International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 5 Issue: 5 1092 – 1097

An Enhanced Bully Algorithm for Electing a Coordinator in Distributed Systems

Minhaj Khan
Shri Ram Swaroop Memorial University
Lucknow, India
minhajkhan7786@gmail.com

Neha Agarwal
Shri Ram Swaroop Memorial University
Lucknow, India
lko.neha@gmail.com

Jeeshan Ahmad Khan
Shri Ram Swaroop Memorial University
Lucknow, India
jeeshan.jak@gmail.com

Abstract—In a distributed system for accomplishing a large complex task, the task is divided into subtask and distributed among processes and coordination among processes done via message passing. To make proper coordination and functioning we need a leader node or coordinator node which acts as a centralized control node. Leader election is the most challenging task in distributed system because it is not necessary that leader node is always same because of crash failure or out of service may occur in the system. Tremendous algorithms have been proposed for elect the new leader. These algorithms use a different technique to elect a leader in distributed system. Bully election algorithm is one of the traditional algorithms for electing a leader, in which the highest node Id is elected as a leader but this algorithm requires lots of message passing for electing a leader that imposes heavy network traffic. Due to heavy network traffic, it creates complexity in message passing and takes more time. In this paper, we introduce a new approach which overcomes the drawback of existing Bully election algorithm. Our proposed algorithm is an enhanced version of Bully election algorithm. Our analytical result shows that our algorithm is more efficient than original Bully Algorithm.

Keywords-*Distributed Systems, Bully election algorithm, Coordinator, message passing*

*****

I. INTRODUCTION

A distributed system is a collection of separate computers which connected together via a network for accomplishing a common complex large job and communication between these computers done via message passing [7, 8]. The main objective of distributed system is to distribute the load among these separate computers for better performance and create a single system image for the user [8]. In distributed system there is no central controlling node, any node can communicate with remaining active nodes in the network and take a correct decision [9]. But it is not necessary that during the decision-making process, all the nodes take the same decision; hence communication among the nodes is time-consuming [9]. Thus for making consistency among all active nodes, a node is selected as a leader and act as a central controlling node [9]. To elect a leader in distributed system is a most challenging task. For selecting a leader different algorithms have been offered. Few of them are in the ring topology, Bully election algorithm, Franklin algorithm Chan and Robert algorithm, Time Slice algorithm, Variable Speeds algorithm etc. [11,12]. This algorithm uses different approaches to electing a leader in distributed system. But these algorithms have some drawback such as message passing, time complexity, redundancy and heavy network traffic. To overcome these problems, we introduce a new algorithm which reduces messages passing for elect a leader. This algorithm is based on some basic assumption which is given below:

a. It is a synchronous timeout mechanism system.
b. Each node is assigned by a unique Id
c. Each node knows the id of another node
d. Nodes don't know which nodes is currently up or down
e. Every node stores the leader node id.
f. A crashed node after the recovery can join the system again.

II. LITRATURE SURVEY

There are several algorithms have been proposed to electing a coordinator in distributed system. In this section we are going to elaborate three significant election algorithms.

A. Bully Algorithm

The Bully algorithm is one of the most popular algorithms proposed by Garcia-Molina in 1982[1] and it is based on some basic assumptions which are given below:

a. The system is synchronous timeout mechanism system.
b. In this synchronous system, each node is distinguished by a unique id.
c. Each node knows the id of another node.
d. There is no concern between the nodes which ones are currently up or which ones are currently down.
e. The process with the highest id is elected as a leader which is agreed by all another node.
f. The node which is failed can join the system again after recovery [2, 3, 14].

Garcia-Molina offered a bully algorithm for electing a coordinator in distributed system. While undertaking this algorithm a number of messages passing increased .i.e. when any node detect the leader crashed or failed, then start an election procedure and the process that having the highest process ID will be elected as a leader. After selecting a leader, the node which has won broadcast as a new leader

 ISSN: 2321-8169
1092 – 1097

among all active nodes. This procedure takes more messages passing which imposes heavy network traffic.

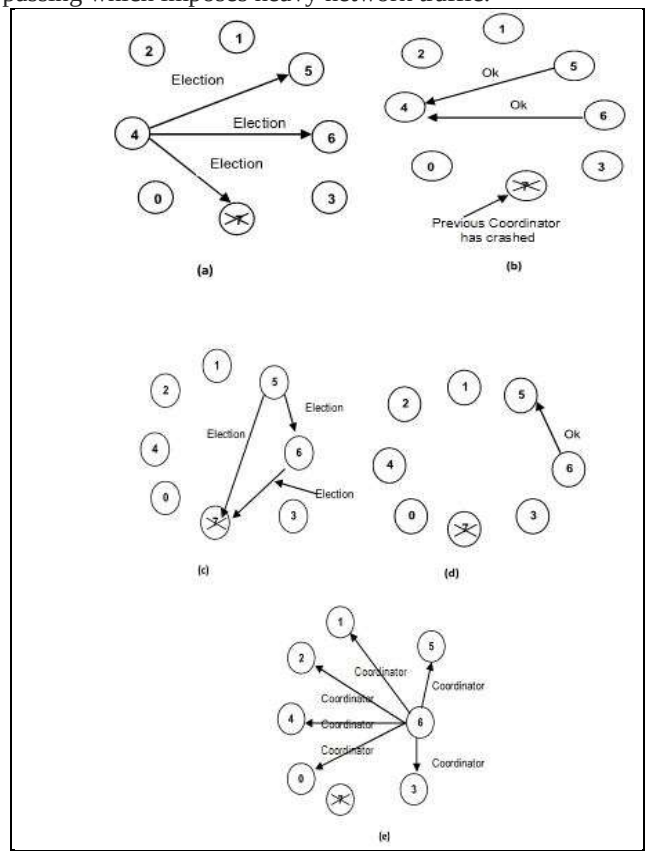

Figure1: Steps of electing a leader in traditional Bully algorithm

a. Node 4 noticed that coordinator 7 has crashed then it sends election message to its highest node i.e. node 5 & 6.
b. After receiving election message node 5 & 6 send an ok message to node 1.
c. After receiving the Ok message, the working of node 4 is stopped and node 5 and 6 will send election message to their highest process number.
d. The process goes on the same way and at last node 6 send an ok message to node 7 but it does not receive any message from node 7.
e. After sending election message node 6 is elect as a coordinator and send coordinator message to all active nodes.

The main drawbacks of the original Bully algorithm are:

a. The main limitation of the Bully algorithm is the highest number of message passing during the election and it has order O(n2) which imposes the heavy network traffic.
b. When any node notices that coordinator down then holds a new election. As a result, there may be n number of elections can be occurred in the system at the same time which imposes heavy network traffic.
c. As there is no guarantee on message delivery, two nodes may declare themselves as a coordinator at the same time. Say, N initiates an election and didn't get any reply message from P, where P has a higher process number than N.At that case, N will announce itself as a coordinator and as well as P will also initiate new election and declare itself as a coordinator if there is no process having higher process number than P.
d. Again, if the system is not working properly for some reason 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. After recovery when coordinator joins the system, so in this case, it is a redundant election.

B. Modified Bully Algorithm by Quazi Ehsanul Kabir Mamun

This algorithm is a proposed by Quazi Ehsanul Kabir Mamun [2]. It is a modified version of the Bully algorithm. This algorithm is based on existing bully algorithm assumption [2, 3]. In this algorithm, the node with the highest Id is elected as a coordinator. When any node notice that the leader is crashed then it sends election message to their highest id node and if receive responses from those highest id nodes then it will select highest id node as a coordinator and broadcast coordinator message to all active nodes. If it does not receive any responses then it elects itself as a leader and sends coordinator message to all active nodes [3].

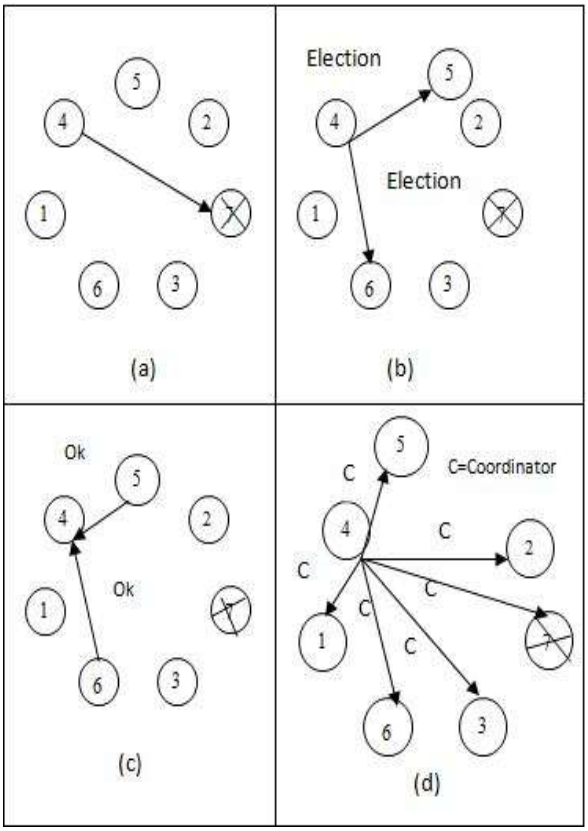

Figure2: Steps of electing a leader in modified Bully algorithm by Quazi Ehsanul Kabir Mamun

a. Node 4 notice the coordinator (node 7) is failed.
b. Node 4 sends election message to node 5 & 6.

 ISSN: 2321-8169
1092 – 1097

c. After receiving election message node5 & 6 send an ok message to node 4.
d. When node 4 get an ok message from node 5 & 6 check highest id node and find node 6 is the highest id node then it broadcast node 6 as a coordinator to all active nodes.

This algorithm overcomes some drawback of the original bully algorithm but this algorithm is also some drawback which is given below:

a. The main drawback of this algorithm is that when any node N crashes after sending the election message to higher Id node or crashes after receiving priority number from higher node, higher node will wait for 3D (D is average propagation delay) time for coordinator broadcasting and if they don't receive any coordinator message then it will start election again. Those are the redundant election.
b. There is no guarantee of coordinator failure.
c. Every redundant election takes more messages passing that impose heavy network traffics.

C. Modified Bully algorithm by M.S. Kordafshari et al.

M. S. Kordafshari et al. proposed a new algorithm to overcome the drawback of synchronous Garcia Molina's Bully Algorithm and modified bully algorithm [5, 6]. In this Algorithm when any node notice the coordinator is failed it immediately start election and send election message to highest id nodes. After receiving the election messages the highest nodes send responses to it. When the node receives the responses, it checks the highest Id node and sends grant message to highest id node. After receiving the grant message, the node who receive the grant message send coordinator message to all active nodes [5, 6, 14].

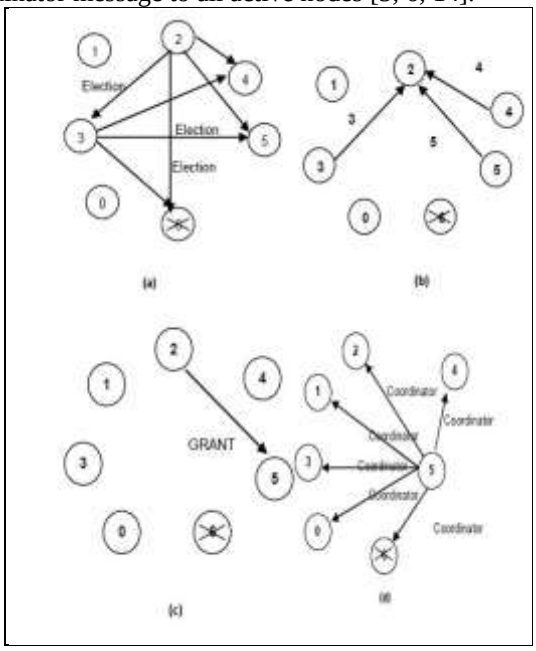

Figure 3.Steps of electing a leader in modified Bully algorithm M.S. Kordafshari et al.

a. When node 2 notice that the coordinator (node 6) is failed then send election message to their highest id nodes.
b. After receiving election message these nodes i.e. node 3, 4 & 5 send responses to node 2.
c. When node 2 receives the ok messages, check highest id node and send grant message to highest id node here highest id node is 5.
d. After receiving grant message the highest id node broadcast coordinator message to all nodes.

This modified Bully algorithm has some drawback which is given below:

a. The main drawback of this algorithm is that when any node N crashes after sending the election message to higher Id node or crashes after receiving priority number from higher node, higher node will wait for 3D (D is average propagation delay) time for coordinator broadcasting and if they don't receive any coordinator message, they will initiate modified algorithm again [5]. If there are q different higher nodes, then there will be q different individual instance of modified algorithm at that time in the system. Those are the redundant election.
b. If node N sends GRANT message to the node with the highest priority number, and N doesn't receive coordinator message from that node, N will repeat the algorithm, which is the redundant election. As after any node with higher priority number compares to coordinator is up, it runs the algorithm, it increases redundant elections.
c. Every redundant election takes resources and imposes heavy network traffics.

III. PROPOSED ALGORITHM

In this algorithm, the node which has the highest process id is selected as a coordinator. If the coordinator is failed then the N-1 node which has the next highest process id is selected as a coordinator. In this algorithm when any node detects the coordinator is crashed send election message to the next highest process id after receiving the election message the next highest process id check coordinator is exactly crashed or not. If the coordinator is crashed then it elects itself as a leader and sends coordinator message to all active nodes.

A. Algorithm

In this algorithm the variables which are used given below:
scp_id->store coordinator process id
rcp_id->recently crashed coordinator process id
ncp_id->new coordinator process id

```
int scp_id,rcp_id,ncp_id
//when any node X detect the coordinator (Node N) is crashed
Create election message and send to N-1 node
Start timer
//After receiving the election message by N-1 node
```

 ISSN: 2321-8169
1092 – 1097

```
Check (scp_id is failed or not)
If (slp_id is failed)
scp_id=ncp_id //ncp-id is the id of node N-1 (second higher ID node (N-1))
broadcast coordinator message (ncp_id, rcp_id)
else
discard the message
```

Figure 4. Pseudo code that is triggered when any node detects the crash of the Coordinator

```
//when node X1, X2, X3.....,Xn detect the node X is failed
Create elections messages by X1, X2, X3.....,Xn
Send to N-1 node
Start timer
//After receiving message by N-1 node
Check (scp_id is failed or not)
If (scp_id failed)
        scp_id=ncp_id //ncp-id is the id of N-1 node
broadcast coordinator message (ncp_id, rcp_id)
else
discard the message
```

Figure 5.Pseudo code that is triggered when more than two nodes detects the crash of the Coordinator

```
When node N-1 node (second higher Id node) detect the leader is crashed then
scp_id = ncp_id \\where ncp_id is the id of N-1 node
broadcast coordinator message (ncp_id, rcp_id)
```

Figure 6.Pseudo code that is triggered when second higher Id node detects the crash of the Coordinator

```
//when at a time node X detect node N1 is crashed
Create election message and send to N2 node
Start timer
//If node X doesn't receive any coordinator message from N2 then send message to next N3 node
//After receiving message by node X
Node N3 check node N1 & N2 is crashed or not
If (N1 & N2 crashed)
 scp_id = ncp_id (ncp_id is the id of node N3)
broadcast coordinator message (ncp_id,rcp_Id)
else
receive reposes from N1 & N2  \\Here N1 > N2)
broadcast coordinator message (scp_id. Null)
```

Figure 7.Pseudo code that is triggered when second higher Id node doesn't response to the coordinator failure detector node

B. Example

When any node x detect that the coordinator is crashed then it sends election message to N-1 (next higher node Id) node. After receiving the election message the node N-1 check the store coordinator process id (scp_id) is exactly crashed or not if it is crashed then it stores their id (ncp_id) as scp_id and send coordinator message the all active nodes. After receiving the coordinator message all nodes compare recently crashed coordinator process id (rcp_id) to store coordinator process id (scp_id). If scp_id = rcp_id then it store new coordinator process id (ncp_id) as scp_id.

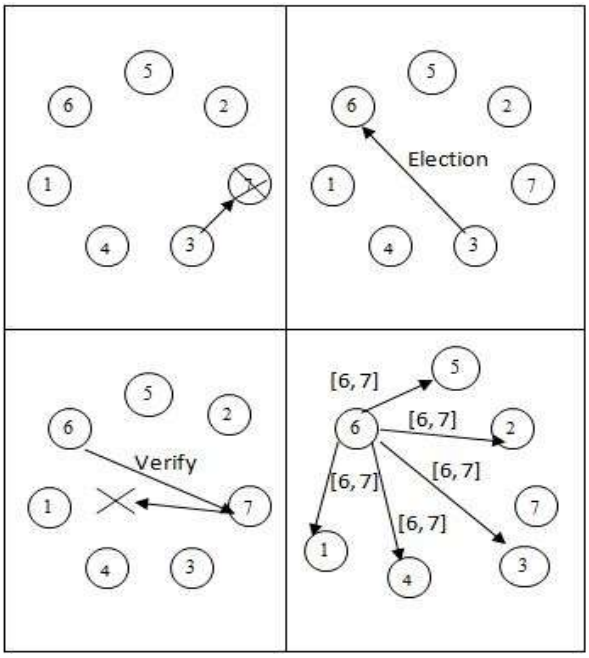

Figure 8.Steps to elect a coordinator when only one node detect the coordinator is failed

In the above example (figure 8) node 7 is the coordinator. Here node 3 notices that the coordinator is failed then it send election message to node 6. After receiving the election message node 6 check its table and find that node 7 is the coordinator and will check whether coordinator is exactly crashed or not. If it finds the coordinator is exactly failed then node 6 store their Id as store coordinator process id (scp_id) and send the message to all active nodes with a message (ncp_id, rcp_id). After receiving the message the nodes check if scp_id(7)=rcp_id(7) then they update their table and store ncp_id(6)as scp_id..

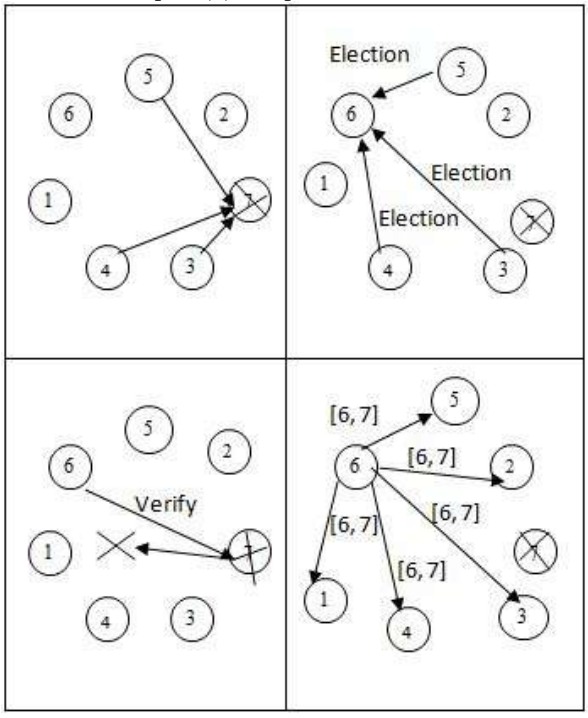

Figure 9.Steps to elect a coordinator when more than two nodes detect the coordinator is failed

 ISSN: 2321-8169
1092 – 1097

In the above example (figure 9) node 3, 4 and 5 detect coordinator (node 7) is failed; they immediately start election and send election message to node 6. After receiving the message node 6 verify whether node 7 is failed or not. If it is failed, then node 6 stores its id as a store coordinator process id (scp_id) and send coordinator message [6, 7] to all active nodes. After receiving message all nodes update their table and store node 6 as a scp_id.

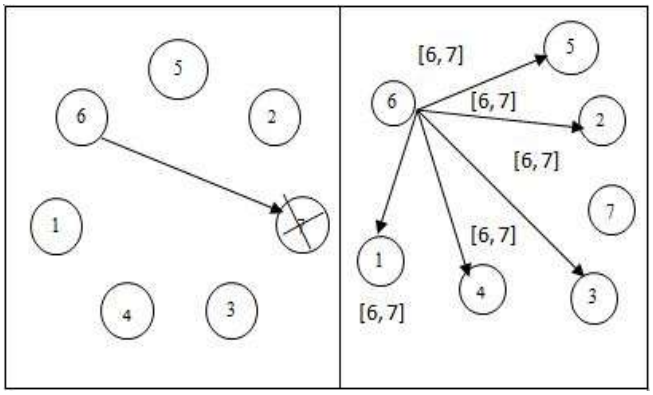

Figure 10.Steps to elect a coordinator second higher Id node detect the coordinator is failed

When node 6 notices that the coordinator is crashed then it becomes the new coordinator. Node 6 stores their id as a scp_id and send the message [6, 7]. After receiving message all active nodes update their table and store node 6 as a scp_id.

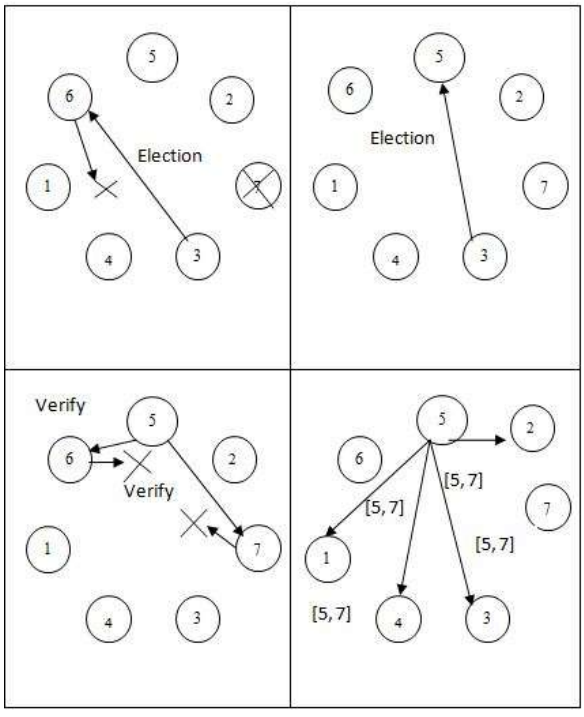

Figure 11.Steps to elect a coordinator when second higher Id node does not give the response

In the above example (fig 7) node 3 notices the coordinator (node 7) is crashed then it sends election message to node 6. After a certain period, if it does not receive any responses from node 6, it sends election message to node 5. After receiving election message, node 5 validate node 6 and node 7 id if it does not receive any responses from node 6 & 7 then it stores its id as a scp_id and send the message to all active nodes [5,7]. After receiving message, all active nodes compare scp_id & rcp_id and find scp_id(7)=rcp_id(7) and update their table and store node 5 as scp_id. If node 5 receives responses from node 6 & 7 then it broadcast coordinator message (7, null).After receiving the messages all active nodes find the scp_id is node 7.

If node 5 does not receive the response from node 7 but receive repose to node 6 then it stores node 6 as a coordinator and broadcast the message [6, 7].After receiving the message all active nodes update their table.

IV. LIMITATION AND ADVANTAGE

Best case: If any node n discovers the coordinator is failed then a number of messages passing between the nodes for electing a coordinator will be 1+2+ (n-2) .Time complexity is O (n).

Average case: If there are n nodes in a network and more than one node (assumed x) discovers coordinator failure then number of message passing between the nodes for electing leader will be 2*x+1+ (n-2). Time complexity is O (n).

Worst case: There are n nodes in a network and all nodes discover coordinator failure then number of message passing between the nodes for electing coordinator will be 3*(n-2) +1. Time complexity is O (n).

V. COMPARISON WITH OTHER ALGORITHMS

In this section, we compare our proposed algorithms with respect to the Bully algorithm and modified Bully algorithm based on their message passing complexity

Table 1: Comparison between previous algorithm and our proposed algorithm

No of nodes in a network	Leader Election Algorithms		
	Bully Algorithm	***Modified Bully Algorithm***	***Proposed Algorithm***
5	24	14	10
10	99	29	25
25	624	74	70
100	9999	299	295
150	22499	449	445

VI. CONCLUSION

Many algorithms have been proposed to electing a coordinator in distributed system. In this paper, we propose an improved algorithm for electing of a coordinator and modified the previous election algorithms. We tried to overcome drawbacks of the original Bully algorithm and modified bully algorithm. Our comparison section prove that our algorithm is more efficient than bully algorithm and modified bully algorithm with respect to message passing, redundant election, and network traffic.

REFERENCES

[1] H. Garcia-Molina, "Elections in distributed computing system," IEEE Transaction Comp., volume.C-31, pp.48-59, January.1982.

[2] Q.H. Mamun,S.H.Masum and M. A.Mustfa, "Modified bully algorithm for electing coordinator in distributed systems," in Proc. 3rd WSEAS International Conference on Software Engineering, Parallel and Distributed Systems pp.22- 28.

[3] Basim Alhadidi, Laith H.Baniata and Mohammad H.Baniata, Mohammad Al-Sharaiah "Reducing Massage Passing and Time Complexity in Bully Election Algorithms Using Two Successors", International Journal of Electronics and Electrical Engineering Vol 1, No. 1, Mar 2013

[4] M.Gholipur, M.S.Kordafshri, M.Jahanshani, A.M.Rahmani " A New Approach For Election Algorithm in Distributed Systems," International Conference on Computer and Information Technology, 2009

[5] M. S. Kordafshari, M. Gholipour, M.Jahanshahi, A.T. Haghighat, "Modified Bully Election Algorithm in Distributed Systems", WSEAS International Conference on Comp., 2005

[6] M. S. Kordafshari , M. Gholipour , M. Jahanshahi , A.T. Haghighat ," Two novel algorithms for electing coordinator in distributed systems based on bully algorithm", the fourth WSEAS International Conference on Software Engineering, Parallel and Distributed Systems,2005

[7] Sathesh B.M," Optimized Bully Algorithm",International Journal of Computer Applications (0975 – 8887), Volume 121 – No.18, July 2015

[8] P BeaulahSoundarabai, Thriveni J, K R Venugopal, L M Patnaik," AN IMPROVED LEADER ELECTION ALGORITHMFOR DISTRIBUTED SYSTEMS**",** International Journal of Next-Generation Networks (IJNGN) Vol.5, No.1, March 2013

[9] Deepali P. Gawali," Leader Election Problem in Distributed Algorithm", IJCST Vol. 3, Issue 1, Jan. - March 2012

[10] Hetal Katwala1, Prof. Sanjay Shah," Study on Election Algorithm in Distributed System**",** IOSR Journal of Computer Engineering (IOSRJCE),Vol-7,Issue 6,pp 34-39,2012

[11] Amit Biswas, Animesh Dutta," A Timer Based Leader Election Algorithm",IEEE Conference,2016

[12] Balmukund Mishra, Ninni Singh and Ravideep Singh," Master-Slave Group Based Model For Co-ordinator Selection, An Improvement of Bully Algorithm", International Conference on Parallel, Distributed and Grid Computing,2014

[13] Mina Shirali, Abolfazl, Mehdi Vojdani," Leader election algorithms: History and novel schemes",3rd International Conference on Convergence and Hybrid Information Technology,2008

[14] M. Rahman, A. Nahar "Modified Bully Algorithm using Election Commission",MASAUM Journal of Computing,vol 1,number 3,pp.88-6,2009

[15] Muneer Bani Yassein, Ala'a N. Alslaity, Sana'a A. Alwidian, "An Efficient Overhead-Aware Leader Election Algorithm for Distributed Systems," IJCA Journal, Vol 49– No.6, 2012

[16] Mohammad Reza Effat Parvar, Nasser Yazdani, Mehdi Effat Parvar, Aresh Dadlani, Ahmad Khonsari,"Improved Algorithms for Leader Election in Distributed Systems,"The 2nd international conference IEEE,Vol 2,April 2010

[17] S. Lee, H. Choi," The Fast Bully Algorithm: For Electing a Coordinator Process in Distributed Systems," International Conference on Information Networking ,2002

[18] M. Mirakhorli, A. A. Sharifloo, M. Abbaspour, "A Novel Method for Leader Election Algorithm". The 7th IEEE International Conference on Computer and Information Technology, 2007.

[19] Sung-Hoon Park, Yoon Kim Jeoung Sun Hwang, "An Efficient Algorithm for Leader-Election in Synchronous Distributed Systems," IEEE Transaction, Vol. 43,no.7,pp.1991-1994,1999.

[20] P Beaulah Soundarabai, Ritesh Sahai, Thriveni J, K R Venugopal, L M Patnaik," Improved Bully Election Algorithm for Distributed Systems", International Journal of Information Processing,2013

[21] Vaibhav P. Gajre,"Comparison of Bully Election Algorithms in Distributed System", International Journal of Scientific and Research Publications, Volume 3, Issue 9, September 2013.