RELAXED MUTUAL EXCLUSION ALGORITHM

IN

MOBILE CELLULAR NETWORK

Submitted in Partial Fulfillment of the Requirement

For the Award of the Degree of

Master of Technology

in

Computer Science and Engineering

By

PREETI VATS
University Roll No. 2K13/CSE/34

Under the Esteemed Guidance of

Mr. Vinod Kumar
Assistant Professor, Computer Engineering Department, DTU

2013-2015

DEPARTMENT OF COMPUER SCIENCE AND ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

DELHI – 110042, INDIA

ABSTRACT

Now a days, Mobile computing networks revolutionize the way computers are used. Mobile hosts have small memory, a relatively slow processor and low power batteries, and communicate over low bandwidth wireless communication links. Distributed dynamic channel allocation in mobile cellular networks and other similar structures is a fundamental resource management problem. It has aroma of distributed mutual exclusion but is not exactly mutual exclusion problem because a channel must be reused in different cells. Existing mutual exclusion algorithms for cellular systems are not suitable to mobile systems due to above limitations. Here, the exact relation being established between the both algorithms. Specifically, we describe the procedure of relaxed mutual exclusion to modularize the problem of distributed channel allocation.

A geographical area is divided into hexagonal cells, every cell uses the same bandwidth, which is divided into channels. Cell phone must use one channel to make a call. Cells have different traffic usage of channels such that every cell could need different quantity of channels. Allocation algorithm should dynamically allocate channels (resources) to cells, to get the most efficient usage of available bandwidth using relaxed mutual exclusion. We propose a general algorithm that guarantees relaxed mutual exclusion for a single resource, prove a necessary and sufficient condition for the information structure, and address the issues that arise in relaxed mutual exclusion, including deadlock resolution, dealing with multiple resources, and design of efficient information structure. Here, improvement based on mutual exclusion solution, such as deadlock minimization and broken cells serving extended to multiple channel case is described to an extent to use the bandwidth efficiently.

Keywords : Relaxed Mutual Exclusion, channel allocation, deadlock minimization, broken cell serving, TSDP(Two-Step Dynamic-Priority), borrowing scheme

ACKNOWLEDGEMENT

First of all, I would like to express my deep sense of respect and gratitude to my project supervisor Mr. VINOD KUMAR for providing the opportunity of carrying out this project and being the guiding force behind this work. I am deeply indebted to him for the support, advice and encouragement he provided without which the project could not have been a success.

Secondly, I am grateful to Dr. O.P. VERMA, HOD, Computer Engineering Department, DTU for his immense support. I would also like to acknowledge Delhi Technological University library and staff for providing the right academic resources and environment for this work to be carried out.

Last but not the least I would like to express sincere gratitude to my parents and friends for constantly encouraging me during the completion of work.

Preeti Vats

University Roll no: 2K13/CSE/34

M.Tech (Computer Science & Engineering)

Department of Computer Engineering

Delhi Technological University

Delhi – 110042

Computer Engineering Department
Delhi Technological University
Delhi-110042
www.dce.edu

CERTIFICATE

This is to certify that the dissertation titled **"RELAXED MUTUAL EXCLUSION IN MOBILE CELLULAR NETWORK "** is a bonafide record of work done by **Preeti Vats, Roll No. 2K13/CSE/34** at **Delhi Technological University** for partial fulfilment of the requirements for the degree of Master of Technology in Computer Science & Engineering. This project was carried out under my supervision and has not been submitted elsewhere, either in part or full, for the award of any other degree or diploma to the best of my knowledge and belief.

(Mr. Vinod Kumar)
Assistant Professor & Project Guide
Department of Computer Engineering
Delhi Technological University

Date: ______

Table of Contents

List of Figures

List of Abbreviations

MH	Mobile host
MSS	Mobile Support System
BTS	Base Transceiver Station
BSS	Base Station controller
ME	Mutual Exclusion
CS	Critical Section
FIFO	First-in ,first out
RME	Relaxed mutual exclusion
DRME	Distributed relaxed mutual exclusion
GRME	Generalized relaxed mutual exclusion
DCA	Dynamic channel allocation
DDCA	Distributed dynamic channel allocation
FCA	Fixed channel allocation
HCA	Hybrid channel allocation
TSDP	Two- step dynamic priority

CHAPTER 1

INTRODUCTION

1.1 Mobile Cellular Network

A Mobile Cellular Network consists of two sets: one is random number set for mobile hosts (MH) and another one is finite set for fixed devices. Some mobile device act like base stations and supports mobile network. These are called as mobile support systems (MSS). In these networks, some mobile host which do not support communication, also considered as MSSs as their cells never communicate to their mobile support system. All communication networks and fixed hosts communicate them with specified communication network. Each mobile support system can directly communicate easily using any wireless medium with mobile hosts located in its cell. Wireless network or Mobile cellular network covers certain geographical area. The mobile support system covers a this geographical area. This area is divided into various small hexagonal cells. BTS (Base Transceiver Station) is located in the centre of each cell. BTS is mobile telephones which are in services. Each BTS is connected with BSC (Base Station Controller) and using wired network such as optical fibre cables, BSCs are connected each other. A support system can communicate directly with mobile device *or* vice versa only when the mobile host is physically occur in the cell which is serve by mobile support system. Each mobile network have its assigned bandwidth. Each BTS shall manage this bandwidth. At random instant of time, a mobile host can occur only one cell. If any host wants communicate another host which do not exist in the same hexagonal area, the source *MH* will connect its local *MSS* which further communicate and send message to local base station of the target device over the wireless network. When a mobile user travel from one area (cell) to another area (cell), a *handoff* algorithm will executed by the base system of two cells. Message propagation delay at physical (wired) medium is finite and arbitrary .The channels uses FIFO procedure for delivery of messages between *MSS* and its local mobile hosts.

A mobile host can travel from one hexagonal area (cell) to another hexagonal area (cell). The communication between the network and mobile host will break if its base station fails. To recover the connection, mobile host must move into another cell covered by an operational base station. So, a mobile host is always directed to another network reconnect to the network before it loses the connection to its base station. A mobile host can fail or disconnect from the network anytime and lost its current volatile state.

Available bandwidth divided into channels and each of channel may busy with only one communication call at one time. A given channel can be used in many cells simultaneously, but cells which are using the same channel must be placed in some distance apart because simultaneous use of channel and its bandwidth will successful when distance between both cells must be greater than minimum reuse distance . When a cell receive a call, it try to serve the call by assigning a channel to a call. The call is blocked or to be awaited if no channel is available. Now if mobile host travel to other cell and channels are not available, then call is dropped.

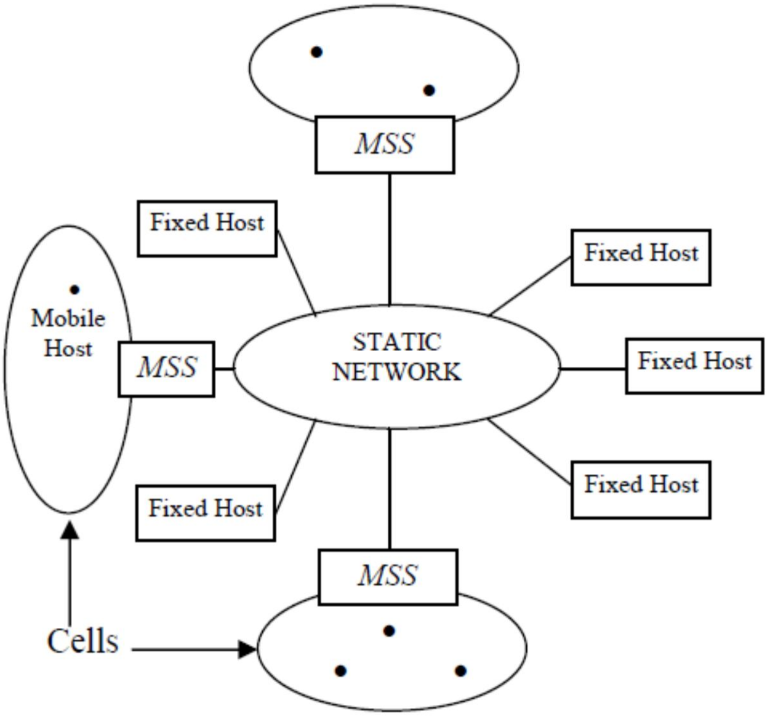


Fig 1.1 : Structure of Mobile Cellular Network

1.2 Mobile Network Model

The cellular network is regular and uniform grid of hexagonal cells with radius *R*. Mobile cellular networks have *p* rows and *q* columns of cells. (Fig.1.2 sketch a 7x7 cellular network.) The cell with row I and column J can be denoted as (I, J) .Every cell have one geographical or physical centre. The x-axis point to the direction of row 0 at the rectangular Cartesian coordinates system, the coordinates *(x,y)* of the center of cell (I,J`) can be calculated as:

$$(x, y) = (c, r)\begin{pmatrix} \sqrt{3R} & 0 \\ \frac{\sqrt{3}}{2}R & \frac{3}{2}R \end{pmatrix} \quad (1)$$

The *distance* between any of two cells in the network, c_1 and c_2, given by $dist(c_1, c_2)$. It is called as Euclidean distance between centres of c_1 and c_2. Thus, the Cartesian coordinates $c_1 = (x_1, y_1)$ and $c_2 = (x_2, y_2)$, the distance between c_1 and $c_2$is given by

$$dist((x_1 y_1), (x_2 y_2)) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}\,. \quad (2)$$

A conservative way is to use *minimum reuse distance* D_{min} , which provide better solution for inference problem. Two cells having distance larger than or equal to D_{min} (minimum reuse distance) between them can use the same channel without interfering each other. It will guarantees interference-free carrier reuse. The value of D_{min} can be choose to fulfil the given prescribed requirement even in worst interfering situation. Every cell has six neighbouring interference cells. In the worst case, ratio is given by

$$C/I = \frac{\left[\left(\frac{D_{min}}{R}\right) - 1\right]^4}{6}$$

Here we assumed fourth-power law attenuation. With $D_{min} = 3\sqrt{3}R$, $[C/I]_{min} \approx 17\,dB$ are reasonable value which are used in practice. DCA strategies does not use any specified value for D_{min} it is assumed $D_{min} = 3\sqrt{3}R$.

The problem is stated as allocating channels to various cells, such that efficient usage of available bandwidth is atmost. The traffic usage can be different for each cell (Fig. 1.3). The basic idea is to move channels between different cells, allocate more channels to a cell so it can handle heavier traffic. The *static allocation* is a simple method. It is assumed that the set of predefined channels is assigned permanently to each cell and the traffic at each cell is constant and permanently to each cell. It is not effective, when usage conditions will change. When traffic is not according to the initial traffic patterns, there may a situation occur, with lack of channels in a given cell and many unused channels in another adjacent cell.

1.3 Basic Terms, Notion And Definition

Cell – small hexagonal shape geographical area. In structure, its position is described as row *i* and column *j* which denoted as (*i*,*j*). In Fig. 1.2 an example of that structure (model) is pictured.

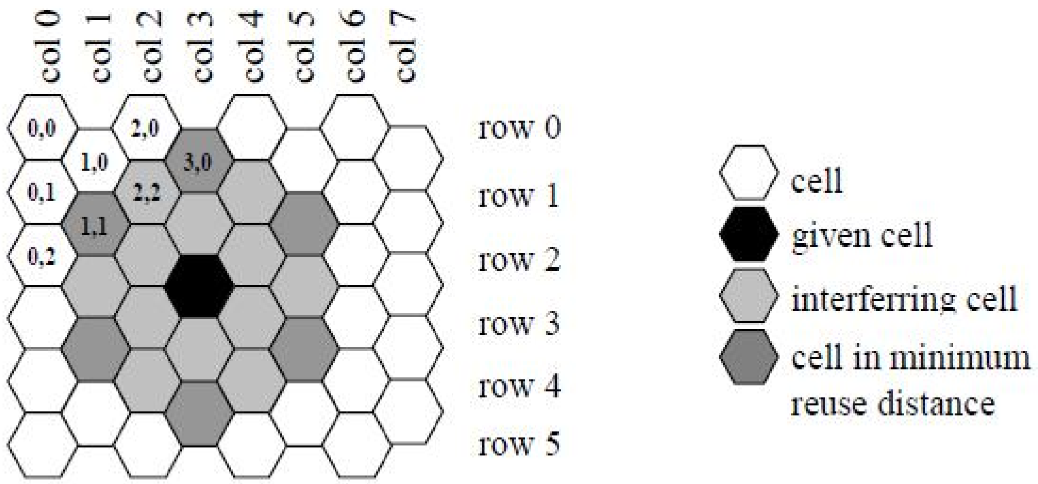


Fig 1.2: Hexagonal Structure of cellular Network

Bandwidth – divided into finite number of channels. It is the set of radio frequencies available to all cells.

Channel – At a single moment of time only one call at one will handle bya channel. It is part of bandwidth.

Distance between two cells – It is uses Euclidean distance with normalized radius of the cell (it ensures that distance between any two adjacent cells == 1).

$$dist(c_1, c_2) = [(i_1 - i_2)^2 + (i_1 - i_2)(j_1 - j_2) + (j_1 - j_2)^2]^{1/2}$$

Minimum reuse distance - minimum distance required between any two cells in network two cells such that each cell use the same channel at particular instance of time. It is symbolize as D_{min}

Co-channel cell: minimum reuse distance D_{min} from cell c.

Interference neighbourhood: the IN_c is set which contain cells in which all cells(sites) in set have distance to particular cell c less D_{min}. Any cell from set IN_c.in set cannot use same channel simultaneously as the channel used by cell c.

Request Set: $denoted\ as\ R_i$ set of cells, send grant message or permission to cell i, before cell *i* will get critical resource.

Pending Flag –*Pending flag* is set to 1 when a cell send request to all the cell in its set . This flag is cleared or set to 0 when the critical resource is acquire or revoke. During this time until flag is cleared, the cell cannot send another requests.

Inform Set denoted as I_i : set of sites, that informed by cell c before c release its critical resource.

Received Request Set given as L_i: set of requests made by the member of cells, received by cell *i*.

Usage List given as CRU_i – the cells using the resource instantly as per knowledge of cell *i*.

Status Set given Si : $J \in S_i\ Si\ \ iff\ i \in I_J$

Traffic in cell – determines no. of channels should be available in cell, to avoid occurrence of insufficiency of channels. Fig.3 describe this feature

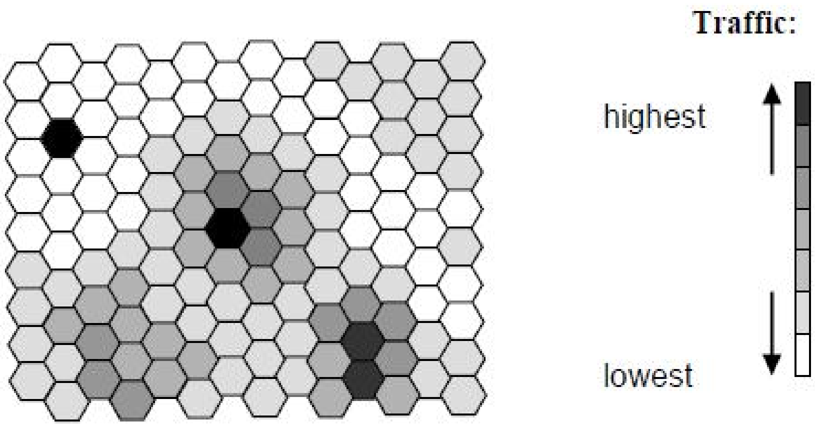


Fig 1.3 : Hexagonal Network of Cell – Traffic

1.4 Motivation

Mutual exclusion in distributed systems has received very consideration of scholars and researchers during last decade. Various algorithms are proposed to solve mutual exclusion in distributed systems. In distributed mutual exclusion algorithms, the site sending the request check every site in the system if site requesting critical section or not. These algorithms are not efficient as sites which not require CS are also receives request messages .hence it will take more time and compiler become too laggy. Such algorithms are inefficient for mobile device which use low frequency in cellular networks.

Here relaxed mutual exclusion problem(RME) has been explained. It is a generalized form of mutual exclusion. It execute the channel allocation problem finely. Mutual Exclusion is a also specialization of channel allocation because mutual exclusion may be further derived to channel allocation. Channel allocation is also one application of relaxed mutual exclusion. The algorithm which is proposed for relaxed mutual exclusion problem will also work for mutual exclusion problem. So, an effective way to design DRME (distributed relaxed mutual exclusion) algorithm can be designed such effectively that it start with existing or new distributed algorithm and generalize them. Here we have distributed mutual exclusion algorithm to generate a generalized algorithm.

Distributed algorithms are categorized in two classes:

a) Token-based: Token-based algorithms are not extensible for solving
b) Non-token-based: As non-token-based algorithms, Sanders algorithm is considered and all existing non token based algorithms are derived from this algorithm.

Other solution given for solving the relaxed mutual exclusion problem is a *centralized allocation*. It is proposed, that a cellular network have one central management system, which divide channels to different cells. This strategy can be very effective, but these kind Hexagonal structures have high centralized traffic overhead, and sensitive for failure and lead to deadlocks . The effective way is *dynamic channel allocation*. Advantages of Distributed Dynamic Channel Allocation algorithms are lying in good efficiency of channel allocation and their high speed. The GRME-algorithm deals with multiple channels and channel selection strategies. Here, improvements of algorithm ensures increased efficiency, avoiding deadlocks and broken cells serving mechanism .

1.5 Research Objective

The communication at cellular network is limited to a few messages such as three messages: a) request a token b) received the token and 3) release the token respectively. Since the mutual exclusion will run by the base stations , due to consumption of mobile hosts ,the CPU time become low. The protocols become more efficient with respect to energy. The protocol must be free from all needed data structures and the number of mobile hosts which can easily managed by the base stations. Therefore, protocol become scalable and cannot be affected by mobile host failures.

Other interesting features needed by the protocol to be satisfied are as follows.

- At the time of mutual exclusion execution, a base station must watch every mobile host within its cell to manage the request messages and the token.
- In mobile network, a handoff algorithm needed to perform mutual exclusion efficiently and correctly.
- Amount of computation calculated by mobile host should be kept low.
- Communication overhead in wireless medium should be minimal.
- Algorithm must be scalable with respect to the number of mobile hosts.
- Algorithm must easily handle the effect of mobile host connections and disconnections

The mutual exclusion algorithm in cellular network execute in two different phases.

Phase I: local mutual exclusion execution phase in which a mobile host execute the CS.

Phase II: global mutual exclusion execution phase in which each base station takes part in the mutual exclusion by sending message to another *MSS*.

1.6 Thesis Organization

We start this discussion with introduction in chapter 1. Chapter 2 clarifies the problems and various issues regarding relaxed mutual exclusion algorithm and describe their relationships with each other. Chapter 3 describes general relaxed mutual exclusion algorithm for single resource (channel), which contains information structure needed to designed the algorithm and an algorithm, the sufficient and necessary conditions for a information structure and proved considering various cases. It also contains improvement of GRME algorithm having two solutions; deadlock minimization and broken cell serving mechanism. Chapter 4 presents the proposed generalized relaxed mutual exclusion algorithm to deal with allocating more than one channels. Chapter 5 describe about channel selection strategies in distributed channel allocation, which is analysis and simulation is carried out. Chapter 6 provide conclusion the thesis and provide future wok

CHAPTER 2

REQUIRED LITERATURE REVIEW

2.1. Information Structure

Information Structure described as to explain different distributed mutual exclusion algorithms for systems. The information structures explain the state information about any other processes. A process must request permission before execution and enters the critical section. Distributed mutual exclusion algorithm always follows same sequence of steps to execute mutual exclusion without considering state of the system because of static nature of these information structures. Therefore, these algorithms cannot get any advantages of the dynamic system.

Here, dynamic information structure exist with time in mutual exclusion algorithm used in distributed system. The information structure of a system must be continuously update by state information.

Dynamic information structures in relaxed mutual exclusion algorithms are interesting features and have scope for further study too. They can adapt with fluctuating conditions of mobile network to increase the efficiency, optimize the performance and help in decision-making problems in mobile network. For designing the dynamic information structure for distributed mutual exclusion algorithms various initial information structures design are used. It must follow the rules for change occurring in the information structures so that mutual exclusion conditions always satisfied.

2.1.1 Information Structure in Algorithm

An information structure are maintains by every site. It considers various sites to send request messages for executing mutual exclusion in cellular network and after the execution of critical section the sites will send reply messages**.** The information structure change accordingly in the site with the state of the system as the site will get respond messages from other sites.

2.1.2 Data Structure at a Site

Information structure of a site Si classified as two groups sets.

I Set: request set denoted by R. It contains the sites from which Si should get permission grant before executing critical section.

II Set: inform set denoted as I. It contains sites to which site Si should send its grant message to execute critical section after executing its own critical section.

When site has finished execution of critical section, it will send permission. As algorithm uses inform set(I) which is changing dynamically results in good presentation dynamic information structure. Using Lamport's rule , every site Si in request set maintains a logical clock C, which will update according to time. Every request assigned with a timestamp for critical section execution current time in the clock, to order the requests. There are two boolean variables for each site - Request and Execute to the state of the site. If the site being requesting or executing CS, these variable become true.

2.1.3 Types of Information Set

Information structure contains two types of sets : *Request Set* and *Inform Set.* For a given cluster of cells, we can specify co-channel cells, if minimum reuse distance D_{min} is known. Each cell has six closest co-channel cells except those cells which are on boundary of mesh, each of them can be specified as follows: Let us consider *c(0,0)* be the cell specified, and let *d(p, q)* be the co-channel cell. Co-channel cell of channel $c_i$can be find as follows:

(i) start from cell *c and* travel *p* cells along,

(ii) move anti-clockwise 60 degrees,

(iii) travel *q* cells along the new direction.

Co-channel cells to cell *i* are considered as a *co-channel set* G_i.

The information structure for a site i consists of three sets R_i , $I_i$and $C_i$and satisfy the given theorem .

Theorem 1 : *Let us assume* $I_i \cap R_i$ *for every site i. The sufficient and necessary condition to ensure relaxed mutual exclusion that any two conflicting sites I and J their information structure which will satisfy the following*

$$[(i \epsilon I_i \cap R_j) \Delta (i \epsilon I_j \cap R_i)] \nabla (I_i \cap I_j \neq \emptyset)$$

Any information structure must satisfy this theorem guarantees that it is correct. It reduce deadlock probability, increase given efficiency coefficients and decrease message complexity.

Some possible information structures are as follows. They satisfy the condition of Theorem1.

a) Centralized structure: $I_i = R_i = \{s\}$, where s is a particular site manage channel allocation in the network. This structure satisfies the theorem 1 condition given in as $I_i \cap I_j \neq \emptyset$ for any i, j.

b) Partially centralized structure: It is assumed that P be fixed and arbitary co-channel set of cells. The algorithms have information structure is: For i ∈ P, $I_i = R_i = \{i\}$ and for $i \notin P$, $I_i = R_i = I N_i \cap P$. This structure is un-symmetric and favours all cells in P. This structure fulfill the given Theorem 1 condition : If (i ∈ $IN_j \wedge j \in IN_j$), then there is a cell r such that i and j are both in IN_r, hence $r \in IN_i \cap IN_j$ and, thus, r ∈ $I_i \cap I_j \neq \emptyset$.

c) Trivial structure : $I_i = \{i\}$ and $R_i = IN_i$. This information structure is very simple. Deadlock probability, message complexity and acquisition delay become high for all conflicting cells in information structure. It fulfil the condition of Theorem 1 : i ∈ $I_i \cap R_i = \{i\}$ and j ∈ $I_j \cap R_i = \{j\}$ if i and j are interfering cells.

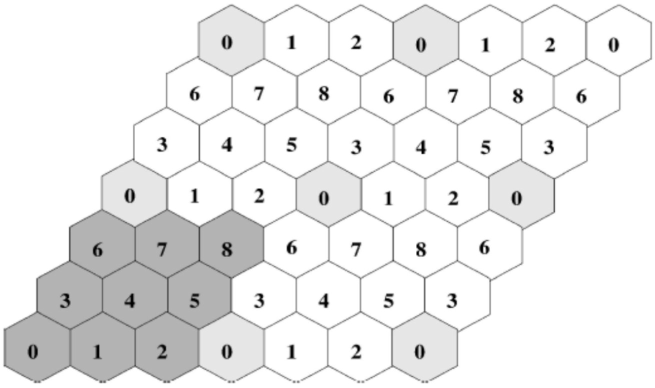


Fig 2.1: Structure of network -With $D_{min} = 3$ and $N = 9$

d) Mercedes information structure: The nearest co-channel cells, labelled as P in one of which is the center of cell. The interference neighbourhood is highlighted by light-shaded area indicates the, IN of $I\,N_i$ of i. The information structure is highlighted by dark shaded area; that is, $I_i = R_i = C_i$ = the Mercedes logo. In general, let $D_{min} = (a^2 + ab + b^2)^{1/2}$ where a and b are two positive integers with $a \geq b$. If b=0, the structure has a size of 3a-2 i.e. $I_i = R_i = C_i = 3a - 2$. In the case of b ≠ 0 the size is $3(a + b) - 2$.(Fig 2.2 shows Mercedes logo Structure)

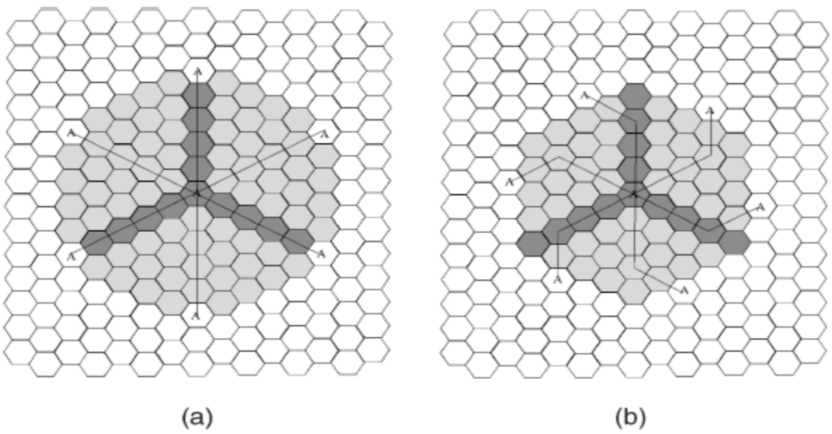


Fig 2.2: The Mercedes Information Logo a) a=5 & b=0; b)a=3 & b=2

e) Blossom structure: The interference neighbourhood of the centre P cell highlighted by light-shaded area. The dark-shaded area of information structure given as $I_i = R_i = C_i$. In general, $I\ N_i$, IN is the interference neighbourhood where i is a hexagonal area. If the radius of $I\ N_i$ is r, then the blossom information structure given by centre cell plus outline of a hexagon of radius [r/2].

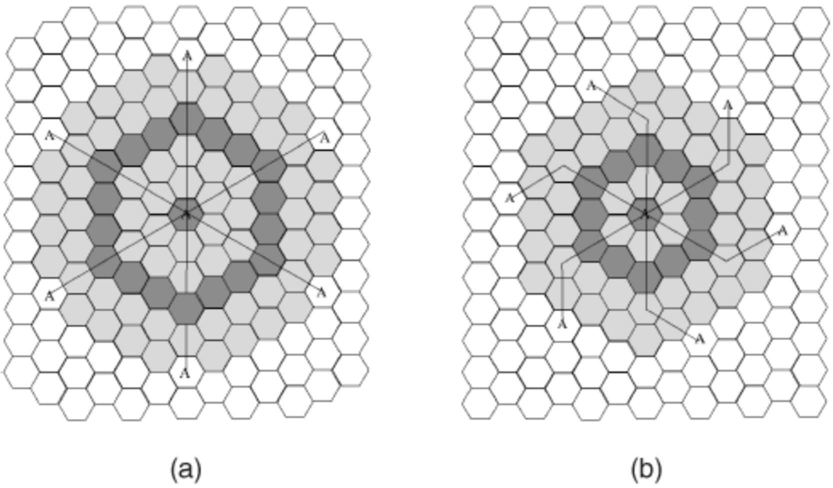


Fig 2.3: The Bolossoms Structure a) a= 5& b = 0, b) a=3 & b=2

2.2. Mutual Exclusion

The characteristics of mobile computing network such as mobility, resource constraints and communication are considered the development of dynamic distributed algorithms are much more difficult than traditional distributed systems for solving distributed control problems. A group of mobile hosts try to execute the critical section in to get profit of exclusive access to the shared (critical) resource , is considered as MUTEX problem. The solution for MUTEX problem are solvable if they must satisfy the following properties:

- *Mutual Exclusion* (safety): at one instance of time ,at most only one host allowed to enter in CS;

- *Deadlock Free* (liveness): If any host waiting for the CS, then in a given finite time for host enters the CS;
- *Starvation Free* (Fairness): If a host waiting for the CS, then in given finite time the host enters the CS.

2.3 Channel allocation

Recent development in wireless network, the mobile communication system is tightly constraint for capacity of proper utilization of channels and available frequency spectrum. in For solving the channel allocation problem (CAP), channel allocation techniques which capable of ensuring efficient channel allocation is essential. The channel allocation problem is defined as allocating the channels to each cell in cellular network. It considered as NP-hard problem. The task of a channel allocation scheme is to minimizing the probability and allocating channels to cells so that incoming calls are restricted or blocked. The channel allocation scheme should satisfy traffic requirement and electromagnetic compatibility constraints, to minimize the call blocking probability.

There are three classifications for the channel allocation problem.

1) Fixed Channel Allocation (FCA): In FCA, each cell has channels are assigned to permanently, based on its channel demand, which is already defined. This uses predetermined assignment strategies which goals to improve average case efficiency and performance. However these schemes cannot adapt with changing environment of traffic congestion. FCA also provides worst case of channel utilization. FCA is the easiest among three. FCA maximize the frequency reuse by allocating channels to each cell permanently In FCA schemes, distance between cells sharing same channels is equal to the minimum reuse distance between the cells.

2) Dynamic Channel Allocation (DCA) : In DCA, as the calls arrive the channel allocate dynamically. It have use various algorithms & also high degree of randomness. Based on the channel requests, channels are dynamically allocated to each cell. This scheme is being designed try to optimize system performance by adapt with traffic variations. It almost removes the requirement of a static and structured frequency reuse pattern and makes available all radio channels available for every call. DCA is more complex than FCA. Mapping CAP problem is to generalized the graph colouring problem, the CAP problem has proven to be NP-complete. DCA goals to design an algorithm for the base stations for exchange information so that it can acquire channels. The goals are to maximizing bandwidth utilization ,restrict interference, but also to minimizing call dropping and blocking .

3) Hybrid Channel Allocation(HCA) :As it name implies, HCA is a hybrid(combination) of FCA and DCA. HCA constitute benefits of both. In HCA,if one set of channel is allocated as DCA and the other set will allocated as FCA.

2.4 Differences between Distributed Mutual Exclusion and Mobile Computing Network

The differences of mutual exclusions between distributed systems and mobile computing networks are as follows:

• The mobile host moves to base station from which it received token, when the mobile host using the resource. In this case, the mobile host send token to its local base station. But On receiving the token, base station follows the rules to maintain the property of the mutual exclusion. When mutual exclusion protocol started, the mobile host which have token will occur in any of the cell. So, every base station should communicate with all other base stations to get the information about the token that which base station covers mobile host holding the token. It causes message congestion among base stations, this is the big difference between mobile computing environments and static distributed systems.

• A handoff algorithm will perform mutual exclusions correctly in mobile computing environment, but it is not required in distributed systems.

• The distributed algorithm solve mutual exclusion executed by the sets of *BTS* to perform the set *G_MH* of mobile hosts , due to resource constraints of mobile hosts and limited bandwidth of wireless links.

2.5 Distributed Dynamic Channel Allocation(DDCA)

In fact, our goal to reduce the DDCA problem into five small sub-problems, each have an interesting design issue for its information structure. The sub problems are which can be solved:

1. implementing RME for a single resource.
2. resolving deadlocks.
3. dealing with multiple channels.
4. designing efficient information structures.
5. implementing strategies for efficient channel selection.

The first four sub-problems deals with neighbouring cells which can not use the same channel at same time, whereas the last sub-problem deals about maximizing channel reuse.

2.6 Relaxed Mutual Exclusion

According to mobile computing system, Relaxed Mutual Exclusion (RME) is explained as follows: Mobile cellular network contains two sets. One for mobiles sites S = $\{s_1, s_2, \dots \dots \dots . s_n\}$ and another for critical resources R = $\{r_1, r_2 \dots \dots \dots \dots, r_n\}$. There must follow a rule, that resource requested by a site is specific (although other systems do not care which resource will assigned to them).

For each site S_i ,there are k_i independent and concurrent processes,both of which alternates execution which require a critical resource and computations which do not require a critical resource. In various given sites, a binary relation is derived which is symmetric and reflexive, but not transitive. Two sites x and y are supposed to be conflicting or interfering with each other iff x Ø y.Here, sites are considered as cells and critical resources are regarded as channels. Critical resources in relaxed mutual exclusion can be reused according to the following rules:

1) At same time, a critical resource can be used by different sites until unless no two of them will mutually interfering.

2) a critical resource cannot be shared by two or more processes, for any specific site.

The problem of distributed relaxed mutual exclusion is designing a protocol which dynamically assigns critical resources to different sites so as it satisfy with above rules.

2.7 Message Passing

In mobile computing system, communication between the cells performed by message passing mechanism. Each cell can send and receive a message to any other cell and response in given timestamp. FIFO queue are used for communication between cells and communication delays are arbitrary. To assign timestamps to events, Lamport's logical clock scheme is used so that events will totally ordered at different sites. Each message in communication has a different and unique timestamp. Timestamp-based priorities are used, to break ties among requests. A higher priority request has smaller timestamp. If site *i* have to respond multiple incoming messages, the message will served first with smallest timestamp and the smallest timestamp . So, it is necessary to assign the priorities of incoming and out going messages.

Sites are communicating about their current state and other condition by passing messages. Sites can communicate to each other such that every site have information about conditions in neighbouring cells. It may be distinct types of messages such as:

Request Message – when site *i* wants to acquire channel ,sent by s_i (site *i)*. This message will send all sites queued in request set R_i.

Grant Message –the response to *Request Message* given to s_i from s_j sent by site *i* .

Release Message - when site *c* releases channel or critical resource, sent by site *c*. This

message received to all cells which are queued in inform set I_i .

Reject Message – cell *i* response as *Reject message* to *j* if cell *i* could not grant a resource to cell *j* .

Revoke Message – when site *i* receives *Reject Message from* site j ,it sends *Revoke Message* to all members(site) in *Ii.*

If there is a cycle of different sites: $(s_1, t_1; s_2, t_2; \ldots \ldots \ldots \ldots \ldots s_n, t_n,)$ each *i*, site s_i *will* wait for a granting resource from t_i*+1* and received a Grant message from t_i,already then possibility of deadlock may occur. Also, each and every site in li , t_n cannot to grant request from s_i-1 because s_i-1 has conflict with s_i. The deadlock avoidance can be handled by grant and reject messages. If site receives a request message, Site has reply to grant or reject message in a limited amount of time.

2.8 Relationship Between Mutual Exclusion, Channel Allocation and Relaxed Mutual Exclusion

RME is a generalized mutual exclusion problem, It serves finely to modularize channel allocation. Mutual exclusion is a special case of Relaxed Mutual Exclusion, where there is only one critical resource, each site has only one process, and every symmetric and reflexive pair of sites (x , y). Mutual exclusion is special case of Channel Allocation, in which there is only one channel in the cellular system and the minimum reuse distance D_{min} which approximately equals to infinity.

Channel Allocation is an application of relaxed mutual exclusion. Relation between channel allocation and RME can be easily described. Let S be the set of cells (or base stations) and define x, y reflexive and symmetric if and only if the distance between site x and site y is less than the minimum reuse distance D_{min}. Allowing the same channels to be used concurrently by different cells makes a special case of RME. But ,some differences are found here. In the Channel allocation problem, a cell requesting a channel, does not care which particular channel allocated too long that it will able to use it without interference from other cells. In RME problem, when a particular site requests a critical resource, it is interested in acquiring a specific resource. Thus, in the CA problem, our concern is choosing among numerous channels that may be available, when deciding which channel to allocate to a requesting cell.

So, ME can be reduced to Channel Allocation, which in turn can be reduced to RME.

CHAPTER 3

Generalized RME for Single Resource

Our aim for this project to design a generalized algorithm which ensure relaxed mutual exclusion for a single resource. This algorithm nearly close of Sanders Algorithm for mutual exclusion in generalized manner. It includes all special cases of non-token based algorithms for mutual exclusion such as different non-token based algorithms are use different information structures. Here we describe the information structure we use for developing algorithm. Then we develop the algorithm. Then, we will prove sufficient and necessary condition that the information structure satisfy the required condition and prove that the algorithm is correct.

3.1 Information structure

The *information structure* for relaxed mutual exclusion algorithm divided into two groups (two sets)for every site i — *request set*, denoted as R_i and the *inform set*, denoted as I_i . We also require one more set called as *check set* c_i . three additional sets are also needed develop the algorithm but they are not the component of information structure. These sets are defined as follows:

a) R_i : *request set* is the set of sites sending request from which i will get permission for using a critical resource and execute the critical section.

b) I_i : *inform set* is the set of sites which site will inform about status.

c) C_i : *check set*, is a set of sites in which site i checks for critical resources. It named as check set because site i *checks* if granting request cause any confliction with the sites in C_i when some site requests to access the critical resource simultaneously.

d) S_i : *status set* site is explained as $j \in S_i \;\; iff \; i \in I_j$. It provide feasibility during the designing of algorithm.

e) Q_i : set of requesting site for critical resource received by site j.

f) CRU_i : according to i's knowledge,the sites in CRU_i using the critical resource at single instance.

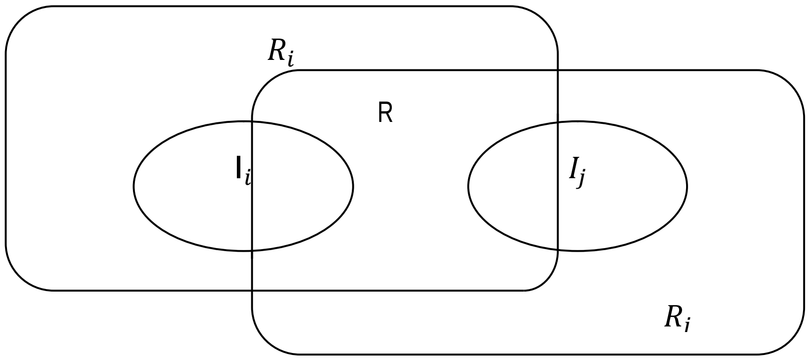


Fig 3.1: The Information Structure

3.2 GRME Algorithm

Mobile computing system consist of *n* sites, each of them implement algorithm in its own site, running in sites at same time. It constitutes one big allocation system. The algorithm denoted as GRME (Generalized Relaxed Mutual Exclusion) is based on relaxed *mutual exclusion*. Each rule describes algorithm for given condition. The rules listed below may be applied to sites in mobile system (mobile cellular network).

1. Requesting channels : let cell *c* require one or more channels, it will send *Request message* to all sites request list R_i where c will exist. Also, site *c* sets its Boolean variable true called as *Pending flag*.

2. Acquiring channels. site *c* will wait for responses after sending Request message. When it get permission as Grant(F_j) message from all cells from R_i it can start searching channel which will appropriate for it.

- The first step is calculating the set of channels $F = \cap \{F_j : j \in R_i\}$.

Channel selection strategy is used to calculate the set of channels F'. This set, F', is acquired by cell *c*. The search fails if set $F' = \emptyset$.

- In other step, site *c* will sends an *Acquire(F ') message* to every site in *Ii* and its P*ending flag* become false.

3. On receiving of Acquire message. site *c* removes cell *k* from CRU_i *(x) for every channel* $x \in F_{ij} - F'$, if site *i* receives an *Acquire(F ') Message* form site k.

4. On receiving of Request message. When cell *c* receives *Request(r) message* from site *k*, site *c* puts its request in Q_i.

5. Grant a request. granting a request for site c (say from site k) in Q_i only if following two conditions are satisfied:

(a) site c is not interfering site k; or if site c interfering site k, the critical resource is not used at current and site c has no pending request of higher priority;

(b) Any sites in CRU_i / C_i do not interference with site j.

When granting k's request, site c performs the following:

(a) Grant message is send to site k;

(b) k's request will removed from Q_i;

(c) add k to CRU_i if j ϵ S_i.

6. Releasing a channel. site *i* sends a *Release(c) message* to all members in I_i when it release a channel

3.3 Necessary and Sufficient Condition

Information, Request and check set are required to validate the algorithm. In GRME algorithm random choices of request, information, and check sets in GRME cannot give good result while correct algorithm is executed. In this section, we have sufficient and necessary condition to check information structure correctly. (A particular information structure is correct for a correct resulting algorithm.) In following theorem, two conditions should be necessary together such that if one is true then the other is necessary. Now we use this theorem in context of GRME

Theorem 1 If i $\in I_i \leq R_i$ for all site i, the two conditions together are necessary and sufficient for the algorithm to guarantee RME:

1. For two conflicting or interfering sites i, j, it must hold the following

   $(I_i \cap I_j \neq \emptyset) \nabla (i \in R_j \Delta j \in R_j)$

2. For any two conflicting and interfering sites i, j such that $i \in R_j$ or $j \in R_i$ and there exist a site $k \in I_i \cap I_j$ so that {i , j} $\in C_k$

Proof. *Sufficiency*. It is assumed that given both condition are mutually satisfied. Let i and j are two mutually conflicting sites.

Using property 1, either $I_i \cap I_j \in \emptyset$ or both i $\in R_j$ and $j \in R_i$.

Two cases conditions are considered separately.

Case 1: Suppose both i $\in R_j$ and j $\in R_i$. Two scenarios are considered. In the first scenario, both i and j will send their request simultaneously. The higher priority site do not send a Grant message unless it executed the critical resource and release it. Considering second scenario, when a site j execute the critical section after it received a Request from and sent a Grant to site i. Site i grant j's request, which has a large timestamp with lower priority than i's request, only after site i itself has released the critical resource. In both cases, i and j will never get access to the critical resource simultaneously.

Case 2: let us assume that $I_i \cap I_j \in \emptyset$;, but either i Rj or j =2 Ri. Using property 2, a site $k \in I_i \cap I_j$ exists such that $C_k\{i,j\}$. Site k will not issue a Grant message to both i and j at same time; for k sends a Grant message to, i, site i placed in CRU_k, and thus i $\in C_k \cap CRU_k$, site k will not grant message the request from j until after i releases its critical resource.

Necessity of Condition 1. It is assumed that the given condition are not true. A situation is constructed such that relaxed mutual exclusion is prohibited. There exist at least two conflicting sites i, j, the condition can be proved false, such that

$$(i \notin I_i \cap R_j \vee j \notin R_i \cup I_j) \wedge (I_i \cap I_j = \emptyset) .$$

Without losing of generality, le us assumed that $i \notin I_i \cap R_j$. Let i and j both have a request of priority(i) and priority(j), respectively. First, if all requests arrive in certain defined order at a site, they both will be granted. We create a situation in which i's and j's requests arrive at each site in the required order. Then, we consider three classes of sites: i, j, and others. Site i is considered. It is assumed $i \notin I_i \cap R_j$ such that $i \notin I_i$ or $i \notin R_j$. This case $i \notin R_j$ become trivial since j cannot acquire permission for i. If $i \notin I_i$, then i will never be in CRU_i which may grant its own request and then grant j's request (either immediately or after acquiring the resource, depending on whether priority(i) is less than or greater than priority(j)). Thus, in given scenario, we will let i's request arrived at i before j's request. Two cases are considered depending on whether j is or is not in I_i.

Case 1: $j \notin I_i$. In this case, $i \notin S_j$ So, j keep no record of grant messages of request of i. It will grant request of i and then grant it by its own.

Case 2: $j \in I_i$. here, since $I_i \cap I_j = \emptyset$, $j \notin I_j$. $j \notin S_j$ and so j can grant its own request and grant request of i (depending on if priority(i) is greater than or less than priority(j)).

Now, consider those sites $(k \in i, j)$. Obviously, if k =2 Ri \ Rj $(k \notin R_i \cap R_j)$or $(k \notin R_i \cup R_j)$, then k cannot prevent i and j from violating relaxed mutual exclusion. Thus, ore concern on those sites in

$$R = (R \cap I_i) \cup (R \cap I_j) \cup (R - I_i - I_j)$$
$$= (R_j \cap I_i) \cup (R_i \cap I_j) \cup (R - I_i - I_j)$$

The sites possibly prevent i and j from violates mutual exclusion for those in $R_i \cap I_j$ and/or those in $R_j \cap I_i$. The following condition, where t1 < t2 < t3, however, will can be fool such sites as follows:

1) At t1: sites i and j each send a request simultaneously;

2) At t2, site i's requests received at the sites in $R_i \cap I_j$, and j's requests will arrives at the sites in $R_j \cap I_i$. Every site which receives a request returns a grant immediately

3) at t3, and j's requests arrive at the sites in $R_i \cap I_j$ and site i's requests arrives at the sites in $R_j \cap I_i$. This way, both i and j will receive a grant from each site in R.

Necessity for Condition 2: Let us consider as condition 2 is false. It is required to construct a condition which results in violation of relaxed mutual exclusion. Let condition 2 is false, there exist two conflicting sites i, j such that (i) $i \notin R_j$ or $j \notin R_i$ and

(ii) $\forall k \in I_i \cap I_j :: \{i, j\} \not\subseteq c_k$ (i.e. $i \notin C_k$ or $j \notin C_k$).

Split the (nonempty) set $I_i \cap I_j$ into two disjoint subsets as follows

$A = \{k : i \in C_k$; where $k \in I_i \cap I_j\}$

$B = (I_i \cap I_j) - A \subseteq \{k : j \notin C_k$ where $k \in I_i \cap I_j\}$

Furthermore, let

$$A' = R_i - B$$

$$B' = R_i - A'$$

Note that $A \subseteq A' \subseteq R_i$, $B \subseteq B' \subseteq R_j$, $A' \cap B' = \emptyset$, and $A' \cup B' = I_i \cup I_j$.

The following situation, where t1 < t2 < _ _ _ < t6, is possible when

1) at t1: sites i , j each generates request for each other.

2) at t2 : the request from i receives at each site in A’ and the request from j arrives at
each site in B’

3) at t3 : every site in A’ sends a Granted to i, and every site in B’ send a Grant message to j

4) At t4: i’ s request will receive by each site in $A'' = R_i - A'$ and j’s request arrives $B'' = R_j - B'$

5) At t5: each site in A'' send grant message to the i’s request, and each site in B'' grants the j’s request;

6) At t6 : site i and j both get grant message from each site in their request set.

Both sites use the critical resource at simultaneously, thus it restrict relaxed mutual exclusion.

3.4 Application to Mutual Exclusion

When every two sites are mutually conflicting or interfering in mutual exclusion algorithm, the variable CRU_i contain not more than one element at single instance of time. In this condition, step 5 in GRME algorithm can be modified as:

5. Granting a Request. A site i can grant a request from site j in Q_i if CRU_i and site i has no pending request for higher priority sites. When granting j’s request, site i performs the following steps:

i. Grant message will send to site j,

ii. j's request is remove from Q_i, and

iii. let CRU_i = { j } if j ∈ S_i.

With rule E can be modified as and becomes special case of GRME and essentially equivalent to the Sander's algorithm , except that

(1) Q_i doesn't have be in a priority queue in GRME and

(2) GRME uses a variable denoted that a site has a pending request.

The main difference is using a priority queue, whereas Q_i should be a priority queue. Priority queues can avoid deadlocks. But they are not required for achieving mutual exclusion.

3.5 Deadlock Resolving and Avoidance

When there is a cycle exist within requesting sites

$$s_1 , t_1; s_2, t_2; \dots \dots \dots \dots \dots . . s_n t_n$$

such that for each i,

a) s_i Receives permission Grant from t_i and waiting for a Grant from t_{i+1}

b) t_i cannot grant the request from s_{i-1} because s_i have confliction with. s_{i-1}

Many deadlock avoidance and resolution techniques described. Most of the algorithms have large execution time and late responses for mutually exclusive resource acquisition and cannot used for real-time systems such as mobile cellular networks.

We use Reject messages which can handle the critical resource to avoiding deadlocks without any confliction. When a site received a request for critical section, it send either grant message or reject message the request in a given timestamp. Thus, the algorithm can be modified as follows:

ALGORITHM GRME_using_Grant/Reject

1-6. Same as GRME.

7. Reject a request. : If site i is unable to grant request in Q_i within a pre-defined timestamp, it will send a Reject message to the requesting site and removes the request from Q_i.

8. On receiving of Reject message: When a Reject message arrives to i, it aborts its request by clear the pending flag to false and send a Revoke message to every site in Q_i.

9. On receiving of a Revoke message: When site i receives a Revoke message from j, it

Follows the step

a. If j's request is CRU_i, send a Reject message to j and remove the request from Q_i.

b. If j ∈ CRU_i, then remove j from CRU_i.

10.Aborting a request

If site i aborts a request, it cannot make another request until it has received a response (grant or reject) from every site in R_i. If site j's request exist in Q_i, it will send a reject message to j and the request is removed from Li.

Correctness. Any site requesting the resource will receives a response (either a grant message or a reject message) from every site in its request set R then deadlock may not occur.

Performance. The critical concern is response time. The advantage of Grant and Reject scheme is short response time—the requesting sites knows easily and if it can use the critical resource or not. Time is critical concern which is very important property in real time application such as distributed channel allocation. There are many algorithms which resolve possible deadlock but response time is very large or not defined. This scheme does not ensure progress among the sites. It is also possible that no site may obtain the resource even though it is available. In worst case situation, all sites have reject messages again send requests and the same situation occurs time to time. One solution can be derived from, slotted-ALOHA protocol, each rejected site will back off with a random large span of time and attempt another request.

3.6 Improvement in GRME Algorithm

GRME-algorithm cannot avoid deadlocks .When all sites in the system want to acquire the resource in the same time and could not send grant messages to other sites, deadlock occurs. When a timestamp for response is used, all sites from request set attempts to gain critical resource in single moment of time again, situation repeats itself. GRME algorithm resolves deadlocks – but only when the deadlock occurs, the resource is not acquired by any site(search for critical resource become incomplete).

3.6.1. DM MECHANISM

In order to improving GRME algorithm with respect to deadlock resolution, a DM (*Deadlock Minimization*) mechanism is described here. When the deadlock occurs, all cells from cycle waiting for *grant message* to acquire the critical resource. GRME can resolves deadlocks by using *reject messages*, cell revokes and try again. The cell is always requesting a channel with random delay then the DM mechanism causes. It minimizes the probability, that deadlock repeat for the same cycle of sites. To observe the influence of deadlock minimization, some simulation experiments were carried out.

Experimentation systems keep count on repeat of forcing deadlocks. It was observed (see Table 1), that deadlocks will not happen again when GRME using DM was used. In contrast, performance of GRME without DM had large number of deadlocks repeating themselves, because each cell processed to achieve a critical resource and execute it at instance of time, after clearing Pending flag. Simulations were conducted in both deterministic and probabilistic data. The random propagation delay has implemented to make the simulation more efficient and accurate .As Results viewed, deadlocks do not occur again, but it requires further investigations and still under researches. Also, searching for critical resources is completed indirectly. Simulation is deterministic, so most of cells requests resource in first time after the P*ending flag* is cleared.

	50 forced deadlock
GRME using DM	0
GRME	41

Table 1:Comparison GRME with DM and GRME

3.6.2. BCS MODIFICATION

Another factor of improving efficiency of GRME consist is applying BCS *(Broken Cells Serving)* mechanism. GRME algorithm will not handle the situation, where one of cells is damaged and cannot send reject or grant message to other sites. Then, a deadlock may occur and it become unresolved, because cell *i* will wait for response from cell *j*, when cell *j* was broken and cannot not send any messages to cell *i*. The time-out mechanism is described. When GRME using BCS is used, and then cell *i* sends a request, timestamp being assigned, and cell waits for response until timer exceeds zero. Next, for all sites from R_i from whose cell *i* did not received response, should ask neighbours of unused cell about channels. Asked neighbours are outside of R_i. This mechanism cannot introduce change information structure, so when broken cell can be reused again, system will not need any modifications to introduce repaired cell to system again.

Simulation experiments were carried out to examine the results of broken cells on the number of acquired channels. We consider two cases :

(i) one cell broken,

(ii) two cells broken.

	1 cell Broken	2 cell Broken
GRME using BCS	24% request successful	12% request successful
GRME	Requested cell wait	Deadlock occur

Table 2: Comparison between GRME using BCS and GRME

Here we observed, that for GRME using BCS, a cell is able to gain a resource even if some other cells were broken. The table show ,that BCS (*Broken Cells Serving*) is promising and does not affect on overall efficiency. The whole channel acquiring process will longer than for GRME without BCS, because of waiting for timeout and repeating request so it is necessary to use BCS. But successful requests can compensate it.

3.7 Proposed Work -General Algorithm using Two Resources

This algorithm is very similar to GRME. The system has two resources to execute the critical section of three different sites. These sites may n in numbers but here we consider only two resources. Three sites are trying to communicate with both of resources to execute their critical section. Now according to GRME algorithm, we start executing critical section over sites using mutual exclusion. When sites start executing their critical section, they send request to both of the resources and wait for the response from both of the resources R. Now resources are granting if they are not busy. If any of the resources is busy, it will send reject message that it cannot be grant request. Now sites are ready to communicate over the network. Site 1 send request to R1. Now s1 is busy with r1. Now site s2 will sending request r1 .But r1 is busy with s1. Now request redirect to r2. Now r2 will execute the critical section of s2. Now if site s3 want to communicate with r1 or r2 but it fails. Because network have only two resources to execute critical section.

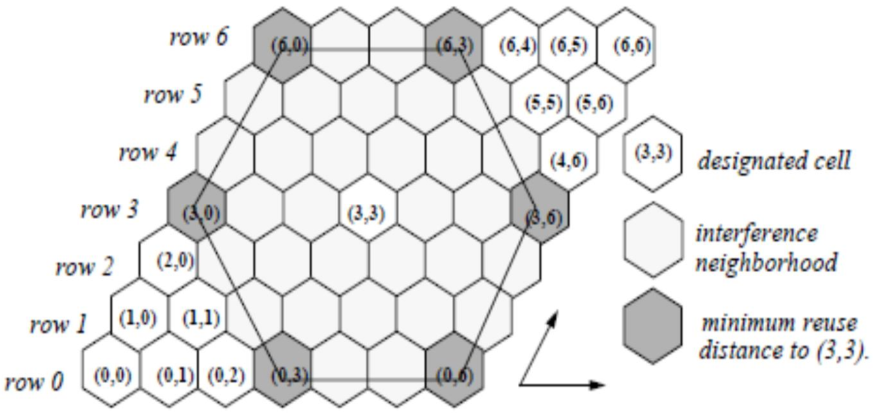


Fig 3.2 : Hexagonal network in GRME

Theorem 1 describes a sufficient and necessary conditions for right information structure. Here, we design efficient information structure for this algorithm. Information structure contains similar set as in GRME for single resources such as *Request Set* and *Inform Set.* If D_{mini} is known for a given grid of cells, we can specify co-channel cells. Every cell have six neighbouring co-channel cells, each of them can be specified as *c(0,0)* be the given cell, and *d(a,b)* be the co-channel cell. To find co-channel cell of channel c,we follow the steps: move *a* cells along from cell c, turn anti-clockwise 60 degrees, and move *b* cells along with new direction. Set of Co-channel cells to cell *i* is denoted as *co-channel set* G_i. if there is a sequence of cells $(c_1 = x, c_2, c_3, c_4 \ldots \ldots .. c_k = y)$ let two cells x and y, x is said to be a co-channel cell of y. Such that ci is a nearest co-channel cell of c_{i+1} $1 < i < i - 1$.

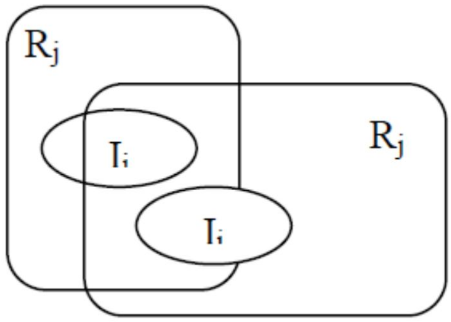


Fig 3.3 Information structure used in proposed algorithm

The Proposed Algorithm

As described above, we have three sites and two resources. All sites will try to communicate with both resources but at a time it can communicate only with one site at time if it executing its critical section. If anyone of the site want to communicate at that time we can start another server. If second server is not respond then deadlock occur. The algorithm can be modified as in some steps like this:

1) Sending Request: Send Request to request list R. All of the sites will wait until response from request list will not receive.
2) Grant resource: Now the resources in request list will grant message to site which is sending the request. Resource which is not executing the critical section will response to the request.
3) Sending response: Now the resource is free to communicate over the sites.
4) Reject Message: If resource is busy it will reject message to site and divert to another site.

The pseudo can be rewritten as :

Step 1: Start both resources and get ready to execute the critical section.

Step 2: Site 1 trying to execute critical Section. Resource R1 grant permission for critical section

Step 3: Site 2 also connect with resource R1. R1 also communicate the site 2 if it is not executing the critical section.

Step 4: If resource R1 executing the critical section then resource r2 will execute the critical section of site 2.

Step 5: If any other site communicate with both of the resource if they executing . they will send reject message.

1) Resource R1: ready

2) Resource R2: ready

3) Site S1 : execute critical section. Send Request message to Request list.

4) If (Ri of Resource list : busy)

Send reject message. Pass token t to Ri+1

Else

Send response to site requesting break the loop. Execute the critical section

5) Site S2: execute critical section. Send request message to Request list.

6) Repeat Step 4;

Simulation and analysis

Here we see that when two resources are available to execute the system is easy to handle all requests in request list. It also take less time in executing the critical section. No deadlocks may occur if any site requesting critical resource receives a response from every site within given timestamp in its request set. The advantage of this scheme required less response time which is particularly very useful in real-time application. There are many deadlock algorithms described that executed in conjunction with the GRME algorithm if response time is not a critical concern. In contrast, performance of GRME using two resources had large number of deadlocks repeating themselves, because each cell was processed to gain a critical resource and execute it immediately, after cleared Pending flag. Simulations were conducted in both deterministic and probabilistic data. The random delay had been implemented to make the simulation more accurate and efficient (in real systems, delay derived from signal propagation and cell internal properties).

As Results showed, deadlocks did not occur again, but this requires further investigations and is still under research. Also, searching for critical resources is completed indirectly. Simulation is deterministic, so most of cells requests resource in first time tick after P*ending flag* is cleared.

The possibility of deadlock become high but can be easily handle by using various mechanism. If time is not essential factor then deadlock can be easily resolved. Further research can be find out the way executing the critical section at a time. Results can be described as follows

	Time taken to execute critical section of two sites
GRME using single resource	10 sec
GRME using two resource	5 sec

Table 3 : Execution time of sites

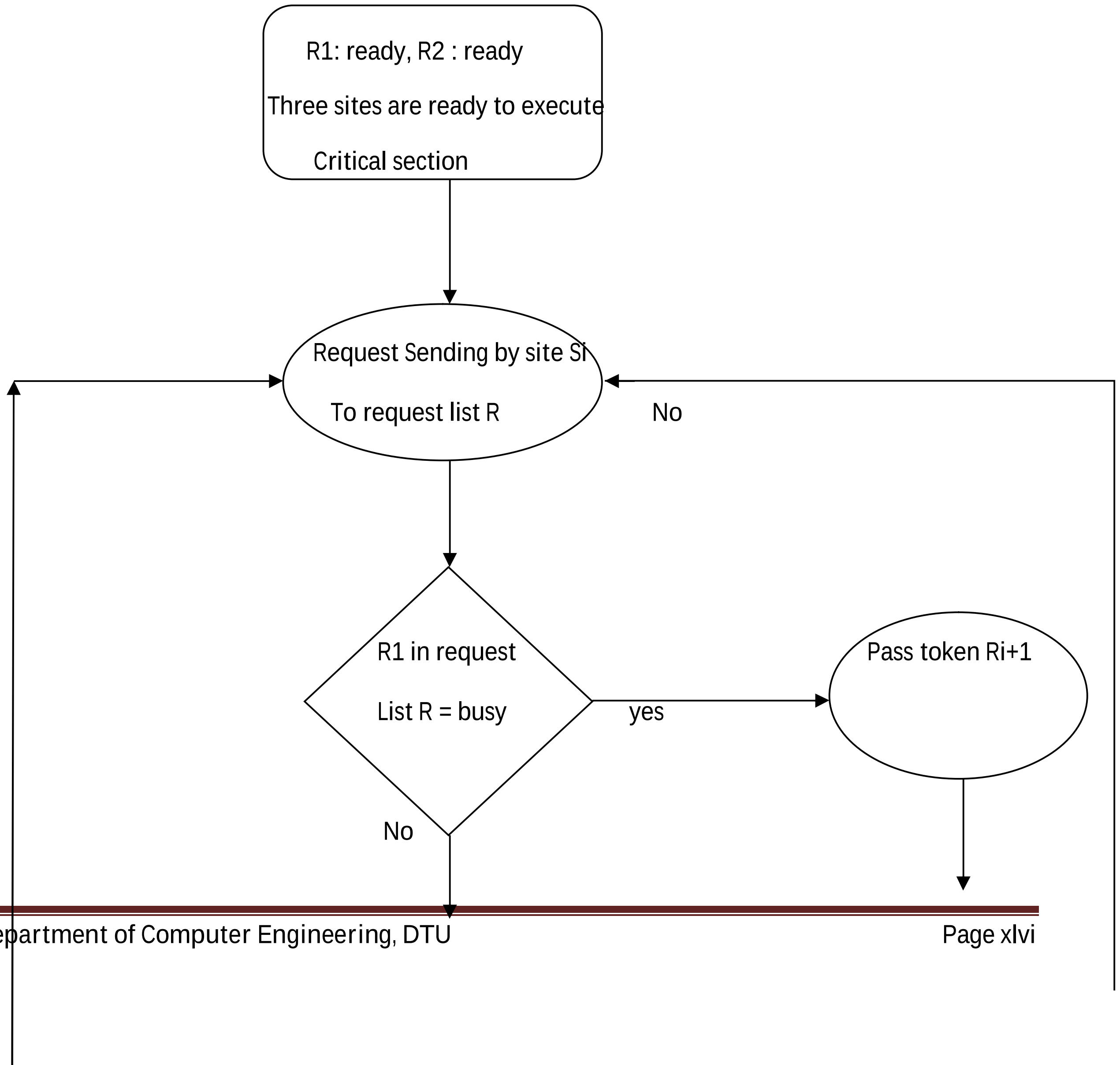
Here we can see that execution time is almost reduced to 50%. Less the execution time, more the efficient is resource. In Mobile communicating system, sites are randomly using resources bind the networks. Less time provide large serving capacity of sites so that network not become too much loaded during communication

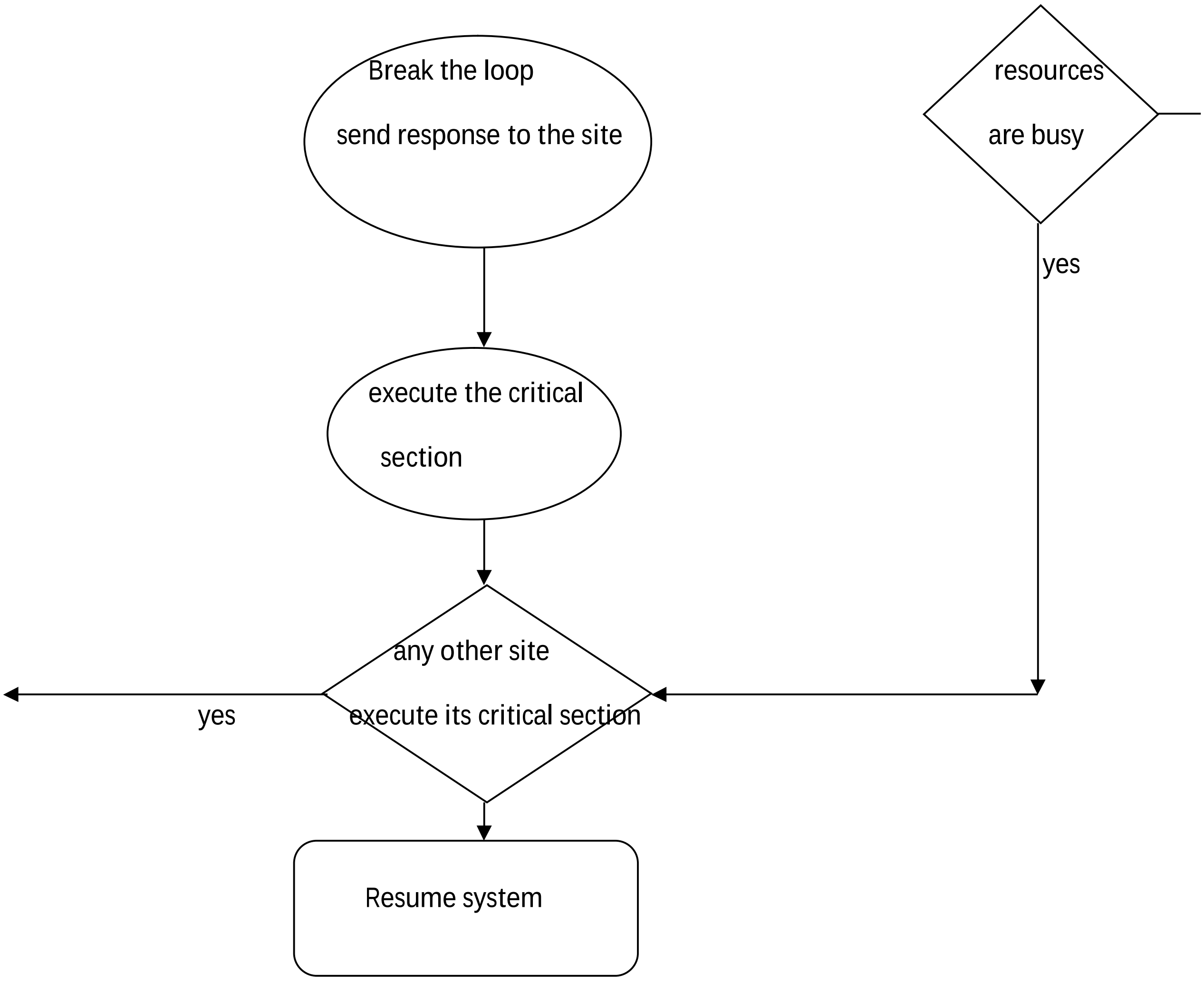
	No. of deadlocks
GRME using single resource	20
GRME using two resource	8

Table 4 : Deadlock occurs at single run of algorithm GRME using two resources

If a set of sites send requests for a critical resource at same time and a deadlock will occur, the use of this algorithm resolves the deadlock, but all requests are rejected and no site acquires the resource even if it is not executing any critical section . In the worst scenario, all rejected sites send requests again and the same situation repeats. An interesting problem can be design a deadlock resolution and avoidance scheme having progress property for future.

Here we need to see the flow chart that how it is using the algorithm to reduce the time in GRME _using_ two resource




Break the loop
send response to the site
resources
are busy
yes
execute the critical
section
any other site
execute its critical section
yes
Resume system

CHAPTER 4

Dealing with Channel Allocation And Multiple Resources

The GRME algorithm considers only one critical resource. Here we deal with multiple resources, required for a distributed channel allocation. Let

$$CR = \{r_1, r_2 \dots \dots \dots .. r_n\}$$

be the set of critical resources. In this section three schemes are described for multiple resources. The following algorithms can be derived as channel allocation schemes by replacing critical resources with channels and sites with cells.

4.1 Sequential Search

Let us assumed any algorithm Algorithm(r) for the single critical resource in relaxed mutual exclusion problem, where r be the critical resource. The sequential search execute as follows: when a site acquire a critical resource, it send message to critical resource r ∈ CR and uses Algorithm(r); if request fails, it tries another critical resource r' ∈ CR and repeat the Algorithm(r'); it works until a resource is acquired or all resources in CR have been searched.

Here two results are found for the given algorithm.

a) The single resource algorithm must have property that within a timestamp every request for critical resource either successful or fails.

b) A site use distinct information structures for various different resources. Thus, number of times Algorithm(r) executes, different information structure (i.e. I_i, R_i, C_i) are required. In application such as channel allocation, this can improve the system performance.

A site i can derive the resource availability from CRU_i for a given information structure. Depends on the information structure used, the information contained in CRU_i may be different at that site. An assumption is made that CRU_i cover basic part of the resource usage information. The long execution delay occur in resource usage or resource availability information. Thus a site has neither complete nor updated information about usage or its availability of resources.

Sometimes, the sequential search has searched two or more resources before a successful search executed. The execution delay for each search may be short as different sites can request for different resources and probability of collision between sites become low. However, if two or more searches have to be executed, the acquisition delay of sequential search may be quite long. Similarly, the message overhead could be heavy if many searches are carried out simultaneously. For efficient sequential search, site must have complete, precise and more accurate information about the resources.

4.2. Parallel Search

The basic concept behind parallel search is that when a site wants to acquire a critical resource, it executes Algorithm(r) same time for all r ∈ CR. There is n copies of the Algorithm(r) are executed in parallel, where n is the number of critical resources. However, execution of n independent copies of the algorithms is inefficient. And one design issue also raised here that minimizing the message overhead and execute n copies of the algorithm collectively, besides independently.

Parallel search, which searches the usage information and availability of all resources, leads to long execution delay.

1) Collecting information about many resources may take large time

2) Probability of collisions in interfering sites is higher as the availability of all resources is sought by each site, resolving collisions requires considerable time.

When interfering sites executing parallel searches concurrently, one site have to wait for another site to finish its execution before it can further processed, or abort its resource searching. One advantage of the parallel search is reduced message overhead and message complexity, yet the size of some messages is larger.

4.3 Hybrid Search

Hybrid search is a combination of both parallel and sequential search. Distributed channel allocations have large number of requirements on channel acquisition delay. For example, a call using handoff algorithm require a new channel in short timestamp. If channel is not found within timestamp, the call will blocked or dropped. A cell may also request more than two channels at a given timestamp. For example, a call required different frequency multiple channels in future third generation multimedia data such as images, videos, audios and docs in mobile network. As reference to previous section, parallel search allows a cell to acquire more than one channel at a time and provide suitable channels such as multimedia applications. The disadvantage of parallel search that when a cell receives a request, it can grant too many channels. When a cell wants to acquire p channels, it requests for q channels ($q \geq p$). The proper value of q is chosen so that after one loop of searching, m or more channels can be acquired by the site. On the other hand, the value of l should be as small as possible so as to reduce the probability of collision among conflicting cells. When l is chosen correctly, both the acquisition delay and the message complexity can be kept low. Exchanging resource usage information can further improve the performance of the algorithm.

CHAPTER 5

DISTRIBUTED CHANNEL ALLOCATION SCHEMES AND ANALYSIS

In this section, we will see how to integrate channel selection schemes (including channel usage information collection) with channel acquisition algorithms, consider how the information structure can affect the performance of a scheme, and discuss ways to use different search algorithms to tackle different requirements of calls (such as handoff calls).

Channel selection strategy is an important issue in both centralized and distributed channel allocation. Now the question raised, for given a set of available channels, which channel should be acquired? Efficient channel selection strategies results in near-optimal reuse of channels between cells. Many previously proposed channel selection strategies were studied for the centralized system. They assumed that every cell have complete and updated knowledge of the channel usage information in a cell's neighbourhood. The selection strategy based on priority, i.e. each channel is associated with an acquisition priority in each cell. When a channel is needed, the potentially available channel with the highest acquisition priority is selected. To maximize channel reuse, the problem of channel selection becomes one of assigning priorities to channels. Here we can use an example, the two-step dynamic-priority (TSDP) channel selection strategy as considered. For any given cell i, channels are classified as primary or secondary based on the degree of reuse of the channel in i's nearest co-channel cells. The primary channels are those that are used by at least λ nearest cochannel cells, where λ $(1 \leq \lambda \leq 7)$ is a constant. Primary channels have higher acquisition priority than secondary channels. The acquisition priority of primary channel become higher if more co-channel cells used it. The acquisition priority of a secondary channel is higher if its cost is lower, i.e., the use of the channel by i will make the channel unavailable to fewer "new" cells ("new" in the sense that the channel is currently available, but will become unavailable if i uses it). The aim behind this scheme is to maximize the use of primary channels among co-channel cells while reusing secondary channels in a compact way. The primary disadvantage of this scheme is the high overhead in calculating channel priorities; it requires channel usage information of all cells in i's covers interference neighbourhood, $ONI_i = \cap (IN_x : x \in IN_i)$.

5.1 Channel Selection Algorithm

5.1.1 TSDP with Mercedes-Logo Information Structure

We use GRME_using_Grant/Reject as the basis of the algorithm. Here we use the Mercedes-logo structure as information structure. For ease of presentation, let us assume

a = 2 and b = 1 in the formula$D_{min} = \sqrt{(a^2 + ab + b^2)}$. The Mercedes-logo information structure thus consists of seven cells.

We choose sequential search for demonstration. A scheme is used to decide which channels are available and, if there is more than one such channel, a policy is needed for selecting one of them. The aim for scheme is to collect channel usage information and the other one, a channel selection strategy. For collecting channel usage information, Cell i wants to know that which channels are available for its use. A simple way is to ask the cells in IN_i which channels they are currently using in mobile cellular network. This will require $2|IN_i|$ messages. Here message efficient method is derived. Recall that each cell j maintains for each channel r a set $CRU_j(r) \subseteq S_j$ such that it represents j's knowledge of which is possibly using channel r. For the Mercedes-logo information structure, the set $\cup (S_j : j \in I_i = R_i)$ contains IN_i. Thus, if every cell j $\in R_i$ sends its CRU_i(r) for every channel r to cell i, then i will know which channels are potentially available — channel r is potentially available for cell i if $\cup\{CRU_j : j \in R_i\}$ has no intersection with IN_i. Based on this idea, we can follow the given scheme for collecting channel usage information: Whenever cell i has to send a message to cell j (where $j \in I_i = R_i$), attach to the message the set $CRU_i(r)$ for every channel r. These sets are small; the Mercedes-logo structure, each $CRU_i(r)$ contains not more than three elements. The information needed for TSDP—namely, the channel usage information at the

nearest co-channel cells—is available, since the channel information coverage.

If request for a channel fails, the cell has to search for another channel. For handoff calls, most desirable is to do one request and succeed. We can use hybrid search also where cell request/search for several channels at one time. The number of channels to search is a design parameter.

5.1.2 TSDP with Trivial Information Structure

The Basic Update Scheme is an instance of GRME_using_Grant/Reject using the sequential search and trivial information structure. It can work with any selection strategy. The TSDP strategy has an interesting and useful *wave* property: if a cell c acquires a primary carrier r becomes primary in some $LOP(c)$ cells. Fig. 4 shows the best case for λ= 1, 2, 3, respectively. In Fig. 4(a) where λ = 1, r becomes primary in all cells in LOP(c) after c acquires the primary carrier r. In Fig. 4(b) where λ = 2, r becomes primary in four cells in LOP(c) (e.g. c_0, c_1, c_2, c_5). In Fig. 4(c) where λ = 3, r becomes primary in three cells in LOP(c) (e.g. c_1, c_3, c_5)

5.1.3 Distributed Borrowing Scheme

In the distributed channel borrowing scheme, each cell i is reallocated a number of channels. These channels are the cell's primary channels. The other channels (which are reallocated to other cells in IN_i) are secondary channels for i. When i wishes to acquire a primary channel, it does not send any request to other cells and, instead, it consults only itself. But, when i wishes to acquire a secondary channel, it submit first queries all cells in IN_i to find out which channels can be borrowed, a channel selected, and borrowing-request was sent to those cells (typically 3 or 4 such cells)which has own the selected channel. This, algorithm can be regarded as an instance of GRME_using_Grant/Reject, where the sequential search and the partially centralized information structure are used and where a cell sends a query to every cell in IN_i to collect channel usage information.

5.2 Analysis and Simulation Results

There are many parameters to evaluate the performance of a channel allocation scheme: call blocking/dropping rate, bandwidth utilization, message complexity, and channel acquisition delay. Call blocking rate and call dropping rate together constitute the call failure rate. Bandwidth utilization is defined as the percentage of system bandwidth capacity used for communication and transmitting useful user packets. Message complexity refers to the

number of messages exchanged for each channel acquisition and release while channel acquisition delay is to measuring the average time for a cell to acquire a channel. In this

section, performance evaluation of the three schemes described in section 5.2.

Let us consider the Mercedes-logo information structure consists of IN_i = 3d cells. Assume the message transmission delay between any two cells to be T. The message complexity and channel acquisition delay can be analyzed as follows:

a)**Distributed Borrowing Scheme**— It is assumed that it takes only one search to acquire a channel; this assumption is realistic because the cell in search for a channel attempts to collect the most-recent channel usage information before proceeding a search. The message complexity and acquisition delay for acquiring a primary channel are both zero. The message complexity for acquiring and releasing a secondary channel is $3d^2 + 3d^2 + 3 + 3 + 3$ or $6d^2 + 9$, and the channel acquisition delay is 4T. If the percentage of secondary channels in the acquired channels is ps, then the average message
complexity and channel acquisition delay are
ps. $6d^2 + 9$. and 4T, respectively.

b)**TSDP with Trivial Information Structure**— It is assumed that on an average, it takes n + 1 searches to successfully acquire a channel, where n+0. The message complexity for each channel acquisition and release is (n+1)* $(3d^2 + 3d^2)$ or (6n + 9)d, and the acquisition
delay is about (2n+1)T.

c)**TSDP with Mercedes-logo Information Structure**— It is assumed that on average it takes n+1 searches to successfully acquire a channel. Then, the message complexity for each channel acquisition and release is (n + 1) * (3d + 3d) + 3d or (6n + 9)d
and the acquisition delay is 2(n + 1)T.

The value of n lies between 0 and 1 in above analysis. This is because after an unsuccessful search, a cell would have collected the most-recent channel usage information that a second search would succeed with a high probability. Especially the presence of congested commu nication links between cells in the simulations and high call-arrival rate, this could severely invalidate any of the other advantages enjoyed by TSDP-trivial. The channel borrowing scheme is shown to have the least overhead; however, it has the highest call failure rate and the lowest bandwidth utilization.

TSDP-Mercedes has a low call failure rate and high bandwidth utilization very close to TSDP-trivial. It has a relatively low message complexity. Although the channel acquisition delay is higher due to possible failed searches, it is not a concern for new arrived calls. Both TSDP-Mercedes and TSDP-trivial use the same channel selection strategy as TSDP. But, in TSDP-Mercedes, a cell does not have the channel usage information of all cells in its OIN. This simulation result shows that this is not critical for channel selection. Overall, the scheme described, TSDP-Mercedes is more efficient than the other two providing high bandwidth efficiency and low overhead.

Longer acquisition delay is not a big concern for new calls, because they are more tolerable to connection delays but critical to handoff calls. We simulated hybrid search for handoff calls with two channels being searched for each hybrid search. Fig. 11 shows the percentage rate of failed hybrid searches. The rate is significantly reduced and the average channel acquisition delay is reduced to a bit longer than 2T, comparing with the failure rate of sequential searches.

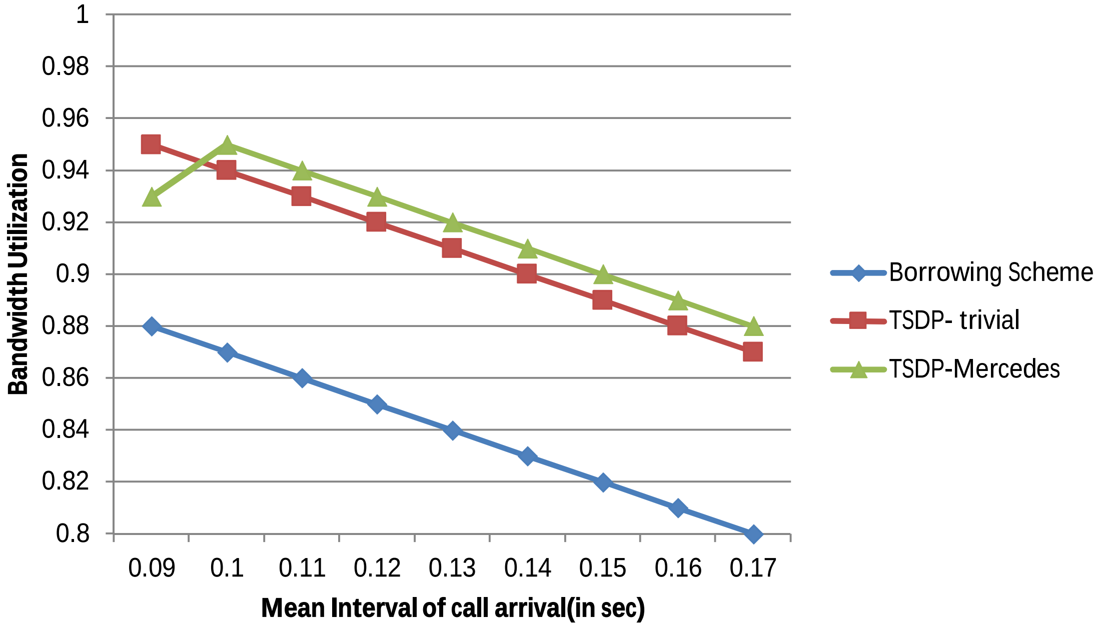


Fig 5.1: Channel Utilization

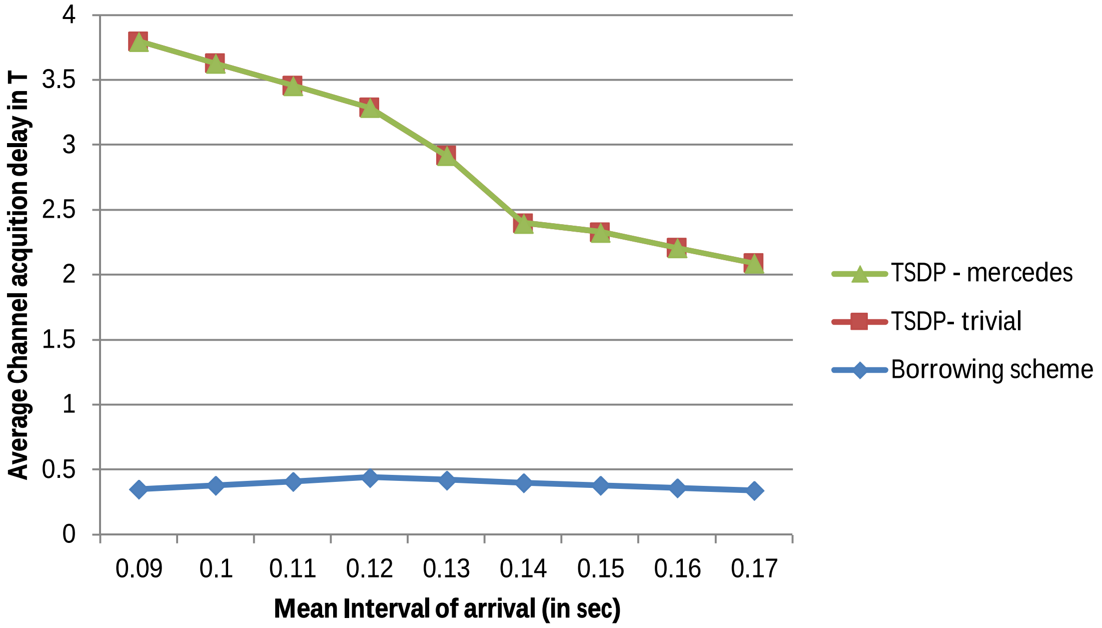


Fig 5.2: Channel acquisition delay

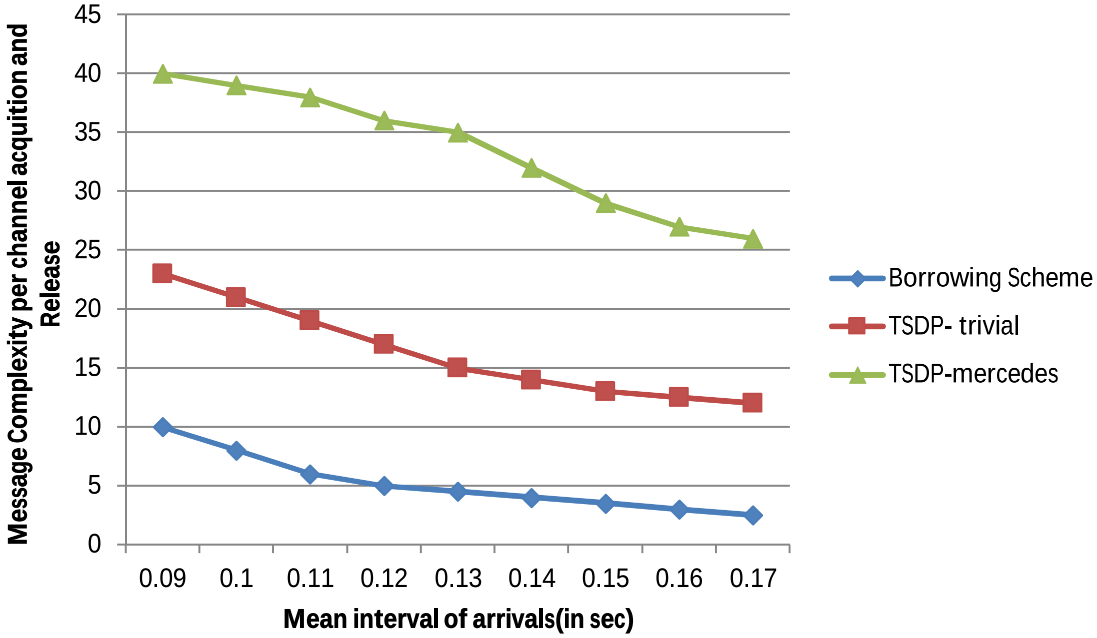


Fig 5.3: Message Complexity rate

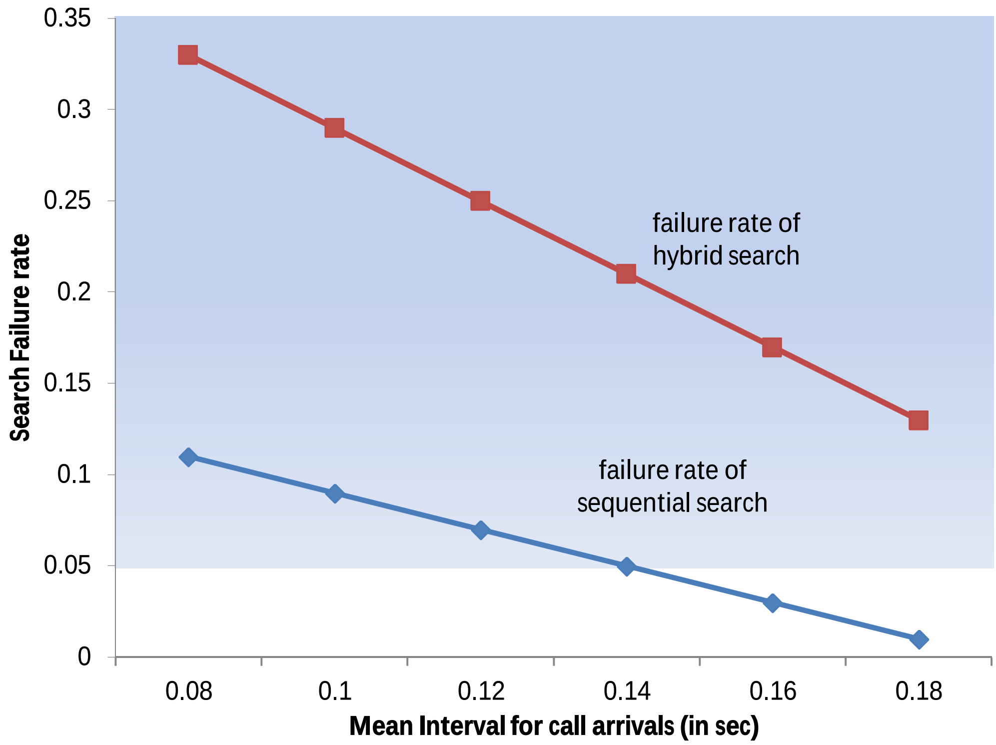


Fig 5.4: Hybrid search failure

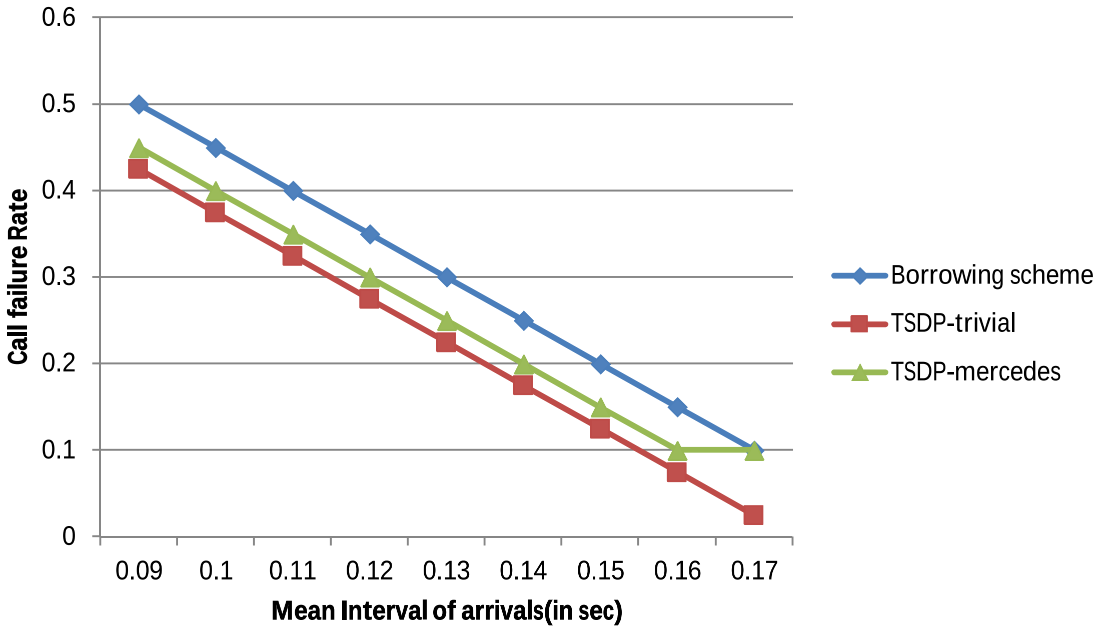


Fig 5.5 : Call failure rate

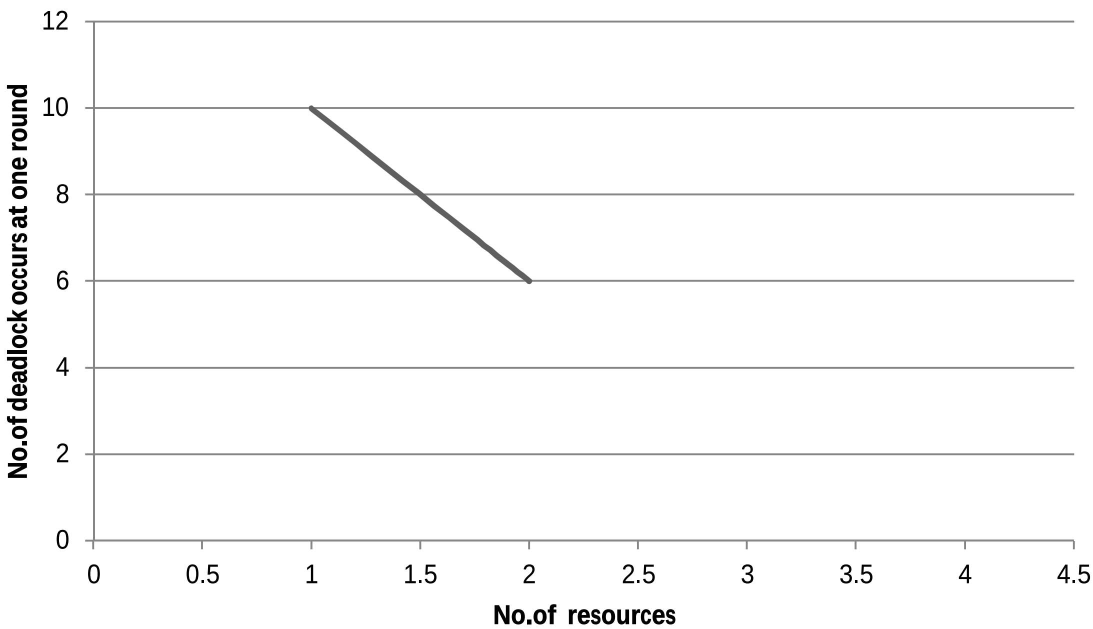


Fig 5.6: GRME using two resources – deadlock occurs

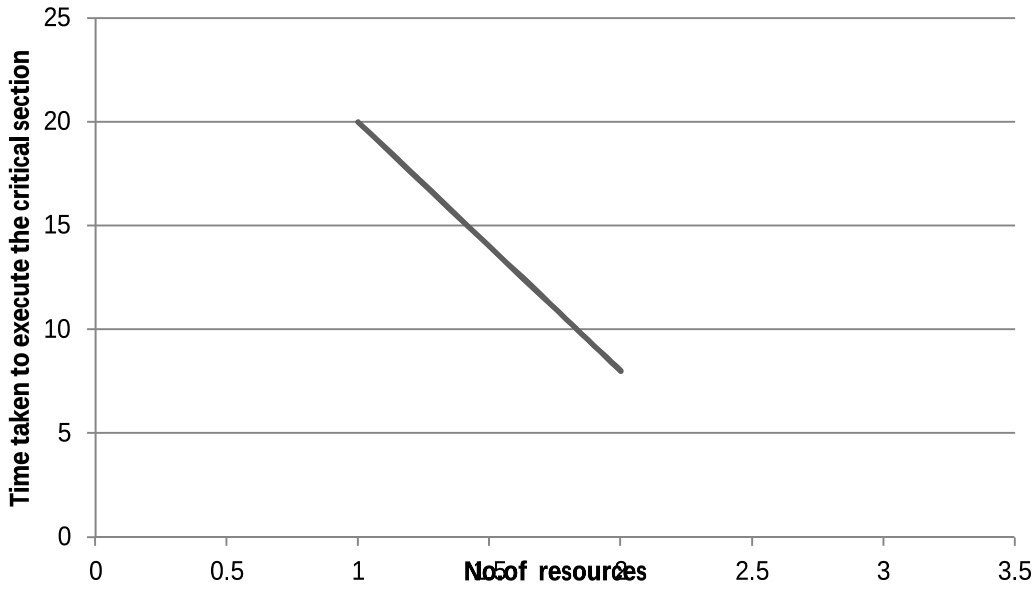


Fig 5.7 : GRME using two resources- time

CHAPTER 6

CONCLUSION AND FUTURE WORK

Optimizing communication over cellular network is very difficult task with implications in real world. Relaxed mutual exclusion including channel allocation is one of the such problem. Here we generalize the previously proposed mutual exclusion algorithm and decompose it in several sub-problem. A general algorithm is proposed for relax mutual exclusion for two source having information structure with necessary and sufficient condition. It also describes the problems which raise during RME including deadlock resolution, dealing with multiple resources and describe various information structure. Distributed dynamic channel allocation problem can be solved in many ways with taking into account various aspects. Deadlock is very crucial problem and it must be control immediately. Deadlock avoidance is very important factor, as it can slow down (or even stop) migration of resources. Broken cells are a hide problem that is to be researched also, because it can force large inefficiency in system. Through this paper, it is concluded that GRME-algorithm including Deadlock Minimization and Broken Cell Serving mechanisms is very promising and results in efficient communication over network.

We considered the mutual exclusion problem over mobile network. Channel allocation is major problem for defining various cell over the network. Locating a mobile phone is another problem. In particular, individual sites will not have up-to date information about the set of mobile phones currently located in the cells serviced by the various base stations. One approach that has been suggested [21] for dealing with this problem is the "quorum-based" approach . The basic idea is that different sites will have information about the location of any given mobile phone, so by depending on information from a "majority" (quorum) of sites, we can reliably locate the phone. Of course, as the mobile phones move about, information about their locations maintained in the various sites would have to be updated in a reasonably timely manner, else even with majority consensus, we will not be able to accurately locate the phones. In any case, it would be interesting to investigate the possibility of combining this type of information with information about channel usage in various cells that we currently maintain. As messages are exchanged between the cells requesting channels, granting channels, or rejecting requests, we could also piggyback information about the changing locations of the mobile phones that the particular site is aware of.

References

[1] Baiocchi, F. D. Priscoli, F. Grilli, and F. Sestini. The geometric dynamic channel allocation as a practical strategy in mobile networks with bursty user mobility. IEEE Trans. Vehicular Tech., 44(1):14–23, Feb. 1995.

[2] R. Beck and H. Panzer. Strategies for handover and dynamic channel allocation in micro-cellular mobile radio systems. In Vehicular Tech. Conf. IEEE, May 1989.

[3] X. Dong and T. H. Lai. Dynamic carrier allocation strategies for mobile cellular networks. Technical Report OSU-CISRC- 10/96-TR48, Dept. of Comp. and Info. Sci., Ohio State University, 1996.

[4] S. M. Elnoubi, R. Singh, and S. C. Gupta. A new frequency channel assignment algorithm in high capacity mobile communication systems. IEEE Trans. Vehicular Tech., 31(3), 1982.

[5] G. Cao and M. Singhal, "Distributed Fault-tolerant Channel Allocation for Mobile Cellular Networks," In IEEE INFOCOM'99, March 1999, pp. 584-591.
Department of Computer Engineering, DTUPage 43

[6] X. Dong, and T. H. Lai, "Distributed Dynamic Carrier Allocation in Mobile Cellular Networks: Search vs. Update," In Proceedings of the 17th International conference on Distributed Computer Systems, May 1997, pp. 108-115.

[7] X. Dong, and T. H. Lai, "An Efficient Priority-based Dynamic Channel Allocation for Mobile Cellular Networks," In IEEE INFOCOM'97.

[8] David J. Goodman, Wireless Personal Communications Systems, Addison-Wesley, 1997.

[9] Kahol, S. Khurana, S. K. Gupta, and P. K. Srimani, "Adaptive Distributed Dynamic Channel Allocation for Wireless Networks," In Proceedings of the

1998 ICCP WORKSHOPS, August 1998, pp.124-133.

[10] Katzela and M. Naghshineh, “Channel Assignment Schemes for Cellular Mobile Telecommunication Systems: A Comprehensive Survey,” IEEE Personal Communications, June 1996, VOL. 3 No. 3, pp. 10-31.

[11] L. Lamport, “Time, clocks and the ordering of events in a distributed system,” Communications of the ACM,21(7):558-565, November 1978.

[12] W. C. Y. Lee, Mobile Cellular Telecommunications: Analog and Digital Systems, McGraw-Hill, 1995.

[13] V. H. MacDonald, “The Cellular Concept,” The Bell System Technical Journal, Vol. 58, No. 1, January 1979, pp. 15-41.

[14] R. Prakash, N. Shivaratri, and M. Singhal, “Distributed Dynamic Channel Allocation for Mobile Computing,” In Proceedings of the 14th ACM Symposium on Principles of Distributed Computing, 1995, pp. 47-56.

[15] A. Sanders, “The Information Structure of Distributed Mutual Exclusion Algorithms,” In ACM Transactions on Computer Systems, Vol. 5, No. 3, August 1987, pp. 284-299.

[16] M. Singhal, “A Taxonomy of Distributed Mutual Exclusion,” Journal of Parallel and Distributed Computing, 1993, pp. 94-101.

[17] JIANG J, LAI T.H., SOUNDARAJAN N., On Distributed Dynamic Chanel Allocation In Mobile Cellular Networks. IEEE Transactions on parallel and distributed systems, IEEE 2003

[18] KATZELA I., NAGHSHINEH M., Channel Assignment Schemes for Cellular mobile Telecommunication Systems, IEEE Personal Comm, June 1996

[19] V.H. MacDonald, “The Cellular Concept,” The Bell System Technical J., vol. 58, no. 1, pp. 15-41, Jan. 1979.

Department of Computer Engineering, DTUPage 44

[20] L. Narayanan and S. M. Shende, “Static Frequency Assignment in Cellular Networks,” Algorithmica, pp. 396-409, 2001.

[21] M. Maekawa, “A NP-Algorithm for Mutual Exclusion in Decentralized Systems,” ACM Trans. Computer Systems, vol. 3, no. 2, pp. 145-159, 1985.

[22] C. Mohan and A. Silberschatz, “Distributed Control—Is It Always Desirable?” Symp. Reliability in Distributed Software and Database Systems, July 1981.

[23] S. Nishio, K.F. Li, and E.G. Manning, “A Resilient Mutual Exclusion Algorithm for Computer Network,” IEEE Trans. Parallel and Distributed Systems, July 1990.

[24] R. Prakash, N. Shivaratri, and M. Singhal, “Distributed Dynamic Channel Allocation for Mobile Computing,” Proc. 14th ACM Symp. Principles of Distributed Computing, pp. 47-56, 1995.