NTBCBT:A DISTRIBUTED MUTUAL EXCLUSION ALGORITHM

Richa[1], Shikha[2] and Pooja[3]

[1] Department of Computer Science, HEC, Jagadhri
richaaggarwal8@yahoo.co.in
[2]. Department of Computer Science, HEC, Jagadhri
getshikha.garg@gmail.com
[3]. Department of Computer Science, HEC, Jagadhri
poojadhiman20@gmail.com

ABSTRACT

In the recent years there is a considerable interest in developing mutual exclusion algorithms. Various algorithms are available to achieve mutual exclusion. Bandwidth, synchronization delay and throughput are the performance factors for these algorithms. Here we present an algorithm NTBCBT (Non Token Based using Complete Binary Tree) that using the concept of complete Binary Tree and provide better bandwidth as compare to other algorithms.

KEYWORDS

Non Token Based, Mutual Exclusion, Complete Binary Tree, Message Passing

1. INTRODUCTION

In distributed systems several computer communicate with each other by interchanging the messages over the communication network without sharing any memory or clock. In these systems one of the basic & interesting problems of mutual exclusion exists. Mutual exclusion gives the assurance that the use of the shared resource is restricted to one site at a time in a distributed computer system. Due to the lack of shared memory and a global clock and unpredictable message delay, the job of designing a distributed mutual exclusion algorithm that is fair, fault tolerance and free from deadlock and starvation, is difficult. Distributed mutual exclusion algorithms are either token-based [2] or nontoken-based. In token-based mutual exclusion algorithms, a unique token exists in the system and only the holder of token can access the protected resource. Examples of token-based mutual exclusion algorithms are Suzuki-Kasami's algorithm [5], (N or 0 messages for each CS), Singhal's heuristic algorithm [4], ([N/2,N] messages), Raymond's tree-based algorithm [6], (log (N) messages), Yan et-al.'s algorithm [21], (O(N) messages), and Naimi et-al.'s algorithm [3], (O(log(N)) messages). Non token-based mutual exclusion algorithms exchange messages to determine which process can access the CS next. Figure 1 shows the concept of Mutual exclusion. Examples of nontoken-based mutual exclusion algorithms are Lamport's algorithm [7,8], (3(N-1) messages), Ricart- Agrawala's algorithm [9], (2(N-1) messages), Carvalho-Roucairol's modification on Ricart- Agrawala's algorithm ([0,2(N-1)] messages),Maekawa's algorithm [12,11], ([3rootN, 5rootN] messages), and Singhal's dynamic data structure algorithm [10], ([N-1, 3(N-1)/2] messages).

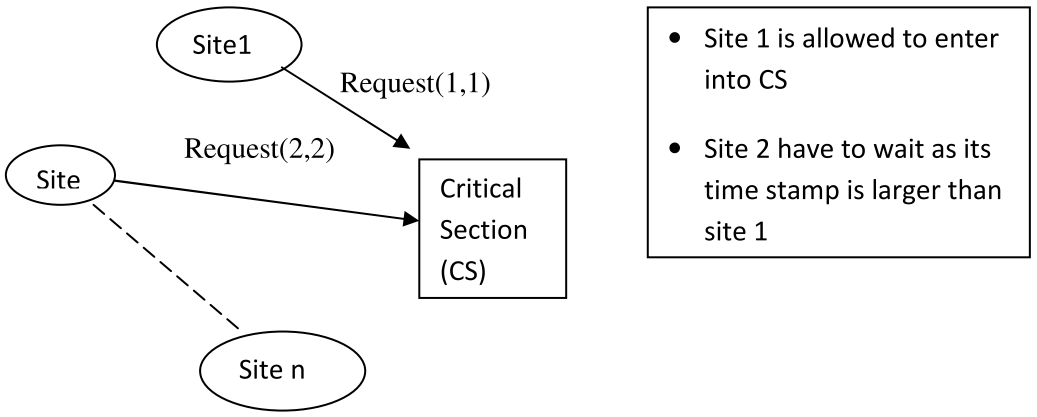
Site1
Request(1,1)
Request(2,2)
Site
Critical Section (CS)
Site n
Site 1 is allowed to enter into CS
Site 2 have to wait as its time stamp is larger than site 1

Figure 1: Concept of Mutual Exclusion

[1] In this paper, we present a simple, efficient and non token based distributed mutual exclusion algorithm for a distributed computer system of *N* geographically isolated computer sites. The aim of this work is to improve the performance of non token based algorithms by considering that sites are arranged as a complete Binary Tree. Here we present an algorithm which reduces the message traffic while maintain synchronization delay. The algorithm is deadlock free and free from starvation. The rest of this paper is organized as follows: In the next section we provide a brief overview of related work. In section 3, we give the distributed system model for our algorithm. The basic idea of the algorithm is given in section 4, followed by the description of the algorithm in section *5.* The proof of the correctness of the algorithm is given in section 6. Section 7 discusses the performance analysis of the algorithm. We conclude with some final remarks.

2. RELATED RESEARCH

During the last decade, a huge amount of effort has been focused on the development of the efficient distributed mutual exclusion algorithms. One of the earliest efforts in the area was by Alsberg and Day [14]. They appoint a central node to manage the access to a shared resource and the sites by enter their critical sections need to get permission from this Central site. This solution is highly efficient in that it requires only three messages per critical section execution, regardless of the size of the system or the compactness of requests. However, an obvious drawback of this solution is that the central site is unfairly burdened with responsibility and with message traffic. There after m many distributed mutual exclusion algorithms have been proposed. In general, these algorithms can be classified into two groups [13, 15] *permission based* and *token based.* Lamport [16] was the first to design a fully distributed permission based mutual exclusion algorithm using logical timestamps. In his algorithm each request set is the entire network. Then, if *N* is the numbei- of sites in the system, and if self-messages are not counted, the algorithm requires *N* - 1 requests, *N* - 1 replies, and *N* - 1 releases; or *3(N* - 1) messages per critical section execution. Ricart and Agrawal [17] realized that if all sites must grant permission by sending replies, then the release messages are superfluous, since a reply involves an implicit release. Therefore, they have reduced the number of messages in Lamport's algorithm to *2(N* - 1). Then a Makewa's [18] describes an algorithm in which permission have to be taken only from a subset of the sites included in the system.they reduced the number of messages to 3√N. But this algorithm is not deadlock free. For deadlock removal it required $5\sqrt{N}$ messages per critical section execution. In the token based algorithms the TOKEN is a "unique and singular" message which circulates among the sites. Only the site which possesses the TOKEN may enter its critical section. The various token based algorithms are distinguished by the methods for determining how a site obtains the TOKEN, where a site sends the TOKEN when it is finished with its critical section. One of the earliest token based mutual exclusion algorithms is by LeLann[19]. Suzuki and Kasami[18] present a token based distributed mutual exclusion algorithm which is an improvement over the well known Ricart and Agrawala's permission based algorithm. There after a series of more improved token based algorithms were introduced .Usually the mutual exclusion problem in a dis tributed system is limited in that only one node can be in its critical section (CS) at any time. Raymond [18] has removed this restriction on the entry to CS by allowing K sites to enter the critical section simultaneously, where $1 < I< < N$ and N is the number of the sites in system. With this change, Raymond has extended the Ricart and Agarwala's distributed mutual.Recently, Srimani and Reddy have proposed another distributed mutual exclusion algorithm which allows K simultaneous entries into a critical section in a distributed system. Their algorithm extends the mutual exclusion algorithm of Suzuki and Kasami which is token based. However, unlike in the original algorithm of Suzuki and Kasami which uses only one token message to provide mutual exclusion, in their algorithm there are K token messages to allow $I<$ simultaneous entries in the critical section. Although Srimani and Reddy's algorithm need the same number of messages per CS invocation as Suzuki and Kasami's algorithm, its response time for requesting CS is prolonged. It is because that more than one token may respond to the same request while other concurrent requests have to suspend until one of the collision tokens is redirected. The synchronization delay is also affected by this problem. More recently, Makki et al. [15]proposed a more efficient token based distributed mutual exclusion algorithm for multiple entries to the critical section. Their algorithm is based on an efficient token based mutual exclusion algorithm of Makki [19]

3. REPRESENTATION FOR DISTRIBUTED SYSTEM

We take a distributed system to mean a collection of N physically dispersed autonomous computer sites that are logically form a complete Binary Tree and communicate with one another only by sending messages. The sites do not share a common memory or global clock. We are **as**suming that the sites are numbered from 1 to *N .Complete network form a tree topology.* it is assumed the underlying net**work is** reliable and sites do not crash. Also we assume that the delay of a message delivery is unpredictable, but it is finite. The messages between any pair of nodes are transferred in the order they were sent.

4 ALGORITHM & ANALYSIS OF NTBCBT

There are four types of messages used in our algorithm. i.e. REQUEST, INFORM, REPLY, and REMOVE. All of them have the same simple format. Every type of messages includes its Sender-ID, a Recipient- ID and a Time-stamp. The four type of messages will be assigned the following semantics. A REQUEST indicates a requesting node request for using the CS.Inform Message is used to inform to children of a node that set their flag variable. Reply message is used to give permission to the requesting node for entering into critical section and REMOVE message is used to exit from critical section.

4.1Algorithm

Here is the algorithm explained in three sections describe following

4.1.1 Requesting CS

If the s_i site want to enter into CS then

```
1   Make an entry in the request queue of ith site.
2   It send a request message to its parent located at i/2
3   Inform its children to set their flag to 1and enter id of its children to the RF Array.
4  On getting a request message site j will perform the following

        a. It will make an entry in its request queue
        b. Repeat the following while j!=root
        c. If(flag[j]=1 )
             Then
                  wait
             Else
                 Set child=j
               Set j=j/2
             End of if structure
             End of loop
        d. if (flag[root] =1)
           Then
                wait
             Else
               set flag=1
                  send the inform message to another child  (other than child) so that it will set its
                 flag  variable to 1 and make its entry to RF array.
               send the reply message to the child pointed by child
         [ End of if structure]

5  On getting reply message the site will set its flag to 1
6  Remove the topmost entry from the request queue and send the reply message
   to the site i
7  Send an inform message to other child to set  its flag variable and make its
   entry in the Rf array.
```

4.1.2Executing CS

```
8   On getting reply message from its parent site the site will enter into the CS
```

4.1.3Release CS

```
9   Set its flag variable to 0.
10 Send a REMOVE message to all the sites in the RF Array and delete them from the
    array.
11  On getting remove message sites will clear their flag variable.
12  Send a release message to its parent if its not ROOT.
13  On getting release message go to step 9
14  If its request queue is not empty then go to step 4B.
```

4.2 Analysis & Results

The performance criteria on which the distributed mutual exclusion algorithms can be analyzed are bandwidth, delay and throughput. Bandwidth can be defined as the total number of messages sent in each entry and exit operation. We have analyzed various distributed mutual exclusion algorithm by implementing them in C. we analyzed these algorithms for n=10,20,30,40.Here n is the no of sites competing for mutual exclusion to enter into critical section.

Table 1: Comparison of Bandwidth of DME Algorithms

	Lamport				Ricart Agrawala				Maekawa				NTBCBT			
No. of Sites(n)	10	20	30	40	10	20	30	40	10	20	30	40	10	20	30	40
Bandwidth	27	57	87	107	18	38	58	78	16	22	27	32	12	16	20	20

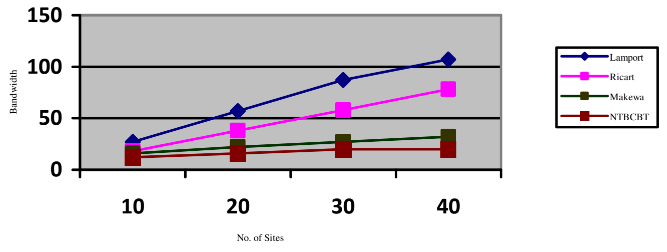

Figure 2: Bandwidth Vs No. Of Sites

Table1 and Figure2 shows the Bandwidth of various algoriths.Another criteria on which these algorithms are analyzed is synchronization delay. It is the time interval between one process exiting the critical section and the next process entering it (when there is only one process waiting). The data collected is given below

Table 2: Comparison of Synch. Delay of DME Algorithms

Name of Algo	Lamport				Ricart Agrawala				Maekawa				NTBCBT			
No. of Sites(n)	2	3	4	5	2	3	4	5	2	3	4	5	2	3	4	5
Synch. Delay	1	1	1	1	1	1	1	1	2	2	2	2	2	4	4	4

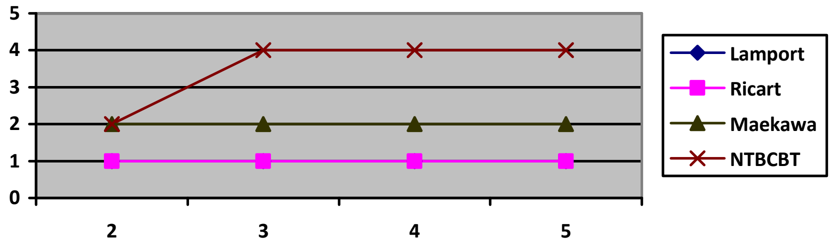

Figure 3: Synchronization Delay Vs No. Of Sites

5. CONCLUSION & FUTURE WORK

We presented a mutual exclusion algorithm for a distributed system with asynchronous message passing. This algorithm requires 4 $\log_2 i$ messages per access to the critical section, and improves upon the Ricart-Agrawala algorithm, which is the best known algorithm, without introducing any additional overhead. Bandwidth is not only the single criteria for checking out the performance of DME algorithms. There are some other factors like Synchronization delay, response time etc. These factors are also considered during design of the algorithm Further we are trying to improve the synchronization Delay and response time. At present these factors are very close to previous algorithms.

REFERENCES

[1] Moharram Challenger, Peyman Bayat,&Mohammad Reza Meybodi (2006)"A reliable optimization on distributed mutual exclusion algorithm",*IEEE Conference: Testbeds and Research Infrastructures for the development of networks and communities - TRIDENTCOM* pp. – 566-574

[2] Toyomura, M., Kamei, S. & Kakugawa, H.:(2003) "A Quorum- Based Distributed Algorithm for Group Mutual Exclusion", *IEEE Trasns. On Distr. and Par. Sys*tem , pp: 742 - 746

[3] M. Naimi, M. Trehel, & A. Arnold, (1996) "A log(N) Distributed Mutual Exclusion Algorithm Based on Path Reversal", *J. Parallel and Distributed Computing,* vol. 34, pp. 1-13,

[4] Singhal M.,(May 1989) "A Heuristically Aided Algorithm For Mutual Exclosion In Distributed Systems", *IEEE Transaction On Computers*, , Vol. 38, No. 5, PP. 651-662.

[5] Suzuki I. And Kasami T., (Nov 1985)"A Distributed Mutual Exclosion Algorithm", *ACM Transaction On Computer Systems*, , Vol. 3, No. 4, PP. 344-349.

[6] K. Raymond, (Feb.1989.)"A Tree-Based Algorithm for Distributed Mutual Exclusion", *ACM Trans. Computer Systems,* vol. 7, pp. 61-77

[7] Lamport, L.:(July 1978) "Time, Clocks, and the Ordering of Events in a Distributed System", *Commun. of the Acm*, vol. 21, pp. 558-564,.

[8] Lamport, L.:(Nov. 1990) "Concurrent Reading and Writing of Clocks",*ACM Trans. on Computer Systems*, vol. 8, pp. 305-310,.

[9] Ricart, G., and Agrawala, A.K.:(Jan. 1981) "An Optimal Algorithm for Mutual Exclusion in Computer Networks", *Commun. of the ACM*,vol. 24, pp. 9-17

[10] M. Singhal,(Jan. 1992.) "A Dynamic Information Structure Mutual Exclusion Algorithm for Distributed Systems", *IEEE Trans. Parallel and Distributed Systems*, vol. 3, no. 1, pp. 121-125

[12] Maekawa, M., Oldehoeft, A.E., and Oldehoeft, R.R.(1987): "Operating Systems Advanced Concepts", *Menlo Park, CA: Benjamin/Cumings*, pp:200-208

[11] Maekawa, M.: "A Square-root(N) Algorithm for Mutual Exclusion in Decentralized Systems"(May 1985), *ACM Trans. Comp. Syst*., vol. 3, no. 4, pp. 145-159

[13] K. Makki, "An Efficient Token Based Distributed Mutual Exclusion Algorithm," *Journal of Computer and Software Engineering*, Vol. 2, No. 4, (December 1994), PP. 401-416.

[14] P.A. Alsberg and J.D. Day (October 1980)"A Principle for Resilient Sharing of Distributed Resources," *Proc. Of the second International Conf. on Software Engineering,* St. Louis, MO, PP. 23-31

[15] K. Makki, K. Been and P. Pissinou, (April 1994)"A simulation study of token-based mutual exclusion algorithms in distributed systems," *International Journal in Computer Simulation,* Vol. 4, No. 1, , PP. 65-88.

[16] L. Lamport, (July 1978),"Time clocks and ordering of events in distributed systems," *Communications of the ACM*, Vol. 21, No. 7, PP. 558-565

[17] G. Ricart and A. Agrawala, "An optimal algorithm for mutual exclusion in computer networks," *Communications of the ACM,* Vol. 24, No. 2, pp.212-215

[18] Mukesh Singhals(2001) "Advanced Concepts IN Operating Systems"*Tata McGraw-Hill Edition,*Vol- 1,*PP 122-132*

[19] N.Pissinou * K. Makki E.K. Park, Z. Hu W. Wong (1996) "An Efficient Distributed Mutual Exclusion Algorithm," *International Conference on Parallel Processing(IEEE):* pg 196-203

AUTHORS

Richa Gupta born in india, on Oct. 29, 1983. After completing, Bachelor of Technology in the field of Computer Engineering from Haryana Engineering College, Jagadhri, Haryana 2005, Kurukshetra University, she has done Masters in Technology in the field of Computer Engineering from Seth Jai Prakash Mukand lal Institute of Engineering and Technology, Radaur, in 2009, Presently, she is working as Lecturer in the Department of Computer Engineering at Haryana Engineering College, Jagadhri, Haryana (India).

Shikha Garg born in india, on Nov. 14 1986. After completing, Bachelor of Technology in the field of Computer Engineering from Haryana Engineering College, Jagadhri, Haryana 2008, she has done Masters in Technology in the field of Information Technology from Karnataka State University. Presently, she is working as Lecturer in the Department of Computer Engineering at Haryana Engineering College, Jagadhri, Haryana (India).

Pooja Dhiman born in india,on jan.. 20, 1986. After completing, Bachelor degree in the field of Computer science from Guru Nanak Khalsa College, Yamunanagar, Haryana 2007, Kurukshetra University, she has done Master Of Computer Applications from Tilak Raj Chadha Institute of Management and Technology, Yamunanagar, affiliated to Kurukshetra University , Kurukshetra , in 2010and pursuing M.TECH ,Presently, she is working as Lecturer in the Department of Computer Engineering at Haryana Engineering College, Jagadhri, Haryana (India).