

Summary of Thesis

FTRRP-A Fault Tolerant Mutual Exclusion Resource Reservation Protocol for Clustered Mobile Ad hoc Networks

Mohammad Moallemi

Department of Computer Science and Engineering
Faculty of Engineering, Ferdowsi University of Mashhad
Mashhad, Iran
mo_mo16@stu-mail.um.ac.ir

1. Introduction

A mobile ad-hoc wireless network (MANET) is a collection of mobile nodes that communicate over paths composed of one or a sequence of wireless links. A wireless link is established only if two nodes are within a certain transmission radius. Moreover, the nodes mobility pattern creates unpredictable wireless links formation and removal, as a consequence a path between two nodes can change very frequently due to topology change. In [9], Badrinath et al. show that also in MANET, implementing distributed algorithms that ensures control and ordering (such as mutual exclusion) is a critical point for many specific problems (e.g. channel allocation). Since these networks presents limited device power supply, the resource consumption required for the execution of distributed algorithms should be carefully controlled. However, in such setting previous specific performance metrics (e.g. power consumption) as well as communication issues (e.g. link failures and recovery) have to be taken into account. One of the most classical paradigms of distributed computing is Distributed Mutual exclusion (DMUTEX). It consists in defining a protocol run by a set of processes which want to coordinate themselves to access a single shared resource (or a piece of code), namely critical section (CS), that can be used only by one process at a time. In the recent years, several DMUTEX algorithms specifically designed for MANETs have already been presented [6, 8, 9, 10, and 11]. These algorithms can adapt their behavior to changes in the physical topology of the network to reduce communication overhead in a mobile environment.

The distributiveness of mobile ad hoc networks makes resource allocation strategies very challenging since there is no central node to coordinate and monitor the activities of all the nodes in the network. Since a single node cannot be delegated to act as a centralized authority due to limitations in the transmission range, several delegated nodes may coordinate the activities in certain zones. This methodology is generally referred to as clustering and the nodes are called clusterheads. The clusterheads employ centralized algorithms in its cluster; however, the clusterheads themselves are distributive in nature. In this thesis we use Entropy based clustering technique [5].

In this thesis a fault tolerant mutual exclusion resource reservation protocol for mobile ad-hoc networks is proposed. Fault tolerance is a key feature for any algorithm in ad-hoc networks, since the ratio of failure in these networks is relatively high in contrast with wired networks. Some main causes of failure in ad-hoc networks are: abundant movement of nodes and small range of radio transmission, use of battery, any hardware or software crash, high ratio of fault in wireless connections, nature of military usage –since military usage is one of the fundamental usage of ad-hoc networks. Thus every algorithm in ad-hoc networks must support fault tolerance to be accepted and applicable in these networks. Yet, no complete fault tolerant algorithm in ad-hoc networks has been introduced.

2. Relation to Previous works

A robust mutual exclusion algorithm that has been proposed in distributed systems is Naimi-Trehels algorithm [3] which is a token-based algorithm. It keeps a logical dynamic tree of all nodes, that the root of the tree is the last node among all requester nodes which receives the token.

Every node A has the following variables:

- The *last* variable, which represents the probable owner of the token.
- The *next* variable, which points to the next node to whom the token will be granted after A leaves the critical section.
- The Boolean *token* variable, whose value is true if the node A owns the token, or false otherwise.
- The Boolean *requesting* variable, whose value is true if the node A requests the token, or false otherwise.

During the initialization process one node is elected (*Elected_Node*) which generates the token for mutual exclusion.

3. Proposed Algorithm

This algorithm is based on Naimi-Trehel's algorithm and considers the delay and distance between nodes, to place them on different clusters. It reduces the intra-cluster messages and gives a higher priority to local nodes in a cluster, for entering the critical section. The idea of distributing nodes in different clusters is given from [2].

This algorithm improves Naimi-Trehel's algorithm based on aggregation of messages and preemption of the token by local nodes. This method uses the following variables besides the already named variables of a node in Naimi-Trehel's algorithm:

- Each node has a *local_cluster* variable which determines, node *i* belongs to which cluster.
- Each cluster C_i has a Clusterhead node, except the cluster which has the *Elected_node* and the token.
- There is a repeated *R_Queue* variable at all the nodes of the cluster which has the token. This queue holds the requests of remote clusters.
- There is an *nb_preempt* variable in the system configuration which specifies the number of local requests that would be served first.
- There is a *token_pos* variable at each node, which points to the cluster in which the token exists. Each time the token is transferred from one cluster to another, the owner of the token broadcasts a *TOKEN_POS_CHANGE* message to all nodes of all clusters, and then they will update their *token_pos* variables. It also broadcasts a copy of *R_Queue* in the token receiver cluster.

At first each node sets its *last* variable to the clusterhead of its cluster, and like Naimi-Trehel's, sends its request to the clusterhead. Then the clusterhead sends the request to the *Elected_node* in the remote cluster and changes its *last* to the requester.

Such as Naimi-Trehel's, each request follows the *last*'s path until it reaches its *local_root* (the node of the same cluster whose *last* variable is set to \emptyset).

The principle of proposed algorithm is based on constructing the *N_Queue* (the next variables chain queue) and protecting the *R_Queue* which is remained in that cluster after the failure. The goal of reconstruction is to maintain the order of token requests and avoid the retransmitting of the requests. On the other hand, if reconstruction is impossible, a new *N_Queue* based on *last_tree* is built. However, the assumption is that in the cluster holding the token, at least one live node exists so the *R_Queue* will survive after failure.

Every time a node S_i sends a token request, node S_j which is the *local_root* or clusterhead in the local cluster, sends a COMMIT message for S_i . By this, S_j tells S_i its position in *N_Queue* and its predecessors. This way, *N_Queue* is ordered so that the lowest position is for the node which owns the token (if the token exists in this cluster), or the node in the cluster which first of all will receive the token (if the token does not exist in this cluster). The COMMIT message contains the following sets of information:

- The closest predecessor of S_i in its local cluster.
- S_i 's position in *N_Queue*, which is S_i 's predecessor's position plus one.

After receiving COMMIT message, S_i periodically checks its closest predecessor's aliveness.

Now we are able to cover M faults in the clusters which do not own the token and $M-1$ faults in the cluster which owns the token. Generally, this algorithm can cover $N-1$ faults in the whole system. (M is the number of nodes in a cluster and N is the number of all the nodes in all clusters).

Every cluster has its own *N_Queue*. Position zero of every *N_Queue* is assigned to the first requesting node (by the local clusterhead).

4. References

Below some of the main resources used in this thesis are listed.

- [1] J. Sopena, L. Arantes, M. Bertier, P. Sens. "A fault-tolerant token-based mutual exclusion algorithm using a dynamic tree". EuroPar 2005, Lisboa, Portugal, September 2005. LNCS.
- [2] M. Bertier, L. Arantes, and P. Sens. "Hierarchical token based mutual exclusion algorithms". In 4th IEEE/ACM CCGrid04, 10 April 2004.
- [3] M. Trehel and M. Naimi. A distributed algorithm for mutual exclusion based on data structures and fault tolerance. In Proc. IEEE 6th International Conference on Computers and Communications, pp, 35-39, 1987.
- [4] Mohammad Moallemi, Yasser Mansouri, Amin Rasoulifard and Mahmoud Naghibzadeh, "Fault-Tolerant Hierarchical Token-Based Mutual Exclusion Algorithm" In Proceeding of ISCIT 2006 Conference, IEEE 2006.
- [5] Kimberly Robinson, Damla Turgut, and Mainak Chatterjee, "An Entropy-based Clustering in Mobile Ad hoc Networks", 2006 IEEE.
- [6] Cheng-Zen Yang. "A Token-based h-out of-k Distributed Mutual Exclusion Algorithm for Mobile Ad Hoc Networks", 0-7803-8932-8/05/\$20.00 © 2005 IEEE.
- [7] F. Mueller. "Fault tolerance for token-based synchronization protocols. Workshop on Fault-Tolerant Parallel and Distributed Systems", IEEE, April 2001.
- [8] R. Baldoni, A. Virgillito, and R. Pehassi, "A distributed mutual exclusion algorithm for mobile ad-hoc networks," in Proceeding of the Seventh International Symposium on Computers and Communications (ISCC 2002),IEEE , July 2002, pp. 539-544.
- [9] B.R. Badrinath, A. Acharya, and T. Imielinski, "Structuring distributed algorithms for mobile hosts," in Proceeding of the 14th International Conference on Distributed Computing Systems, IEEE, June 1994, pp. 2 1-28.

- [10] M. Benchba, A. BouabdaUah, N. Badache, and Ahmed. Nacer, "Distributed mutual exclusion algorithms in mobile ad hoc networks; an overview," ACM Operating Systems Review, vol. 38, no. 3, July 2004, pp.74-89.
- [11] Weigang Wu, Jiannong Cao, Jin Yang, "A Scalable Mutual Exclusion Algorithm for Mobile Ad Hoc Networks", 0-7803-9428, 2005 IEEE.