ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 1, Issue 7, September 2012

A Comparative Analysis in Terms of Message Passing & Complexity of Different Coordinator Selection Algorithms in Distributed System

Ms. Sachi Choudhary, Mr. Dipesh Sharma

Abstract— **In distributed systems, many of the algorithms that have been used are typically not completely symmetrical, and some node has to take the lead in initiating the algorithm. The main role of an elected coordinator is to manage the use of a shared resource in an optimal manner. Consequently, it is sometimes necessary that, from a set of nodes, a node must be selected as a leader or coordinator. So, to achieve this, several coordinator selection algorithms have been proposed so far. This paper proposes a comparative study and analysis of the different algorithms discussed, efficiency in terms of number of messages exchanged in each case and complexity of the various coordinator selection algorithms in distributed system.**

Index Terms— **Election algorithm, Coordinator, Distributed System, Message Passing, Nodes, Algorithms, Process table, Crash failure.**

I. INTRODUCTION

A leader election is defined as getting a set of processes to agree on a unique leader. The leader (coordinator) typically knows all the other processes (participants) in the group. Several distributed algorithms require that there be a coordinator node in the entire system that performs some type of coordination activity needed for the smooth running of other nodes in the system. As the nodes in the system need to interact with the coordinator node, they all must unanimously who the coordinator is. Also if the coordinator node fails due to some reason (e.g. link failure) then a new coordinator node must be elected to take the job of the failed coordinator [8] .In a classic paper, Garcia-Molina in 1982 specifies the leader election problem for synchronous and asynchronous distributed systems with crash failures and gives an elegant algorithm for each type of system; these algorithms are called the Bully Algorithm and Invitation Algorithm respectively[1][5][6]. After that Silberschatz And Galvin in 1994 proposed an algorithm for coordinator selection between the nodes organized in logical ring. In this algorithm the processes are arranged in a logical ring, each process knows the structure of the ring. [2][3] In this the election message circulates over the ring from one active node to another and eventually returns back to initiating node. Among the list, it elects the node with the highest priority as the new coordinator and then circulates a coordinator message over the ring to inform the other active nodes.
In distributed systems, many algorithms have been proposed for electing coordinator among set of nodes in the system This paper proposes a comparative study and analysis of the different algorithms discussed, efficiency in terms of number of messages exchanged in each case and complexity of the various coordinator selection algorithms in distributed system.

Manuscript received Sep 15, 2012.

Ms. ***Sachi Choudhary****, Computer Science & Engineering,, CSVTU Bhilai/RITEE Raipur, India,+919098534283.*

Mr. Dipesh Sharma*, Reader Computer Science & Engineering,, CSVTU Bhilai/RITEE Raipur, India.*

II. COORDINATOR SELECTION ALGORITHM

In a distributed computing system, a node is used to coordinate many tasks. It is not an issue which node is doing the task, but there must be a coordinator that will work at any time. An election algorithm is an algorithm for solving the coordinator election problem. Various algorithms require a set of peer nodes to elect a leader or a coordinator. **A leader is required to make synchronization between different node.** Elections may be needed when the system is initialized, or if the coordinator crashes or retires[1][2][3].

Assumptions [1[2][3]]:

a. Each node in the system has a unique priority number or unique ID.

b. Every node in the system should know the values in the set of ID numbers, although not which node is up or down.

c. Whenever an election is held, the node having the highest priority number among the currently active node is elected as the coordinator.

d. On recovery, a failed node can take appropriate actions to rejoin the set of active node.

III. DISTRIBUTED SYSTEM

Tanenbaum and van Renesse: A distributed system is one that looks to its users like an ordinary, centralized, system but runs on multiple independent CPUs.

A Distributed system is a collection of autonomous computing nodes which can communicate with each other and which cooperate on a common goal or task. For example, the goal may be provide the user with a database management

system, and in this case the distributed system is called a distributed database.

Or

A distributed system is a collection of processors interconnected by a communication network in which each processor has its own local memory and other peripherals and the communication between them is held by message passing over the communication network.

A. Features of Distributed System:

1. Inherently distributed applications
2. Information sharing among distributed users
3. Resource sharing
4. Better price performance ratio
5. Shorter response times and higher throughput
6. Higher reliability
7. Extensibility and incremental growth
8. Better flexibility in meeting users needs

B. Need of Election

Several distributed algorithms require that there be a coordinator node in the entire system that performs some type of coordination activity needed for the smooth running of other nodes in the system. As the nodes in the system need to interact with the coordinator node, they all must unanimously who the coordinator is. Also if the coordinator node fails due to some reason (e.g. link failure) then a new coordinator node must be elected to take the job of the failed coordinator.

IV. Different Election Algorithms

A. Bully Algorithm by Garcia Molina[5][6]

Bully algorithm is one of the most famous election algorithms which were proposed by Garcia-Molina [5] in 1982. This algorithm is established on some basic **assumptions** which are:

1. It is a synchronous system and it uses timeout Mechanism to keep track of coordinator failure detection .
2. Each node has a unique number to distinguish them .
3. Every node knows the node number of all other nodes .
4. Nodes do not know which nodes are currently up and which nodes are currently down.
5. In the election, a node with the highest node number is elected as a coordinator which is agreed by other alive nodes.
6. A failed node can rejoin in the system after recovery.

In this algorithm, there are three types of message and there is an election message (ELECTION) which is sent to announce an election, an answer (OK) message is sent as response to an election message and a coordinator (COORDINATOR) Message is sent to announce the new coordinator among all other alive nodes . When a node P determines that the current coordinators crashed because of message timeouts or failure of the coordinator to initiate a handshake, it executes bully election.

algorithm using the following sequence of actions[5][6]

.

1. P sends an election message (ELECTION) to all other nodes with higher node numbers respect to it. If P doesn't receive any message from nodes with a higher node number than it, it wins the election and sends a COORDINATOR Message to all alive nodes.

2. If P gets answer message from a node with a higher node number; P gives up and waits to get COORDINATOR message from any of the node with higher node number. Then new node initiates an election and sends ELECTION message to nodes with higher node number than that one. In this way, all nodes will give up the election except one which has the highest node number among all alive nodes and it will be elected as a new coordinator.

3. New Coordinator broadcasts itself as a coordinator to all alive nodes in the system.

4. Immediately after the recovery of the crashed node is up, it runs bully algorithm.

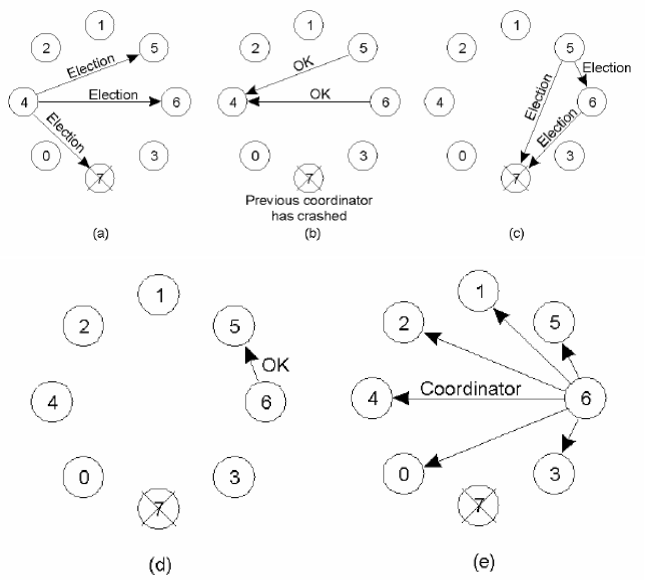

Figure 1: Bully algorithm example (a) process 4 detects coordinator is failed and holds an election, (b) process 5 and 6 respond to 4 to stop election, (c) each of 5 and 6 holds election now, (d) process 6 responds to 5 to stop election, (e) process 6 winds and announces to all.

Bully algorithm has following limitations:

1. The main limitation of bully algorithm is the highest number of message passing during the election and it has order O (n2) which increases the network traffic.

2. When any node that notices coordinator is down then holds a new election. As a result, there May n number of elections can be occurred in the system at a same time which imposes heavy network traffic.

3. As there is no guarantee on message delivery, two nodees may declare themselves as a coordinator at the same time. Say, P initiates an Election and didn't get any reply message from Q, where Q has a higher node number than P. At that case, p will announce itself as a coordinator and as well as Q will also initiate new election and declare itself as a

coordinator if there is no node having higher node number than Q.

4. Again, if the coordinator is running unusually slowly (say system is not working properly for some reasons) or the link between a node and a coordinator is broken for some reasons, any other node may fail to detect the coordinator and initiates an election. But the coordinator is up, so in this case it is a redundant election. Again, if node P with lower node number than the current coordinator, crashes and recovers again, it will initiate an election from current state.

B. Modified Election algorithm by M.S. Kordafshariet al[6][7]:

Modified Bully algorithm by Quazi Ehsanul Kabir *Mamun et al.* Quazi Ehsanul Kabir Mamun et al. described an efficient version Bully algorithm to minimize redundancy in electing the coordinator and to reduce the recovery problem of a crashed process.

a. Assumption[6] :

There are five types of message. An election message is sent to announce an election, an ok message is sent in response to an election message, on recovery, a process sends a query message to the processes with process number higher than it to know who the new coordinator is, a process gets an answer message from any process numbered higher than it in response to a query message and a coordinator message is sent to announce the number of the elected process as the new coordinator.

b. Algorithm[6][7] :

a) When a process p notices that coordinator is down, it sends an election message to all processes with higher number. If no response, p will be the new coordinator.
b) If p gets ok message, it will select the process with highest process number as coordinator and send a coordinator message to all process.
c) When a crashed process recovers, it sends query message to all process with higher process number than it.
d) And if it gets reply then it will know the coordinator and if it doesn't get any reply it will announce itself as a coordinator.

c. Limitations :

Although this algorithm reduces redundant election on some extent, it still has some redundant elections and also has high message complexity. Some of the limitations are given below:
a) On recovery, it sends query message to all processes with higher process number than it, and all of them will send answer message if they alive. Which increases total number of message passing and hence it increases network traffic.
b) It doesn't give guarantee that any process p will receive only one election message from processes with lower process number. As a result there may be q different processes with lower process number can send election essage to p and p will send ok message to all of them. This increases number of election and also number of message passing.
c) It doesn't give any idea if p will crash after sending an election message to all processes with higher process number.
d) It also doesn't give any idea if a process with the highest process number will crash after sending ok message to p.

C. Election algorithm using election commission :

Algorithm :

a) When process P notices that the coordinator is down, it sends an ELECTION message to Election Commission.
b) FD of Election Commission verifies ELECTION message sent by P. If the sending notice of P is not correct, then Election Commission will send a COORDINATOR message to P with process number of the current coordinator.
c) If the sending notice of P is correct and if the highest process number is P, then Election Commission will send a COORDINATOR message to all processes with process number of P as a new coordinator. If the highest process number is not P, Election Commission will simply find out the alive process with the highest process number using HP and sends a COORDINATOR message to all processes with the process number of that process as a new coordinator.
d) If any process including last crashed coordinator is up, it will send a QUERY message to the Election Commission. If the process number of the newly entranced process is higher than the process number of the current coordinator, Election Commission will send a COORDINATOR message to all processes having the process number of new coordinator.
e) If not, Election Commission will simply send a COORDINATOR message to newly entranced process having process number of the current coordinator.
f) If more than one process sends ELECTION message to Election Commission at the same time, then Election Commission will consider the process with higher process number which ensure less message passing to find out the highest process number using HP.

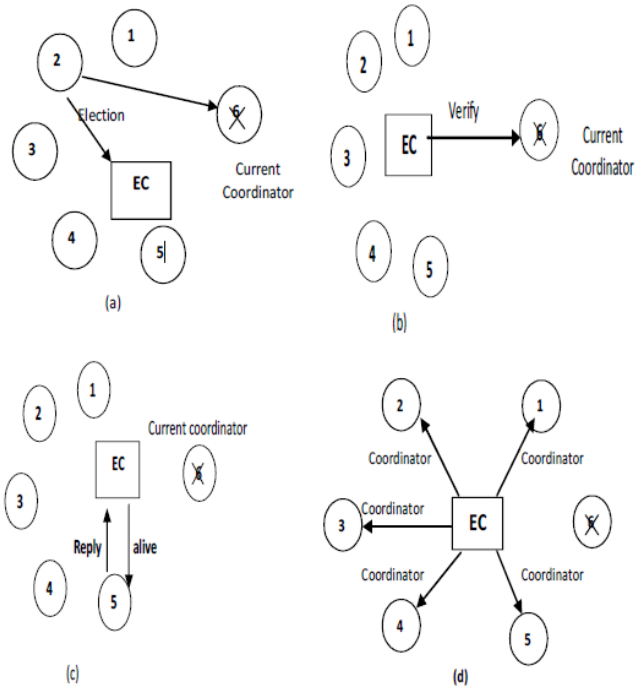

Figure 2 : Election Procedure: (a) Process 2 detects current coordinator is down and sends an election message to EC, (b) EC verifies either the

coordinator is really down or not,(c) EC finds the alive process with highest number using alive message,(d) EC sends coordinator message to all process having process number of currently won.

B. Description :

Figure 2 represents regular election procedure of the proposed algorithm. Here, the system consists of six processes with process number 1 to 6. Current coordinator is the process 6. But it has just crashed and process 2 first notices this. So it sends an election message to the EC in Figure 2(a).In Figure2(b), EC sends verify message to the current coordinator to be sure about the election message sent by process 2. After verification, In Figure 2 (c), EC sends alive message to process 5 (the next highest process number) to check either the current highest process is alive or not. And EC gets a reply message from 5. In Figure 2(d), EC select 5 as new coordinator and sends coordinator message to all processes having 5 as a new coordinator of the system.

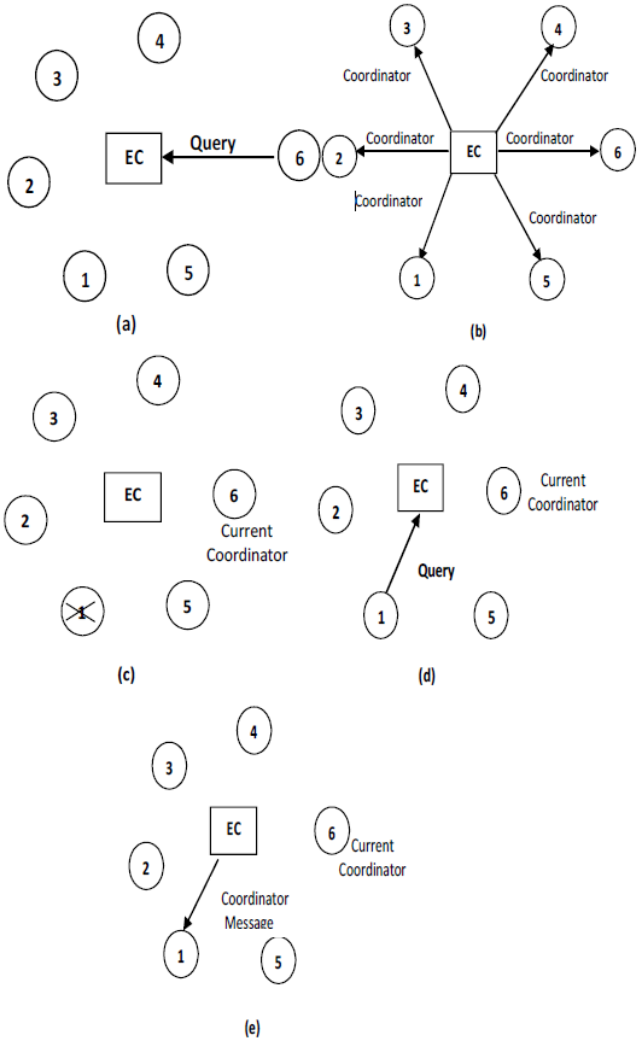

Figure 3. Query after Recovery: (a) Last crashed coordinator 6 is up and sends a query message to the EC, (b) EC selects 6 as new coordinator and sends coordinator message to all processes, (c) Now process 1 is crashed, (d) Again process 1 is up and sends query message to EC,(e) EC sends coordinator message to process 1 having the current coordinator).

Figure 3 represents the steps when a crashed process is up. In Figure 3 (a), last crashed coordinator 6 is up and sends a query message to EC. As process number of 6 is higher than the current coordinator of the system, in Figure 3 (b), EC sends coordinator message to all processes with process number 6 as new coordinator. In figure 3 (c), process 1 is now just Crashed. In figure 3 (d), process 1 is just up after crashed, and it sends a query message to EC. EC checks that process number of newly entranced is lower than the current coordinator. So in Figure 3 (e) EC sends coordinator message to only process 1 having the process number of current coordinator of the system.

V. COMPARISON & ANALYSIS OF DIFFERENT ALGORITHMS

In **Bully algorithm**,[1][4][7] when the process having the lowest priority number detects the coordinator's failure and initiates an election, in a system of n processes, altogether (n-2) elections are performed. All the processes except the active process with the highest priority number and the coordinator process that has just failed perform elections. So in the worst case, the bully algorithm requires O(n2) messages. When the process having the priority number just below the failed coordinator detects failure of coordinator, it immediately elects itself as the coordinator and sends n-2 coordinator messages. So in the best case, it has O(n) messages.
During recovery, a failed process must initiate an election in recovery. So once again, Bully algorithm requires O(n2) messages in the worst case, and (n-1) messages in the best case.

For the case of **modified bully algorithm[6][7]** there will be need of or O(n) message passing between processes. In worst case that is the process with lowest process number detects coordinator as failed, it requires 3n-1 message passing. In best case when p is the highest process number, it requires (n-p) + n messages.
For the case of **election algorithm with election commision** there will be need of 1 election message to inform EC, 2 verify message to ensure the failure of coordinator, and say r is the highest alive process then alive and reply message to find out the highest alive process and so total or O (n) message passing between processes. If the process with lowest process number detects coordinator as failed it will not change total message. In worst case it may happen that our algorithm needs to check up process to p+1 to find out highest alive process. Only at that case it requires message passing between processes. However, in best case, our algorithm may find the highest alive process with only one alive and one reply message that is highest alive process in the system is process with process number n-1. In that case, our algorithm requires only 1+2+2+n messages. When p is the highest process number, it requires only 1+2 + n messages.
If a process crashes and recovers again, it sends a query message to all processes higher than that process to know the current coordinator which requires 2*(n-p) message passing. But in our algorithm, any process after recovery will only send query message to EC and EC will send a coordinator message having process number of current coordinator which requires only 2 messages passing.

VI. CONCLUSION

In this paper, a comparison between election algorithms in a distributed system is done. The comparision is done on the

basis of message passing and time complexity parameters of algorithms. This paper also focuses on limitations of different algorithms for coordinator selection.

.

REFERENCES

[1] Sandipian Basu ”An Efficient Approach of Election Algorithm in Distributed Systems” Post graduate Department of Computer Science,St. Xavier’s College, Kolkata INDIA. Indian Journal of Computer Science & engg. ISSN : 0976-5166 Vol. 2 No. 1.

[2] Tanenbaum A.S, Distributed Operating System, Pearson Education, 2007. PP. 140-142

[3] Silberschatz Galvin Gagne, “Operating Systems Concepts “ ,Seventh edition, page no. 265 topic : Election algorithm.

[4] Heta Jasmin Javeri, Sanjay Shah “A Comparative Analysis of Election Algorithm in Distributed System” IP multimedia communication, A special issue from IJCA

[5] Garcia – Molina “Elections in a distributed computing system”, IEEE transactions on computers, vol C-31, No 1,pp 48-59., Jan 1982.

[6] S. Mahdi Jameii, “A Novel Coordinator Selection Algorithm in Distributed Systems” Department of Computer-Engineering, Islamic Azad University, Shahr-e-Qods Branch, Tehran, Iran IJAEST Vol No. 9, Issue No. 2, 310 -313

[7] Quazi Ehsanul Kabir Mamun, Salahuddin Mohammad Masum , **“**Modified Bully algorithm for electing coordinator in distributed system” . CD proceedings of 3rd WSEAS international conference on software engg. , parallel & distributed system (SEPADS) 2004 feb 13-15, Australia

[8] “Message Efficient Leader Election in Synchronous Distributed System with Failure Detectors “Sung-Hoon Park School of Electrical and Computer Engineering, Chungbuk National Unvi. Cheongju ChungBuk 361-763, Korea

[9] Sinha P.K, Distributed Operating Systems Concepts and Design, Prentice-Hall of India private Limited, 2008.

*Ms. **Sachi Choudhary**, Computer Science & Engineering,, CSVTU Bhilai/RITEE Raipur , India,+919098534283.*

***Mr. Dipesh Sharma**, Reader Computer Science & Engineering,, CSVTU Bhilai/RITEE Raipur , India.*