Reducing Massage Passing and Time Complexity in Bully Election Algorithms Using Two Successors

Basim Alhadidi
Al-Balqa' Applied University, salt, Jordan
Email:hadidi72@hotmail.com

Laith H.Baniata and Mohammad H.Baniata
University of Hail, Saudi Arabia
Email: baniata999@hotmail.com; mhm_baniata@hotmail.com

Mohammad Al-Sharaiah
University of Jordan, Amman, Jordan
Email: mohabdal2003@yahoo.com

***Abstract*—In distributed system environment like human life environment we need a manager to coordinate the job of all processes. We call this process as a Leader. The problem occurs when the coordinator process fails; so that these processes must work together in order to elect a leader. There is many algorithms discussed leader's election such as bully algorithm and modified bully algorithm. In our paper we will propose a new election algorithm to solve leader failure. The proposed algorithm is based on using successor (Vice coordinator (leader)) in order to reduce massage passing and time complicity and overcome the disadvantages and limitations of previous algorithms**

***Index Terms*—bully algorithms, coordinator, election massage, ok massages.**

I. Introduction

In distributed computing environment like human life environment we need a manager to coordinate processes in the entire system in order to organize system's tasks. It is not an issue which process is doing the task, but there must be a coordinator that will work any time. So electing a coordinator or a leader is very fundamental issue in distributed computing. In order to select a process to be a coordinator we should use an election algorithm. There are many algorithms that are used in election process, several algorithms had been conducted such as bully, ring algorithms, and some other modified algorithms based on these two basic algorithms [1].

Election algorithm is an algorithm for solving the coordinator election problem by choosing a process among a group of processes on different processors in a distributed system to act as the central coordinator, It can be necessary to determine a new leader if the current one fails to respond, Provided that all processes have a unique identification number, the process can be called for an election at most one time, There might be a concurrent calls for the processes ,the non-failed process with the best election attribute value (e.g., highest id or address,.) is elected[2].

The structure of this paper begins with abstract and introduction that describe the main subject of this paper, short description about election algorithm, problem formulation and the proposed solution. Secondly, the background describes the bully algorithm and its advantages and limitations. Related work described the modified version of bully algorithm .In main work we will present proposed election algorithm (Vice coordinator (leader)) that overcomes the disadvantage of other election algorithms, our proposed algorithm reduces massage passing and time complexity.

II. Background

Bully algorithm is one of the most famous election Algorithms which was proposed by Garcia-Molina in 1982[3]. The bully algorithm is based on basic assumptions:

The system use timeouts to detect process failure (coordinator), each process has a unique number in the system; every process knows the process number of all other processes and which processes have the higher number, Processes do not know which processes are currently up and which processes are currently down. Once an election is held a process with the highest process number is elected as a coordinator which is agreed by other processes [4].

The bully Algorithm has three types of messages: 1- election message (inquiry) which is sent to announce an election.2- an answer (ok) message is sent as Response to an election messages.3- coordinator (victory) Messages

Manuscript received January 5, 2013; revised March 5, 2013.

doi: 10.12720/ijeee.1.1.1-4

are sent to announce the new coordinator among all other alive processes[5]. When a process determines that the coordinator is crashed it executes bully election algorithm using the following steps:

1- Process P sends an election message to all other Processes with higher process number.

2-If P doesn't receive any messages from processes with a higher process number than it, it wins the election and sends a coordinator messages to all other processes. 3-If P gets answer message from a process with higher process number; P gives up and waits to get coordinator message from any process with higher number.4- Then new process initiates an election and sends election message to processes with higher process number, In this way, all processes will give up the election apart the process which has the highest process number.5- The new coordinator announces its victory by sending all processes a message telling them that it is a coordinator. Bully Algorithm is illustrated in Fig. 1

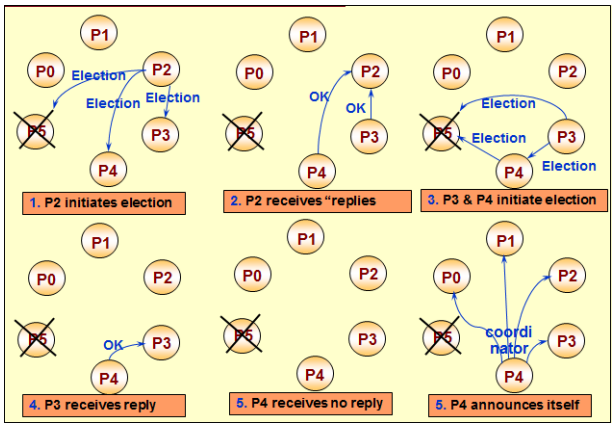

Fig. 1.

Limitations:

There are no guarantees on message delivery order, and the recipient of messages may not conclude which is the coordinator. Safety condition may be Broken if the assumed time out turn to be accurate-the process that detect the failure is unreliable.

Best case scenario: The process with the second highest id notices the failure of the coordinator and elects itself. *N-2* coordinator messages are sent. it require n-2 messages .A failed process is previously a coordinator; it must sends coordinator message to all other processes on recovery. Hence, bully algorithm requires n-1 messages in best case.

Worst case scenario: When the process with the lowest id detects the failure, N-2 processes altogether begin elections, each sending messages to processes with higher ids, Hence in the worst case, the bully algorithm require $O(N^2)$ messages. A failed process is not previously a coordinator. It must start an election on recovery .the bully algorithm requires $O(N^2)$ in worst case.

III. Related Work

In "A Modified bully algorithm for Electing Coordinator in Distributed Systems" [6], the suggested algorithm is based on the assumptions of the existing bully algorithm. There are five types of message in this algorithm. using election message that's sent for announcing the election, an ok message is sent in response to an election message, on recovery, a process sends a query message to the higher process than it to know who the new coordinator is, a process gets an answer message from any process higher than it in response to a query message and a coordinator message is sent to announce the id of the elected process as the new coordinator [6].

Algorithm:

1-When a process p notices that coordinator is down; it sends an election message to all processes with higher id. If no response, p will be the new coordinator. If p gets ok message, it will select the process with highest process id as a coordinator and sends a coordinator messages to all other processes.

2-When a crashed process recovers, it sends query message to all processes with higher process id than it. 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. Modified bully algorithm is illustrated in Fig2.

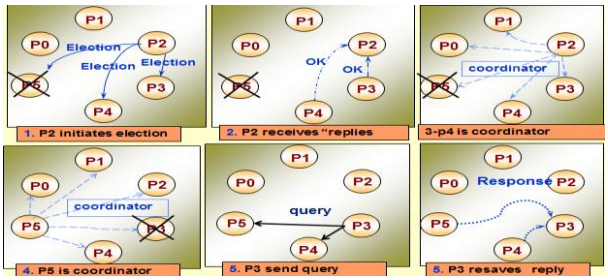

Fig. 2.

Limitations:

Best case scenario: when the process with the second highest id detects that the coordinator has failed; it immediately elects itself as a coordinator and sends n-2 coordinator messages. It requires n-2 messages. A failed process is previously a coordinator. On recovery this algorithm requires n-1 messages in best case.

Worst case scenario: When the process with lowest process id detects the coordinator failure, it require 3n-1 message passing .this algorithm requires O(n) messages. A failed process is not previously a coordinator. On recovery this algorithm requires 2(n-1) messages in worst case.

IV. Main Work

The proposed Vice Coordinator Election algorithm:

The idea of this proposed algorithm is simple and direct, it avoids the problem by assigning a vice coordinator to replace the main coordinator when the coordinator crashes, for each coordinator there are two sub coordinators or successors; the main successor and the sub successor. When the main coordinator crashes the main successor becomes a coordinator. We assume in our election algorithm that it is impossible for three highest processes to be crashed at same time. There are two types of messages in this algorithm. Coordinator

Messages are sent from the coordinator to the two successors (main and sub successors) informing them that they are the successors. Election messages are sent when the coordinator crashes.

The coordinator messages:

The coordinator sends two messages to the successors (successors 'appointment messages) to check if these two successors are not crashed (alive).The two successors reply to the coordinator by an ACK messages to inform the coordinator that they have been informed. The coordinator sends (n – 3) messages to the rest of nodes, these massages contain successors' ids; hence, n is the total number of nodes

The election messages:

When a certain node discovers that the coordinator is dead, it sends an election message to the main successor. The main successor checks the main coordinator, if it's actually dead, then it sends a (n-2) new coordinator messages to all nodes. On recovery, the process sends query messages to the 3 highest process ids to know which process is a coordinator.

How does the algorithm work?

When the system starts, the process with the highest ID is assigned as a coordinator, the new coordinator sends 2 messages to two processes with the highest ID's informing them that they have been assigned as a main successor and a sub successor. The two successors replies with an ACK message informing the coordinator that they received the offer, this step is done only once and the cost is 4 messages. The coordinator sends messages to the rest of the nodes inform them the new successors, this cost (n – 2) messages. It is shown in Fig. 3.

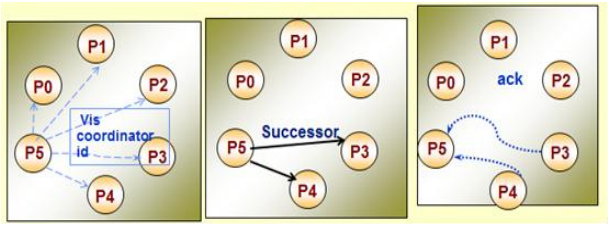

Fig. 3.

When a node discovers that the coordinator is down it sends a message to the main successor to inform it that the coordinator is down (if no coordinator message was sent from the main successor then the node will send the same message to the sub successor); number of messages is 1 and at worst case are 2 messages .

When the main successor knows that the coordinator is down, it checks with the main coordinator; if it is down then it sends (n – 2) messages to the rest of the nodes announcing itself as a coordinator and selects two successors; it cost (n – 2) + 1 messages. On recovery the node sends 3 messages at most and receives 3 responses. This process is shown in fig 4.

Suppose system consists of six processes numbered from 0 to 5 and all process are alive except process 0 and previously process 5 was the coordinator, but it has just crashed.

1- Process 2 detects that the coordinator is failed and then it sends an election massage directly to the main successor process number 4.

2- If main successor (process 4) does not respond then process 2 sends another election massage to sub successor (process 3).

3-the main successor (process number 4) checks if the coordinator is crash by sending election massage to it , if the coordinator respond then Process 4 stop election, else it announce itself as coordinator by sending massage to all processes.

4-conseder that processes 5 recover from failure, as process 5 know that it the heights number it just send coordinator Message to all process, and become coordinator.

5-suppose that process 0 recover from failure it will and send only at maximum three query message to process 5,4,and 3 instead of holding election

6- Process 0 resave answer message from process 5, 4, and 3 it now that process 5 is the coordinator. The entire process is shown in Fig 4.

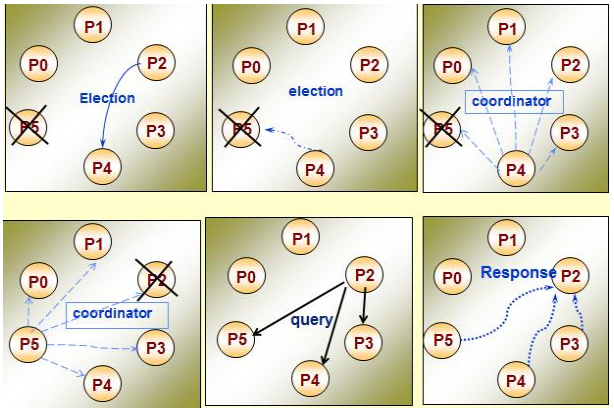

Fig. 4.

Limitation and advantage:

Best case: The main successor discovers that the coordinator is down and therefore it sends (n – 2) messages directly announcing itself as the new coordinator, only (n – 2) messages will be sent. Time complexity is O (n).

Average case: Any node other than the main successor discovers that the coordinator is down, it sends (n – 2) + 1 + 1 messages. Tim complexity is O (n).

V. Conclusion

In our paper we proposed an improved election algorithm and modified the previous election algorithms by using two successors for the coordinator. Our improved algorithm reduced massage passing and time complexity of the previous election algorithm and it solved coordinator failure.

References

[1]. S. Tanenbaum, *Distributed Operating Systems*, Pearson Education Singapore Pte. Ltd., 2002, ch4, pp. 140– 142.

[2]. D. Taj, "Divided ring leader election algorithm", *ICA3PP LNCS 5574,2009,.© Springer-Verlag Berlin Heidelberg*, pp. 90– 99, 2009.

[3]. H. Garcia-Molina, "Elections in distributed computing system," *IEEE Transaction Computer*, vol.C-31, pp.48- 59, Jan.1982.

[4]. H. Sayeed, M. Abu-Amara, and H. Abu-Avara, "Optimal asynchronous agreement and leader election algorithm for

complete networks with byzantine faulty links", *Distributed Computing*, vol.9, no.3, pp.147-156, 1995.

[5]. M.Rahman and A. Nahar, "Modified bully algorithm using election commission". *MASAUM Journal of Computing*, vol.1 no.3, pp. 88- 96, October 2009.

[6]. Q. mamun,S.masum and M. 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.

Dr. Basim Alhadidi was born in Jordan in June 1972; He earned his Ph.D in 2000, in Engineering Science (Computers, Systems and Networks) He received his M.Sc in 1996 in Engineering Science (Computer and Intellectual Systems and Networks). He is presently associate professor at the Computer Science Department at Al-Balqa' Applied University, Jordan.

Dr. Alhadidi published many research papers in many topics such as: computer networks, image processing, and artificial intelligence. He is a reviewer for several journals and conferences. He was appointed in many conferences as keynote speaker, reviewer, track chair and track co-chair.

Mr. Laith Hasan Baniata was born in Jordan in September 1985, He earned a bachelor degree in computer science from Al-Balqa' applied University in Salt, Jordan in 2007 and he got a Master degree in computer science from the same university in 2009. He has been working as a LECTURER in computer science department at university of Hail in Saudi Arabia since September 2009.His predominant research interests have evolved into three categories that he believe are interconnected. The three areas of interest are Data mining classification methods,, computer Networks and VPN and Agent oriented reverse engineering.

Mr. Mohammad Hasan Baniata was born in Jordan in August 1987, He holds a bachelor degree in computer Information systems from University of Jordan Amman, Jordan in 2009 and he earned a Master degree in computer information systems from the same university in2011. He has been appointed as a lecturer in computer science department at University of Hail in Saudi Arabia since September 2011.His research interests are Operating systems Algorithms, Networking and GIS.

Mr. Mohammad Al-Sharaiah was born in Jordan in 1980, He earned a bachelor degree in computer science from Philadelphia University in Amman, Jordan in 2003 and he got a Master degree in computer science from the Al-Balqa' Applied University, Jordan in 2010. He has been working as a LECTURER in computer science Department at university of Jordan in Jordan since 2004.His predominant research interests have evolved into three categories. The three areas of interest are Bioinformatics, computer Networks, mobile agent security, algorithms and VPN.