AN EFFICIENT METHOD OF FAULT TOLERANT PERMISSION BASED CLUSTERED MUTUAL EXCLUSION ALGORITHM IN DISTRIBUTED SYSTEM

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

Dr. Sarvottam Dixit
Professor
Deptt. Computer Science & Engineering,
Mewar University, Chittorgarh

ABSTRACT: The selection of a 'good' mutual exclusion algorithm, for the design of distributed systems, is of great importance. A number of mutual exclusion algorithms, with different techniques and varying performance characteristics, are available in the literature. These algorithms can be broadly classified into token based algorithms and non-token based algorithms. A number of survey papers for non-token based mutual exclusion algorithms exist. Although, some of them include discussion on token based mutual exclusion algorithms too, however, none of them include any discussion on the newer variants of classic mutual exclusion, like k-mutual exclusion and group mutual exclusion. we present the Permission-Based clustered Mutual Exclusion Algorithm in distributed systems. This algorithm provides better result than Reddy algorithm. The algorithm is depend on token-based technique and requires a lot number of messages to enter the critical section (CS) and to detect the system failures. We analyze the algorithm, provide performance results of the implementation, and discuss the fault tolerance and the other algorithmic extensions.

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

1. INTRODUCTION

The problem with mutual exclusion which occurs is when some two or more processes or devices try to work at same instant of time simultaneously [1]. When they compete for the critical section at the same time, no one of the devices get chance to share resources, and none of them can't use it fruitfully [2, 3, 4].So, to prevent from this issue, a distributed algorithms is designed to manage the critical region [5]. Critical section (CS) is a code in which sharing of resources is done or accessed. Now practically sharing common resources simultaneously is not possible and cannot be synchronized for sharing resources. So, if two nodes try to access the critical section can lead to crisis. Mutual exclusion is to ensure that at a time only one of the concurrent processes are allowed to access the common or shared resources at an instant of time. In case of distributed systems, where multiple sites are considered it is named as distributed mutual exclusion (DME). MANETs has no restrictions when it comes to topology and the sites are free to move within a region i.e. the critical region. This free moving of the sites in the region can generate link failure which is a main issue for us discussing Mutual Exclusion. Also the nodes use the battery power for the processing which makes them totally relay on those batteries. This is the major problem which is observed in MANETs than static networks. Another factor which has to be considered is to satisfy the Combinatorial Stability. It is a state when a node is waiting for making a decision of whether to enter critical section or not, at the same time the underlying topology of the network can change. So, the processing time of the algorithm should be fast enough that a change in the network layout does not affect the objective of the algorithm. It is a concern when it comes to token based DME algorithms or permission-based algorithms [8]. Distributed Mutual Exclusion Algorithm can be classified into two major categories: Token-based algorithm and Non token-based/Permission-based algorithm. Token-based algorithm depends on the site entering to critical section by accessing a token. Further it can be classified as circulating and requesting method. In the circulating method a token is passed among all the participating sites and the site which possess the token gets the chance to enter the critical section and after that it releases the token back in the circulation. The other method is requesting one in which it a site requests the other participants for entering into the critical section. Suzuki-kasami's algorithm is an example of tokenbased mutual exclusion algorithm for distributed systems. In the case of Permission-based a site desiring to enter to critical section must first get permission from all the sites before it enters in the critical section or from some nodes, it varies according to the algorithm. Lamport, RicartAgrawala,Roucairol and Carvalho's

algorithm are some examples of Non token-based mutual exclusion algorithms for distributed systems. Distributed Mutual Exclusion Algorithm can also be classified by three approaches named: Token-based approach, Non Token-based approach and Quorum based approach [9]. Below are listed the main requirements for mutual exclusion algorithm

2. RELATED WORK

Suzuki and Kasami's algorithm [1] reduced it to N (number of sites) number of messages in comparison to the Ricart-Agrawala Algorithm. It is a token-based algorithm in which mutual exclusion is achieved by maintaining a token among all nodes for entering the critical section.

Meaekawa [2] proposed one which uses the 'quorum' for making the decision regarding taking permission for using the critical section than taking permission from each site, it was called as quorum-based distributed mutual exclusion algorithm. In this it was proposed that the nodes which are participation in the process are in the quorum. The use of voting technique by Thomas is based on a majority attained by a node and requires that a node requesting mutual exclusion obtain a permission vote from only a majority of the nodes regardless all the nodes in the critical region.

Ricar-Agrawala [3] it requires 2 (N - 1) messages exchange for getting entry to critical section, while in Suzuki and Kasami [16] they reduced it to just N messages. Then Maekawa [11] improved the algorithm and reduced the minimum messages to O$\sqrt{N}$ (O - quorum). After that Raymond and Kerry [19] came with algorithm which enhanced the performance by using a tree topology and reduced the message required to O (log N)

Singhal, el al. [4] came with an idea that if a site is not competing for critical section then is it necessary to take permission from that site? So, they observed that a site need not consult other sites that are not currently in a need of critical section. They introduced an idea of 'look-ahead technique' in which before sending REQUEST message a site identifies that which or how many sites are concurrently competing for critical section and then enforcing mutual exclusion on those sites which are competing rather than to all the sites.

Wu et al. [5] introduced three new messages DOZE, DISCONNECT and RECONNECT. They also introduced FIFO (First-In-First-Out) service which was not feasible in case of MANETs as the sites in region have no fixed locations or topology to follow. The important problems which are to be considered in case of tolerance is link failure and host failure which is the very frequent in MANETs. If we use timeout and retransmission of *REQUEST* message then this link failure and host failure can be removed or minimized.

Sahel Alouneh, et al.,[6] has proposed and focused on path protection fault tolerance schemes as well as issues in MPLS (Multi-protocol label switching) network and steps to resolve it. MPLS networks are prone to failures thus fault tolerance is important as it focuses on factors like utilization, recovery time and packet loss. MLPS provide VPN functionality to increase security. The advantages of this method are high speed packet delivery, reduction in PSL (Path switching LSR), fast and cost effective notification. But packet loss factor depends upon the various approaches without any guarantee from VPN and it is the disadvantage of this method. Further works needs to be data integrity, confidentiality, authentication in MPLS and multi path routing.

Moushumi Sharmin,et al., [7] has proposed to design a resource discovery unit used to maintain privacy, resource sharing and modification scheme in an effective computing environment. Fault tolerance techniques uses secret sharing in which the process will not keep a single copy of resource manager to ensure the optimum use of tiny storage. For the security & reliability resource provider provides privacy and security to a specified resource and resource holder is responsible for providing it to approved person only. Resource manager manages resources, facilitates the best match and stores the address in the hash table. Disadvantages are in this increases seek time, wastage of resources & memory, low privacy of service information and domain identity.

M.AI-Kuwaiti et.al. [8] has proposed address issues of complex infrastructure such as information, reliability and availability with some fault tolerance and security features. In fault tolerance the system behaves normally in case of any hardware or software fault and fault masking is another method that can be used to tolerate fault. Reliability refers to any failure free operation during an interval and security protects confidentiality & integrity..

Ben Hardekopf,et al., [9] has proposed a system with security and fault tolerance in case of congestion and protocols to create a secure voting algorithm. Durability & Redundancy is used to reduce the risk of any single component in operating flawlessly is the advantage of this approach. Future work is that reduce network congestion, reliability & security for the WAN.

Jitendra K. Rout, et.al.,[10] has proposed a fault tolerance paradigm that can identify the black holes attack which degrades the performance of the network by dropping the packets. In the fault tolerance dividing the routing protocols into zones, if an error occurs in one then the other zones does not get effected and a node connection algorithm is used to detect where the fault has actually occurred. For the security purpose network connection algorithm used for fault tolerance and provide reliability. Its advantages are that many routing protocols have been discovered because of which data transmission and maintenance has been enhanced. Due to this increase in the number of routers, bandwidth and cost will be effective

Steven E.Crerwinski et.al. [11] has proposed to handle the failure automatically by hiding the complexity of fault recovery and to determine whether the communication among the components is secure or not. For the fault tolerance whenever a packet gets dropped, cryptographic methods provide strong authentication at end points. SDS server is used to calculate the number of clients in the system and verify that the security features of the system does not reduced and Authenticated RMI implementation is used to encrypt the data sent over the network. Its advantages are that it scalable, secure information repository and fault tolerant. Lack of access control, service information cannot be granted and cost in effective

Yaron Minsky et al., [12] has proposed a variety of applications in the internet and other large distributed systems. Replication and voting are not sufficient for improving the performance of agent computation. It involves fault tolerance, if there exists no faulty host in the pipeline then the remaining hosts separate the computations by sending an agent. The correctness of the present state depends upon the correctness of predecessor. It uses chain of authentication to prevent masquerading and when two hosts combine then voting will determine which faulty host is encountered and which is not, it is used for security factor .

Stephen Bohacek et.al.,[13] has proposed simulation that improves routing security, minimize the impact of link router and approaches to failure prevention and recovery. Used dynamic routing protocol to try to reduce the failure, when a fault is generated in one layer then transmission will keep following the same route and continue interception. Security contains protest against some form of interception; provide end to end security mechanism to other layers. Its advantages are to improve security and fault tolerance; proactive approaches to achieve connectionless value enable failure. Demerits are increasing throughput and complex multiple parts. For future work, scalable algorithm is required to compute next hop probabilities, decrease packet transmission delay and increase throughput.

Bharat B. Madan,et.al., [14] has proposed assessment of security attributes for an intrusion tolerates system. Fault tolerance ensures the effective recovery from failures and allows a finite probability; the system security may be breached. SMP model deals with security, Integrity and authorized actions. Merits are it is able to detect faults in the system and detect the insertion of the security attacks into a system. Various subsystems communicate with each other. But it can't compromise with data integrity and DOS attacks, consume large amount of service resources..

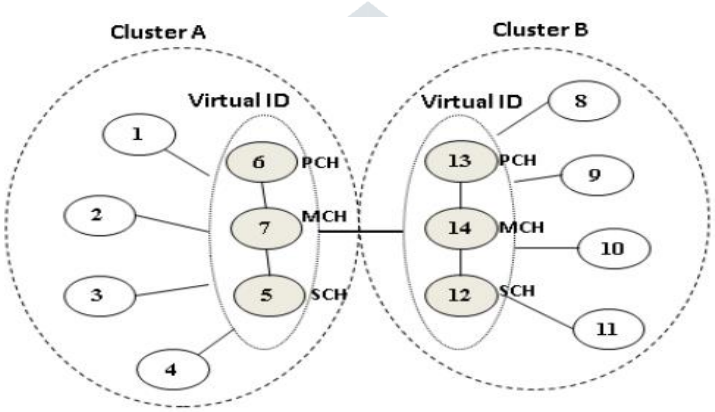
3. PROBLEM IDENTIFICATION

Distributed system is a collection of distinct processes that do not share memory or clock. These processes interconnected by a communication network in which each process has its own local memory and other peripherals. The communication between any two processes of the system takes place by message passing over the communication network. These processes in the distributed system run concurrently. The purpose of the distributed system is to provide an efficient and convenient environment for sharing of resources. In this paper, we present an efficient fault-tolerance algorithm for Mutual Exclusion (ME) in distributed systems. This algorithm is considered as an enhancement to Reddy algorithm. The algorithm is based in token-based technique and requires a lot number of messages to enter the critical section (CS) and to detect the system failures. In Reddy algorithm the process must request to enter the CS from all process all the times. The presented algorithms also can be seen from another point of view, the fault tolerant; we said that the algorithm is fault tolerant if it treats the failure and still function without any problems. There are many algorithms that take into account several types of failures and treat them [8]. By contrast, we cannot consider that the system is reliable and it is error free so to build an algorithm over this suggestion is not practical and they cannot be classified with the others that take in their accounts the error and any failures in the system. The fault tolerant algorithms could be different in their efficiency and complexity. For example, in token-based algorithm the efficiency depends on the number of messages needed for entering the CS, for detecting loss of token, and for generate new token.We present in this thesis an efficient Permission-Based clustered Mutual Exclusion Algorithm that reduces the number of messaged needed to enter the CS, the number of messages to detect the loss of token and to generate a new token in case of failures

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

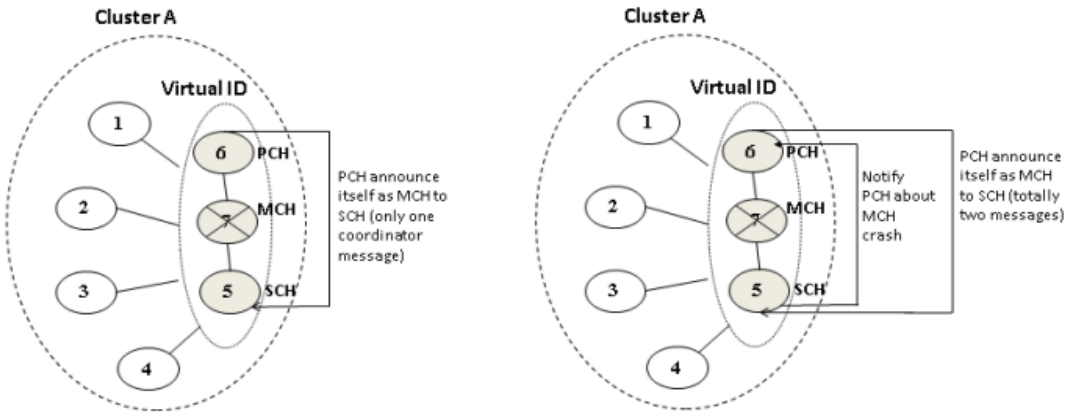
4.1. Proposed Technique

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(2): Failure of Main Cluster Head noticed by Primary Cluster Head

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

5. PERFORMANCE ANALYSIS

When we applied proposed algorithm in normal situation and in another case if there is one request to enter the critical section. We find that the number of messages transmitted between processes in this algorithm as follows.

(N-1) request message + one token message = (N Message)

Whereas: (N) number of processes in the system.

But when is more than one request to enter the CS we found the number of message in our suggestion as follow.

Whereas:

N: Number of processes in the system.

M: Number of processes wants to execute the critical section.

As shown in the following table:

Number of processes wants to execute the critical section.	Number of messages exchange between processes in our suggestion	Number of messages exchange between processes in Reddy's Algorithm
1	15	9
2	22	18
3	35	27
4	40	36
5	56	45
6	60	54
7	70	63
8	82	72

Table 1: The table shows No. of Message of Proposed approach and Reddy's algorithm

We cannot account the number of transmission messages between the processes if there any failure case in the system because the creator for this algorithm didn't take in the consideration the treatments operation in failure case.

When we tested both algorithms in failure situation and in case if there is one or more than process posses copy of the token after failure situation is done. We found that the number of messages transmitted between processes in both two algorithms as follows. First: Reddy et al's Algorithm

(M (N-1)) Request Message + (M (N-1)) Is-Token-Lost Message + (F M) Reply Message + (M) Generating Message + (F) Copy of the Token Message.

Whereas:

N: Number of processes in the system.

M: Number of processes wants to execute the critical section.

F : Number of processes posses' copy of the token

Second: Our Suggestion

(N-1) Request Message + X "is-token-lost" message + X Reply Message + one generating token message +. Whereas:

N: Number of processes in the system.

X: Number of process in the global queue at most 3 processes.

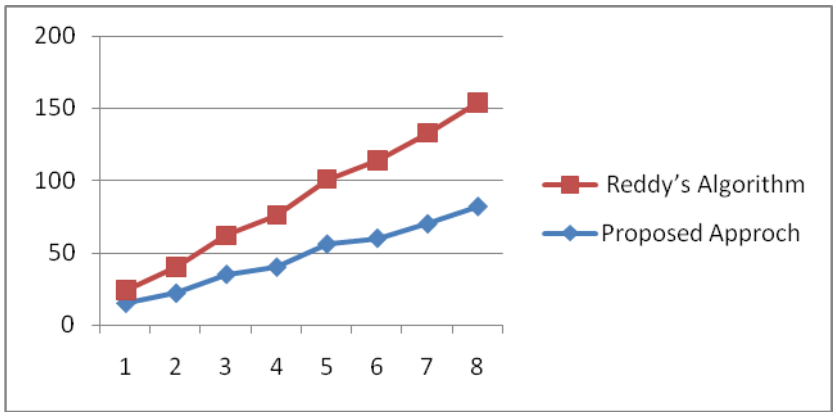


Figure 4: Number of Messages in Both Algorithm in failure situation.

6. CONCLUSION

We have presented in this paper a new distributed mutual exclusion algorithm which considerably reduces the maximum waiting time of low priority requests without increasing the number of priority inversions and reduce the messages complexity. Such a behavior is due to the "awareness" approach of the algorithm which gives more flexibility for priority increments. Furthermore, level functions allow to better configure the threshold between the maximum waiting time and the number of priority inversions. We have also observed that, thanks to the distance mechanism, the algorithm message complexity does not degrade.

REFRENCES

1. I. Suzuki, T. Kasmi, A distributed mutual exclusion algorithm, ACM Transactions on Computer Systems l3 (4) (1985) 344 – 349.
2. M. Maekawa, "A√N Algorithm for Mutual Exclusion in Decentralized Systems", ACM Trans. Computer Systems, vol. 3, no. 2, pp. 145-159, May 1985.
3. G.Ricart and A. K. Agrawala, "An Optimal Algorithm for Mutual Exclusion in Computer Networks," Communications of the ACM, pp 9-11, 1981.
4. M. Singhal, "A Class of Deadlock-Free Maekawa-Type Algorithms for Mutual Exclusion in Distributed Systems" Distributed Computing, vol. 4, no. 3, pp.131-138, 1991.
5. Weigang Wu, Jiannong Cao, Jin Yang, "A Scalable Mutual Exclusion Algorithm for Mobile Ad Hoc Networks," Proc. of the 14th International Conference on Computer Communications and Networks (ICCCN2005), San Diego, USA, Oct. 17-19, 2005
6. Sahel Alouneh, Sa'ed Abed "Fault Tolerance and Security Issues in MPLS Networks" pub. ISSN: 1792-4863 ISBN: 978-960-474- 231-8.
7. Moushumi Sharmin, Shameem Ahmed, and Sheikh I. Ahamed " SAFE-RD (Secure, Adaptive, Fault Tolerant, and Efficient Resource Discovery) in Pervasive Computing Environments".
8. M. Al-Kuwaiti, N. Kyriakopoulos "A Comparative Analysis of Network Dependability,Fault-tolerance, Reliability, Security, and Survivability" pub.in IEEE communication surveys & tutorials, VOL. 11, NO. 2, SECOND QUARTER 2009.
9. Ban Hardekopf,Kevin Kwiat"Secure and Fault –Tolerant Voting in Distributed Systems" published by 0-7803-6599-2/01/$10.00 _ c 2001 IEEE.
10. Jitendra Kumar Rout, Sourav Kumar Bhoi "SFTP: A Secure and Fault-Tolerant Paradigm against Blackhole Attack in MANET" pub.in International Journal of Computer Applications (0975 – 8887) Volume 64– No.4, February 2013.
11. Steven E. Czerwinski, Ben Y. Zhao "An Architecture for a Secure Service Discovery Service" pub. In Mobicom '99 Seattle Washington USA Copyright ACM 1999 I-58113-142-9/99/08.
12. Yaron Minsky, Robbat Van Revesse "cryptographic support for fault tolerant distributing computing".
13. Stephan Bohacek, Jo.ao Hespanha" Game Theoretic Stochastic Routing for Fault Tolerance and Security in Computer Networks".

14. Bharat B. Madan, Katerina Goševa-Popstojanova "A method for modeling and quantifying the security attributes of intrusion tolerant systems" pub.in 0166-5316/$ – see front matter © 2003 Published by Elsevier B.V.
15. Kelvin J.Rowett "Method and apparatus for fault tolerance connection of a computing system to local area network", published by US005448723A Pub.Date: Sep. 5, 1995.
16. David Tipper, Teresa Dahlberg "Providing Fault Tolerance in Wireless Access Networks" pub.in IEEE Communications Magazine • January 2002.