

# AN EFFICIENT AND ROBUST ACTIVE INFORMATION SHARING SCHEME FOR MOBILE ADHOC NETWORKS

## A DISSERTATION

*Submitted in partial fulfillment of the requirements for the award of the degree of*

**INTEGRATED DUAL DEGREE**

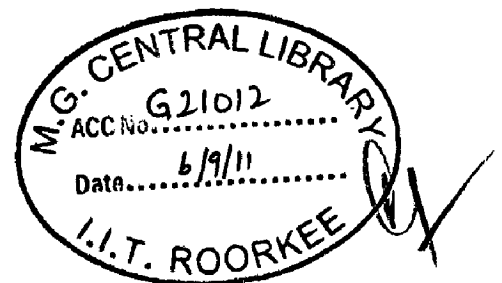
in

**COMPUTER SCIENCE AND ENGINEERING**

(With Specialization in Information Technology)

By

**NITESH YADAV**



**DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING  
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE  
ROORKEE -247 667 (INDIA)**

**JUNE, 2011**

## CANDIDATE'S DECLARATION

---

I hereby declare that the work being presented in the dissertation work entitled "An Efficient and Robust Active Information Sharing Scheme for Mobile Adhoc Networks" towards the partial fulfillment of the requirement for the award of the degree of **Integrated Dual Degree in Computer Science and Engineering (with specialization in Information Technology)** and submitted to the **Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee, India** is an authentic record of my own work carried out during the period from May, 2010 to May, 2011 under the guidance and provision of **Dr. Manoj Misra, Professor, Department of Electronics and Computer Engineering, IIT Roorkee.**

I have not submitted the matter embodied in this dissertation work for the award of any other degree and diploma.

Date: June, 2011

Place: Roorkee

  
(NITESH YADAV)

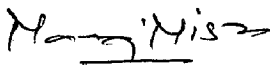
## CERTIFICATE

---

This is to certify that the declaration made by the candidate above is correct to the best of my knowledge and belief.

Date: June, 2011

Place: Roorkee

  
**Dr. Manoj Misra**  
**Professor,**  
**E&CE Department**  
**IIT Roorkee, India**

## ACKNOWLEDGEMENTS

---

I would like to take this opportunity to extend my heartfelt gratitude to my guide and mentor **Dr. Manoj Misra**, Professor, Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee, for his trust in my work, his able guidance, regular source of encouragement and assistance throughout this dissertation work. I would state that the dissertation work would not have been in the present shape without his inspirational support and I consider myself fortunate to have done my dissertation under him.

I also extend my sincere thanks to **Dr. S.N. Sinha**, Professor and Head of the Department of Electronics and Computer Engineering for providing facilities for the work.

I would like to thank all my friends who supported and encouraged me to finish this work.

Finally, I would like to say that I am indebted to my parents for everything that they have given to me. I thank them for sacrifices they made so that I could grow up in a learning environment. They have always stood by me in everything I have done, providing constant support, encouragement, and love.

(NITESH YADAV)

## ABSTRACT

---

A Mobile Ad hoc Network (MANET) is a network of mobile devices, which are connected by wireless links. Its ability of being deployable in an infrastructureless scenario creates an interest among researchers and opens the scope for large number of applications. The characteristics like random mobility and dynamic wireless links between mobile nodes poses challenges for MANET application developer.

Active Information Sharing Scheme is a scheme, which provides the capability of tactical information collaboration among the soldiers in the battlefield. The instantaneous information of battlefield has great importance for the military operations. As the information is shared by all soldiers so, it requires consistency among the updates that take place in the battlefield. This consistency should be maintained even in presence of faults. In this dissertation, a fault tolerance distributed mutual exclusion technique is proposed for MANETs. Extensive simulation is conducted, results shows that the proposed scheme is efficient and effective in various scenarios under different load conditions. Finally the thesis is concluded by pointing out some open issues and possible direction of future research relating to the distributed mutual exclusion problem in MANETs.

## LIST OF FIGURES AND TABLES

---

Figure 2.1	A Sample Shared Battlefield Map.....	8
Figure 2.2	Classification Tree of DME Alorithms .....	11
Figure 2.3	Node $N_1$ and $N_2$ each make a request for the CS. ....	12
Figure 2.4	Node $N_1$ enters the CS.....	13
Figure 2.5	Node $N_1$ exits the CS and sends the REPLY message to $N_2$ 's deferred request. ....	13
Figure 3.1	Application Scenario .....	19
Figure 3.2	System Diagram .....	19
Figure 3.3	Leadership Management Scheme States Diagram .....	22
Figure 3.4	Group Management Scheme .....	23
Figure 3.5	Flow Diagram of Position Based Grouping Algorithm.....	24
Figure 3.6	Critical Section Diagram .....	25
Figure 3.7	Timeline Demonstration of the Algorithm .....	26
Figure 4.1	Life Cycle of Packet in Protocol Stack of QualNet .....	28
Figure 4.2	Flow Diagram of Router functionality .....	30
Figure 4.3	Host State Transition Diagram.....	31
Figure 4.4	Function Calls on Events.....	33
Figure 4.5	Packet Handler Diagram.....	35
Figure 4.6	Router Handle Packets Diagram .....	35
Figure 4.7	Host Handle Packet Diagram.....	36
Figure 5.1	Synchronization Delay .....	38

Figure 5.2	Response Time .....	38
Figure 5.3	Message Complexity in Low load condition .....	39
Figure 5.4	Message Complexity in high load condition .....	40
Figure 5.5	Message Complexity at low load condition (Faulty nodes) .....	40
Figure 5.6	Message Complexity at High Load condition (Faulty nodes) .....	41
Figure 5.7	Response Time on low-load condition .....	41
Figure 5.8	Response Time on High load Condition.....	42
Figure 5.9	Response Time in low-load condition (Faulty Nodes).....	43
Figure 5.10	Response Time in High load condition (Faulty nodes) .....	43
Table 3.1	Message Format for Membership Messages.....	20
Table 3.2	Message Format for Lock Management Messages .....	21
Table 5.1	Simulation Configuration.....	37

## LIST OF ACRONYMS

---

<b>Acronyms</b>	<b>Descriptions</b>
MANET	Mobile Ad hoc Networks
AISS	Active Information Sharing System
DME	Distributed Mutual Exclusion
CS	Critical Section
MH	Mobile Host
MSS	Mobile Support Station
GQ	General Query

# TABLE OF CONTENT

---

---

CANDIDATE'S DECLARATION.....	i
ACKNOWLEDGEMENTS .....	ii
ABSTRACT.....	iii
LIST OF FIGURES .....	iv
LIST OF ACRONYMS.....	vi
TABLE OF CONTENTS .....	vii
<b>CHAPTER 1: INTRODUCTION.....</b>	<b>1</b>
1.1 Mobile Ad hoc Networks .....	1
1.2 Sharing Active Information.....	1
1.3 Motivation .....	2
1.4 Problem Statement .....	3
1.4.1 Problem Description .....	3
1.5 Dissertation's structure.....	4
<b>CHAPTER 2: LITERATURE REVIEW .....</b>	<b>5</b>
2.1 MANET Characteristics.....	5
2.2 MANETs Application .....	6
2.3 AISS Architecture.....	8
2.4 Distributed Mutual Exclusion.....	9
2.4.1 Requirements of Distributed Mutual Exclusion Algorithms.....	10



2.4.2	Distributed Mutual Exclusion Algorithms .....	10
2.4.2.1	Permission Based Approach .....	11
2.4.2.2	Quorum Based Approach .....	14
2.4.2.3	Token Based Approach .....	15
2.5	Research Gaps .....	17
<b>CHAPTER 3: PROPOSED SCHEME .....</b>		<b>18</b>
3.1	System Model .....	18
3.2	System Assumptions .....	20
3.3	Message Structure Used .....	20
3.4	System Architecture .....	22
3.4.1	Leadership Management Scheme .....	22
3.4.2	Group Management Scheme .....	23
3.4.3	Position based Grouping .....	24
3.4.4	Update Manager .....	25
<b>CHAPTER 4: IMPLEMENTATION DETAILS .....</b>		<b>27</b>
4.1	Modelling Protocol in QualNet .....	27
4.2	Network Model .....	28
4.3	Design and Development of Basic Simulation System .....	29
4.3.1	Router Functionality .....	29
4.3.2	Host Functionality .....	31
4.3.3	Event Handlers descriptions .....	33

4.3.4	Messages Types .....	33
4.3.5	Timer Handler Functions .....	34
4.3.6	Packet Handler Functions.....	35
<b>CHAPTER 5: RESULTS .....</b>		<b>37</b>
5.1	Simulation Setup.....	37
5.2	Simulation Analysis .....	37
5.2.1	Performance Parameters .....	38
5.3	Performance Evaluation .....	39
5.3.1	Message Complexity when Nodes are faultless .....	39
5.3.2	Message Complexity when Nodes are faulty .....	40
5.3.3	Response Time when Nodes are faultless.....	41
5.3.4	Response Time when Nodes are faulty.....	42
<b>CHAPTER 6: CONCLUSION .....</b>		<b>44</b>
6.1	Conclusion.....	44
6.2	Suggestions for future work .....	44
<b>REFERENCES .....</b>		<b>45</b>

# CHAPTER 1

---

## INTRODUCTION

---

### 1.1 Mobile Ad hoc Networks

Wireless communication has grown fast in the past few years and with the current growth rate, it can be easily estimated that future information technology will be mainly based on wireless technology. Traditional cellular and mobile networks are still, in some sense, limited by their need for infrastructure. This limitation of traditional cellular and mobile networks is eliminated by Mobile Ad hoc Networks (MANETs). MANET is a network of mobile devices which are connected by wireless links and hence they are infrastructureless. In MANET, each device is free to move independently in any direction therefore, nodes can frequently changes their dynamic links. Each device also act as a router since it has to forward traffic unrelated to its own use. Equipping each device to continuously maintain the information required to properly route traffic is one of the primary challenge in building a MANET.

Applications such as rescue missions in situation like natural disasters, law enforcement operations, commercial and educational use of sensor networks, Personal Area Networks (PAN) are just a few possible examples of MANETs. Prime applications of MANETs include situation management and more precisely battlefield situation management since MANETs provide necessary characteristics for these applications.

MANETs consist of dynamic collection of nodes with rapidly changing topologies of wireless links. They have the problems of bandwidth optimization, transmission quality, discovery, ad hoc addressing, self-routing and power control. Power control is a very important issue in MANETs because nodes are powered by batteries only. Therefore, amount of communication should be minimized to avoid a premature drop out of a node from the network.

### 1.2 Sharing Active Information

In the digital age, the military effectiveness depends to a large extent on the information quality, availability, and on reactive information sharing. The characteristics of information

are of great importance for the military. Information produced by soldier in operation, has to be available at the right moment to the decision maker. This decision produces new information to be delivered to highly mobile battle groups. In a rapidly changing environment, the soldier needs constantly updated information and constantly creates new information. This new operational concept supported by a digital environment will provide the soldiers a constantly updated vision of the battlefield environment.

This application makes it possible to share a map of the battlefield between soldiers and allow them to update the tactical information of their region. The major originalities of scheme are that it works in a totally distributed environment and that it handles what we call active information. Active Information is the information that is updated by the different entities of the network.

### 1.3 Motivation

The creation of Active Information Sharing Scheme (AISS) provides significant information about the current scenario of battlefield and rescue operation. Unfortunately, deployment of such service in MANET has its own set of challenges, the most significant of which are consistency of updates and node failure. Here, integrity violations occur when concurrent processes accessing the shared common resource are not synchronized. Synchronization of shared common resource means that only one process can update this shared resource. The problem of mutual exclusion is to ensure that only one of these concurrent processes are allowed to updates the common shared resource at any given point of time. The context of distributed systems, where the processes can reside in multiple sites, the problem is named as distributed mutual exclusion (DME) [2]. Hence, DME for concurrent updates must be achieved in the sense that the tactical information, including information about the enemy, terrain, weather, local populace, and many other aspects that will affect operations must be consistent. The movements of node in MANETs are random which can cause frequent link failures. Moreover, these nodes may also go to a "SLEEP" mode, shutting down all non-essential components to save power, if the battery charge level drops below some threshold. The DME problem is aggravated by the fact that node failures and link failures happen more frequently in MANETs than static networks. Therefore, it is critical to develop a suite of elaborate and carefully designed DME to achieve consistency in the AISS deployed in MANET environment.

In the literature, for fixed networks there are two popular approaches used to solve the mutual exclusion problem, viz. Permission-based approach [3] [4] and Token-based approach [5] [6]. The DME algorithms are extensions of these classic algorithms to suit the requirements of distributed system. However, only a limited number of previously reported studies [7] [8] have tackled DME problem for MANETs. This thesis focuses on addressing DME problem in information sharing. The basic idea behind this is to deal with the issues of DME problem through permission-based approach. The goal is to resolve the cases when critical section acquired node incur a fault using multicast grouping protocol. This protocol should be capable of dealing with DME problem and at the same time it should also provide a reliable consistency of shared information.

## 1.4 Problem Statement

The aim of this dissertation work is to develop and implement an Active Information Sharing Scheme which provides efficient and robust Distributed Mutual Exclusion (DME) on shared information in Mobile Ad hoc Networks (MANETs) using multicast grouping protocol.

### 1.4.1 Problem Description

Distributed Mutual Exclusion for MANETs essentially requires Safety property (that states at any instant, only one process can execute in the critical section) [2]. There are other important properties which should be satisfied in our scheme like Liveness property, Fairness and Fault Tolerance [2]. Further, the map can be further divided into different regions which make it feasible to concurrently update the independent regions and increase the efficiency of the scheme.

Hence, the problem statement can be divided into sub-problems:

- *To divide the region based on different multicast groups where shared information can be updated independently.*
- *To provide Distributed Mutual Exclusion for every update in each individual region.*
- *To propose an efficient scheme to handle those cases of fault where a node acquires the critical section and crashes or leaves the group during modification of the shared information.*

## **1.5 Dissertation's structure**

This report comprises of six chapters including this chapter that introduces the topic and states the problem. The rest of the dissertation report is organized as follows.

Chapter 2 provides a brief description of previous work done on Distributed Mutual Exclusion in distributed systems and MANETs. The other topic includes multicast grouping protocols for MANETs, and research gaps.

Chapter 3 provides a detailed description of the proposed scheme for resolving the fault case during DME while maintaining other properties.

Chapter 4 gives the brief description of the implementation of the proposed scheme.

Chapter 5 discusses the performance parameters and results based on these parameters. It also provides an analysis on these results.

Chapter 6 concludes the work and gives the directions for future work.

# CHAPTER 2

---

## LITERATURE REVIEW

---

An Active Information Sharing Scheme provides a methodology for collaborating the shared information between different entities existing in the Mobile Ad hoc Network environment.

### 2.1 MANET Characteristics

Mobile Ad hoc Network is a network of mobile devices connected by wireless links. It is a self-configuring infrastructure less network where the devices are free to move about arbitrarily. These devices also act as a router for forwarding the others traffic. A MANET is an autonomous system of mobile nodes where the mobility of nodes causes frequent change in the dynamic link between devices. The system may operate in isolation, or may have gateways to and interface with a fixed network. MANETs have several silent characteristics [1]:

1) **Dynamic topologies**

The network topology may consist of both unidirectional and bidirectional links. It is typically multi-hop where in nodes are free to move arbitrarily and hence, it may change randomly and rapidly at unpredictable times.

2) **Frequently disconnected network (Intermittent connectivity)**

While the two nodes are transmitting information the link between two nodes can quickly disappear, these dynamic topology results in frequently disconnected network. The problem is further exacerbated when DME is required, in the cases where any node can be disconnected from the network. In addition to this system halt is caused if information transmitting node become unreachable. Thus, a robust grouping protocol is required for MANETs which can recognize frequent disconnectivity and at the same time it should provide an alternative in order to achieve synchronization of shared information.

3) **Bandwidth-constrained, variable capacity links**

Wireless links will continue to have significantly lower capacity compared to hardwired links. Moreover, after accounting for the effects of multiple access, fading,

noise, and interference conditions, etc. as compared to radio's maximum transmission rate the realized throughput of wireless communications is often much less.

#### 4) **Energy-constrained operation**

In order to fulfil the energy requirements, some or all of the nodes in MANET may rely on batteries or other exhaustible means. In such cases, energy conservation may be the most important system design criteria for optimization towards these nodes.

#### 5) **Limited Physical Security**

Compared to fixed-cable nets, mobile wireless networks are generally more prone to physical security threats. Thus, for protocol design increased possibility of spoofing, eavesdropping and denial-of-service attacks should be taken into consideration carefully. A set of underlying assumptions and performance concerns are created for protocol design by these various characteristics of mobile wireless networks. Although, a protocol guiding the design of routing within the higher-speed, semi-static topology of the fixed Internet is available, mobile wireless networks demands for protocol design which extend beyond the available protocol for fixed internet.

## 2.2 **MANETs Application**

Adhoc networking is gaining importance with the progress in wireless communication and with the increase of portable devices. Ad hoc networking have number of wide spread applications because it can be applied anywhere, in places where there is no sufficient infrastructure or the existing infrastructure is expensive or inconvenient to use. The applications set for MANETs is diverse, ranging from small static networks that are constrained by power sources to large-scale, mobile, highly dynamic networks since ad hoc networking allows the devices to maintain connections to the network along with the scalability that is ease of adding and removing devices to and from the network. With the upcoming new environments there is large scope of generating new services for these new environments besides the legacy applications that move from traditional infrastructure environment into the ad hoc context.

#### 1) **Military Battlefield**

AISS provides efficient and robust communication in the modern digital battlefield. In primitive forms of communication the communication devices are mostly installed in mobile vehicles, tanks, trucks etc. Also soldiers could carry telecomm devices that



could talk to a wireless base station or directly to their telecom devices if they are within the radio range.

## 2) **Sensor Networks**

Sensor networks are one of the applications of MANETs. Sensor networks can be defined as a network composed of a very large number of small sensors which can be used to detect various properties like temperature, pressure, toxins, pollutions, etc. of an area. The capabilities of each sensor are very limited, and each must rely on others in order to forward data to a central computer. Its applications include forecast of earthquakes and measurement of ground humidity for agriculture.

## 3) **Automotive Applications**

Automotive networks typically consist of a hierarchy of networks. These are widely discussed currently. A hierarchical approach helps optimize applications that have multiple application requirements. For instance, cars should be enabled to talk to the road, to traffic lights, and to each other, forming ad-hoc networks of various sizes. Here, the network will provide the drivers with information about road conditions, congestions, and accident-ahead warnings etc. which will be helpful in optimization of traffic flow.

## 4) **Commercial Sector**

Emergency rescue operations take place where communications infrastructure does not exist or it is damaged and rapid deployment of a communication network is needed. For example situations like fire, flood, or earthquake. In such emergency/rescue operations for disaster relief Mobile Ad hoc Network can be very useful for relaying information from one rescue team member to another over a small handheld device.

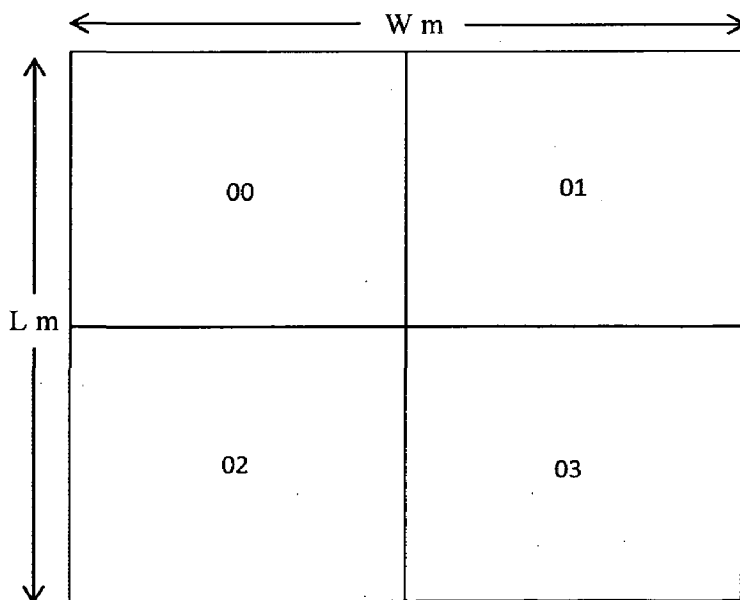
## 5) **Personal Area Network**

Personal area networks are formed in an ad-hoc manner between various mobile (and immobile) devices. It can be constructed either with cables or wirelessly. Simplest example of PAN includes interconnecting various devices, at home for creating a home network which can remain an autonomous network. But rather than limiting PAN to home, connecting it to a larger network makes it more worthy.

## 2.3 AISS Architecture

The Active Information Sharing Scheme provide the capability to mobile nodes to collaborate and shared the common resource like map between them, and still ensure consistency of updates that can be done by anyone of them. Also, provide the opportunity that several users can work at same time on independent part of document.

In a military scenario, a task can to be considered where soldiers, tanks and military vehicles need to collaborate in an area for generating the tactical information. Consider a battlefield map as shown in Figure 2.1. MANETs can be created between these entities (soldiers) acting as an individual node. Further, by splitting the map into different region which are independent of each other. We can achieve the concurrency of updates in each individual region independent of each other. Each region has its own different group address. And the nodes present in any locations join the group based on their location in the map. Nodes are enabled with GPS locator and hence, they can easy find their own location and the group they have to join.



**Figure 2.1** A Sample Shared Battlefield Map

In each region we have multicast grouping protocol, for forwarding the data to different member of the groups. It also provide the logic for creating the group leader in each group which coordinate the synchronization of updates within the group and also, provide the

Distributed mutual exclusion between the nodes present in the group. As each node contains one process per node, so process and node can be used interchangeably.

## 2.4 Distributed Mutual Exclusion

To ensure that only one member of the group updating the shared information within the group, this becomes the Distributed Mutual Exclusion problem within each group. Each update is like a Critical Section (CS) [2], it should be ensure that concurrent modification of this resource is mutually exclusive i.e. executed in serialized manner. In a distributed system like MANETs, shared variables (semaphores) or a local kernel cannot be used to implement mutual exclusion. Hence, message passing is the sole means for implementing distributed mutual exclusion. The decision of allowing access to the next CS is based on message passing, in which each process learns about the state of all other processes in some consistent way. The design of distributed mutual exclusion algorithms is complex because these algorithms have to deal with unpredictable message delays and incomplete knowledge of the system state.

There are three basic approaches [2] for implementing distributed mutual exclusion:

1. Permission-based approach.
2. Token-based approach.
3. Quorum-based approach.

In the Permission-based approach, two or more successive rounds of messages are exchanged among the nodes to determine which node will enter the CS next. A node enters the critical section (CS) when an assertion, defined on its local variables, becomes true. Mutual exclusion is enforced because the assertion becomes true only at one node at any given time.

In the token-based approach, a unique token is shared among the nodes. A node is allowed to enter its CS if it possesses the token and it continues to hold the token until the execution of the CS is over. Mutual exclusion is ensured because the token is unique. The algorithms based on this approach essentially differ in the way a node carries out the search for the token.

In the quorum-based approach, each node request permission to execute the CS from a subset of nodes (called a quorum). The quorums are formed in such a way that when two nodes

concurrently request access to the critical section, at least one node receives both the requests and this node is responsible to make sure that only one request executes the CS at given time.

### 2.4.1 Requirements of Distributed Mutual Exclusion Algorithms

A distributed mutual exclusion algorithm should satisfy the following properties [2]:

1. **Safety property:** The safety property states that at any instant, only one process can execute the critical section. This is an essential property of a mutual exclusion algorithm.
2. **Liveness property:** This property states the absence of deadlock and starvation. Two or more nodes should not endlessly wait for messages that will never arrive. In addition, a node must not wait indefinitely to execute the CS while other nodes are repeatedly executing the CS. That is, every requesting node should get an opportunity to execute the CS in finite time.
3. **Fairness:** Fairness in the context of mutual exclusion means that each process gets a fair chance to execute the CS. In mutual exclusion algorithms, the fairness property generally means that the CS execution requests are executed in order of their arrival in the system (the time is determined by a logical clock).
4. **Robustness:** The failure of a link, the failure of a node, and loss of message are the most common failures in the MANETs. The scheme is must be robust, to ensure the detection any of these failures, re-configure the system such that the computation may continue and recover when a link or site is repaired after a finite time.

The first property is absolutely necessary and the other properties are also considered important in distributed mutual exclusion algorithms.

### 2.4.2 Distributed Mutual Exclusion Algorithms

The DME Algorithms for message passing distributed systems like MANETs can be classified into three groups [2] as shown in Figure 2.2:

1. Permission-based approach.
2. Token-based approach.
3. Quorum-based approach.

Each approach has number of algorithms differ from each other in terms of the way messages are passed in the system.

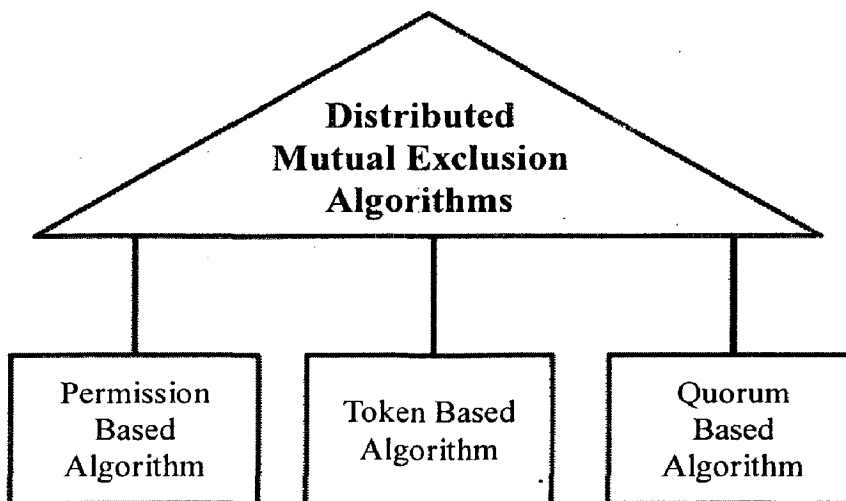


Figure 2.2 Classification Tree of DME Algorithms

#### 2.4.2.1 Permission Based Approach

The Lamport's in [3] presented a first methodology for clock synchronization in the distributed system. The algorithm is fair in the sense that the requests for CS are executed in the order of their timestamps and time is determined by logical clocks. When a site processes a request for the CS, it updates its local clock and assigns the request a timestamp. The algorithm executes CS requests in the increasing order of timestamps. The algorithm achieves mutual exclusion for distributed system and it also provides fairness. Lamport's algorithm [2] requires  $3(N-1)$  messages per CS invocation. The synchronization delay in the algorithm is  $T$ , where  $N$  denotes the number of processes or nodes involved in invoking the critical section and  $T$  denotes the average message delay.

Ricart and Agrawala improved the performance of Lamport's algorithm. The algorithm [4] [2] uses two types of messages: REQUEST and REPLY. A process sends a REQUEST message to all other processes to request their permission to enter the critical section. A process sends a REPLY message to a process to give its permission to that process. Processes use Lamport-style logical clocks to assign a timestamp to critical section requests. Timestamps are used to decide the priority of requests on basis of equation (2.1) [4] in case of conflict – if a process  $p_i$  that is waiting to execute the critical section receives a REQUEST

message from process  $p_j$ , then if the priority of  $p_j$ 's request is lower,  $p_i$  defers the REPLY to  $p_j$  and sends a REPLY message to  $p_j$  only after executing the CS for its pending request. Otherwise,  $p_i$  sends a REPLY message to  $p_j$  immediately, provided it is currently not executing the CS. Thus, if several processes are requesting execution of the CS, the highest priority request succeeds in collecting all the needed REPLY messages and gets to execute the CS.

Figures 2.3 to 5 illustrate the operation [2] of the Ricart–Agrawala algorithm. In Figure 2.3, nodes  $N_1$  and  $N_2$  are each making requests for the CS and sending out REQUEST messages to other nodes. The timestamps of node  $N_1$  and  $N_2$  requests are (1,1) and (1,2) respectively. In Figure 2.4,  $N_1$  has received REPLY messages from all other nodes and, consequently, enters the CS. In Figure 2.5,  $N_1$  exits the CS and sends a REPLY message to node  $N_2$ . In Figure 2.5, node  $N_2$  has received REPLY from all other nodes and enters the CS next.

Condition for the timestamp  $(tN_i, i)$  checking as,

$$(tN_i, i) < (tN_j, j) \text{ iff } (tN_i < tN_j) \text{ or } ((tN_i = tN_j) \ \&\& \ (i < j))$$

(2.1)

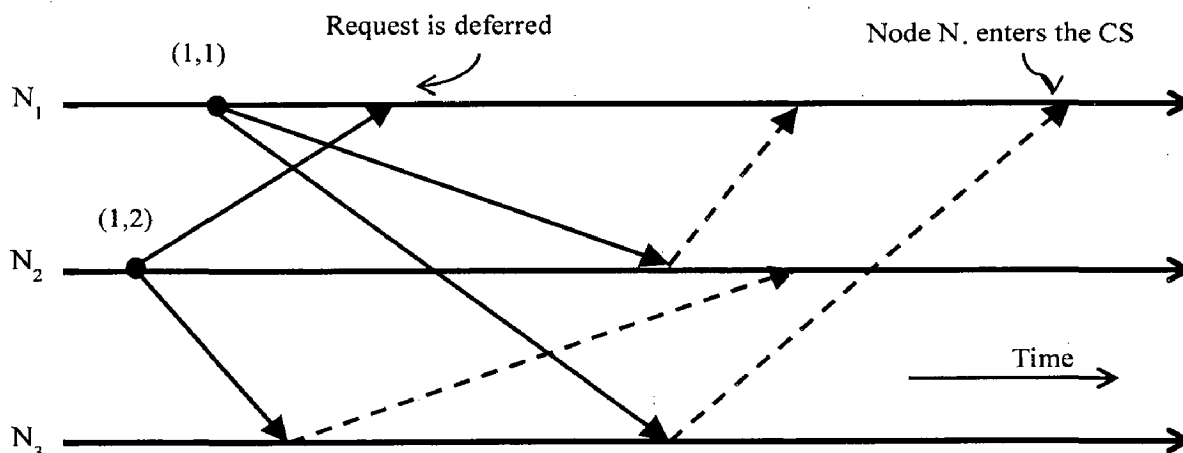


Figure 2.3 Node  $N_1$  and  $N_2$  each make a request for the CS.

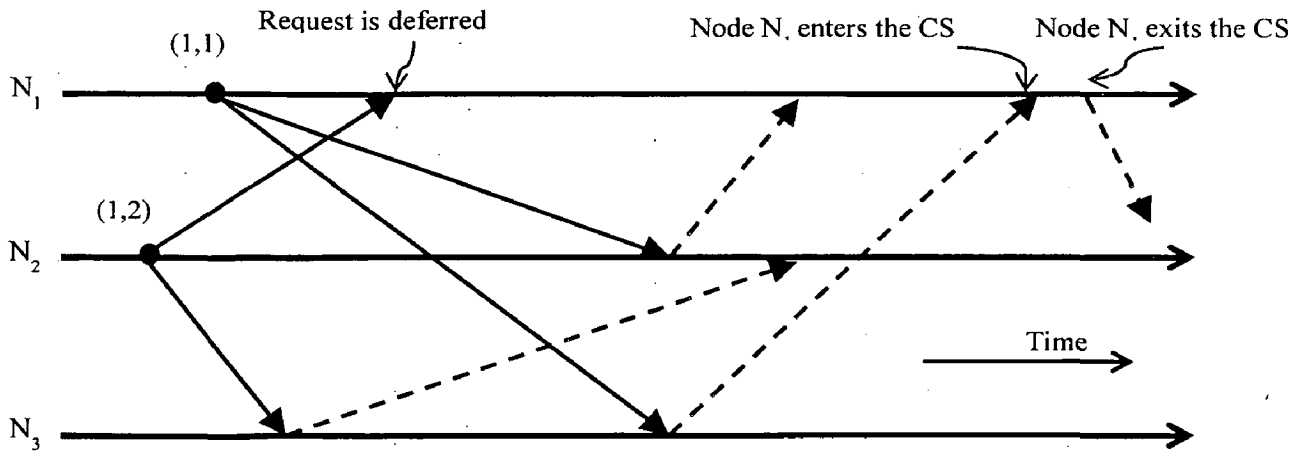


Figure 2.4 Node  $N_1$  enters the CS.

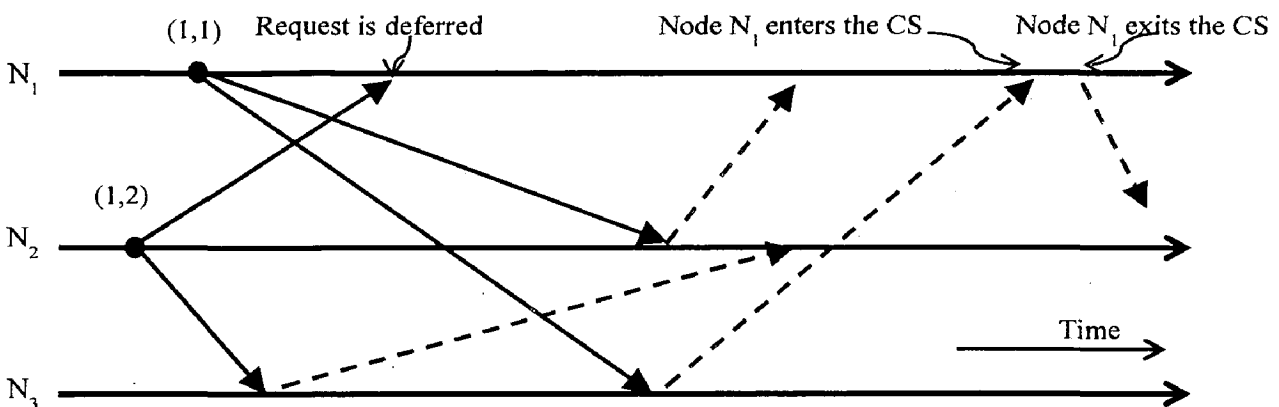


Figure 2.5 Node  $N_1$  exits the CS and sends the REPLY message to  $N_2$ 's deferred request.

In this way, Ricart-Agrawala algorithm achieves mutual algorithm and fairness. The performance is improved to  $2(N-1)$  messages per CS executions [2].

The number of variants [5] using the basic Ricart-Agrawala algorithm are proposed in recent years. In [6] Singhal and Manivannan proposed a dynamic Ricart-Agrawala Algorithm, to reduces the message cost by enforcing the DME algorithm only among the hosts that are currently competing for the CS. But this algorithm makes an assumption that a host had a prior knowledge of its disconnection. And inform his mobile support station about its own disconnection. This algorithm is developed for infrastructured mobile networks consisting of Mobile Host (MH) and powerful Mobile Support Stations (MSS).

In [7] Wu, Cao, Yang, based on the "look-ahead" techniques [6], proposed a new scheme for MANETs, each node  $S_i$  contains two sets: *Info-set<sub>i</sub>* and *Status-set<sub>i</sub>*. *Info-set<sub>i</sub>* includes the IDs of those nodes which  $S_i$  needs to inform when it requests to enter CS, and the *Status-set<sub>i</sub>*

includes the IDs of the nodes which would inform  $S_i$  when they request to enter CS. They also made an assumption when a node wants to disconnection. The introduced three new messages DOZE, DISCONNECT and RECONNECT. When a node wants to go into sleep mode to save power, it sends a DOZE message to others. And when it sees that going out of range it voluntarily send the DISONNECT message and RECONNECT for reconnection to the network.

In [8] Bouillageut, Arantes and Sens, proposed an algorithm under the condition where Responsiveness Property [8] satisfied. Their algorithm can tolerates  $f$  failures, where  $1 < f < k$  for  $k$ -mutual exclusion algorithms. Their algorithm does not rely on any timer, or any failure detectors. But their algorithm has a constraint on the network topology i.e., they considered the interconnection of the cluster such as grids and so cannot be used in MANETs.

#### 2.4.2.2 Quorum Based Approach

Quorum-based mutual exclusion algorithms [2] significantly reduce the message complexity of invoking mutual exclusion by asking permission from only a subset of nodes. These algorithms are based on the notion of “Coterie” and “Quorums”. [2] A coterie  $C$  is defined as a set of sets, where each set  $g \in C$  is called a quorum.

The following properties [2] hold for quorums in a coterie:

- **Intersection property:** For every quorum  $g, h \in C$ ,  $g \cap h = \emptyset$ . For example, sets  $\{1,2,3\}$ ,  $\{2,5,7\}$ , and  $\{5,7,9\}$ , cannot be quorums in a coterie because the first and third sets do not have a common element.
- **Minimality property:** There should be no quorums  $g, h$  in coterie  $C$  such that  $g \supseteq h$ . For example, sets  $\{1,2,3\}$  and  $\{1,3\}$  cannot be quorums in a coterie because the first set is a superset of the second.

A simple protocol works as follows [2]: let “a” is a site in quorum “A.” If “a” wants to invoke mutual exclusion, it requests permission from all sites in its quorum “A.” Every site does the same to invoke mutual exclusion. Due to the Intersection property, quorum “A” contains at least one site that is common to the quorum of every other site. These common sites send permission to only one site at any time. Thus, mutual exclusion is guaranteed. And Minimality property ensures efficiency rather than correctness. There exist a variety of quorums and a variety of ways to construct quorums.



Maekawa's algorithm [2] [9] was the first quorum-based mutual exclusion algorithm. It is based on forming the logical structure on the network. The basic idea is that a process who wants to access the critical section must obtain permissions from every member of a quorum, while a quorum node can give permission to only one process. In this scheme, a set of nodes is associated with each node, and this set has a nonempty intersection with all sets corresponding to the other nodes. Since the size of each set is  $\sqrt{n}$ , the algorithm incurs  $\sqrt{n}$  order of cost. The algorithm deadlock prune, additional messages are required to handle the deadlock. To avoid the deadlock, algorithm cost increased to  $5\sqrt{n}$  messages per critical section.

Agarwal and El Abbadi [2] [10] by introducing tree quorums, a simple and efficient mutual exclusion algorithm was developed. Using hierarchical structure of the networks they created a novel algorithm for constructing tree-structured quorums. Their mutual exclusion algorithm is independent of the underlying topology of the network. A process can enter a critical section whenever it receives permission from every process in its quorum. The quorum 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 sub tree and one root-to-leaf path on the right sub tree. This algorithm tolerates both node failures and network partitions. In the best case, this algorithm incurs logarithmic cost considering the size of the network. However, the cost increases with the increase of node failures. In MANETs, construction of quorum is not efficient and costly [7].

### 2.4.2.3 Token Based Approach

Token-based algorithms [2] achieve mutual exclusion using a token, which is shared among the nodes. A node can enter CS if and only if has possession of the token. The uniqueness of token guaranteed the mutual exclusion. Fair scheduling of token among competing nodes, detecting the loss of token and regenerating a unique token are some of the major design issues of the token-based mutual exclusion algorithms. The token based algorithms differ from each other based on the searching criteria they follow to look for the token before entering the critical section.

In Suzuki-Kasami's algorithm [2] [11], if a site that wants to enter the CS does not have the token, the site broadcasts a REQUEST message for the token to all other sites. Upon receipting the REQUEST message, a site that possesses the token sends it to the requesting

site. If a site receives a REQUEST message when it is executing the CS, it sends the token only after it has completed the execution of the CS.

Raymond's tree-based mutual exclusion algorithm [2] [12] uses a spanning tree of the computer network to reduce the number of messages exchanged per critical section execution. The algorithm exchanges only  $O(\log N)$  message under light load, and approximately four messages under heavy load to execute the CS, where  $N$  is the number of nodes in the network. A minimum spanning tree is created with the root node holding the token and all other nodes trying to access the CS. Any requesting nodes will send their requests to the root. Only the requesting node can access the CS and it retain the token until someone else requested it.

Walter & Kini in [13], proposed a token-asking algorithm, which is derived from Raymond's tree-based algorithm [2] [12] with the improvement to handle broken links caused by host mobility. Using a Direct Acyclic Graph (DAG) of token-orientated pointers, hosts maintain multiple paths leading to the token holder. Requests for CS are forwarded to the token holder along a path in the graph. The token is delivered along the reverse path to the requesting host. Host and link failures are handled by adapting the graph topology. The advantage of this algorithm is that it requires hosts to keep information only about their immediate neighbours. Though, the algorithm is designed with consideration that node mobility is slow which is not realistic. And also it does not consider token loss and network partitioning and merging. Walter, Welch & Vaidyain [14], proposed a revised version of the solution [12] which uses fewer message and improved the execution.

Baldoni, Virgillito & Petrassi, [15] proposed an algorithm based dynamic logical ring and combining the token-circulating and token-asking approaches. Their focus is to reduce the meaningless control messages when no host request to access the CS. The structure of the logical ring is computed on-the-fly, and there is a coordinator for each round. Each node that wants to enter the CS, must request to the coordinator and waits for the token. The coordinator inserts the request in a PendingRequest(P) set ordered according to a certain policy  $P$ . When the PendingRequest(P) of the current coordinator  $c_{k-1}$  becomes not empty following the reception of a request message, the first node in that set, called  $c_k$ , will become the coordinator of round  $k$ . Under low load conditions, this algorithm can greatly reduce the message exchanged. It made assumption about the absence of permanent network partitioning.

## 2.5 Research Gaps

The algorithm present in the previous section 2.4, tries to solve the distributed mutual exclusion problem in infrastructure based distributed systems. Permission based algorithms [3] [4] [5] [6] satisfy most of the requirement of the DME but they incur high message cost per critical section and have no fault tolerance. In [7] Wu, Cao, Yang, using “look-ahead” techniques provide a solution for disconnection due to mobility and power saving behaviour, but they made an assumption that a node had prior knowledge about its disconnection, which is not always true for MANETs. In [8] Bouillaguet, Arantes and Sens, provides a solution for disconnection with assumption that node has permanent disconnection or crashes so, the reconnection is not possible in contrary to MANETs.

In Token-based algorithms [11] [12], the hosts only need to keep information about their neighbour's and message overhead is low, but they have poor failure resilience [16]. Compared to infrastructure based networks the maintenance of a tree or ring topology is much more difficult in MANETs. The mobility and frequent disconnections of node make token loss a more serious problem in MANETS, and token based algorithms suffer from the fatal problem of token loss and so, they are not robust. In [17] Lubowich and Taubenfeld, favour permission-based algorithm over token-based algorithm since the redundancy of messages in permission-based algorithm can tolerate failures.

In Quorum based algorithm, when a host wants to enter critical section just sends a request only to the hosts in the quorum rather than all hosts. However, in MANETs constructing a quorum efficiently is not trivial and is costly [7]. Moreover, quorum-based algorithms are usually prone to deadlocks [18]. In order to break deadlocks, additional message cost is unavoidable.

# CHAPTER 3

---

## PROPOSED SCHEME

---

This chapter gives the complete architecture, design and working of the Active Information Sharing Scheme for implementing Distributed Mutual Exclusion algorithm in MANETs which helps in collaborating the shared information in the distributed environment.

### 3.1 System Model

We considered a scenario where a digital tactical map of battlefield is to be updated by the soldiers in same battlefield. Active Information Sharing scheme ensures that the updates done by each soldier is consistent and satisfies the all distributed mutual exclusion conditions stated in the section 2.4.1. The overall architecture of MANETs considered in this dissertation includes the Leadership Management Scheme, Group Management Scheme, Position based Grouping and mutually exclusive updates by any node which follows distributed mutual exclusion algorithm.

First, the Application partitions this digital tactical map into small equal size regions such that the network partitioning does not happen inside a region. These small regions as shown in Figure 3.1 are shared by the soldiers (act as a mobile node) present in these regions. The soldier present at any region can only update the information for their own region by creating a multicast group in that region. For updating the information, a soldier or node must have the lock of that group such that only one node is modifying this shared information. Update operation of each partition of the map is a Critical Section in this system. So, the number of locks is equal to the regions in the system. This type of architecture provides the system a capability of concurrently updating the information in different regions and still ensuring the consistency of information within its own regions. Complete system diagram is shown in Figure 3.2. Each individual component present in the system is discussed in the next sections 3.3.

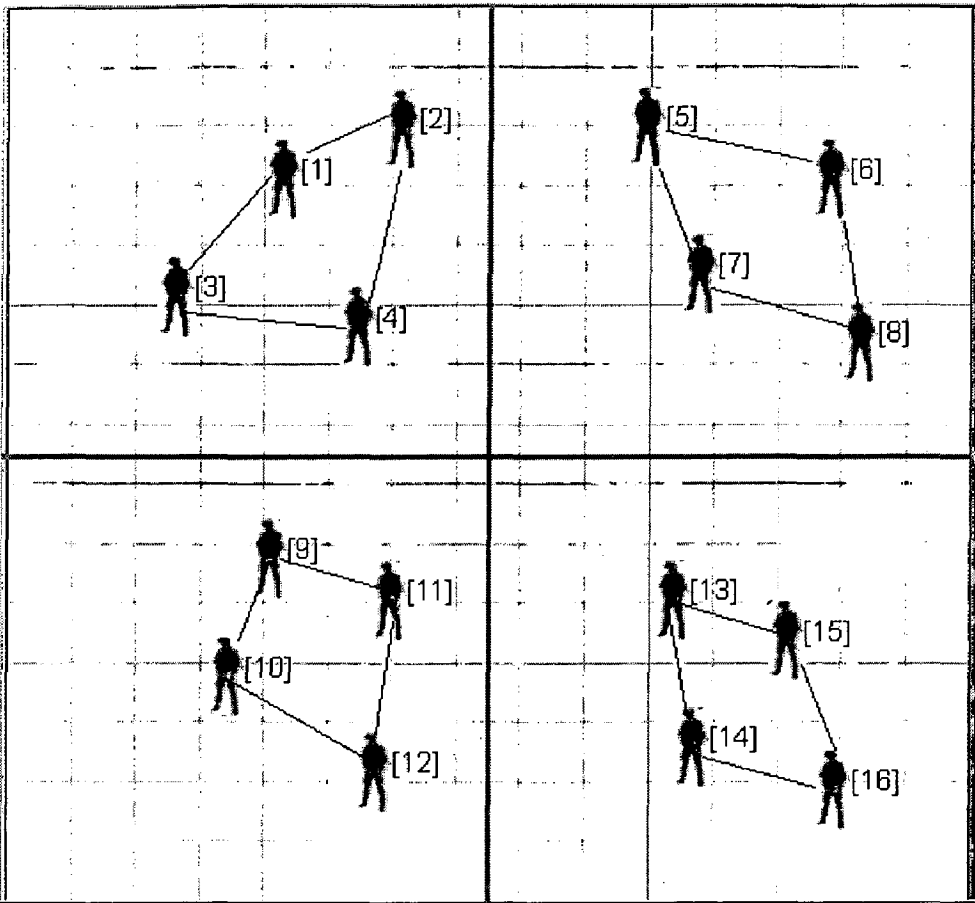


Figure 3.1 Application Scenario

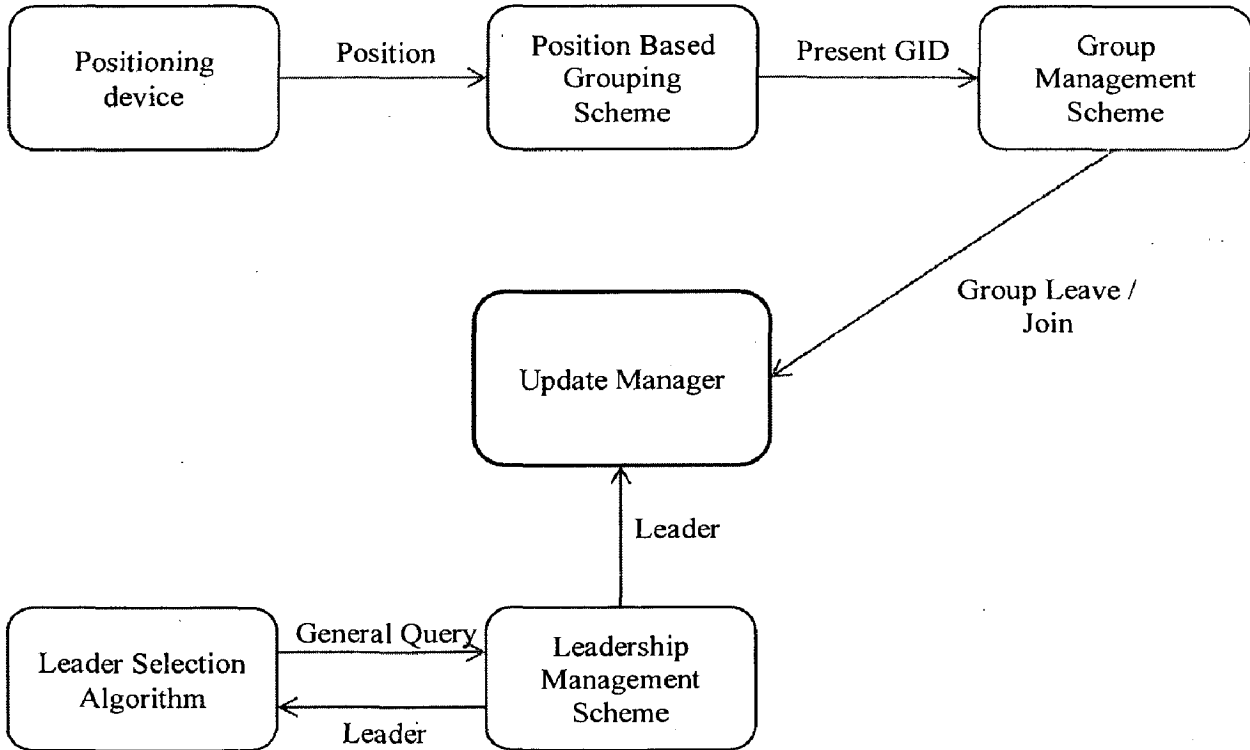


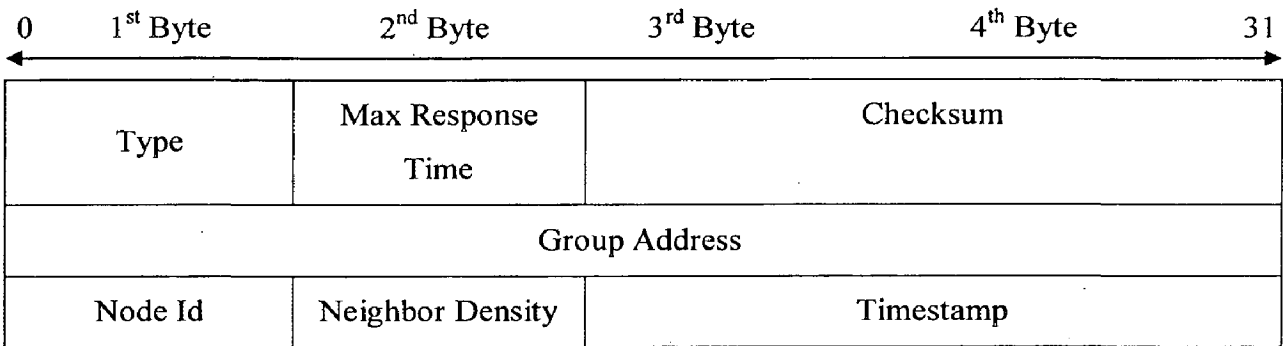
Figure 3.2 System Diagram

### 3.2 System Assumptions

- Inside each region we did not consider the network partitioning, this ensures that each working member of the group receives the data from other members. The assumption is necessary because according to “Brewer’s Theorem” [19] in an asynchronous network, it is impossible to achieve all three. Consistency, availability and partition tolerance simultaneously. So, we itself partition the whole map into different regions where only we only requires consistency and availability.
- Number of lock in the whole system = Number of regions in the system.

### 3.3 Message Structure Used

1) Message Format: For query and report messages.



**Table 3.1** Message Format for Membership Messages

**Type:** There are three types of Membership messages

0x01 = General Query used to learn for selecting the leader in the group.

0x02 = Group Specific Query used to manage the group by learning the status of the member of the group.

0x03 = Report Message in response of the general query or the group specific query to give the status of the member of the group.

**Max Response Time:** The Max Response Time field is meaningful only in Membership Query messages, and specifies the maximum allowed time before sending a responding report. In all other messages, it is set to zero by the sender and ignored by receivers.

**Checksum:** The checksum is the 16-bit one's complement of the one's complement sum of the whole message. For computing the checksum, the checksum field is set to zero. When

transmitting packet, the checksum MUST be computed and inserted into this field. When receiving packets, the checksum MUST be verified before processing a packet.

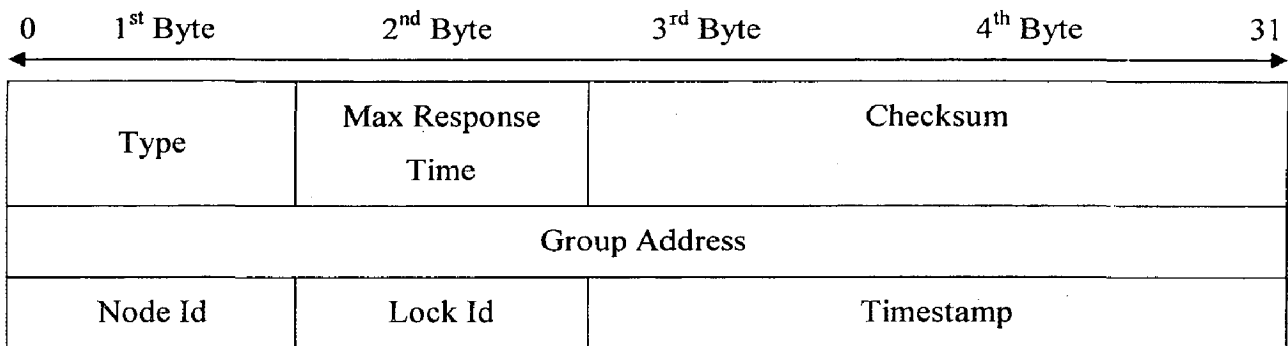
**Group Address:** The Group Address is set to the group address being queried when sending a General Query or Group-Specific Query. It is the multicast group address of the group.

**Node Id:** The Node Id is the unique identifier of each node used in General Query message.

**Neighbour Density:** Number of neighbour present in the communication range of the node. This value can be calculated by number of the HELLO packets (used by routing protocol) received by the node.

**Timestamp:** Timestamp is used to identify the duplicate packets.

2) Message Format: For lock management messages.



**Table 3.2** Message Format for Lock management Messages.

**Type:** There are four lock management messages

0x04 = Lock Query Message used to request for the lock

0x05 = Lock Grant Message used to grant lock to enter the critical section.

0x06 = Lock Release Message used to Release the lock.

0x07 = Wait message is sends by the node executing the critical section on receiving the request for lock query message from other members. It informs others members about the time remaining to complete the critical section. In this case, Max Response time is used as the time remaining to complete the critical section.

**Lock Id:** The Lock Id is used to learn about the node Id of the node who has the lock. It is zero when the leader has the lock or lock is released form.

### 3.4 System Architecture

#### 3.4.1 Leadership Management Scheme

Initially each node starts as Leader in its own group, and sends the General Query (GQ) only in that group. The nodes within the group receive these general queries from other members, they look for the lowest order node identifier group member and that member becomes the group Leader. This group leader will coordinate the distributed mutual exclusion algorithm within the group. To make the leader more efficient and stable, the neighbour density around any node is used as the additional parameter for finding the more centric leader. So, the highest neighbour density node is elected as a leader.

This algorithm provides a stable group Leader, and if any group Leader crashes or leaves the group, the algorithm will replace it with new leader from the set of presently working member of the group. So, that DME algorithm will work consistently. Figure 3.3 shows the state diagram of Leadership Management Scheme.

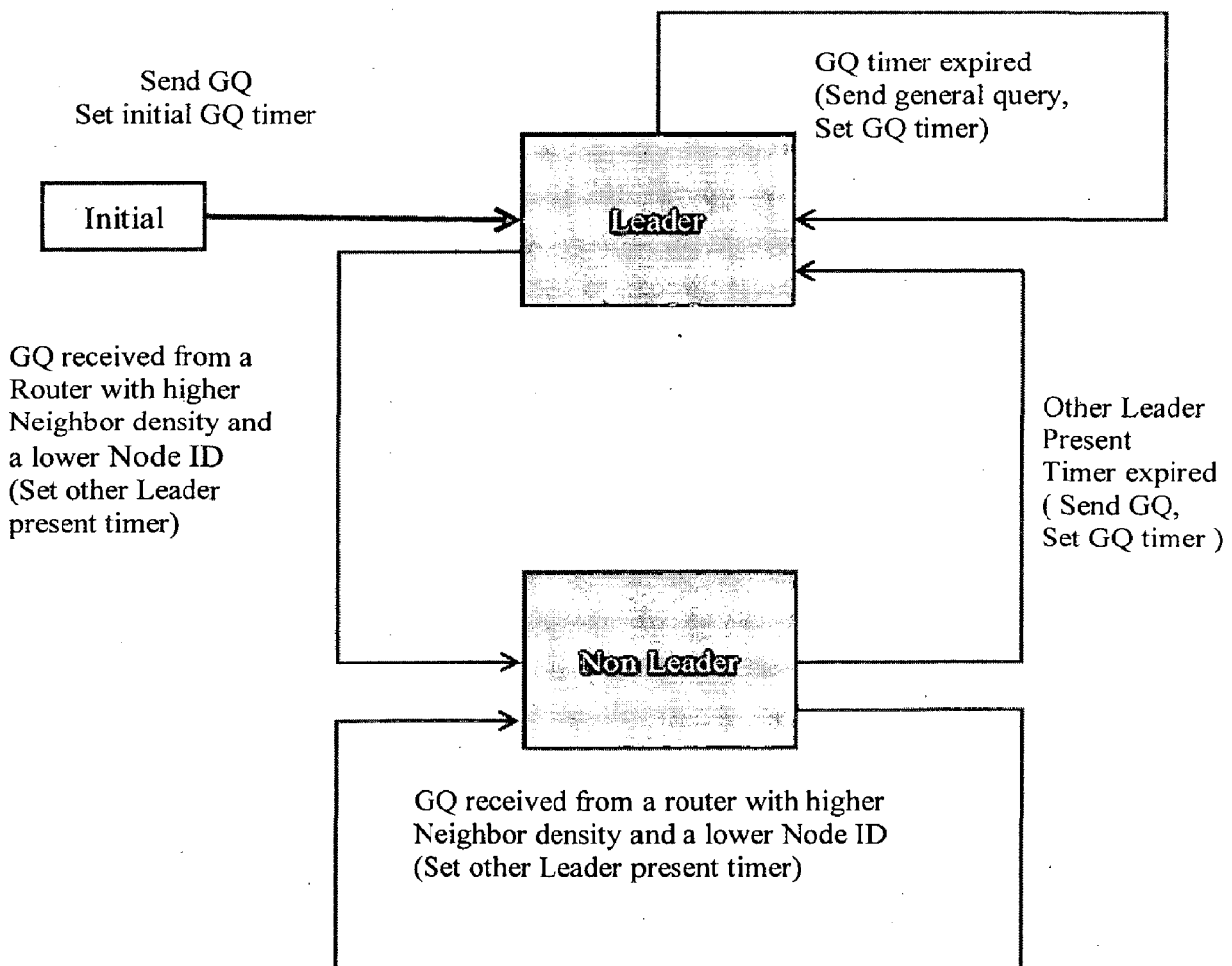


Figure 3.3 Leadership Management Scheme States Diagram



### 3.4.2 Group Management Scheme

The leadership management scheme ensures that there will be only one leader present in each group at any time. This leader has responsibility to manage the group using membership messages and forward the multicast packets to all group members by direct or multi-hop fashion.

The leader periodically broadcast the Group Specific query messages within the group to maintain the group membership information. Other members of group reply to these requests with the Report message. Also when any node enters or leaves the group, it must send the join or leave request to the leader respectively. Since each packet is multicast based membership information is also maintained by each router in the group. So, that they don't have to restart the same membership process, in-case the leader fails. Figure 3.4 only shows the leader functions and the complete flow chart is given inside section 4.3.1 in Figure 4.2.

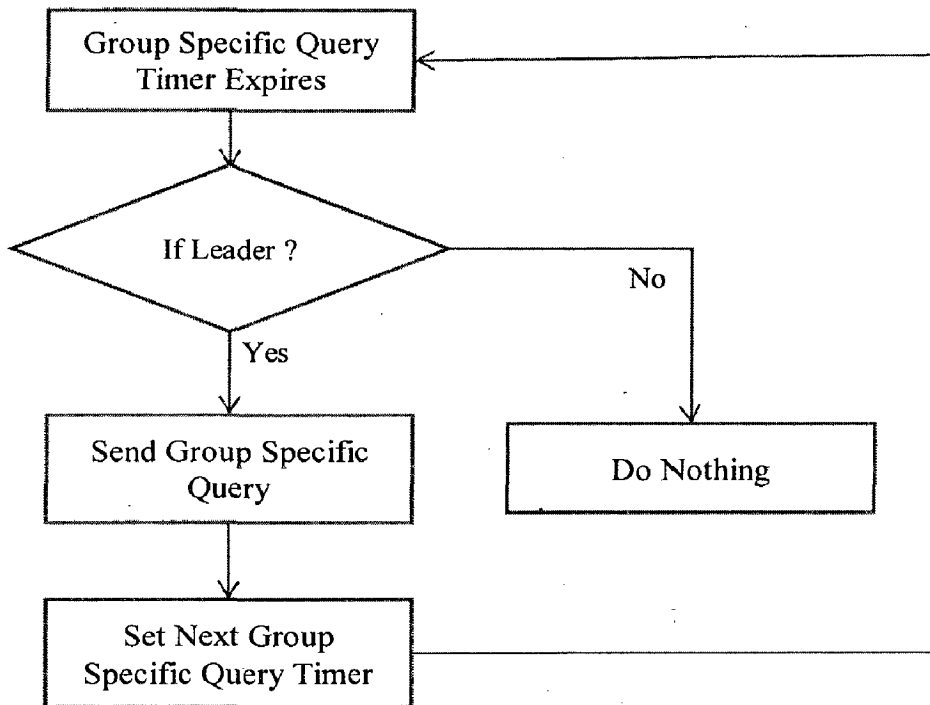


Figure 3.4 Group Management Scheme

If any member leaves the group, it sends the group leave message to group and leader removes the corresponding member from the group. When any new member enters the groups it should send the group joining message to the leader, so the leader adds this node into the group communication.

### 3.4.3 Position based Grouping

The groups considered in our system are based on the position of the node in the map. So, the node present at any region must join the group of that region.

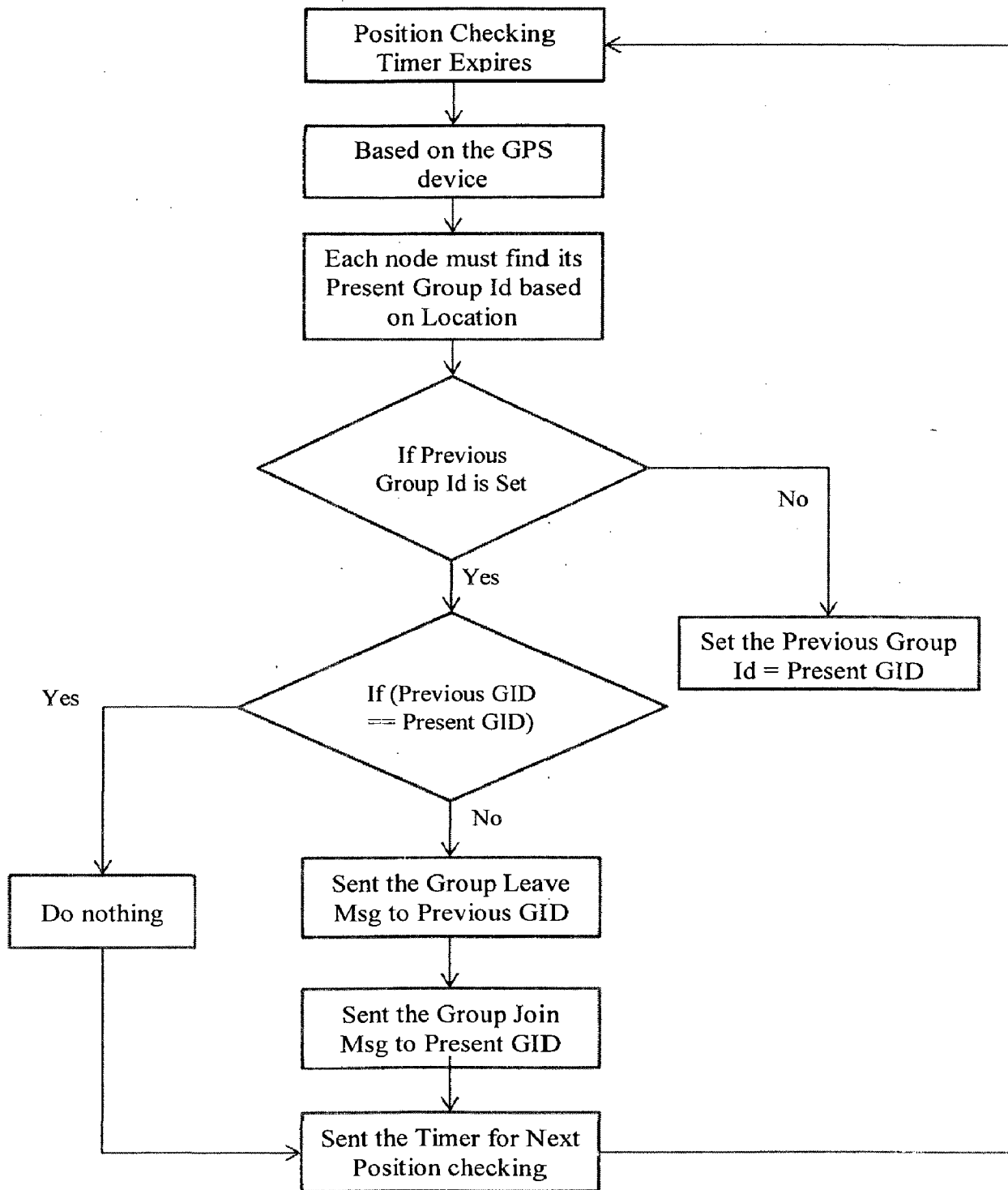


Figure 3.5 Flow Diagram of Position Based Grouping Algorithm

The nodes are mobile in nature and hence, they must check their location at regular intervals. Using some positioning device (e.g., a GPS) must maintain its group membership based on its location. If any node moves outside its region it must send the group membership leave message to the previous group and group joining message to the current group. The whole process can be explained on the basis of flow diagram shown in Figure 3.5.

### 3.4.4 Update Manager

Any node, who wants to update the shared tactical information, must have a lock in its own group. To get this lock, a distributed mutual exclusion algorithm is required so that only one process updates the shared information in a consistent manner.

The approach used here is based on permission-based approach and has been modified so that it can detect node failures and can resolve them. The algorithm must satisfy every required condition of mutual exclusion given in section 2.4.1. The critical section in this case is the updating and transmitting the tactical information within the group. Lock acquisition is the entry point of critical section. And lock release is the exit point of the critical section as shown in Figure 3.6.

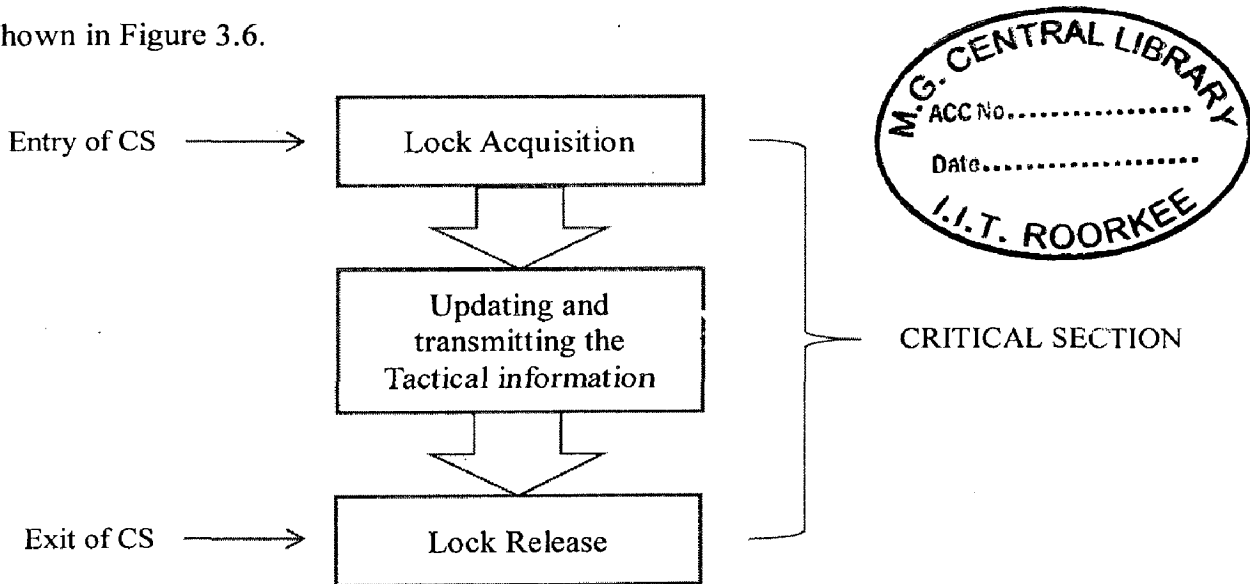


Figure 3.6 Critical Section Diagram

Every node who wants to enter the critical section or wants to acquire a lock must request the permission from the group Leader through the multicast Lock Query Message. Depends on the availability of lock leader may grant the lock or may enqueue the request. If the leader has the lock then it replies to the first member of the queue with the permission to enter the critical section. The permitted node enters the critical section modifies the tactical

information and transmit this information to other members of its own group. After executing the critical section the node which has the lock, can releases the lock. Releasing the lock means the leader got the confirmation that the node has completed the CS, and now the leader has the lock. In this way the tactical information is updated in the consistent way within a region. Only one node i.e. leader can grant the lock to a particular node, it will ensure the safety property stated in the section 2.4.1.

If one node is in the critical section and some other node requests for the lock, then the node inside the critical section replies with wait message to group about the estimated time left to complete the critical section. So the requesting node has an idea of minimum time left after which it can acquire the lock. This message solves the bigger problem of node fault. Considering the scenario, when the node which acquired the lock gets lost or incurs some fault then the absence of this message explains to other member of group that the node is faulty and the leader should take necessary action to re-invoke the lock in the group. In this way, the fault can be detected and removed from the systems. The Figure 3.7, show the timeline demonstration of the algorithms as discussed.

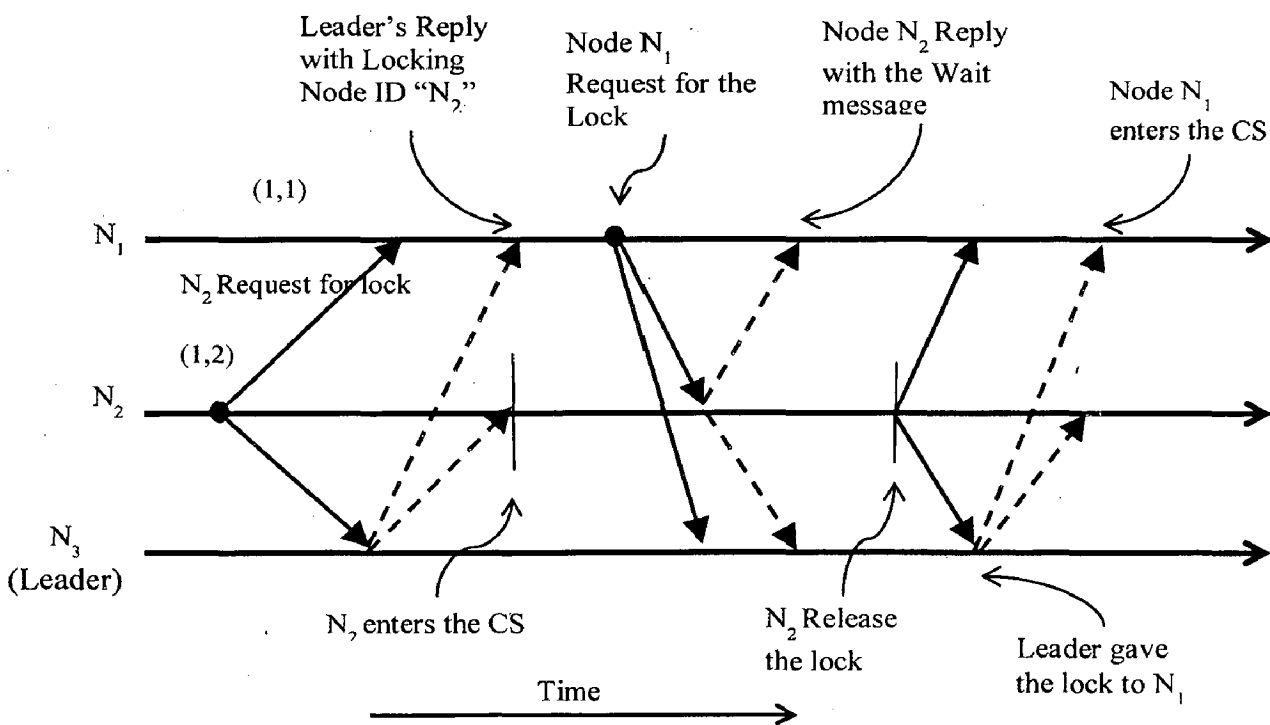


Figure 3.7 Timeline Demonstration of the Algorithm

# CHAPTER 4

---

## IMPLEMENTATION DETAILS

---

The simulation of the proposed scheme was done using QualNet 5.0.2 simulator [21] [22]. QualNet is designed for Network simulation with built-in scenario designer, analyser, packet tracer and file editor. It contains most of the network model libraries for wired and wireless networks.

### 4.1 Modelling Protocol in QualNet

QualNet is a discrete event simulator, [21] it is modelled as it evolves over time by a representation in which the system state changes instantaneously when an event occurs, where an event is defined as an instantaneous occurrence that causes the system to change its state or to perform a specific action. It maintains an event queue associated with each event is its event timeline. Events in the event queue are sorted by the event time. The simulator also maintains a simulation clock which is used to simulate time. The simulation clock is advanced in discrete steps. Thus provide considerable speedup.

There are two types of events in QualNet [21]:

- **Packet events:** Packet events are used to simulate exchange of data packets between layers or between nodes. Packet events are also used for modelling communication between different entities at the same layer.
- **Timer events:** Timer events are used to simulate time-outs and are internal to a protocol.

In QualNet, the data structure used to represent an event is called a message. A message holds information about the event such as the type of event, and the associated data.

Figure 4.1 shows the message passing in QualNet protocol stack. It provide inbuilt library for creating and sending the packet in QualNet.

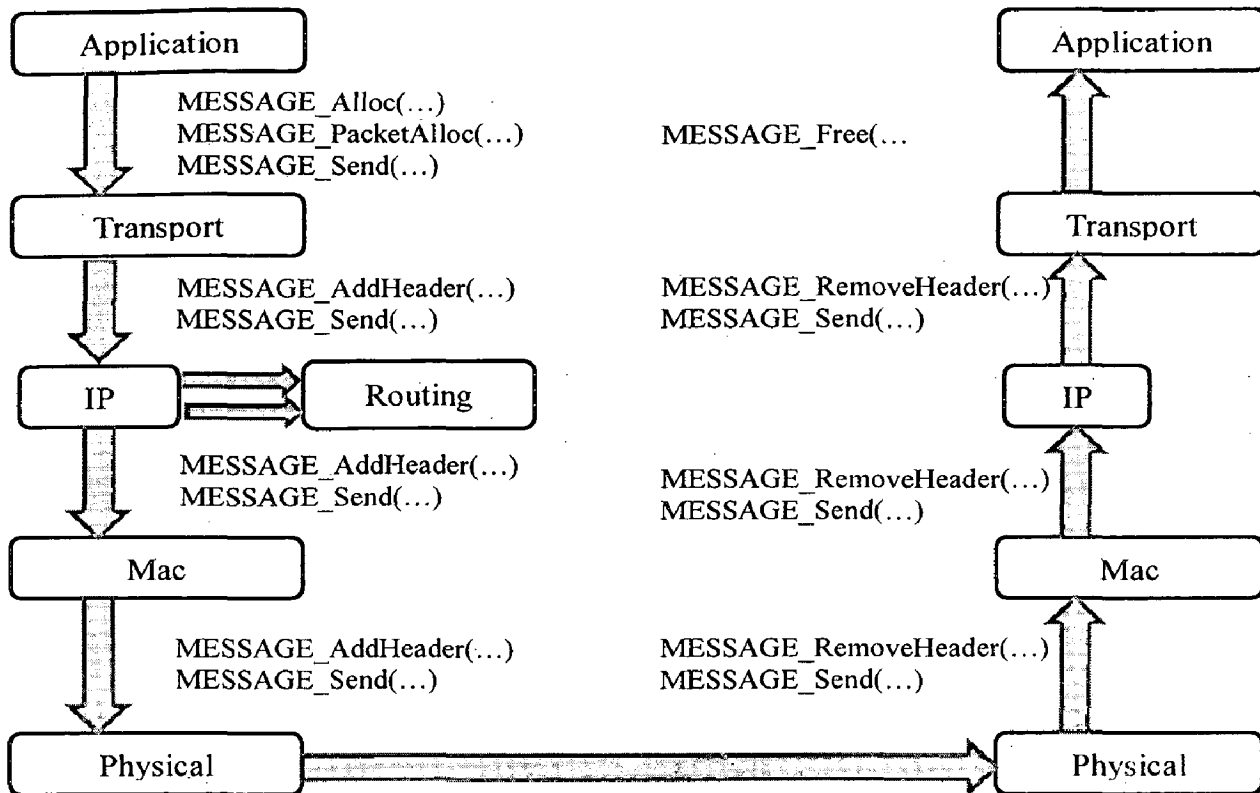


Figure 4.1 Life Cycle of Packet in Protocol Stack of QualNet

## 4.2 Network Model

The individual element of the network is defined below:

**Node:** The MANET consists of mobile nodes. A node can be any portable handheld device or the equipment embedded in a soldier or vehicle. A node has a capability of transmitting and receiving the packets. A node has limited storage space and computing power to perform its operations as it depends upon the battery power. All the task of collecting data, processing data and routing information in the network is done by the node. Nodes are wirelessly transmitting the data to other node either directly or multi-hop fashion through neighbours. The maximum speed of any node is finite and is same for every node.

**Node ID:** A Node in the network also has an identifier, which uniquely recognizes it. The nodes use these identifiers in communication, to address their messages. This ID can be assigned before initializing the network or it can be attained by the nodes in an ad-hoc fashion using some standard protocol.

**Field:** The nodes move in a large rectangular field of finite dimensions  $L \times W$  (as shown in Figure 2.1). The nodes do not leave this area. The nodes are equipped with a GPS device,

thus they know their locations at any given moment. Considering the right-bottom corner of the field as origin, a node can find its position relative to the field.

**Lock:** Each node has a lock field which tell the node about the lock status on the group in which the node belongs. Initial it is set to zero i.e. locked by the leader. The node who wants to access the critical section must acquire the lock from the leader.

Each node is considered to have two types of entities (data structures):

- **Host:** A host is an entity at each node of network who wants to make updates or access the CS. It provides the necessary functionality to node for requesting the access to the critical section.
- **Router:** A router contains all the functionality of host and additional functionalities for selecting the Leader in the group. It also provides the functionality for maintaining the group.

### 4.3 Design and Development of Basic Simulation System

The simulation system consisted of a tactical-map of 1500m X 1500m area with soldiers as mobile nodes. The functionality of each component is explained as follows.

#### 4.3.1 Router Functionality

Router has two defined states:

- **Leader:** This is the state of router, when it is periodic transmitting the group specific query messages in the group.
- **Non-Leader:** This is the state of routers, when there is already one leader present in the group. It learns the presence of leader in the group from the group specific query messages send by the leader.

The transition diagram for these states is shown in Figure 3.3 in section 3.4.1.

Router provides two functionalities: leader selection algorithm and group management scheme as explained in the section 3.4.1 and 3.4.2 respectively. Each router maintains the information about the membership of the group.

The complete description of leader selection algorithm at routers is shown in below Figure 4.2.

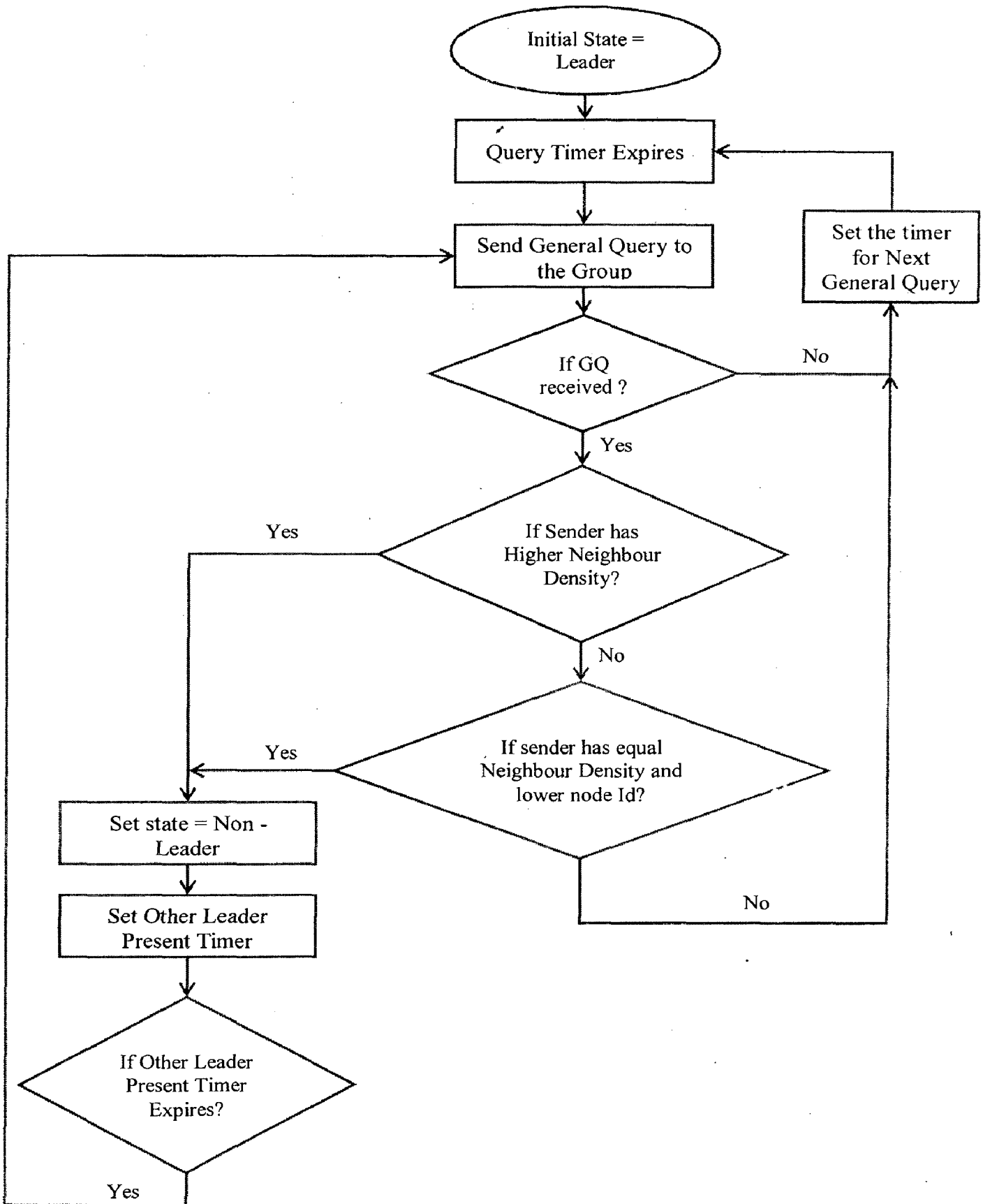


Figure 4.2 Flow Diagram of Router functionality



### 4.3.2 Host Functionality

Host has four defined states:

- **Non-Member:** This is the initial state for a host, when it does not belong to the group.
- **Delaying-Member:** This is the state when the host belongs to the group and has a report delay timer running for that membership.
- **Idle Member:** When host belongs to the group and does not have a report delay timer running for that membership.
- **Locking Member:** When host has acquired the lock and executing the critical section.

Host can exist in one of these possible states, and each state corresponds to specific functionalities. Figure 4.3 the state transition diagram for a host.

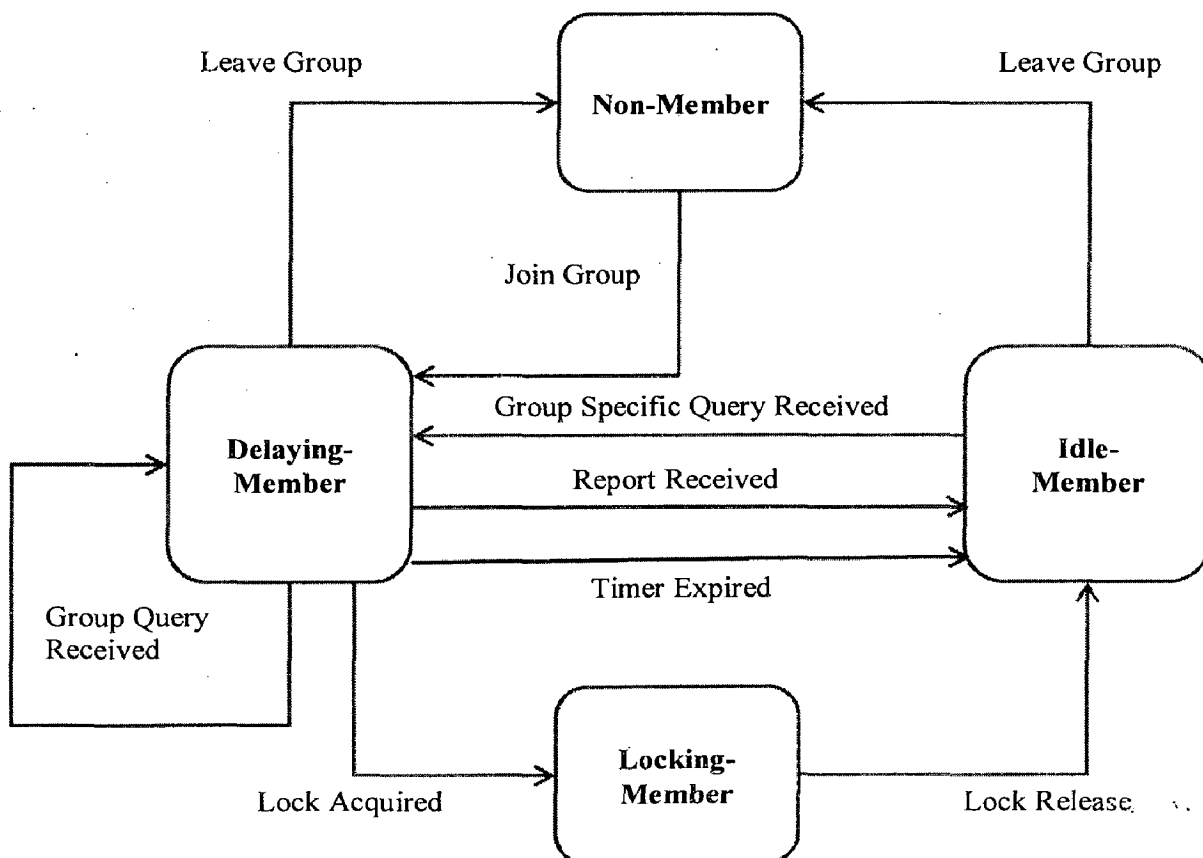


Figure 4.3 Host State Transition Diagram

The router (leader) periodic broadcast the group specific query messages within the group, so the host remains in this group sends back the Report message. If every host reply to the query message then the network get congested due to large number of Report messages. So, to

reduce the traffic, we use message suppression technique [20], such that only one report is sufficient in the group for informing the router that some host are still interested in listening group's data.

There are seven possible events at host that can cause the state transitions:

- **Join Group:** When host decides to join the group, it may occur only in non-member state.
- **Leave Group:** When the host decides to leave the group. It may occur only in the Delaying Member and Idle Member states.
- **Group Query Received:** When the host receives Group-Specific Membership Query message. These queries contain maximum response time for reply the query messages. So, that router is aware of present of host in the network. Queries are ignored for memberships in the Non-Member state.
- **Report Received:** When the host receives a Membership Report message. A Membership Report applies to the maintaining membership in the group. It is ignored in the Non-Member or Idle Member state.
- **Timer Expired:** When the report delay timer for the group expires. It may occur only in the Delaying Member state.
- **Lock Acquired:** When the host acquired the lock in the group. Then the node starts executing the critical section.
- **Lock Release:** When the host release the lock in the group after executing the critical section. This release of lock means to lock opened at the leader.

Initially a host is in the non-member state. When it joins the group, host changes its state to delaying member. Depending upon the query message and report message receive by the host, it changes states from delaying member to idle member or vice versa. When the host is in idle state and receives the group specific query, it changes states to delaying member and starts a timer with random value between zero and maximum reply response time. In delaying member, it waits for timer to expire for sending the report message in reply of query message. But if the other member of the group's reply before the node timer expires and it receives the report message for the same group by other member and then again it changes states back to idle member.

### 4.3.3 Event Handlers descriptions

To implementation all these functionalities, number of timers and packet handler has been implemented in QualNet at each node. Depending upon the type of event occur, either it invokes a specific timer handler for timer event or packet handler for packet events.

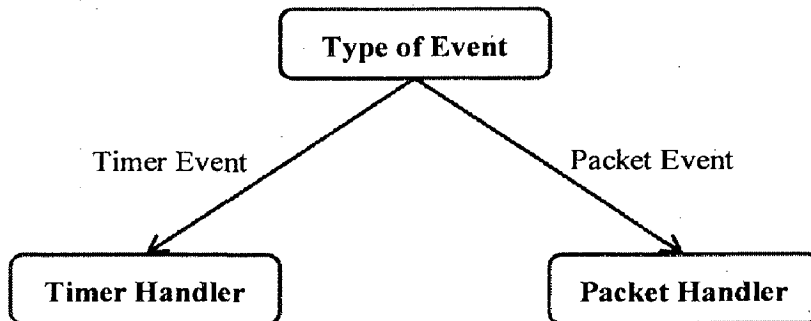


Figure 4.4 Function Calls on Events

#### *TimerHandler()*

This function is used to invoke the particular handler for each timer event take place during simulation. There are number of timer events in this scenario. We will discuss each one of them.

#### *PacketHandler()*

This function is used to invoke the particular packet handler for each packet received event take place at any node. There are numbers of packet event in this scenario. We will discuss each one of them in the section 4.3.5.

### 4.3.4 Messages Types

There are numbers of messages (QualNet events) used for specifying the types of events occur during simulation. They are as follows:

- GENERAL\_QUERY\_MSG
- GROUP\_QUERY\_MSG
- REPORT\_MSG
- JOIN\_GROUP\_MSG
- LEAVE\_GROUP\_MSG
- LOCK\_GROUP\_MSG
- LOCK\_GRANT\_MSG

- LOCK\_RELEASE\_MSG
- WAIT\_MSG

### 4.3.5 Timer Handler Functions

#### *HandleQueryTimer()*

This function is called when the query timer expires at a router structure of the node. It checks the status of the router and if the router is in leader state then only it multicast the group specific query message within the group and sets the timer for next group specific query.

#### *HandleOtherLeaderPresentQueryTimer()*

This function is called when the other leader present timer expires at any router when it is the non-leader state and if it does not receives the general query messages from the leader of the group after the `other_leader_present_interval`. Then the router changes its own state from non-leader to leader in that group and it will start transmitting the general query within the group such that other member knows about its state transition.

#### *HandleGroupReplyTimer()*

This function is called when the host is in the delaying member state after receiving the group query and the reply delay timer is expired. So, the host reply with the report message to the group query.

#### *HandlePositionCheckTimer()*

This function is called at host to validate its group membership based on the position of the node. Its group membership status has to be checked after a fixed interval of time. This fixed time interval is called as `position_refresh_interval`.

#### *HandleLockQueryTimer()*

This function is called when the host want to acquire the lock to get access of the critical section. To simulate the lock acquisition behaviour, we implemented this timer. At the start of the simulation the node start this timer and when the timer expires, the node start the lock acquisition process by sending `Lock_group_msg` and set the next lock query timer for next access to the critical section.

### 4.3.6 Packet Handler Functions

In this scheme, each node can either act a host or router as shown in Figure 4.5. Router has most of the functionalities of host thus, it contains some of the functions of host with its own additional functions as shown in Figure 4.6.

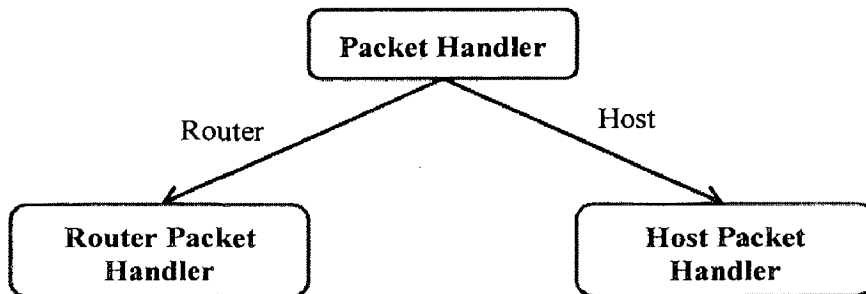


Figure 4.5 Packet Handler Diagram

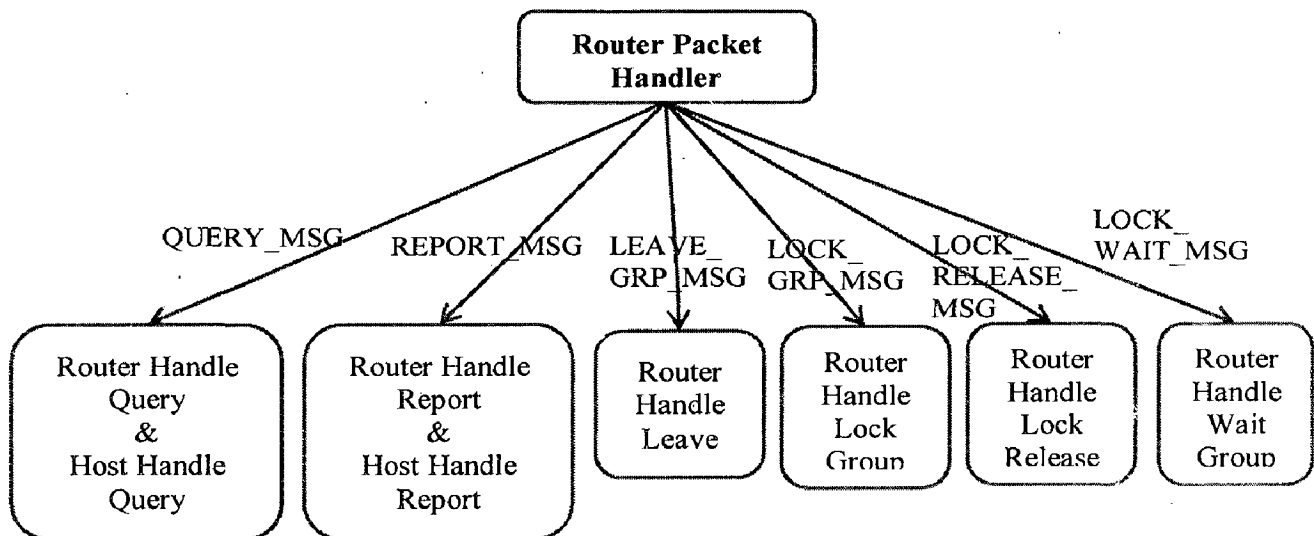


Figure 4.6 Router Handle Packets Diagram

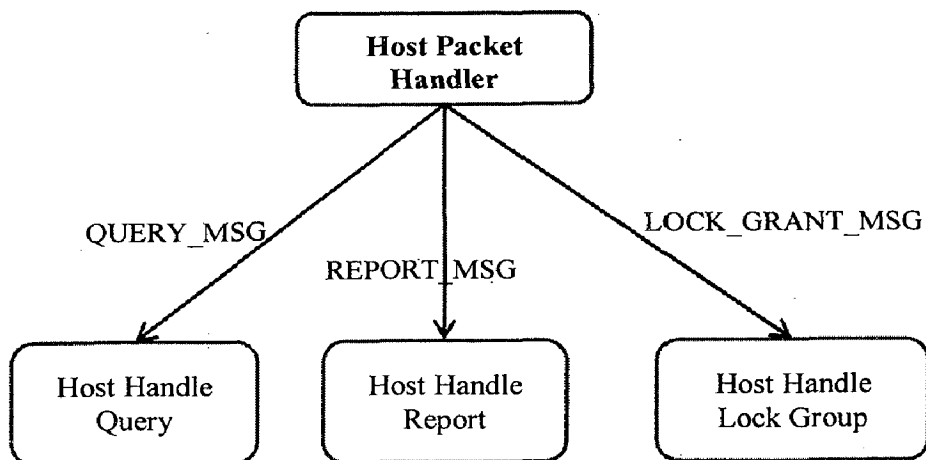


Figure 4.7 Host Handle Packet Diagram

# CHAPTER 5

## RESULTS

The following sections discuss the performance of the proposed active information sharing scheme and compares it with the Wu Scheme [7].

### 5.1 Simulation Setup

The simulator used is QualNet 5.0.2 [21] [22] by Scalable Network Technologies, a platform widely used for simulations of MANET. The simulation parameters are same as those used in [7] performance monitoring.

**Simulation Configuration**

<b>Simulation Scenario</b>	Battle Field Environment
<b>Battlefield Area</b>	1500m X 1500m
<b>Channel Bandwidth</b>	2Mbps
<b>Avg. Mobility of Nodes</b>	20 mps
<b>Mobility Model</b>	Random-way Point Model
<b>No. of nodes</b>	4, 8, 12, 16, 20
<b>Simulation Time</b>	3000 min

**Table 5.1** Simulation Configuration

### 5.2 Simulation Analysis

The performance of a distributed mutual exclusion algorithm is analysed under two special load conditions:

- **Low-load:** In low-load conditions, there is not often more than one request for critical section simultaneously in the system.
- **High-load:** In high-load conditions, there are always one or more pending requests for critical section.

### 5.2.1 Performance Parameters

The performance of mutual exclusion algorithms is generally measured by the following metrics:

- **Message complexity:** This is the number of messages that are required per CS execution by a node.
- **Synchronization delay:** After a node leaves the CS, it is the time required, before the next site enters the CS as shown in Figure 5.1.
- **Response time:** This is the time interval a request waits for its CS execution to be over after its request messages have been sent out as shown in Figure 5.2. Thus, response time does not include the time a request waits at a node before its request messages have been sent out.

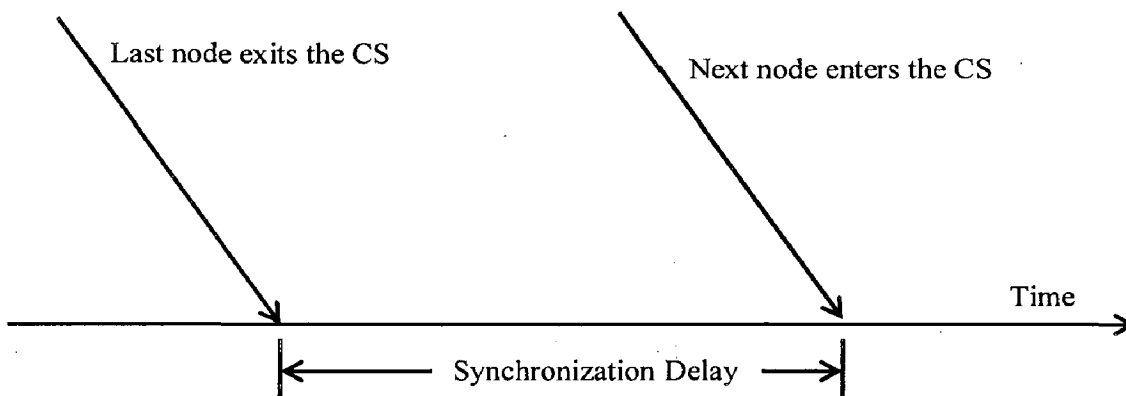


Figure 5.1 Synchronization Delay

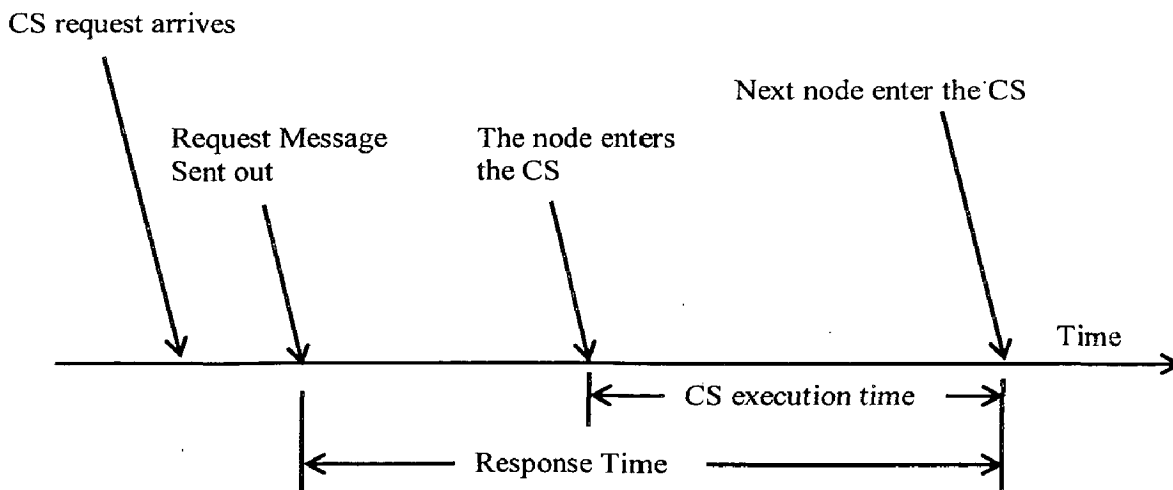


Figure 5.2 Response Time



### 5.3 Performance Evaluation

Two set of simulations were conducted to analyze the impact of different load conditions on our algorithm performance parameters.

At any node arrival of the CS request is considered to follow Poisson distribution with mean  $\lambda$ , which represents the mean number of CS requests generated by a single node per second.

For Low-Load Condition:  $\lambda = 1.00E(-4)$  or 0.0183156389

For High-Load Condition:  $\lambda = 1.00E(-2)$  or 0.718281828

For generating the CS request at these rates, we initialize the Lock\_Query timer with the value generated by Poisson distribution and every time the timer expires, we set the next timer with next value of this distribution.

#### 5.3.1 Message Complexity when Nodes are faultless

The number of nodes has a major impact on the system performance, since it is related to the total number of CS requested.

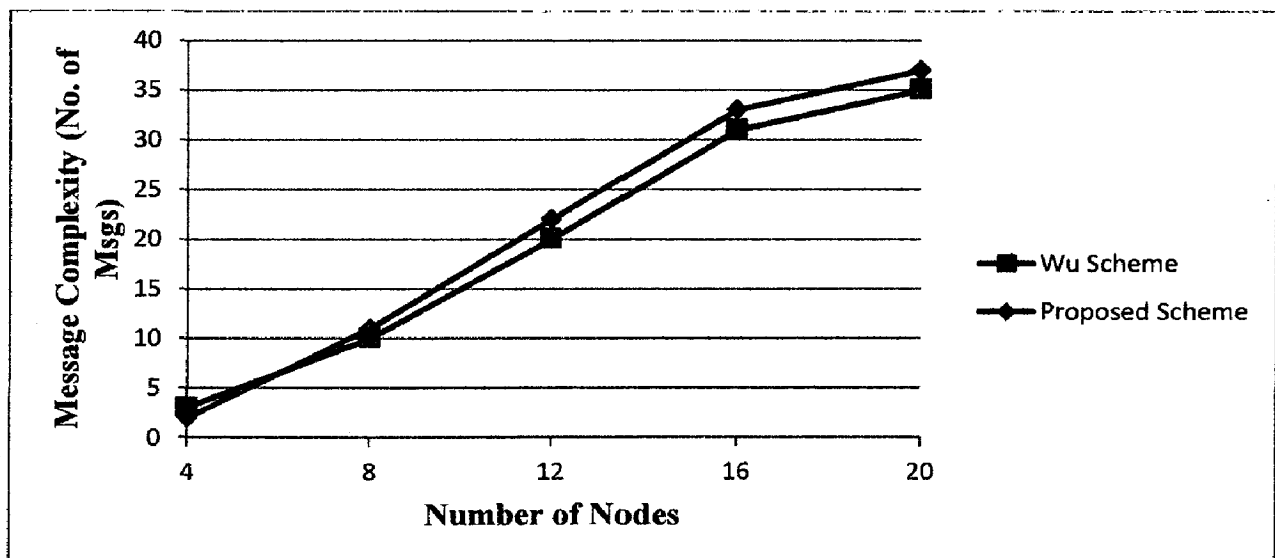


Figure 5.3 Message Complexity in Low load condition

We first considered the case when nodes are faultless. Figure 5.3 and Figure 5.4 show the increase in the message complexity with the number of nodes due to extra wait messages sent by the locking node to inform the time left to complete the CS to the others nodes requests the lock. It can be seen that Wu scheme has lower message complexity because of extra wait message used in our scheme.

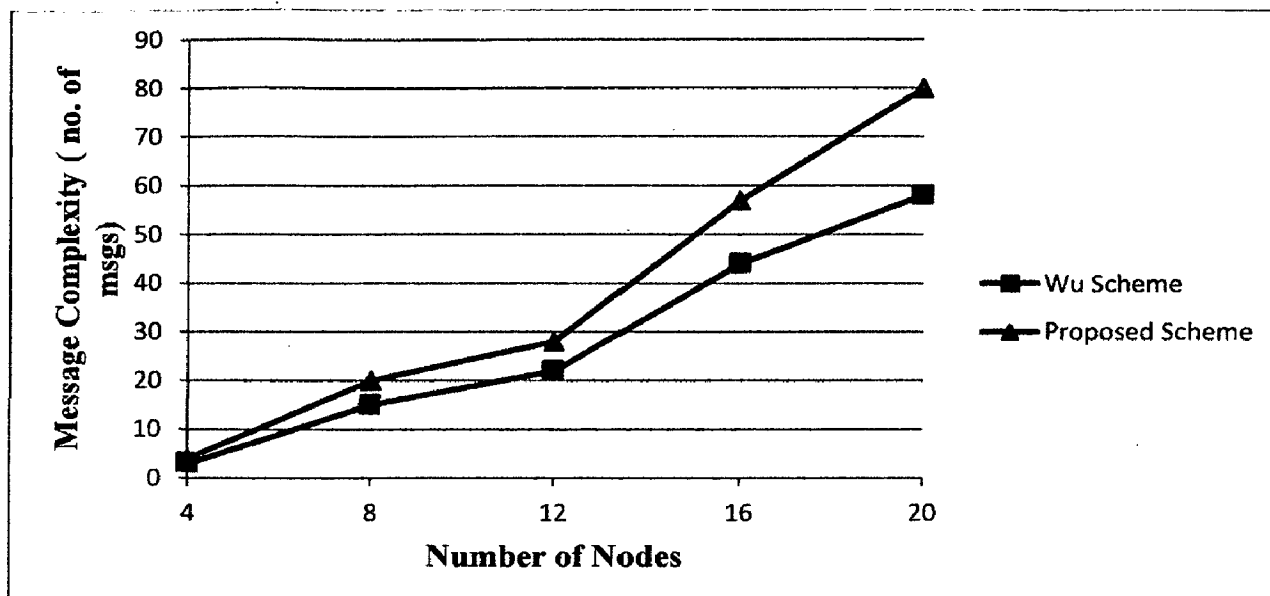


Figure 5.4 Message Complexity in high load condition

### 5.3.2 Message Complexity when Nodes are faulty

In our second set of experiments we considered that the fault rate is 10%, i.e. percentage of node failure is 10%.

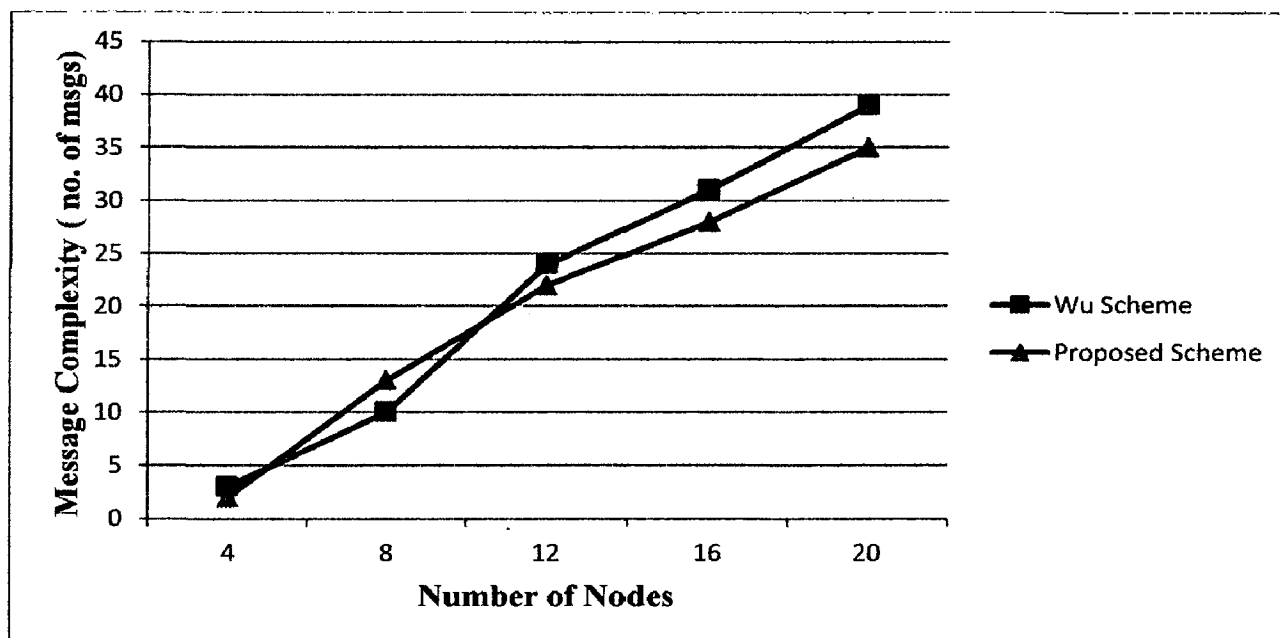


Figure 5.5 Message Complexity at low load condition (Faulty nodes)

The benefit of extra wait message can be seen in case when the locking node incurs a fault and crashes during CS. Figure 5.5 shows that in low load condition less number of messages are required in our scheme once the number of nodes in region becomes more than 10.

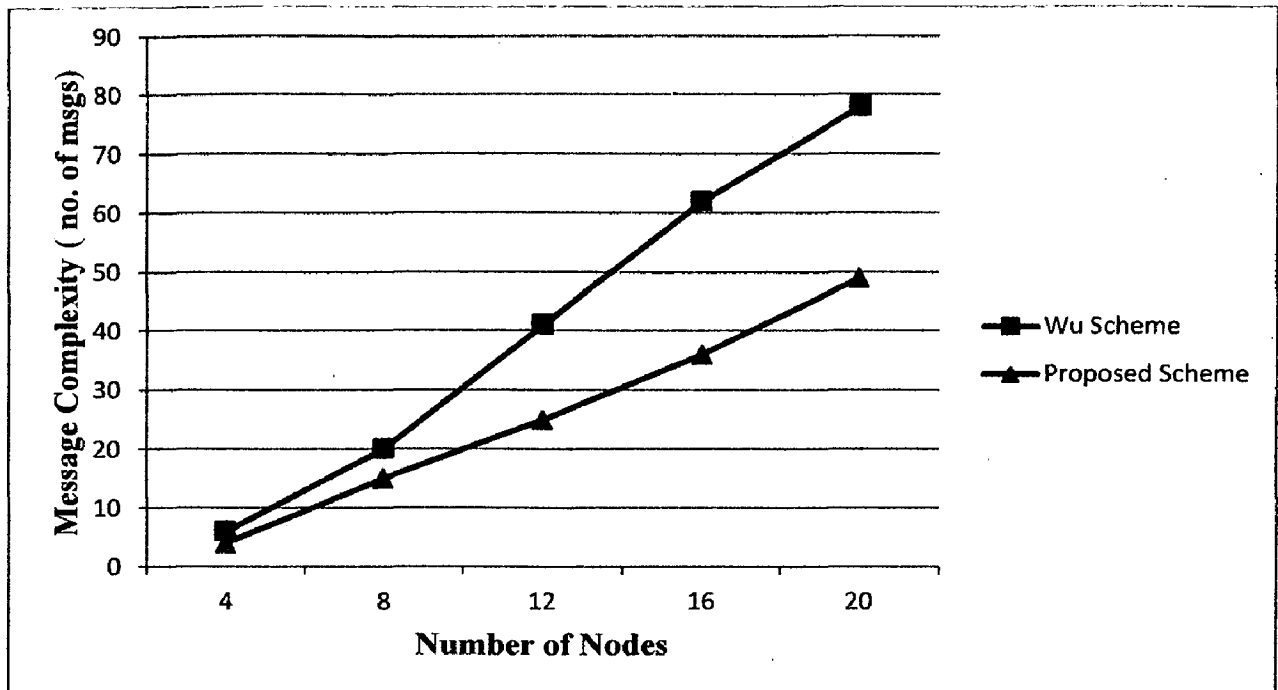


Figure 5.6 Message Complexity at High Load condition (Faulty nodes)

Figure 5.6, shows our scheme has out-perform Wu scheme in term of message complexity in the case of high load condition. Since, wait message efficiently reduces the extra messages traffic generated by time-out event in Wu scheme.

### 5.3.3 Response Time when Nodes are faultless

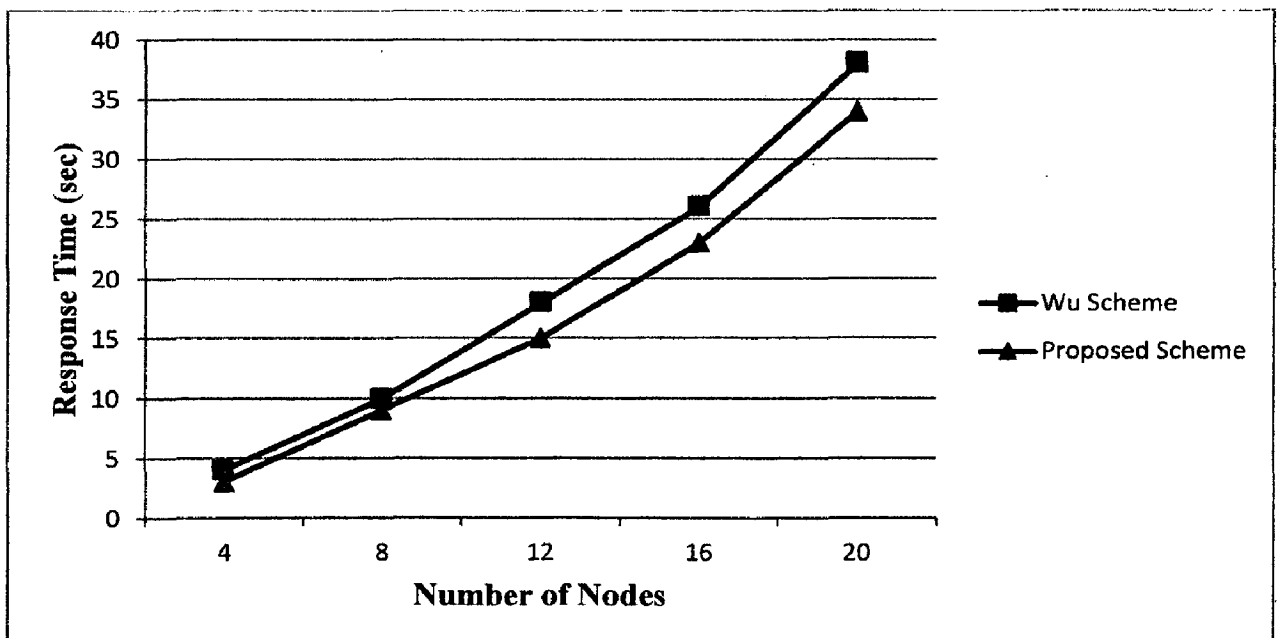
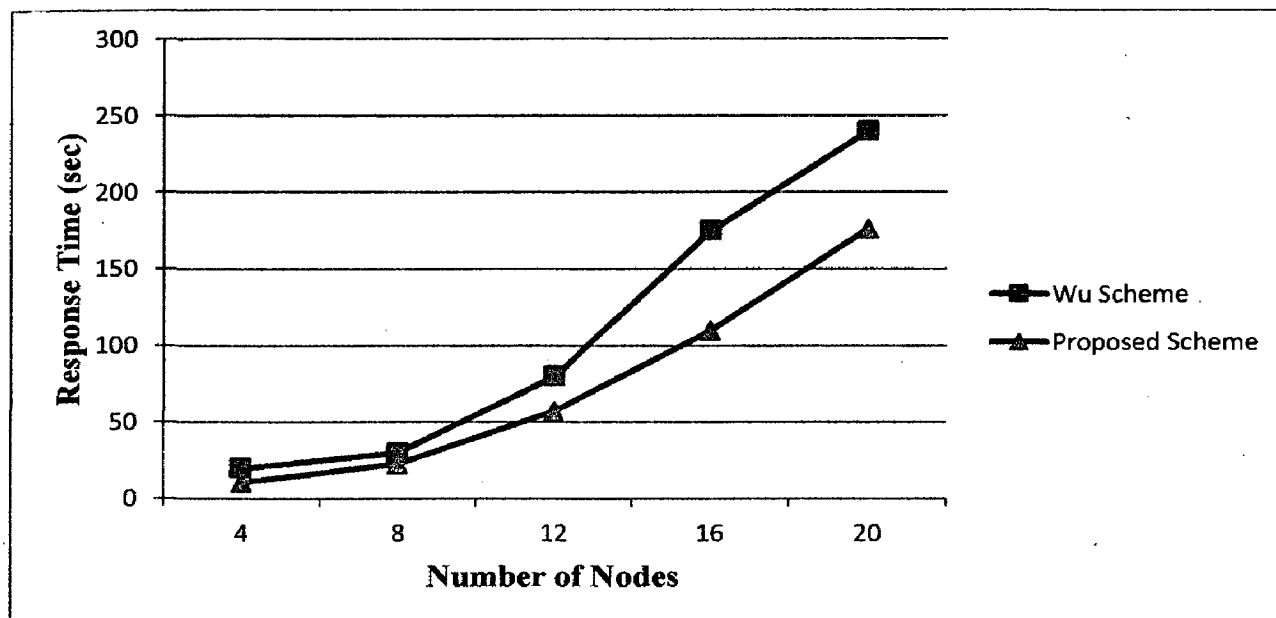


Figure 5.7 Response Time on low-load condition

As shown in Figure 5.7, response time in our case is slightly less than the Wu Scheme because in low-load condition the time-out of Wu scheme and wait message of our scheme behaves almost same.

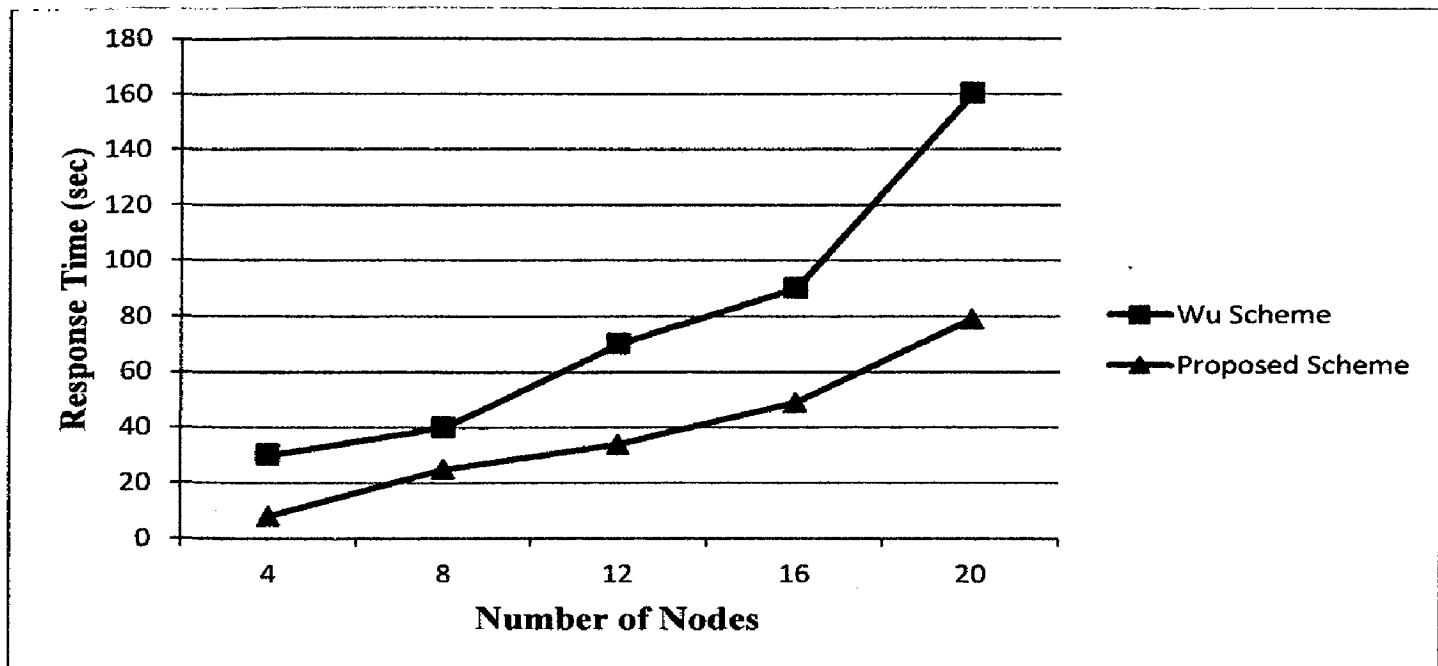


**Figure 5.8** Response Time on High load Condition

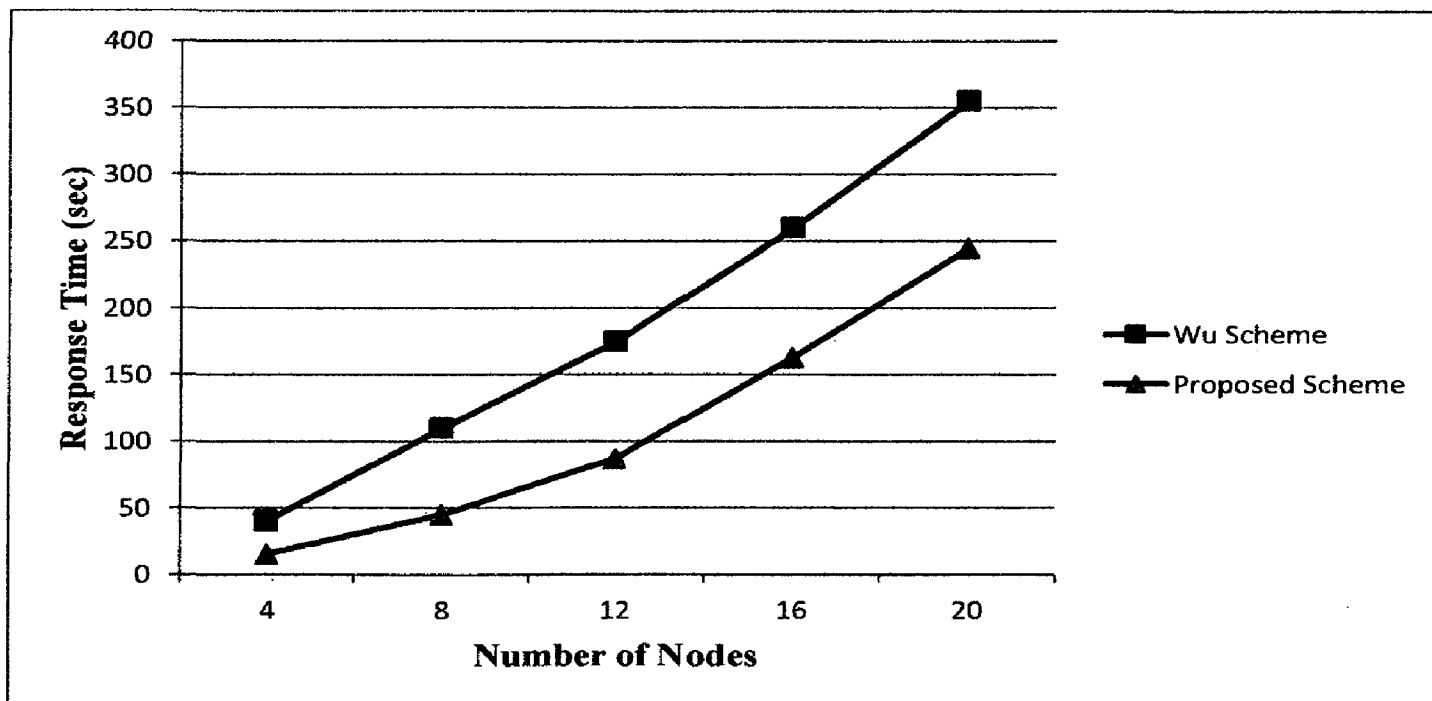
Figure 5.8, shows the response time for high load condition. It can be observed that the response time is lesser in our scheme because of the multicast request and response functionality.

### 5.3.4 Response Time when Nodes are faulty

Figure 5.9 and 5.10, shows that our scheme give considerable improvement in response time in case of faulty nodes, because of the response message give better functionality than the time-out used in Wu Scheme.



**Figure 5.9** Response Time in low-load condition (Faulty Nodes)



**Figure 5.10** Response Time in High load condition (Faulty nodes)

Figure 5.10 shows the significant improvement in response time, which validates our scheme work better in scenario where mobile nodes exhibits the faulty behaviour which is generally possible in mobile ad hoc networks.

# CHAPTER 6

---

## CONCLUSION

---

### 6.1 Conclusion

In this dissertation, a robust Active Information Sharing scheme based on permission based distributed mutual exclusion algorithm is proposed. The main idea of the scheme is that each node which acquired the critical section should reply to the request of CS from other members with wait message to provide the estimated time remaining to leave CS. This scheme satisfies the requirements of distributed mutual exclusion like safety, liveness, fairness and robustness through the use of report message by CS executing node. Extensive simulation has been conducted on both faultless and faulty node systems to demonstrate that the message complexity and response time can be kept low, even in the presence of large number of fault occurs at the node executing the CS. Performance comparisons show that the proposed scheme is efficient and suitable for mobile ad hoc networks.

### 6.2 Suggestion for Future Work

Applications of MANETs are open area of research, the following issues may be addressed in future:

- 1) The scheme proposed in this dissertation considered an assumption that network partition does not take place inside any region while achieving consistency and availability. In future the other options can be explore where network partition is considered with Eventual consistency.
- 2) In our algorithm to achieve fault tolerance, we made a slight trade-off with the message complexity. In future, one possible solution is to reduce the number of members whom the wait message should be send to improve the message complexity of the solution.
- 3) In our solution, the fixed region based groups are created. A node joins the group based on their location. In future, another possible solution is to create dynamic groups on the basis of distance between the nodes. So, the nearby nodes form a cluster and the node can belong to multiple clusters.

# REFERENCES

---

- [1] M. Corson and J. Macker, "RFC2501: Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations", Informational, <http://www.ietf.org/rfc/rfc2501.txt>, 1999.
- [2] Ajay D. Kshemkalyani and Mukesh Singhal, "Distributed Computing Principles, algorithms and Systems", Cambridge University Press, 2008.
- [3] L. Lamport, "Time, Clocks and the Ordering of Events in a Distributed System", Communications of the ACM, 21(7), pp558–565, July 1978.
- [4] G. Ricart, A.K. Agrawala, "An optimal algorithm for mutual exclusion in computer networks", Communications of the ACM 24 (1981) 9–17.
- [5] I. Suzuki and T. Kasami, "A distributed mutual exclusion algorithm", ACM Transactions on Computer Systems, 3(4), 344–349, 1985.
- [6] K. Raymond, "A Tree-based algorithm for distributed mutual exclusion", ACM Transactions on Computer Systems, Feb. 1989, pp. 61–77, 1989.
- [7] 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, 2008, pp. 139-160, 2008.
- [8] M. Bouillaguet, L. Arantes, and P. Sens, "A Timer-Free Fault Tolerant K-Mutual Exclusion Algorithm", 2009 Fourth Latin-American Symposium on Dependable Computing LADC 09, September 2009, pp. 41-48, 2009.
- [9] M. Singhal, "A dynamic information structure mutual exclusion algorithm for distributed systems", IEEE Transactions on Parallel and Distributed Systems 3 (1) (1991) pp. 121–125, 1991.
- [10] M. Singhal and D. Manivannan, "A distributed mutual exclusion algorithm for mobile computing environments", In Proceedings of IASTED International Conference on Intelligent Information Systems'97, IEEE Computer Society, 1997, pp. 557–561, 1997.
- [11] M. Maekawa, "A  $\sqrt{N}$  Algorithm for Mutual Exclusion in Decentralized Systems", ACM Transaction on Computer Systems, vol. 3, No. 2, pp. 145–159, 1985.
- [12] D. Agrawal, A.E. Abbadi, "An efficient and fault-tolerant solution for distributed mutual exclusion", ACM Transactions on Computer Systems (1991) 1–20.

- [13] J.E. Walter, S. Kini, "Mutual exclusion on multihop, mobile wireless networks", Technical Report, Dept. of Computer Science, Texas A&M University, Dec. 1997.
- [14] J.Walter, J.Welch, N. Vaidya, "A mutual exclusion algorithm for ad hoc mobile networks", *Wireless Networks* 9 (2001) 585–600, 2001.
- [15] R. Baldoni, A. Virgillito, and R. Petrassi, "A distributed mutual exclusion algorithm for mobile ad hoc networks", in *Proceedings of the 7th IEEE Symposium on Computer and Communications (ISCC 2002)*, Toarmun, Italy, July 2002, pp. 539–545, 2002.
- [16] S. M. Masum, M. M. Akbar, A. A. Ali, and M. A. Rahman, "A consensus-based  $\ell$ -Exclusion algorithm for mobile ad hoc networks," *Ad Hoc Networks*, Vol. 8, No. 1, 2010, pp. 30-45, 2010.
- [17] Y. Lubowich and G. Taubenfeld, "On the Performance of Distributed Lock-Based Synchronization", in *Proceedings of the 12th International Conference, ICDCN 2011*, Bangalore, India, January 2-5, 2011, pp. 131, 2011.
- [18] M. Singhal, "A taxonomy of distributed mutual exclusion", *Journal of Parallel and Distributed Computing* 18 (1993) pp. 94–101, 1993.
- [19] Seth Gilbert and Nancy Lynch, "Brewer's Conjecture and the Feasibility of Consistent, Available, Partition-Tolerant Web Services", *ACM SIGACT News*, 33(2), June 2002.
- [20] W. Fenner, "RFC 2236 – Internet Group Management Protocol", Standard track, Version 2, <http://www.ietf.org/rfc/rfc2236.txt>, Nov 1997.
- [21] Scalable Network Technologies Inc., "QualNet 5.0.2 Programmer's Guide", March 2010.
- [22] Scalable Network Technologies Inc., "QualNet 5.0.2 User's Guide", March 2010.