FAULT TOLERANT PERMISSION BASED CLUSTERED MUTUAL EXCLUSION ALGORITHM IN DISTRIBUTED SYSTEM

Rishikesh Rawat
PhD Research Scholar
Mewar University, Chittorgarh,
Rajasthan, India

Dr. Santosh Kumar Gandhi
OSD
Directorate of Technocal Education
Bhopal (M.P.),India

Abstract: Mutual exclusion (MUTEX) refers to a way of making sure that if one process is using shared modifiable data or resources then the other processes will be excluded from doing the same thing at the same time. A number of mutual exclusion algorithms are available in the literature, with different performance metrics and with different techniques. The Selection for a "good" mutual exclusion algorithm is a key point. These mutual exclusion algorithms can be broadly classified into token and non-token based algorithm. For distributed mutual exclusion problem in mobile environment we presented two algorithms. Both the algorithms are token based. The first algorithm is providing tokens to the nodes which are in critical section and the second one is hierarchical clustering based. A mobile ad-hoc networks (MANET) is hierarchically (two level) clustered to get logical tree network through which the token is passed from one node to another. However, we simulate the second algorithm only. The evaluation of the proposed algorithms shows that its message requirement is optimal. The proposed algorithm reduces the message complexity as well as response time, synchronization delay and it is also ascertained by simulation results

KEYWORDS: Mutual Exclusion, Permission based algorithm, Fault tolerance, Distributed system

1. INTRODUCTION

Mutual exclusion defines the need of knowing that no two parallel nodes are in the critical section at the same time. Parallel nodes should be clocked simultaneously to obtain the split resources. If more than one node is in the critical section it leads to breaking of unity. If there are no fixed infrastructure means, then the mobile devices communicate over wireless links and cooperate in a distributed manner, a mobile ad-hoc network is an independent collection of mobile devices. A critical section is the part of a program that accesses the shared resources. It leads to Distributed Mutual Exclusion (DME), if more than one node wishes to enter CS for accessing the critical resources simultaneously. Methods for DME problem can be categorized into two groups, based on the Way of selection of node to enter in CS, are described below:

- Token-based algorithms.
- Permission-based algorithms.

In permission based algorithm, if the node needs to enter the CS must collect the permission from all other participating nodes. In the later algorithm, nodes which itself has token is allowed in CS and then the token is passed in other nodes in the network. In the proposed algorithm, two types of tokens i.e.

- Primary token.
- Secondary token.

In the mutual exclusion algorithm proposed by Lamport , when a node needs to enter in its CS, it sends the request to neighbor nodes and it waits for responses. When the node moves out from its CS, it sends a release message to neighbor nodes. This algorithm needs 3*(N-1) messages per CS entry. Author[2] suggested that a quorum needs not to consult other quorums that are not currently present for CS look ahead

technique was introduced to decrease the message complication. Dynamic information sets, consists of Info set and Status set, to keep the set of quorums that are presently involved in Ricart And Agarwal[1] introduced a distributed mutual exclusion algorithm which needs 2*(N-1)messages per CS entry. The time a node enters the CS, it sends the request message to neighbor nodes of the network and waits for the reply message. If it receives the bond from all of these nodes, it enters the CS Response Time is calculated using Lamport clocks. Maekawa5 has proposed the algorithm that needs c*(√N) messages has to enter into the critical section. It uses a logical structure, which has a collection of nodes involved with each of the nodes and this in turn has a set of node has a non-null intersection with every set of node involved in each of the nodes. It allows each of the nodes, which to access its CS, to have the permission only from the each member of the set of nodes involved into it. Singhal increased the performance of the Suzuki and Kazami algorithm, till N messages are in heavy loads. In this method, the heuristic method used to select which the nodes of a system, that are possibly holding or to have the token, the so token request message is sent only to those nodes other than to all the neighbor nodes. When token returns to the requester along the reverse link, it avoids the cycle.

2. MODEL AND PROBLEM DEFINITION

2.1 System Model

In general, most of mutual exclusion algorithms use a common model. In this model, a distributed system is a set of independent and autonomous computers. These computers are called as node or site and connected via a communication network. Each node has abstract view of whole system that communicates with message passing [4]. The most important purposes of distributed system are assigned as providing an appropriate and efficient environment for sharing resource, having an acceptable speed and high reliability and availability.

2.2. The Mutual Exclusion Problem

Mutual exclusion problem in distributed systems has received great consideration in recent 3 decades. This problem ensures that concurrent processes access common resource and data sequentially. In addition each process that executes in its critical section for a finite time, must do without interfering with other processes. Also, when no process is in a critical section, any process that request entry to its critical section must be permitted to enter without delay [3]. Eventually mutual exclusion must be without deadlock and starvation.

3. RELATED WORK

Ricart and Agrawala proposed the fair algorithm [1] that need 2(n-1) messages for a node to use the critical section. This algorithm is the first permission-based ME algorithm where a node need to collect permission from all other node for CS access. Though the algorithm is deadlock and starvation free, it is vulnerable to node and communication There is elegant class of permission-based algorithms [6] that use concept of quorums to achieve mutually exclusive access of CS. A node needs to achieve permissions from all the nodes of a quorum to access CS. Quorum based algorithms are resilient to node and communication failures and often network partitioning and usually have lower communication cost. Communication cost of these algorithms is proportional to the quorum size. Therefore these algorithms try to achieve the two goals: small quorum size with high degree of fault tolerance. Its basic idea is to collect enough permission (votes) to guarantee the mutual exclusion. The majority quorum algorithm [8] can be considered as the first algorithm of this kind, where to attain mutual exclusion, a node must obtain permission from a majority of nodes in the network. Maekawa [4], proposed an ME algorithm by imposing a logical structure on the network. In this scheme, a set of nodes is associated with each node, and this set has a nonempty intersection with all other sets corresponding to the other nodes, which guarantee the ME. The size of each of these sets is n and so the algorithm cost n order.

Garcia-Molina and Barbara [2] have properly defined the concept of quorums with the notion of coterie. A coterie is a set of sets with the property that any two members of a coterie have a nonempty intersection and the minimality property. Combining the idea of logical structures and the notion of coteries, an efficient and fault tolerant quorum generation algorithm for ME is proposed by Agrawal and Abbadi [5]. Here, the nodes form a logical binary tree which is used to generate quorums. The quorum forming in this algorithm is recursive. It can be regarded as attempting to obtain permissions from nodes along a root-to-leaf path. If the root fails, then the obtaining permissions should follow two paths: one root-to-leaf path on the left

subtree and one root-to-leaf path on the right subtree. The algorithm tolerates both node failures and network partitions while in the best case incurring logarithmic costs considering the size of the network. But the cost increases with the increase of node failures.

Walter-Cao-Mohanty [3] presented fault-tolerant token based distributed k-mutual exclusion algorithm which adjusts to node mobility. The algorithm has been specifically designed to suit ad-hoc environment, since; it requires nodes to communicate with only their current neighbors. A “token forwarding" modification to the basic algorithm is shown to lower the time each node waits to enter the CS by circulating unused tokens among participating processors[5,3].

Masum et al. [4] presented a consensus-based mobility aware l-Exclusion (LE) algorithm that operates asynchronously and copes explicitly with arbitrary (possibly concurrent) topology changes associated with such networks. The algorithm is consensus based and a mobile node intending to enter CS tries to collect enough consensuses, and uses diffusing computations for this purpose. This paper presents a simulation to demonstrate that the proposed algorithm, as compared to the k-Reverse Link (KRL) algorithm, is quite effective to variety of operating conditions.

4. TOKEN BASED ALGORITHM

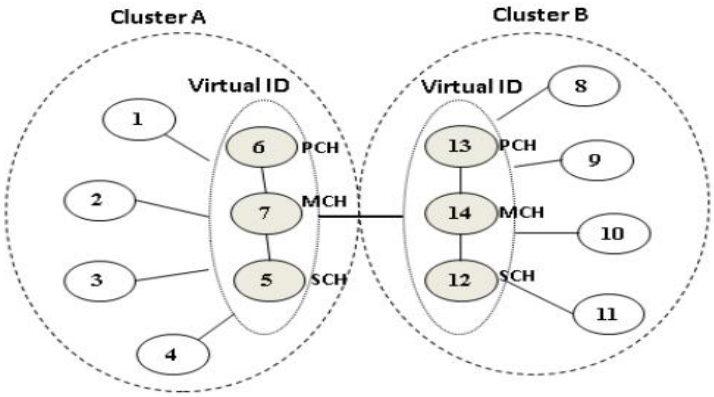
A network having N number of nodes is splitted into M groups. We consider for simplicity M = N. A node can execute the critical section if that node holds the token. We assume that there is only one token in the network. In this scheme to pass the token within a group, every members of that group execute an algorithm at the same time and the token is passed from one member to another member within a group without any message passing. Each member of the group knows that which member of the group holds the token. Any member can execute the critical section if it gets the token after completion of that algorithm. If at that time any other members of a group want to execute the critical section and if that node is not a token holding node then it sends a request message to that node which have the token and belongs to the same group. If at that time the token holder is not executing the critical section, then that node can grant permission to the member whose request received first. Every node in the network holds a list. The list receives request from other members within its group or from other groups. Size of the list is one, so one node receives only one token-request. When execution of the critical section is completed then the token holding node will sends release message to every member of that group. After receiving the release message every member of the group including the token holding node again start the algorithm. After completion of execution of the algorithm the token will pass to a new node without any message passing. New token holding node will send a message to the last token holder of other groups that he is the new token holder of that group. New token holding node can execute the critical section or can grant permission to first requesting member of the group to execute the critical section. If the token holder has already grant permission to other member of the group to execute the critical section, then that member will send a release message to the token holder after executing the critical section. After receiving that release message the token holder sands release message to every member of that group. The release message contained the ID of last token holding node of other groups. If the token is not within a group and if any last token holders of that group want to execute the critical section then it sends a request message to the token holding node in other group. If at that time the token holder remains free then he will release the token to one of the last token holder of other groups whose request received first. If other members of that group want to execute the critical section then it has to sends a request to the last token holding node of its group and then that node sends a request message to the node in other group who holds the token

5. AN EFFICIENT ALGORITHM FOR DISTRIBUTED MUTUAL EXCLUSION IN DISTRIBUTED SYSTEM

5.1. Proposed Technique

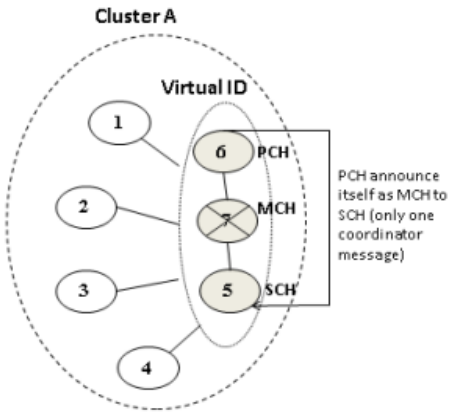
The prime motivation of our proposed algorithm is to guarantee that the new cluster head can be elected with less number of messages, less number of participants in the election process and also reduce the single point of failure. The proposed fault tolerant permission based clustered mutex algorithm comprises virtual cluster head election algorithm to tolerate cluster head failure. The set up consists of a Main Cluster Head

(MCH), two backup heads namely Primary Cluster Head (PCH) and Secondary Cluster Head (SCH). Node with highest ID is selected as MCH and the next two highest ID nodes within the cluster will be selected as PCH, SCH. The main cluster head coordinate the cluster members i.e., handle the CS request and primary, secondary cluster heads observe the main cluster head. A virtual ID is given to cluster heads, the members will send CS request message to this ID and initially virtual ID is mapped to main cluster head. The cluster members are not aware of this virtual group.

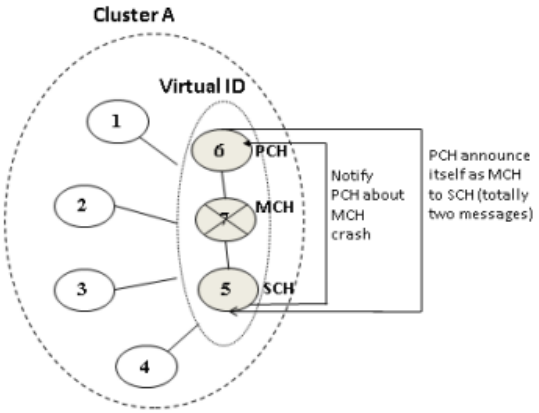


Fig(1): Virtual Cluster Head Election Algorithm

Fig.1 shows the model of virtual cluster head election algorithm. Nodes are first grouped into cluster then cluster heads are selected. For cluster A, the highest ID node is 7 and it is selected as Main Cluster Head, next highest ID 6 is selected as Primary Cluster Head and next ID 5 is chosen as Secondary Cluster Head. A virtual ID is created and it is mapped to MCH. Node 5 and 6 continuously monitor node 7. The bold line between two clusters indicates the connection between cluster heads i.e., cluster A‟s virtual head ID to cluster B virtual head ID. The primary and secondary cluster head constantly monitor the main cluster head by sending „Are You Alive (AYA)‟ message periodically and the MCH responds by sending „Yes I am Alive (YIA)‟ message. If there is no response from MCH (best case), primary will take over the job of main cluster head, map its ID to virtual ID and intimate to secondary cluster head as a new head and it is represented in Fig.3. If the failure is detected by secondary cluster head (worst case) it will intimate to primary cluster head about main cluster head and the primary will take over further operation, denoted in Fig.4. If primary doesn't respond properly, the secondary cluster head will coordinate the members. In this way the election is done by using only three nodes in the cluster. This will reduce the participant "number in the election process and also the number of messages needed for declaration of a new cluster head.



Fig(3): Failure of Main Cluster Head noticed by Primary Cluster Head



Fig(3): Failure of Main Cluster Head noticed by Primary Cluster Head

4. CONCLUSION

A fault tolerant permission based clustered mutex algorithm for Mobile Ad-hoc networks has been proposed. Data or resource sharing is one of the central objectives of distributed systems. When more than one distributed nodes tries to access a common resource it leads to inconsistency. To prevent this so many mutex algorithm exists. The proposed algorithm considerably reduced the overall communication required per mutually exclusive access and also has considerable response time, synchronization delay when compared to existing ones. We first include clustering approach to reduce the message complexity and since cluster head plays a vital role, it may leads to failure. To handle cluster head failure a virtual cluster head election algorithm has been proposed. It reduces the number of participants in the election process so that only constant numbers of messages are required for conducting election and declaration of new cluster head

REFRENCES

1. G. Ricart, and A. K. Agrawala, "An Optimal Algorithm for Mutual Exclusion in Computer Networks," Communications of the ACM, Vol. 24, No. 1, pp. 9-17, 1981
2. Garcia-Molina,"Elections in a distributed computingsystem," IEEE Transactions on Computers, January 1982.
3. J. Walter, G. Cao, and M. A. Mohanty, "k-mutual exclusion algorithm for ad hoc wireless networks," in Proceedings of the 1st Annual Workshop on Principles of Mobile Computing, 2001, pp. 29-39.
4. Mamun, Q.E.K.; Masum, S.M.; Mustafa, M.A.R.,"ModifiedBully Algorithm for ElectingCoordinator in DistributedSystems," International Conference on Software Engineering,Parallel and Distributed Systems, 2004
5. A.S. Tanenbaum and M.V. Steen, "Distributed systems: principles and paradigms," 2nd edition, Prentice Hall of India, 2006.
6. Anchal, C. Ramakrishna, P. Saini, "A survey on permission based distributed mutual exclusion algorithms in mobile ad-hoc networks," International Conference on Computer Network and Information Technology, 20-21 March, 2014.
7. Sung-Hoon Park, Seoun-Hyung Lee, "Token-Based Mutual Exclusion Algorithm in Mobile Cellular Networks,"International Conference on Advanced Information Networking and Applications Workshops, 2012.
8. Garcia-Molina,"Elections in a distributed computingsystem," IEEE Transactions on Computers, January 1982.
9. Greg, N. F., Nancy A. L., "Electing a leader in a synchronousring," Journal of ACM, 1987.
10. Candido Caballero-Gil, Pino Caballero-Gil, and JezabelMolina-Gil, "Self-Organized Clustering Architecture for Vehicular Ad Hoc Networks," International Journal of Distributed Sensor Networks, Volume 2015, Article ID 384869and Computer Security (CCCS) Newcastle University, Newcastle upon Tyne, NE1 7RU, UK,2016.
11. H. Garcia-Molina, and D. Barbara, "How to assign votes in a distributed system," Journal of the ACM, Vol. 32, No. 4, pp. 841-860, 1985.
12. Basim, A.,Laith, H.B, Mohammad, A., "Reducing MassagePassing and Time Complexity in Bully Election AlgorithmsUsing Two Successors," International Journal of Electronicsand Electrical Engineering, March 2013
13. S. Masum, M. M. Akbar, A. Ali, and M. A. Rahman, "A consensus-based l-exclusion algorithm for mobile ad hoc networks," Ad Hoc Networks, Vol. 8, 2010, pp. 30-45.