A Novel Token based DME Algorithm in MANET

Mandeep Kaur
M.Tech Student
Lovely Professional University
Phagwara, Punjab 144411

Shabnam Sharma
Assistant Professor and
Research Scholar
Lovely Professional University
Phagwara, Punjab 144411

Aditya Prakash
Lecturer
Lovely Professional University
Phagwara, Punjab 144411

ABSTRACT
Mutual exclusion among the nodes waiting for critical resources is considered as one of the major area of research in MANET. Mutual Exclusion allows mobile nodes to share resources among them. Formation of quorum is required for delivery of data with common intermediate node in between them. While communication, data transmission between quorums, is carried out using an arbitrator that is common to both regions. The main function of arbitrator is to grant the permission to incoming requests so as to enter the CS, by forwarding incoming requests to node, that is having the the primary token, which in turn will reduce the response time, synchronization delay and message complexity.

General Term
Mutual exclusion

Keywords
Distributed system, critical section, and mutual exclusion

1. INTRODUCTION
Mutual exclusion refers to the requirement of ensuring that no two concurrent nodes are in the critical section at the same time. Concurrent nodes must be properly synchronized to access the shared resources. If more than one node enters CS,it will lead to integrity violations.A mobile ad hoc network is an autonomous collection of mobile devices that communicate with each other over wireless links and cooperate in a distributed manner, in order to provide the necessary network functionality in the absence of a fixed infrastructure. A Critical Section is the part of a program that accesses shared resources. In case, more than one node wishes to enter CS for accessing the critical resources simultaneously, it leads to distributed mutual exclusion problem (DME).

Solutions to DME problem can be classified into two groups, based on the criteria of selection of node to enter in CS, are described below:

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

In Permission based algorithm, the node wishes to enter the CS must gather the permission from all other participating nodes. In later algorithm, nodes which owns token can enter the CS and further token is passed to other nodes in the network. In the proposed algorithm, two types of tokens i.e. Primary Token and Secondary Token are used.

2. LITERATURE REVIEW
In the algorithm proposed by Lamport [4], when a node wants to enter its CS, it sends a request to all other nodes and waits for reply messages. When it exits its CS, it sends a release message to all other nodes. This algorithm requires 3*(N-1) messages per critical section entry.Singhal, et al [6] suggested that a quorum needs not to consult other quorums that are not currently contending for CS. To further reduce the message complexity, look-ahead technique was introduced. Dynamic Information Sets, comprising of Info set and Status set, to keep track of quorums that are currently involved in CS. Ricart and Agrawala [1] proposed a distributed algorithm which requires 2*(N-1) messages per critical section entry. When a node wants to enter the CS, it sends a request message to all the nodes of the network and waits for response. If it receives an agreement of all these nodes, it enters the CS. Response Time is computed using Lamport clocks.Maekawa [5] has proposed an algorithm which requires c*(√N) messages to enter the Critical Section. The algorithm uses a logical structure which is defined by a set of nodes associated with each node and this set has a non-null intersection with every set associated to each node. This structure allows each node, which to access its CS, to have ermission only from each member of the set associated to it.

Singhal [7] improved the performance of the Suzuki and Kazami algorithm, to at most N messages in heavy loads. In this method, heuristic method is used to guess what nodes of the system are probably holding or are likely to have the token, So token request message is sent only to those nodes rather than to all the nodes. To achieve this, the knowledge of each node about the requesting nodes is passed through the token. Suzuki and Kazami [2] proposed an algorithm in which the queue of requesting nodes is piggybacked within the token and the queue is updated by a local queue of each visited node in an ascending node number. Raymond [3] proposed an algorithm, based on a logical tree on the network rooted by the token holder node. Where tree is maintained by the logical pointers distributed over the nodes and directed to the node that is holding the token. When a node wants to access its CS, it enqueues its identity and sends a request message to the next node in the direction of the token holder. The token is sent back over the reverse path to the requesting node. The direction of the link of the token sending nodes must be reversed so to point always the token holder. Chang, Singhal and Liu [8] proposed an algorithm that improves Raymond's algorithm, which tolerates link and node failures by maintaining multiple paths to search the token. The algorithm also tries to avoid cycles when the token returns to the requester along the reversal links.

3. PROPOSED METHODOLOGY
In the proposed algorithm, token based approach is implemented at quorum level, algorithm consists of two classes of tokens: Primary token and secondary token. The primary token is maintained as a unique identifier in the network and is circulated between two quorums through arbitrator. A secondary token is generated by the node which owns primary token.

3.1 The Principle of Proposed Algorithm

3.1.1 Request Sending Phase

When node Ni wants to enter in CS it first sets timestamp request Ts to the current time CT and send the request for CS to the nodes which are present in info_set as well as to arbitrator and wait for reply message.

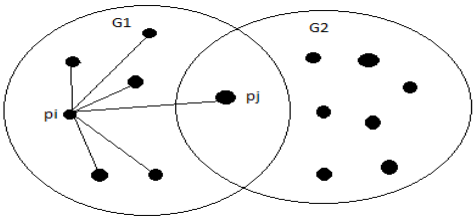

Fig 3: Requesting for token

3.1.2 Request Receiving Phase

Upon receiving request message from node Ni, Arbitrator will send the primary token to node Ni, if it is not in CS. If any other node, except Arbitrator, is receiving request from node Ni, then it will send the reply to node Ni, if it is not in CS or have high priority, otherwise node will store, the received request in Request Queue R_q.

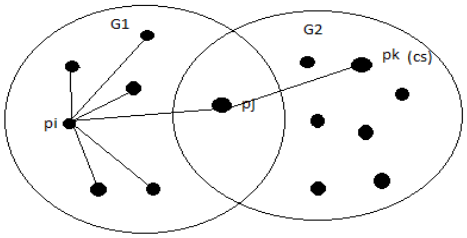

Fig 4: Forwarding request message

3.1.2.1 Each request in the queue is granted according to the following rules

If no node is present in the CS, then primary token is assigned to the requesting node.

If more than one node wishes to enter in the CS, arbitrator sends a secondary token to node Ni and places it in the Request Queue of the current node, till no further requests are received from other nodes present in either the same quorum or different quorum, having higher priority than the requested node.

If node Ni is the owner of the primary token, then it will decrement the queue length by one after exiting CS.

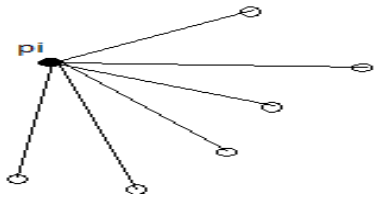

Fig 5. Releasing the CS

4. PERFORMANCE

We performed comparative evaluation of the proposed DME algorithm, i.e permission based and token based algorithms with the help of a simulator MATLAB. All the three algorithms were tested by creating networks randomly for different number of nodes. We have used the following three performance metrics for comparison. Response time, synchronization delay and message complexity. In fig.6 we are comparing delay value in micro seconds with next hop by node. Each time node will choose different hop to forward request to arbitrator. So we will check delay value in each hop taken by node to forward request. In fig 7, we are considering message complexity in both cases. It shows the comparison of number of messages transferred by each node in single hop. In fig 8, Response time is considered by each hop. We will check the response time taken by each node and check that which path takes less response time.

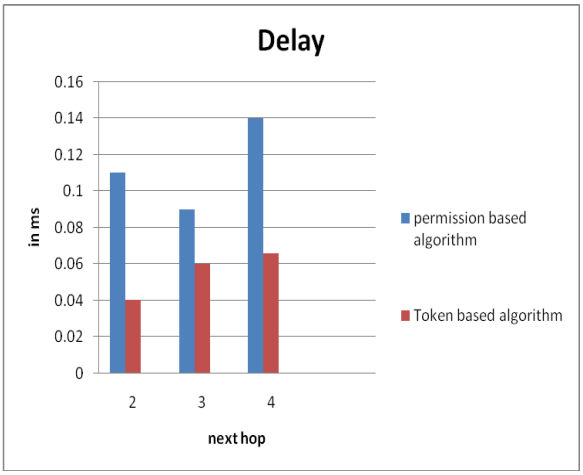

Fig 6. Delay vs next hop graph

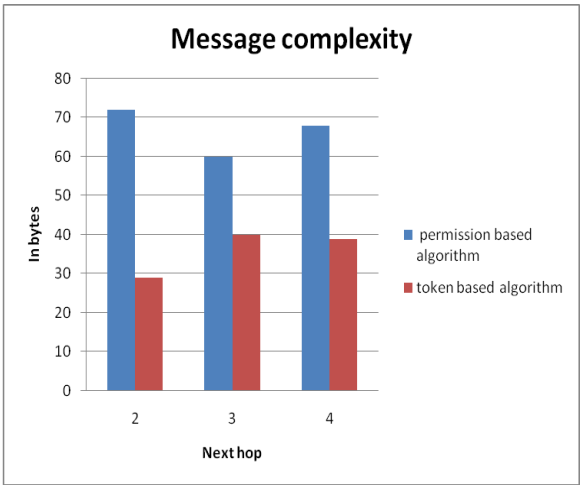

Fig 7 Message complexity vs next hop graph

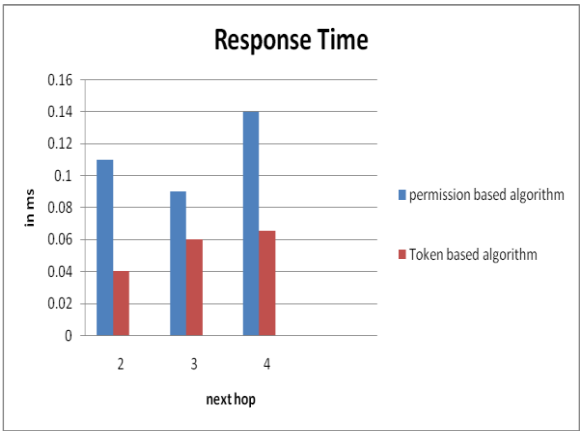

Fig 8. Response time vs next hop graph

5. CONCLUSION

Mobile Adhoc is a decentralized type of wireless network which allows to nodes to move independently. Distributred mutual exclusion allows mobile nodes to allocate resources between them. We proposed a distributed group mutual exclusion algorithm based on tokens for permissions and quorums for requesting communication. Although the proposed algorithm is fully distributed and the management of pending requests is centralized at the holder of the primary tokens The proposed algorithm achieves high concurrency, low response time and synchronization delay, reduce message complexity as compared to permission based algorithms. By concerning future scope, we will work on queues that on what basis the queue will grant the permission to incoming requests rather than time stamps.

6. REFERENCES

[1] A. Derhab and N. Badache, "A distributed mutual exclusion algorithm over multi-routing protocol for mobile ad hoc networks," International Journal of Parallel, Emergent and Distributed Systems, Vol 23 no 3, June 2008, pp 197-218

[2] B. A. Sanders. "The Information Structure of Distributed Mutual Exclusion Algorithms".ACM Transactions on Computer Systems, 5(3):284-299, August 1987.

[3] G. Ricart and A. K. Agrawala, "An optimal algorithm for mutual exclusion in computer networks," Communication ACM, vol. 24, no. 1, pp. 9–17, Jan. 1981. [Online]. Available: http://doi.acm.org/10.1145/358527.358537

[4] Ichiro Suzuki and Tadao Kasami, "A distributed mutual exclusion algorithm," ACM Transactions on Computer Systems, vol. 3, no. 4, pp. 344–349, Nov. 1985

[5] K. Raymond, "A tree-based algorithm for distributed mutual exclusion," ACM Trans. Comput. Syst., vol. 7, no. 1, pp. 61–77, Jan. 1989. [Online]. Available: http://doi.acm.org/10.1145/58564.59295

[6] L. Lamport, "Time, clocks, and the ordering of events in a distributed system," Communications of the ACM, Vol. 21, N°7, July 1978, pp. 558-565.

[7] M. Maekawa, "A n algorithm for mutual exclusion in decentralized systems," ACM Trans. Comput. Syst., vol. 3, no. 2, pp. 145–159, May1985.[Online].

[8] M. Singhal and D. Manivannan, "A distributed mutual exclusion algorithm for mobile computing environments," in Intelligent Information Systems, 1997.IIS '97.Proceedings, 8-10 1997, pp. 557 –561.

[9] M. Singhal, "A heuristically-aided algorithm for mutual exclusion in distributed systems," IEEE Trans. On Computers Vol. 38 5, may 1989, pp. 651-662

[10] Y. Chang, M. Singhal, and M. Liu, " A fault tolerant algorithm for distributed mutual exclusion," In Proc. of 9th IEEE Symp. On Reliable Dist. Systems, pp. 146-154, 1990.

[11] Murali Parameswaran, Chittaranjan Hota, "A Novel Permission-based Reliable Distributed Mutual Exclusion Algorithm for MANETs", 978-1-4244-7202-4/10/$26.00 ©2010 IEEE.

[12] W. Wu, J. Cao, and J. Yang, "A fault tolerant mutual exclusion algorithm for mobile ad hoc networks," Pervasive and Mobile Computing, Vol. 4, No 1, February 2008, pp 139-160, doi:10.1016/j.pmcj.2007.08.001.