

# Consistent Global Data Structure in MANET

K.K.Basheer  
Asst.Prof. CSE Dept.,  
VITS, Proddatur.  
Kadapa (dist), A.P. INDIA.

V.Venkata Ramana  
Asst.Prof. CSE Dept.,  
SSITS, Raychoti,  
Kadapa (dist), A.P. INDIA.

A. Rama Mohan Reddy  
Professors, Dept. Of CSE,  
SVU College of Engineering,  
Tirupati, A.P., INDIA.

## ABSTRACT

Our ultimate goal is to provide application developers with the ability to maintain a consistent global data structure in a setting in which mobile hosts come and go as they please and engage in reliable transient collaborative activities. In this context, the group membership service needs to provide an accurate snapshot of the membership view all the time, and a message entrusted in a view shall be guaranteed to be delivered to members in that view, despite motion and motion-induced disconnections. This property makes the group membership service useful for many mobile applications, such as those we mentioned earlier. Next, we seek to formally define the group membership problem.

## General Terms

MAC, Consistent, GPS.

## Index Terms

Adhoc network, Distributed systems, Asynchronous, Fake system.

## 1. INTRODUCTION

Prakash and Baldoni [ PB98] present an architecture for group communication in the context of mobility. Three different types of mobile networks are considered: (1) A cellular and (2) a virtual cellular model in which base stations are mobile, too, and (3) a fully mobile ad hoc network without base stations. All models apply only to connected networks.

### 1.1 Mobile Systems

The location of nodes plays an important role. Hence, the authors propose a “proximity layer” that links

#### Mobile Host:

Initial membership view: A host always installs itself as the only member in its view when it starts  $mem(p, t_{init}) = p$ . The host with the smallest identifier in a group is chosen as the group’s leader and the host identifier (id) of the leader to serve as the name for its group

(gid).

R- transmitting range

id- Node Identifier

gid-Group Identifier

r – Group transaction seq no. The following are some of the methods briefed.

#### 1.1.1 Consistent Group Membership in Ad hoc Networks

The design of ad hoc mobile applications often requires the availability of a consistent view of the application state among the participating hosts. Such views

the group membership layer with the underlying medium access control (MAC) layer. The MAC layer of the mobile network provides point-to-point communication and beacons ‘I am alive’-messages every  $t$  time units within transmission distance  $d$ . Then, a D-proximity test tries to find all nodes within distance  $D$  from a given node  $p$ . If  $D = d$  then a node  $p$  determines its D-proximity set of nodes by just listening to the location stamped beacon messages. For  $D = d$ , this corresponds to our neighborhood service which uses heartbeat messages without the location. The group construction protocol works on top of the proximity layer. There, a process  $p$  identifies all group members in D-proximity. Again, for  $D = d$  this is analogous to our localized group membership service. The authors propose a three round protocol to solve the group construction task. First,  $p$  sends REQUEST messages to an a priori known superset  $S$  of the group members. Second, nodes receiving the request answer with acknowledge (ACK) or negative acknowledge (NACK) messages depending on their distance to  $p$ . In the third round, upon receiving all ACKs and NACKs from the set  $S$ ,  $p$  sends JOIN messages to all nodes from which it has gotten positive acknowledgments. Although the ideas of D-proximity and group construction are similar to our approach with a neighborhood and a localized group membership service, this work rather focuses on the solution of an informally stated group membership problem. Moreover, the limiting assumption of full connectivity does not hold in our model. Finally, the proposed architecture requires routing capabilities of the underlying network, which in turn is an active field of research in ad hoc networks. Routed messages in wireless ad hoc networks can be arbitrarily delayed or even lost which possibly prevents the proposed protocol to terminate.

are important because they simplify both the programming and verification tasks. Essential to constructing a consistent view is the ability to know what hosts are within proximity of each other, i.e., to form a group in support of the particular application. In this paper we propose an algorithm that allows hosts within communication range to maintain a consistent view of the group membership despite movement and frequent disconnections. The novel features of this algorithm are its reliance on location information and a conservative notion of logical connectivity that creates the illusion of announced disconnection. Movement patterns and delays are factored in the policy that determines which physical connections are susceptible to disconnection.

#### 1.1.2 Newton: A Fault-Tolerant Group Communication Protocol

A general purpose group communication protocol suite called Newton is described. It is assumed that processes can simultaneously belong to many groups, group size

could be large, and processes could be communicating over the Internet. Asynchronous communication environment is therefore assumed where message transmission times cannot be accurately estimated, and the underlying network may well get partitioned, preventing functioning processes from communicating with each other. Newton can provide causality preserving total order delivery to members of a group, ensuring that total order delivery is preserved for multi-group processes. Both symmetric and asymmetric order protocols are supported, permitting a process to use say symmetric version in one group and asymmetric version in other.

### 1.1.3 Timed Asynchronous Distributed System Model

Depending on whether the underlying communication and process management services provide “certain communication”, distributed systems can be classified as either *synchronous* or *asynchronous*. By *certain communication* we mean that 1) at any time there is a minimum number of correct processes, and 2) any message sent by a correct process to a correct destination process is received and processed at the destination within a known amount of time, i.e. the probability that is not received and processed in time is “negligible”. The authors of explain what it means for failures to be negligible. A synchronous system guarantees certain communication. All other systems are asynchronous. The *timed asynchronous distributed system model* (or, shorter the *timed model*) which we define formally in this paper assumes that 1) all services are timed: their specification prescribes not only the outputs and state transitions that should occur in response to inputs, but also the time intervals within which a client can expect these outputs and transitions to occur, 2) Inter-process communication is via an unreliable datagram service with omission/performance failure semantics: the only failures that messages can suffer are omission (message is dropped) and performance failures (message is delivered late, 3) processes have crash/performance failure semantics: the only failures a process can suffer are crash and performance failures, 4) processes have access to hardware clocks that proceed within a linear envelope of real-time, and 5) no bound exists on the frequency of communication and process failures that can occur in a system.

#### 1.1.3.1 Group Communication Specifications

View-oriented group communication is an important and widely used building block for many distributed applications. Much current research has been dedicated to specifying the semantics and services of view-oriented Group Communication Systems (GCSs).

#### 1.1.3.2 View-Synchronous Group Communication

Fekete et al. [FLS97, FLS] propose a partitionable group communication service VS and an application using VS. The specification is split into safety requirements and performance and fault-tolerance requirements. The safety requirements are expressed by an abstract state machine that associates preconditions with acts. The performance and fault-tolerance requirements are a set of properties that apply to executions of the state machine. The VS specification is tailored to the application of a totally ordered broadcast.

However, the authors claim that other applications have used the proposed service as well.

#### 1.1.3.3 Group Membership in Wide Area Networks

Keidar et al. [KSMD99] study the task of group membership in the context of wide area networks. Here, the membership service does not evolve from existing services in local area networks—in contrast, it resides on dedicated servers which are not involved in the communication among the group. This approach makes the service scalable both in terms of the number of groups and in the number of members in each group. The membership service on dedicated servers uses an underlying network event notification service that handles failure detection of neighboring servers and local clients. Also, the communication between servers is reliable in the sense that a message either eventually arrives at its destination or else the notification service reports the link to be faulty. Upon receiving an event from the notification service, a membership server multicasts a “proposal” message to all other servers. This indicates that the system requires an client, scalable multicast service.

#### 1.1.3.4 Partitionable Light-Weight Groups

Rodriguez and Guo [RG00] describe a light-weight group service that is able to operate in partitionable networks. A light-weight group service maps multiple user groups onto a smaller number of instances. Then, a virtual synchronous implementation of a group membership service works on these fewer instances. Virtual synchrony ensures that all processes in the group receive consistent information about the group membership. Such algorithms do not scale well for a large number of groups. Therefore, the light-weight group service manages a pool of groups that are able to share common resources in order to enhance the performance of the virtual synchronous membership service. In the case of partitions, it is impossible to ensure the consistency of mapping decisions made in distinct parts of the network.

#### 1.1.3.5 Partitionable Systems

At the University of Bologna, the group communication paradigm has been studied and implemented for example in the Jgroup [Jgr] project. Advances in “partition-aware” group communication systems are reported in [BDMS98, BDM98, Mon00]. “Partition-aware” applications continue operating without blocking when the network fragments and reconfigure themselves when partitions merge. Babao glu et al. [BDM98] specify a partitionable group membership service that guarantees liveness and excludes trivial solutions. They give an implementation that satisfies the specification in distributed systems with a certain stability.

## 2. Problem Definition

The group communication paradigm [Bir93] embodies a prominent technique in fault-tolerant and reliable distributed computing. Groups of member processes therein interact and communicate in order to achieve a common goal. A group communication system usually integrates a group membership service with a reliable multicast service.

The task of the group membership service is to keep members consistently informed about the current membership of a group by installing views. Processes can join and leave the group or even crash—all resulting in dynamic changes of the membership. Installed views consist of a set of members and reflect the perception of the group’s membership. This requires the members to agree on the composition of a view.

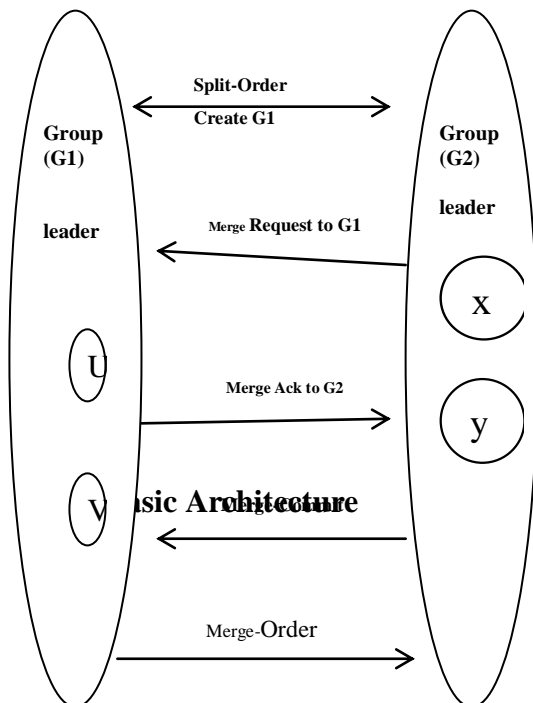
In recent years, several approaches to group communication and to building fault-tolerant toolkits have been reported, including Transis [Tra], Ensemble [Ens], Newton [EMS95], Group [Jgr], and Spread [Spr]. However, no final agreement yet exists about a general specification of a partitionable group membership service.

### 2.1 Partitionable Group Membership or Reconfiguration Protocol or Maintenance Protocol:

Each Partitionable group has one leader. Each partitionable group uses safe distance as a criteria for finding out who is close enough to be a merge candidate and they report any positive discoveries to their leader. The host with the smallest identifier in a group is chosen as the group’s leader.

### 2.2 Merging

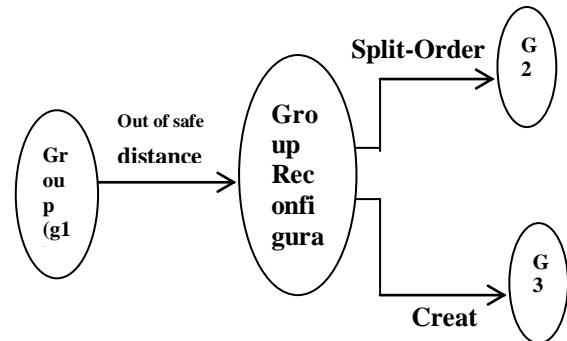
A group to merge with another group, it must first be able to discover which other groups are present in its vicinity. The discovery protocol carries out this function and serves as a supporting layer for the group membership maintenance protocol (reconfiguration protocol). In our discovery protocol, hosts in each group use safe distance.



**Fig.2: Group Merge**

By using location and mobility information about the mobile host in the region the membership service is able to guarantee to the application layer a reliable

The leader of a group frequently checks the member’s locations to make sure that only those that are guaranteed to stay connected with the group for at least  $t + t'$  more units of time remain in the group, where  $t$  is the time specified by the application layer and  $t'$  is the time bound for configuration changes.

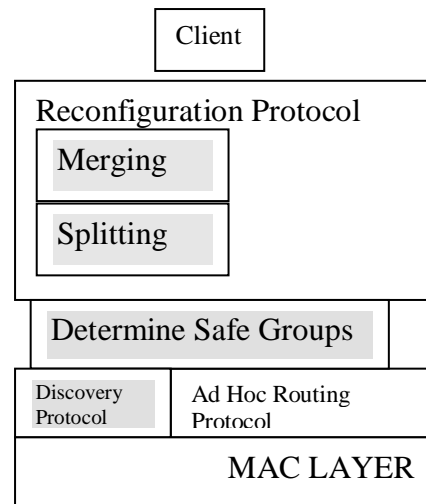


**Fig.1: Group Split**

### 3. Solution Strategy

In mobile computing systems the network configuration changes due to node mobility.

This paper identifies the issues a group communication service has to take into account in order to handle node mobility. These include the need to identify the location of a node, and the ability to cope with inaccuracies in the determination of a group membership. A multilevel architecture for group communication in mobile systems is presented.



**Fig.3: Complete picture of the System**

message delivery service to group members in the presence of mobility-induced unannounced disconnection, given appropriate system assumptions. This approach represents

A new direction in fault-tolerant distributed computing, one that factors into protocols information about mobility and space. This work also provides a practical solution to masking mobility induced unannounced disconnections in ad hoc mobile systems.



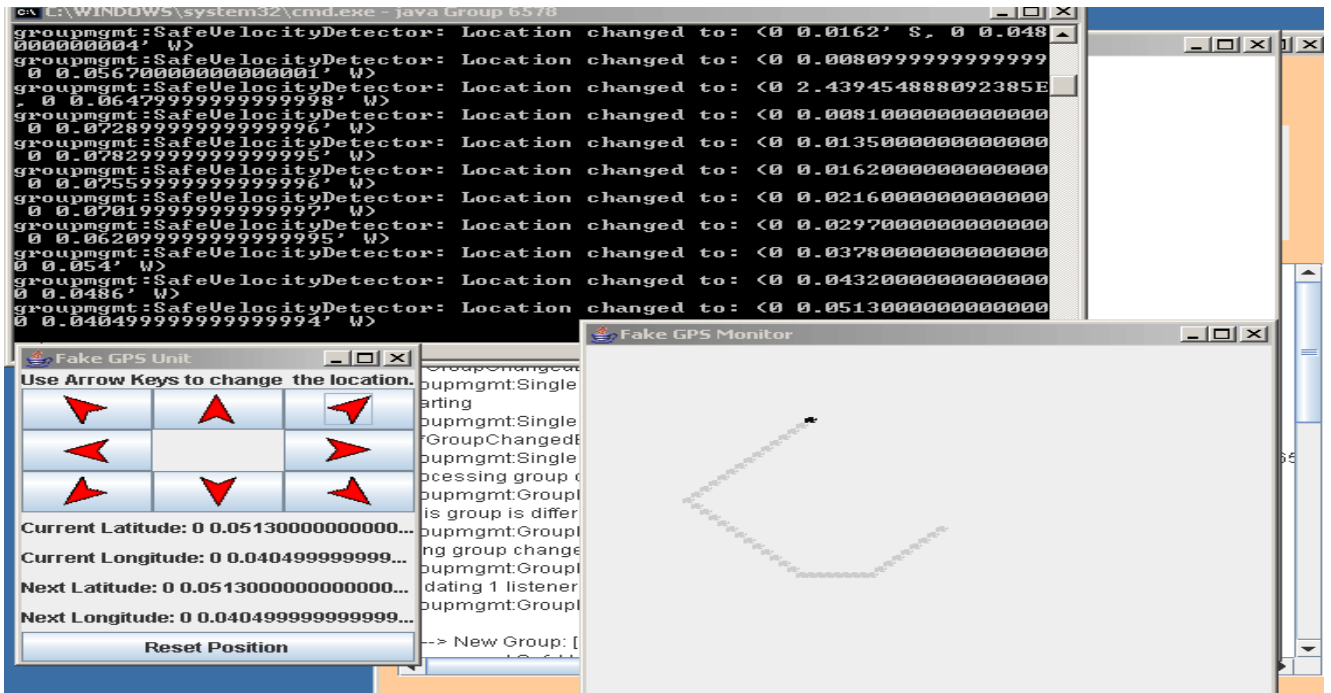


Fig.7: An Example of Fake GPS Monitor

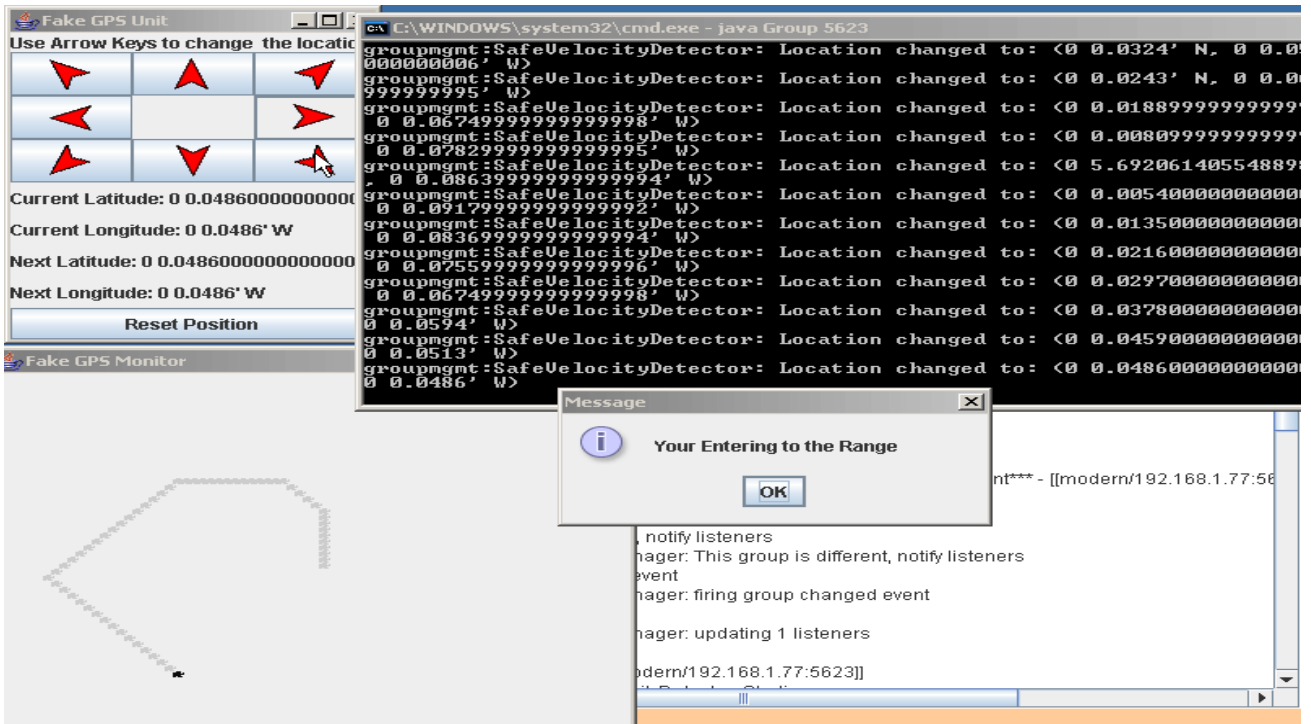


Fig.8: An example for Safe Distance in a consistent view

### Conclusion

The motivation for this work rests with our desire to provide data consistency in applications that execute over ad hoc networks. Yet maintaining a consistent view of the global state in a distributed network is difficult in general and essentially impossible in the presence of unannounced disconnections in ad hoc mobile systems, mobility-induced unannounced disconnection occurs frequently as a part of the normal operation of the network. This makes the development of fault-tolerant systems on top of ad hoc networks very challenging our goal of assisting software developers in their efforts to design and build reliable

mobile applications leads us to define a new partitionable group membership service with strong consistency requirements we have also presented a strategy and an algorithm to implement the service given appropriate system assumptions the novel feature of the algorithm is its ability to create the illusion of announced disconnection.

### **Future Enhancement**

The future enhancement will be considering factors about the various atmosphere factors such as humidity, temperature etc. It should not act according to the various internal factors such as network traffic, congestion control etc. We have given a detailed service with strong consistency requirements.

We have also presented a strategy and an algorithm to implement the service, given appropriate system assumptions. The novel feature of the algorithm is its ability to create the illusion of announced disconnection. By using location and mobility information about the mobile hosts in the region, the membership service is able to guarantee to the application layer a reliable message delivery service to group members in the presence of mobility-induced unannounced disconnection, given appropriate system assumptions. This approach represents a new direction in fault-tolerant distributed computing, one that factors into protocols information about mobility and space. This work also provides a practical solution to masking mobility induced unannounced disconnections in ad hoc mobile systems. The main work behind this concept is that the Hand-Off problem will be fully reduced so that the message sent will be delivered to the destination with full reliability. The frequent disconnection of the mobile systems will be reduced

### **REFERENCES**

[1] F. Christian, "Relying on safe distance to achieve strong partitionable group membership specifications in AdHoc networks," *Distributed Computing*, vol. 4, no. 4, pp. 175-188, 1991.

[2] F. Christian, "Reaching agreement on processor-group membership in synchronous distributed systems," *Distributed Computing*, vol. 4, no. 4, pp. 175-188, 1991

[3] Y. Amir, D. Dolev, S. Kramer, and D. Malki, "Transis: A communication subsystem for high availability," in *FTCS-22: 22nd International Symposium on Fault Tolerant Computing*, Boston, Massachusetts, 1992, pp.76-84, IEEE Computer Society Press

[4] K. P. Birman, "The process group approach to reliable distributed computing," *Communications of the ACM*, vol. 36, no. 12, pp. 37-53, December 1993.

[5] A. Ricciardi and K. Birman, "Process membership in asynchronous environments," Tech. Rep. 93-1328, Cornell University, Department of Computer Science, February 1993.

[6] Y. Amir, L. E. Moser, P. M. Melliar-Smith, D. A. Agarwal, and P. Ciarfella, "The Totem single-ring ordering and membership protocol," *ACM Transactions on Computer Systems*, vol. 13, no. 4, pp. 311-342, 1995

[7] P. D. Ezhilchelvan, R. A. Macedo, and S. K. Shrivastava, "Newtop: A fault-tolerant group communication protocol," in *Proceedings of the International Conference on Distributed Computing Systems*, 1995, pp. 296-306

[8] L. E. Moser, P. M. Melliar-Smith, D. A. Agarwal, R. K. Budhia, and Colleen A. Lingley-Papadopoulos, "Totem: A fault-tolerant multicast group communication system," *Communications of the ACM*, vol. 39, no. 4, pp. 54-63, 1996

[9] R. van Renesse, K.P. Birman, and S. Ma\_eis, "Horus, a exible group communication system," *Communications of the ACM*, vol. 39, no. 4, pp. 76-83, April 1996.

[10] R. Prakash and R. Baldoni, "Architecture for Group Communication in Mobile Systems," in *Proceedings of the IEEE Symposium on Reliable Distributed Systems (SRDS)*, October 1998, pp. 235-242.

[11] O. Babaoglu, R. Davoli, and A. Montresor, "Group communication in partitionable systems: Specification and algorithms," *IEEE Transactions on Software Engineering*, vol. 27, no. 4, pp. 308-336, April 2001