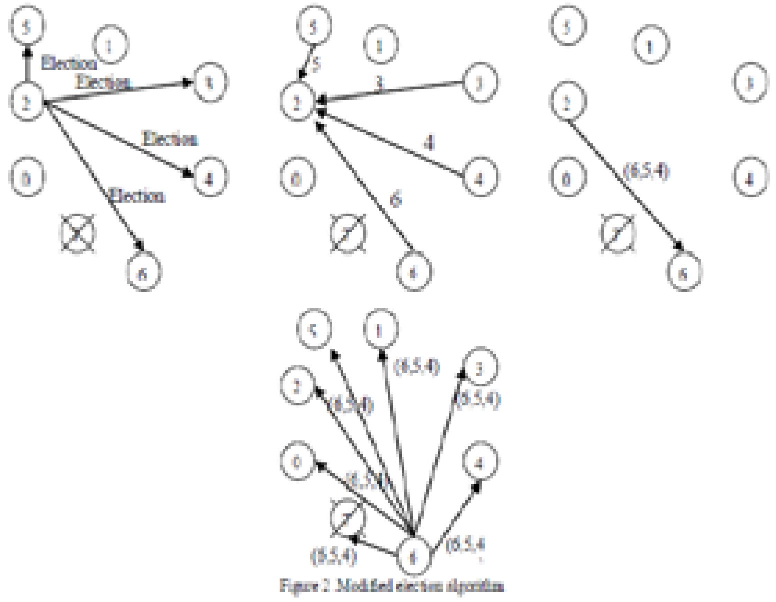

Fig2: modified election algo

Disadvantages

- A modified algorithm is also time bounded.
- It is better than bully but also has o (n2) complexity in worst case.
- It is necessary for all process to know the priority of other.

VI. RING ALGORITHM

This election algorithm is based on the use of a ring. We assume that the processes are physically or logically ordered, so that each process knows who its successor is [9].

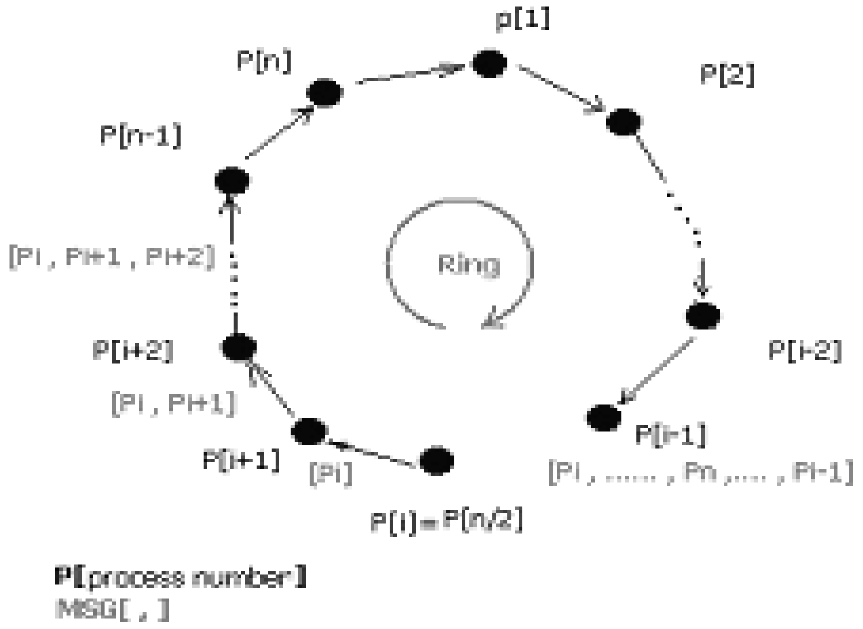

Fig3. Ring algo

When any process notices that the coordinator is not functioning, it builds an ELECTION message containing its own process number and sends the message to its successor. If the successor is down, the sender skips over the successor and goes to the next number along the ring, or the one after that, until a running process is located. At each step, the sender adds its own process number to the list in the message. Eventually, the message gets back to the process that started it all. That process recognizes this event when it receives an incoming message containing its own process number. At that point, the message type is changed to COORDINATOR and circulated once again, this time to inform everyone else who the coordinator is (designated by the list member with the highest number) and who the members of the new ring are. When this message has circulated once, it is removed and everyone goes back to work. From fig. 3, Process P[i] sends one token with “ELECTION” title to the next process in the ring. (to P[i+1]). If next process doesn’t reply, then process P will suppose that next process is down and will send “ELECTION” message to second next process in the ring (to P[i+2]). P[i] repeats this works until finds next process that is not in down status and has Received the “ELECTION” message. Founded process repeats this work too, and this will go on till election message returns back to the process P[i]. Each process when receives “ELECTION” message; before forwarding that, puts its process number in that message. When the “ELECTION” message returns back, then P[i] will choose the biggest process number as coordinator (between process numbers in the “election” message). Then it wheels a “COORDINATOR” message in the ring to announce new coordinator. In this case, we have two rounds of “message passings” in the ring. First for “CRASH” and “ELECTION” massages and second for “COORDINATOR” message. So we have 2 steps and the number of messages and average time are as below:
Number of messages = 2*n =2n => O(n)
Latency = 2*n =2n => O(n)

VII. MODIFIED RING ALGORITHM

In fig (a) nodes 2 & 4 notices that the coordinator has crashed simultaneously. In fig (b) they send their IDs into the ring. In fig (c,d), the greatest ID always remains in the ring. In fig (e), 5 is declared as Leader.

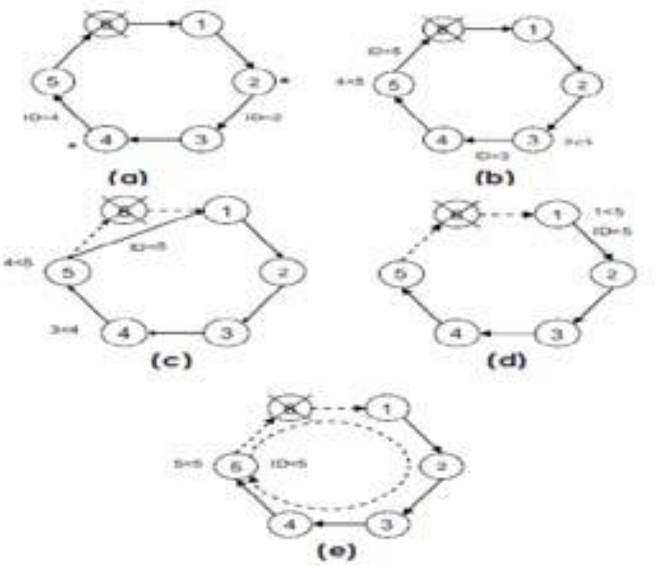

Fig4. Modified ring algo.

When a node notices that the leader has crashed, it sends its ID number to its neighboring node in the ring. Thus, it is not necessary for all nodes to send their IDs into the ring. At this moment, the receiving node compares the received ID with its own, and forwards whichever is the greatest. This comparison is done by all the nodes such that only the greatest ID remains in the ring. Finally, the greatest ID returns back to the initial node. If the received ID equals that of the initial sender, it declares itself as the leader by sending a coordinate message into the ring. It can be observed that this method dramatically reduces the overhead involved in message passing. Thus, if many nodes notice the absence of the leader at the same time, only the message of the node with the greatest ID circulates in the ring thus, preventing smaller IDs from being sent. If n{i1,i2,··· ,im} is the number of nodes that concurrently detect

the absence of the crashed coordinator and n is the number of nodes in the ring, then the total number of messages passed with an order of O(n2) is as follows:
T = n{i1i2,··· ,im} × n.

VIII. CONCLUSION

A distributed system is a collection of computers. These computers can work together, but all systems are independent. A leader's election is a basic necessity for distributed systems. When a system is chosen as a leader, it should operate as a system's management; make final decisions and the like. There are several election algorithms available in Distributed system. In this paper we discussed the concept of some existing algorithms. In wireless networks, leader election has a variety of applications such as: key distribution, routing coordination, sensor coordination, and general control.

REFERENCES

[1] Mohammad Reza Effat Parvar, Nasser Yazdani, Mehdi Effat Parvar, Aresh Dadlani, Ahmad Khonsari,"Improved Algorithms for Leader Election in Distributed Systems".
[2] Tai Yun Kim,"A Leader Election Algorithm in a Distributed Computing System", Department of Computer Science, Korea University-1, Ga, Anam: Dong, Seoul, pp. 136-701, KOREA
[3] Gurdip Singh,"Efficient Distributed Algorithrris for Leader Election in Complete Networks", Department of Computer Science SUNY at Stony Brook.
[4] Yuan-Chieh Chow Kenneth C.K. Luo Richard Newman- Wolfe,"An Optimal Distributed Algorithm for Failure-driven Leader Election in Bounded-Degree Networks".
[5] Sung-Hoon Park, Yoon Kim Jeoung Sun Hwang, "An Efficient Algorithm for Leader-Election in Synchronous Distributed Systems".
[6] S.Park, Y.Kim and J.S.Hwang, "An Efficient Algorithm for Leader-Election in Synchronous Distributed Systems,"1999 IEEE TENCON.
[7] J. Brunekreef, J.-P. Katoen, R. Koymans, and S. Mauw,"Design and analysis of Dynamic leader election protocols in broadcast networks," *Distributed Computing*, vol.9
[8] Tai Woo Kim , Eui Hong Kim, Joong Kwon Kim " A Leader Election Algorithm in a Distributed Computing System" IEEE 1995
[9] Mahdi Zargarnataj,"New Election Algorithm based on Assistant in Distributed Systems".