



uOttawa

L'Université canadienne  
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES  
ET POSTDOCTORALES



FACULTY OF GRADUATE AND  
POSTDOCTORAL STUDIES

Kaouther Abrougui

AUTEUR DE LA THÈSE / AUTHOR OF THESIS

Master of Computer Science

GRADE / DEGREE

School of Information Technology and Engineering

FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

QoS and Fault Tolerant Distributed Channel Allocation Protocols for Wireless and Mobile Networks

TITRE DE LA THÈSE / TITLE OF THESIS

Azzedine Boukerche

DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

EXAMINATEURS (EXAMINATRICES) DE LA THÈSE / THESIS EXAMINERS

Chung-Horn Lung

Amiya Nayak

Gary W. Slater

LE DOYEN DE LA FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTDOCTORALES /  
DEAN OF THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

# **QoS and Fault Tolerant Distributed Channel Allocation Protocols for Wireless and Mobile Networks**

By

**Kaouther Abrougui**

A Thesis submitted to the  
Faculty of Graduate and Postdoctoral Studies

In partial fulfillment of  
The requirements for the degree of  
**Master of Computer Science**

Ottawa-Carleton Institution for Computer Science  
School of Information Technology and Engineering  
University of Ottawa  
Ottawa, Ontario, Canada



Library and  
Archives Canada

Bibliothèque et  
Archives Canada

Published Heritage  
Branch

Direction du  
Patrimoine de l'édition

395 Wellington Street  
Ottawa ON K1A 0N4  
Canada

395, rue Wellington  
Ottawa ON K1A 0N4  
Canada

*Your file* *Votre référence*  
*ISBN: 978-0-494-16973-5*  
*Our file* *Notre référence*  
*ISBN: 978-0-494-16973-5*

#### NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

#### AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

---

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

  
**Canada**

© Kaouther Abrougui, Ottawa, Canada, 2006

The following publications by the author are relevant to this thesis:

### **Journal**

1. A. Boukerche, T. Huang and K. Abrougui. "A Fault-Tolerant Dynamic Channel Allocation Protocol for Cellular Networks". Accepted to *Wiley Journal of Wireless Communications and Mobile Computing*.
2. A. Boukerche, T. Huang and K. Abrougui. "QoS and Fault-Tolerant based Dynamic Channel Allocation Protocol for Mobile Networks". To be submitted.
3. A. Boukerche and K. Abrougui. "Channel Allocation Algorithms for Mobile Networks: A Taxonomy". To be submitted.

### **Conference**

1. A. Boukerche, T. Huang, K. Abrougui and J. Williams. "A Fault-Tolerant Dynamic Channel Allocation Protocol for Cellular Networks". In *IEEE International Conference on Wireless Networks, Communications, and Mobile Computing 2005*, Maui, Hawaii, Jun. 2005.
2. A. Boukerche, T. Huang and K. Abrougui. "Design and Performance Evaluation of a QoS-Based Dynamic Channel Allocation Protocol for Wireless and Mobile Networks". In *13th IEEE ACM International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS) 2005*, pages 445-452, Atlanta, USA, Sep. 2005.
3. A. Boukerche, T. Huang and K. Abrougui. "QoS and Fault-Tolerant Based Distributed Dynamic Channel Allocation Protocol for Cellular Networks". In *the Proceedings of the 30th IEEE International Conference on Local Computer Networks (LNC)*, pages 59-67, Sydney, Nov. 2005.



# Acknowledgements

I would like to give special thanks to my supervisor Dr. Azzedine Boukerche for his confidence in my abilities, his stimulating suggestions and encouragement that helped me to complete this work.

I would like to thank Dr. Tingxue Huang, a post doctoral research assistant in Paradise Laboratory in University of Ottawa, for his assistance and his cooperation to accomplish this work.

This work was partially supported by NSERC, Canada Foundation for Innovation Grants, Canada Research Chair Program, and OIT/Distinguished Researcher Award.

I would like to show special gratitude to my brother in law Mourad Elhadeif for his advices, his help and his continuous support for me.

I would like to thank my sister Aida Abrougui, she has been supportive in every way.

I would also like to thank my parents for their continuous support and constant encouragement.

# Contents

<b>Acknowledgements</b>	<b>iii</b>
<b>List of Figures</b>	<b>x</b>
<b>List of Tables</b>	<b>xi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Mobile Cellular Communication Networks . . . . .	1
1.1.1 An Architecture for a Mobile Cellular Network . . . . .	1
1.1.2 Channel Allocation . . . . .	1
1.1.3 Channel Allocation versus Mutual Exclusion . . . . .	2
1.2 Problem Statement . . . . .	3
1.3 Motivation . . . . .	4
1.4 Thesis Outline . . . . .	4
<b>2 Overview of Distributed Channel Allocation Protocols for Wireless and Mobile Networks</b>	<b>5</b>
2.1 Classification of Channel Allocation Algorithms . . . . .	5
2.1.1 Fixed Channel Allocation Protocols . . . . .	5
2.1.2 Dynamic Channel Allocation Protocols . . . . .	6

---

2.1.3	Hybrid Channel Allocation Protocols . . . . .	7
2.2	CAWM Class of Dynamic Channel Allocation Schemes . . . . .	8
2.2.1	Geometric Dynamic Channel Allocation . . . . .	8
2.2.2	Borrowing with Directional Channel-Locking (BDCL) Strategy . . . . .	11
2.2.3	Nanda-Goodman Strategy . . . . .	13
2.2.4	Two-Step Dynamic-Priority Dynamic Channel Allocation Strategy . . . . .	15
2.3	Cell-Based Distributed Dynamic Channel Allocation Schemes . . . . .	16
2.3.1	Local Packing Dynamic Distributed Channel Assignment . . . . .	17
2.3.2	LP-DDCA with Adjacent Channel Interference Constraint . . . . .	18
2.3.3	Locally Optimized Dynamic Assignment Strategy . . . . .	19
2.4	Signal Strength Measurement-Based Distributed Dynamic Channel Allocation Schemes . . . . .	20
2.4.1	Sequential Channel Search . . . . .	20
2.4.2	Minimum Signal-to-Noise Interference Ratio . . . . .	20
2.4.3	Dynamic Channel Selection . . . . .	21
2.5	Biologically-Inspired Channel Allocation Protocols . . . . .	21
2.5.1	Simulated Annealing-Based Channel Allocation Protocols . . . . .	22
2.5.2	Neural Network-Based Dynamic Channel Allocation Protocol . . . . .	24
2.5.3	Genetic-Based Dynamic Channel Allocation Protocol . . . . .	25
2.5.4	ANTS Algorithm . . . . .	27
2.5.5	Dsatur . . . . .	29
2.6	Algorithms Based on Cells Borrowing Mechanisms . . . . .	29
2.6.1	Simple Borrowing Algorithm and its Two Options . . . . .	30
2.6.2	Dynamic Channel Allocation Based on Borrowing from the Richest First Available . . . . .	31

2.6.3	Dynamic Channel Allocation Algorithm Based on Dynamic Load Balancing Strategy . . . . .	32
2.7	Mutual Exclusion Based Dynamic Channel Allocation Algorithms . . . . .	34
2.7.1	The Prakash-Shivaratri-Singhal Algorithm . . . . .	34
2.7.2	The Choy and Singh Algorithm . . . . .	36
2.7.3	The Cao and Singhal Algorithm . . . . .	37
2.7.4	The Boukerche-Hong-Jacob Algorithm . . . . .	37
2.8	Graph Coloring Based Dynamic Channel Allocation Algorithms . . . . .	38
2.8.1	Deterministic Distributed Approach . . . . .	38
2.8.2	Randomized Distributed Approach . . . . .	40
2.9	Fault-Tolerant Channel Allocation Algorithms . . . . .	40
2.9.1	Prakash, Shivaratri and Singhal's Channel Allocation Protocol . . . . .	41
2.9.2	Cao and Singhal's Fault-Tolerant Channel Allocation Protocol . . . . .	42
2.9.3	Yang, Jiang, Manivannan and Singhal's Fault-Tolerant Enhanced Channel Allocation Scheme . . . . .	42
<b>3</b>	<b>Design and Performance Evaluation of a QoS-Based Dynamic Channel Allocation Protocol for Wireless and Mobile Networks</b>	<b>45</b>
3.1	System Model . . . . .	46
3.2	Distributed Dynamic Channel Allocation Algorithm . . . . .	47
3.2.1	Mutual Exclusion Paradigm . . . . .	49
3.2.2	Storage Information for Allocation Channels . . . . .	49
3.3	QoS-based Distributed Dynamic Channel Allocation Algorithm . . . . .	50
3.3.1	A Timer for Channel Reservation . . . . .	50
3.3.2	Prediction of the Moving Direction of MHs . . . . .	50
3.3.3	Dynamic Channel Reservation for <i>Handoffs</i> . . . . .	51
3.4	Formal Description of the QoS-based Dynamic Channel Allocation Algorithm	53
3.5	Simulation Experiments . . . . .	55

---

3.5.1	Performance Metrics . . . . .	58
3.5.2	Experimental Results . . . . .	59
3.6	Conclusion . . . . .	65
<b>4</b>	<b>A Fault-Tolerant Dynamic Channel Allocation Protocol for Cellular Networks</b>	<b>66</b>
4.1	Fault-Tolerant Distributed Dynamic Channel Allocation Algorithm . . . . .	67
4.2	Fault Tolerance . . . . .	68
4.2.1	MH Failures and Wireless Link Failures . . . . .	68
4.2.2	BS failures . . . . .	69
4.2.3	Communication Link Failures . . . . .	69
4.3	Formal Description of the Fault-Tolerant Dynamic Channel Allocation Algorithm . . . . .	70
4.4	Illustrative Examples . . . . .	73
4.5	Simulation Experiments . . . . .	76
4.5.1	Performance Metrics . . . . .	78
4.5.2	Experimental Results . . . . .	79
4.6	Conclusion . . . . .	87
<b>5</b>	<b>QoS and Fault-Tolerant Based Distributed Dynamic Channel Allocation Protocol for Cellular Networks</b>	<b>88</b>
5.1	Introduction . . . . .	89
5.2	Formal Description of the QoS and Fault-Tolerant Based Dynamic Channel Allocation Algorithm . . . . .	89
5.3	Simulation Experiments . . . . .	93
5.4	Conclusion . . . . .	103
<b>6</b>	<b>Conclusion</b>	<b>105</b>
6.1	Summary of Contributions . . . . .	105
6.2	Future Works . . . . .	106



# List of Figures

2.1	Channel Allocation Classification . . . . .	7
2.2	Channel Allocation For Fault-Tolerant Enhancement . . . . .	8
2.3	Cell to Label Affection in a Mobile Cellular Network with $\gamma = 9$ . . . . .	9
2.4	Carrier Acquisition Preference Lists for 9 Labels . . . . .	10
2.5	Channel-Borrowing and Directional Locking . . . . .	12
2.6	Cost/Reward Computation for Hexagonal Cells . . . . .	13
2.7	Identification of the Locally-Optimal reuse Pattern of C: LOP(C) . . . . .	15
2.8	ACO Matrix at Base Station $i$ . . . . .	17
2.9	Structure of the Population Strings . . . . .	25
3.1	Cellular System . . . . .	47
3.2	Dropping Rate . . . . .	61
3.3	Denial Rate . . . . .	62
3.4	Acquisition Time . . . . .	63
3.5	Message Complexity . . . . .	64
4.1	Denial Rate with BS Failures . . . . .	80
4.2	Acquisition Time with BS Failures . . . . .	81
4.3	Message Complexity with BS Failures . . . . .	82
4.4	Denial Rate with Link Failures . . . . .	83
4.5	Acquisition Time with Link Failures . . . . .	84

---

4.6	Message Complexity with Link Failures . . . . .	85
4.7	Partition Split with 400 Channels . . . . .	86
5.1	Dropping Rate with BS Failures . . . . .	97
5.2	Dropping Rate with Link Failures . . . . .	98
5.3	Denial Rate with BS Failures . . . . .	99
5.4	Denial Rate with Link Failures . . . . .	100
5.5	Acquisition Time with BS Failures . . . . .	101
5.6	Acquisition Time with Link Failures . . . . .	102
5.7	Message Complexity with BS Failures . . . . .	103
5.8	Message Complexity with Link Failures . . . . .	104

# List of Tables

3.1	Simulation Parameters . . . . .	57
4.1	Simulation Parameters . . . . .	77
5.1	Simulation Parameters . . . . .	94

# Chapter 1

## Introduction

### 1.1 Mobile Cellular Communication Networks

#### 1.1.1 An Architecture for a Mobile Cellular Network

Generally speaking, the geographical area of a mobile cellular network is partitioned into many adjacent regions, called cells. In real life, the size and shape of each cell are not regular because of the signal strength and some obstacles. In theory, all cells could be simplified and represented as equal sized hexagons. In each cell, there is one and only one *base station* (BS) that serves a number of mobile users, referred to as *mobile hosts* (MH's). Neighboring base stations are grouped into sets, each controlled by a *base station controller* (BSC) that executes some centralized functions. Several BS/BSC are served by a *mobile switching center* (MSC) managing a large coverage area. Typically, the links between BSs and their BSC and between two BSs are wired while the links between MH's and their BS are wireless.

#### 1.1.2 Channel Allocation

The channel allocation problem is one of the resource allocation problems in cellular networks. When a MH initiates a communication session, it sends first a communication request to the nearest BS controlling the cell. The BS executes the channel allocation algorithm and

assigns an available channel satisfying this request. In order to avoid any co-channel interference, BSs responsible for neighboring cells must use different communication channels as long as the coverage areas of two neighboring cells overlap. When the geographic distance between two cells is more than a threshold called the *minimum channel reused distance* ( $D_{min}$ ), the same communication channels could be reused.

### 1.1.3 Channel Allocation versus Mutual Exclusion

The concept of mutual exclusion has received a considerable attention mainly in the domains of operating systems and distributed computing. The mutual exclusion paradigm prevents processes from deadlocks and conflicts to acquire shared resources. Mutual exclusion has been exploited in operating systems to permit processes to acquire exclusively a shared resource and in distributed computing to administer the access to a common database.

The mutual exclusion paradigm can be used to solve the channel allocation problem. In fact, in a mobile cellular network, several communication sessions can be initiated at the same time. Since bandwidth and communication channels are bounded, it is probable that two neighboring cells select the same channel from the bounded set of channels. Consequently, to prevent from co-channel interference, a channel should be exploited exclusively between two adjacent cells. As a result, the mutual exclusion paradigm can be used to solve this problem.

The algorithms based on mutual exclusion to solve the channel allocation problem can be classified into three approaches:

1. ***Timestamp-based approach:*** The algorithms use a *timestamp*. This approach was adopted by Prakash-Shivaratri-Singhal and Cao-Singhal algorithms [8, 35]. The main idea consists of sending a *timestamped REQUEST* by a base station that wants to acquire a channel from its neighbors. The base station with the sooner *timestamp* acquires the channel in case there is a conflict on the same channel.
2. ***Competition-based approach:*** In the second approach, there is a use of a *competition*

variable. This approach is used by the Distributed Dynamic Resource Allocation protocol (DDRA) [6]. In this case, if two base stations want to acquire the same channel, the decision is made after comparing their *competition* values. The larger is the competition value, the prior is the base station holding this value to acquire the channel. The defeated base station ameliorates its competition capability in the next round by incrementing its *competition* value by one.

3. **Priority-based approach:** In the third approach, a priority is set for each base station. This approach is used by Choy-Singh algorithm [12]. In this case, when two base stations want to acquire the same channel, the channel is assigned to the one holding the larger priority.

## 1.2 Problem Statement

Recent demand for mobile telephone services has been growing rapidly [20] while the electro-magnetic spectrum of frequencies allocated for this purpose remains limited. Any solution to the channel assignment problem is subject to this limitation, as well as the interference constraint between adjacent channels in the spectrum. The recent research focused on distributed channel allocation schemes due to their high reliability and scalability. In these schemes, a base station has to consult with its neighboring base stations in order to assign a channel to a call. If it cannot communicate with its neighbors, it fails in allocating a channel. However, it is a common phenomenon that a base station fails in communicating with its neighboring base stations due to some reasons, such as heavy traffic load. Besides, in most of the cases in the congested cells, ongoing calls are subject to be dropped during inter-handoff, thus annoying the QoS for mobile users.

## 1.3 Motivation

In order to ensure and guarantee the continuous service to mobile users, an efficient adaptive channel reservation schema is required to provide continuous QoS support. Besides, to ensure that a channel allocation algorithm works well under mobile host failures, base station failures and communication link failures, a fault-tolerant channel allocation protocol is needed. Finally, to benefit from both previous features at the same time an efficient QoS and fault tolerant based protocol for dynamic channel allocation is required.

## 1.4 Thesis Outline

The remainder of this thesis is organized as follows:

- Chapter 2 presents related work on distributed dynamic channel allocation protocols for wireless and mobile networks and present our classification of these protocols based on characteristics they have in common.
- Chapter 3 presents a design and performance evaluation of our QoS-based dynamic channel allocation protocol for wireless and mobile networks.
- Chapter 4 presents our fault-tolerant dynamic channel allocation protocol for cellular networks.
- Chapter 5 presents our QoS and fault-tolerant based distributed dynamic channel allocation protocol for cellular networks.
- Chapter 6 concludes the thesis and provides some directions for future research.

## **Chapter 2**

# **Overview of Distributed Channel Allocation Protocols for Wireless and Mobile Networks**

### **2.1 Classification of Channel Allocation Algorithms**

The channel allocation algorithms have been well studied in the literature [1, 3, 10, 13, 14, 19, 25, 32, 33, 34, 37, 39, 40, 43, 44, 45] and are still under consideration due to their importance [5]. Three categories of channels are distinguished depending on the way they are assigned to cells: *Fixed Channel Allocation (FCA)*, *Dynamic Channel Allocation (DCA)* and *Hybrid Channel Allocation (HCA)*.

#### **2.1.1 Fixed Channel Allocation Protocols**

In the literature, many protocols considered the Fixed Channel Allocation approach (FCA) [32, 40, 44, 45]. The basic property of the FCA protocols is that channels are assigned to cells in a static and nearly equitable way. Nevertheless, this approach witnessed low performance and high blocking rate [45]. The main reasons for this disadvantage is that in real life traffic

loads differ from cell to cell. In fact, in the mobile cellular network, cells can be *cold cells* [15] to depict the cells with low traffic load or *hot cells* [15] to represent the cells with high traffic load. An improvement of this approach consisted in allocating channels to cells relatively to their traffic loads. However, this still exhibits inefficiency due to the unpredictable and unstable characteristics of call distributions. As a consequence, another approach for channel allocation known as dynamic channel allocation emanated to deal with this problem.

### 2.1.2 Dynamic Channel Allocation Protocols

To deal with the problem of unstable and unpredictable distribution of calls in a mobile cellular network, several Dynamic Channel Allocation (DCA) approaches have been proposed [3, 10, 13, 14, 33, 34, 37]. In DCA, any channel can be used by any cell under some restrictions. Besides, there is no fixed allocation of channels to cells as in the FCA approach.

Recently, many DCA protocols were developed. Some of them [34] deal only with the present information concerning the channel usage to make a decision for channel allocation while the others [3, 34] require the current and the previous channel usage at the same time. The former schemes are known as the *call-by-call* DCA schemes while the later ones are known as the *adaptive* schemes.

Another classification can be used for DCA depending on their implementation. In case all the channels are put in a central pool and a centralized controller is responsible to assign them, the schemes are classified as centralized schemes [13, 14]. However, when each base station is responsible to allocate a channel to a mobile host in its range based on its local information collected from the environment or the other base stations, the schemes are classified as distributed DCA schemes [10, 36]. The information used by the base stations can be *cell-based* [10, 11] or *signal-strength* information [36, 38] given birth to two other different classes of DCA schemes.

### 2.1.3 Hybrid Channel Allocation Protocols

In the literature, many simulations and analysis proved that the performance of FCA schemes and DCA schemes is tightly related to the traffic load in the mobile cellular network [13, 25]. Thus, when the traffic load is uniform and heavy, the DCA schemes outperform the FCA schemes. While, in case the traffic load is non-uniform but low or moderate FCA schemes give better results. To profit from the advantages of both the FCA and the DCA schemes, many authors proposed the Hybrid Channel Allocation (HCA) schemes [25, 39, 43]. In this later approach, channels are separated in two sets: *fixed* and *dynamic*. Channels belonging to the fixed set are allocated using the FCA approach and channels belonging to the dynamic set are designated to be used by all the base stations in the mobile cellular network.

Figures 2.1 and 2.2 give the main algorithms that we will present in the rest of this chapter categorized according to their basic characteristics.

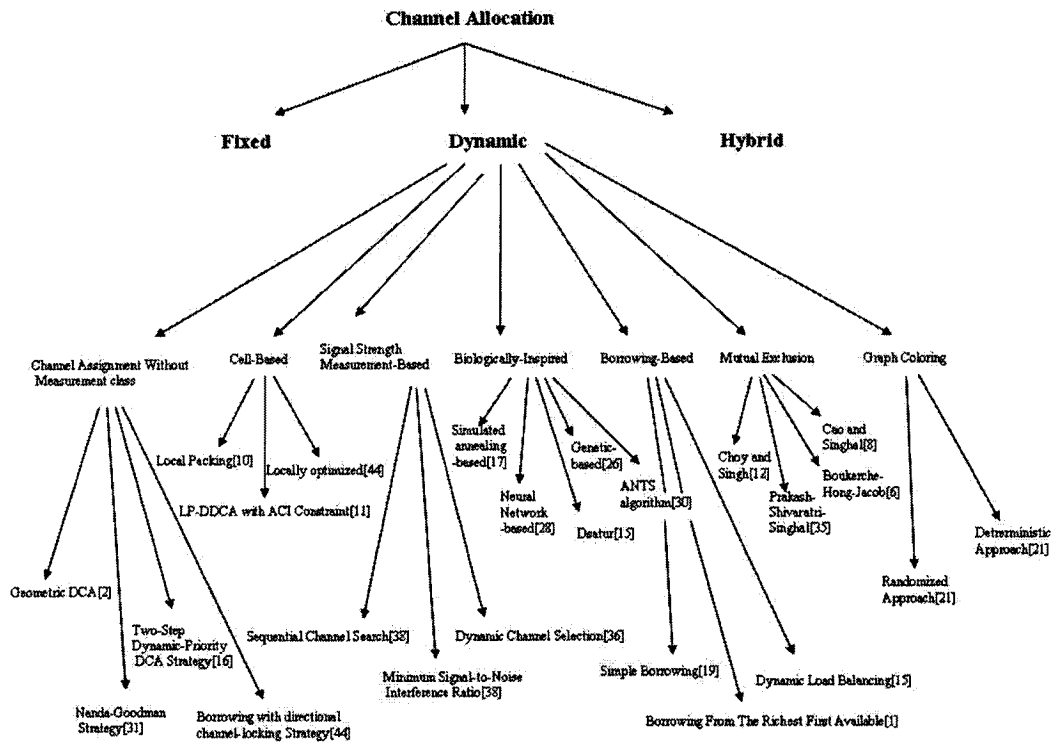


Figure 2.1: Channel Allocation Classification

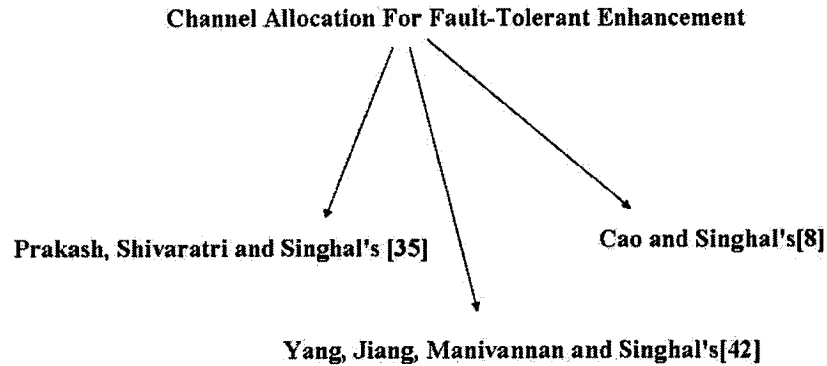


Figure 2.2: Channel Allocation For Fault-Tolerant Enhancement

## 2.2 CAWM Class of Dynamic Channel Allocation Schemes

Channel Assignment Without Measurement (CAWM) [2, 44] is a class belonging to the dynamic channel allocation schemes. For this class, when a base station holds a carrier of channels, it assigns all its channels to mobile hosts in its cell. In some cases, even if a cell  $C1$  needs only one channel in a carrier  $r$ , it holds all the channels in the carrier and any other cell  $C2$ , in the interference neighboring of  $C1$ , cannot hold the same carrier  $r$  to exploit another channel on it. In the following we present some algorithms belonging to this class.

### 2.2.1 Geometric Dynamic Channel Allocation

Geometric Dynamic Channel Allocation (GDCA) was proposed in [2] and is basically based upon the *label* and the *pool of carrier* concepts.

- **Cell Labeling:** The concept of labeling the cells consists of affecting a label to each cell in the mobile cellular network with respect to the two following rules:
  1. Any two cells belonging to the same interference neighborhood region should be affected two different labels.
  2. The total number of distinct labels assigned in the whole cellular network should be minimized with respect to the first rule.

Figure 2.3 gives an illustrative example of a cellular network. The cellular network is partitioned into hexagonal cells of equal size and the reuse distance is fixed to  $3\sqrt{3}R$ , where  $R$  is the radius of the cell. These cells are labeled using capital letters (A,B,C,...). The number of different labels is designed by  $\gamma$ . In Figure 2.3, the illustrated cellular network requires 9 different labels ( $\gamma = 9$ ) (from A to I).

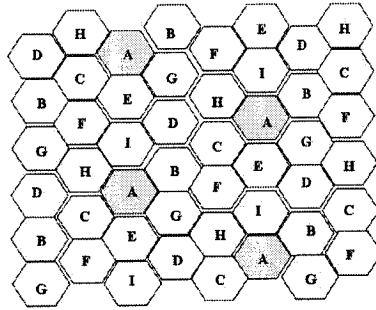


Figure 2.3: Cell to Label Affection in a Mobile Cellular Network with  $\gamma = 9$

- **Pools of Carriers:** In the GDCA strategy the set of carriers is partitioned into  $\gamma$  pools of carriers.  $P(i)$  represents the  $i$ -th pool of carriers, where  $i$  ranges between 0 and  $\gamma$ . According to the example shown in Figure 2.3, the number of pools in the illustrated example is equal to 9. Besides, each pool consists of an ordered list of carriers.

The basic idea behind the GDCA technique is to acquire or release a carrier with respect to preference lists of carriers. Preference lists differ with regards to the base stations' labels and are the same for base stations holding the same label. Suppose  $LA(X)$  is the carrier acquisition preference list corresponding to the label  $X$ .  $LA(X)$  consists of a classified sequence of pools of carriers where the first pool in the list corresponds to the label  $X$  and the first carrier in that pool holds the highest priority. The algorithm consists of assigning numbers with respect to the labels in the following way:  $A$  is substituted by 0,  $B$  is substituted by 1,  $C$  is substituted by 2 and so on...

Formally:

$$LA(X) = \{P(i), P([i + 1] \bmod \gamma), P([i + 2] \bmod \gamma) \dots P([i + \gamma - 1] \bmod \gamma)\}$$

The carrier release preference list  $LR(X)$  is the reverse of  $LA(X)$  and can be formulated as follows:

$$LR(X) = \{P([i + \gamma - 1] \bmod \gamma), \dots, P([i + 2] \bmod \gamma), P([i + 1] \bmod \gamma), P(i)\}$$

When a mobile host initiates a communication it sends a request to the base station  $BS$  in its cell. The base station  $BS$  with label  $X$  tries to acquire the first available carrier in its  $LA(X)$  which is the carrier with the greatest priority. Once it finishes, it releases the carrier with the inferior priority using its  $LR(X)$  list. Figure 2.4 shows the carrier acquisition preference lists when  $\gamma = 9$ .

Suppose that there are 36 carriers and that  $ri$  depicts the  $i$ -th carrier where  $i$  is in the range [1..36]. The algorithm will correspond to each label an integer in the following way:  $A \leftrightarrow 0$ ,  $B \leftrightarrow 1$ ,  $C \leftrightarrow 2$ ,  $D \leftrightarrow 3$ ,  $E \leftrightarrow 4$ ,  $F \leftrightarrow 5$ ,  $G \leftrightarrow 6$ ,  $H \leftrightarrow 7$ ,  $I \leftrightarrow 8$ . Besides, the algorithm constructs the pools in the following way:

$$P(i) = \{c_{4i+1}, c_{4i+2}, c_{4i+3}, c_{4i+4}\}$$

Carrier acquisition preference lists									
Cell label	Choice 1	Choice 2	Choice 3	Choice 4	Choice 5	Choice 6	Choice 7	Choice 8	Choice 9
A	$P(0)$	$P(1)$	$P(2)$	$P(3)$	$P(4)$	$P(5)$	$P(6)$	$P(7)$	$P(8)$
B	$P(1)$	$P(2)$	$P(3)$	$P(4)$	$P(5)$	$P(6)$	$P(7)$	$P(8)$	$P(0)$
C	$P(2)$	$P(3)$	$P(4)$	$P(5)$	$P(6)$	$P(7)$	$P(8)$	$P(0)$	$P(1)$
D	$P(3)$	$P(4)$	$P(5)$	$P(6)$	$P(7)$	$P(8)$	$P(0)$	$P(1)$	$P(2)$
E	$P(4)$	$P(5)$	$P(6)$	$P(7)$	$P(8)$	$P(0)$	$P(1)$	$P(2)$	$P(3)$
F	$P(5)$	$P(6)$	$P(7)$	$P(8)$	$P(0)$	$P(1)$	$P(2)$	$P(3)$	$P(4)$
G	$P(6)$	$P(7)$	$P(8)$	$P(0)$	$P(1)$	$P(2)$	$P(3)$	$P(4)$	$P(5)$
H	$P(7)$	$P(8)$	$P(0)$	$P(1)$	$P(2)$	$P(3)$	$P(4)$	$P(5)$	$P(6)$
I	$P(8)$	$P(0)$	$P(1)$	$P(2)$	$P(3)$	$P(4)$	$P(5)$	$P(6)$	$P(7)$

Figure 2.4: Carrier Acquisition Preference Lists for 9 Labels

For example, the carrier acquisition preference list of a base station  $C$  is:

$$LA(C) = \{c_9, c_{10}, c_{11}, \dots, c_{36}, c_1, c_2, \dots, c_8\}$$

and the carrier release preference list of the same base station  $C$  is:

$$LR(C) = \{c_8, \dots, c_2, c_1, c_{36}, \dots, c_{11}, c_{10}, c_9\}$$

Which is the reverse of  $LA(C)$ .

### 2.2.2 Borrowing with Directional Channel-Locking (BDCL) Strategy

In [44], Zhang and Yum proposed an improvement of the Borrowing with Channel Ordering strategy (BCO) [18]. The basic disadvantage of the BCO strategy is that a channel cannot be borrowed if it is at least occupied in one of the three immediate co-channel cells. This leads to miss many opportunities to reuse channels and to reduce the blocking rate. In the algorithm proposed by Zhang and Yum, the locking of the borrowed channel in the co-channel cells is limited only to cells concerned by this borrowing. This is achieved by using a concept called *directional locking of borrowed channel*, see Figure 2.5. Suppose a cell  $X$  borrows a channel  $i$  from cell  $C1$ . In this case, channel  $i$  should be blocked only in directions 3, 4 and 5 in cell  $C2$  and directions 4, 5, 6 and 1 In cell  $C3$ . The cells in the remaining directions can acquire the channel  $i$ . This improves the availability of channel  $i$  and reduces the blocking rate in the mobile cellular network. The proposed algorithm deals with three type of lists managed by the Mobile Telephone Service Office (MTSO). The three type of lists are mentioned below:

- UC: organized list of unoccupied nominal channels
- OC: organized list of nominal channels occupied by local calls
- LC: organized list of nominal locked channels as well as their locking directions and the cells causing their locking

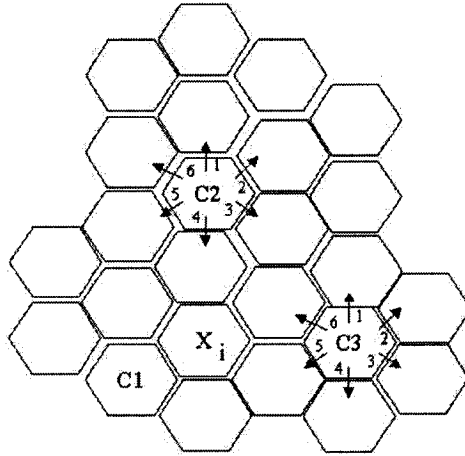


Figure 2.5: Channel-Borrowing and Directional Locking

When a call reaches a cell  $C1$ , the algorithm proceeds in the following way to serve the coming call:

- Step 1:** If  $UC(C1)$  is not empty, then affect the first available channel  $i$  to the call and switch the channel  $i$  to the list  $OC(C1)$ .
- Step 2:** If  $UC(C1)$  is empty, the MTCO determinates the set  $S1$  of all the possible assignable channels from the neighbors of  $C1$  and the locked channels where  $C1$  is in the non-locking direction. If  $S1$  is vacant, the call is blocked.
- Step 3:** Choose from  $S1$  another set  $S2$ .  $S2$  contains all the unoccupied channels in their two closest co-channel cells, and the locked channels by cells distant from  $C1$  by at least three cell units. If  $S2$  is vacant, the call is blocked.
- Step 4:** The assigned channel  $i$  is the one that satisfies the two following conditions in  $S2$ :
1.  $i$  belongs to the cell with the maximum number of unoccupied channels.
  2.  $i$  is the last ordered channel in the cell depicted in the first condition.
- Step 5:** After allocating  $i$  to the call,  $i$  is locked by the three nearest co-channel cells in the convenient directions. Then the lists are updated:

- First by switching channel  $i$  from the list  $UC$  to the  $LC$  lists of the three cells.
- Second by registering  $CI$  in the list  $LC$ , since it is the cause for locking channel  $i$ .

To release a channel, the channel is reallocated using a reallocation procedure [44], and the three lists:  $UC$ ,  $OC$  and  $LC$  are updated to keep the recent information.

### 2.2.3 Nanda-Goodman Strategy

In [31], Nanda and Goodman have proposed a distributed dynamic resource acquisition algorithm where each cell selects individually a carrier to acquire. The selection is based on a *cost-reward metric*. Each cell collects the carrier usage information from its neighbors to determine the cost-reward information.

Suppose cell  $C$  wants to acquire a carrier and that the reuse factor is  $R$ . The set of cells in the interference neighborhood of the cell  $C$  is denoted by  $IN$ . The set of cells that have their  $IN$  overlapping with  $IN(C)$  is denoted by  $INN$ . If we suppose that the mobile cellular network is composed of hexagonal cells, the algorithm needs the carrier usage information of all the cells situated one and two hops from the considered cell  $C$ . All the carriers being in use by the cells situated one hop from the considered cell  $C$ , cells in  $IN(C)$ , cannot be acquired by  $C$  since

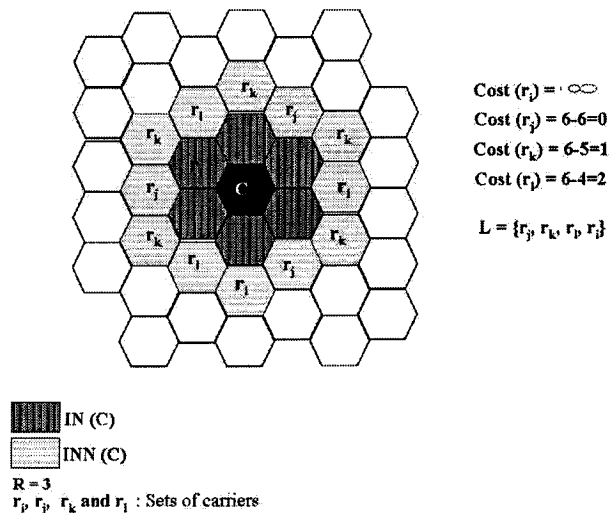


Figure 2.6: Cost/Reward Computation for Hexagonal Cells

they are in its interference neighborhood. All the other carriers, the candidate carriers, are considered and a *cost/reward metric* is computed for each one of them. The candidate carrier with the minimum cost will be acquired by cell  $C$ . The cost metric used in this algorithm is determined by computing the number of cells that are not allowed to acquire the carrier  $r$  if  $r$  is allocated to the considered cell  $C$ . It is obvious that a carrier acquired by  $C$  should not be acquired by cells in its interference neighborhood  $IN(C)$ . Consequently the cost of acquiring a carrier  $r$  by a cell  $C$ , if no other cell in  $INN$  is using the same carrier, is equal to the number of adjacent cells to  $C$ . Otherwise, if at least one cell in  $INN$  is using the same carrier  $r$ , some additional computations must be performed.

Suppose that  $SN$  is the set of cells in  $INN(C)$  using the carrier  $r$ . First, the algorithm finds the number  $N1$  of cells in the intersection between the interference neighborhood of  $C$  and the interference neighborhood of each element of  $SN$ :

$$N1 = |(IN(C) \cap IN(\{SN\}))|$$

Second,  $N1$  is reduced from the number of cells in the interference neighborhood of cell  $C$   $IN(C)$ . The result  $N2$  is the cost of acquiring carrier  $r$  by cell  $C$  and is given by:

$$N2 = N1 - IN(C)$$

The same computation is done for each candidate carrier. Then, the candidate carriers are sorted in increasing order in a list  $L$ .

- $C$  acquires the first carrier in  $L$  not already in use by  $C$ .
- $C$  releases the last carrier in  $L$  in use in  $C$ .

The objective of this algorithm is to maximize the packing. An illustrative example is shown in Figure 2.6. It illustrates how the cost and reward computations are done.

### 2.2.4 Two-Step Dynamic-Priority Dynamic Channel Allocation Strategy

In [16], Dong and Lai proposed the Two-Step Dynamic-Priority (TSDP) strategy. In their algorithm, two kinds of status should be assigned dynamically to each carrier: *primary* or *secondary* status. For *primary carriers*, the priorities are calculated such that the reuse of carriers is optimal. For *secondary carriers*, the priorities are computed in a way to ensure maximum packing. When a cell  $C$  wants to acquire a carrier  $r$ , the TSDP algorithm proceeds as described in the following steps:

**Step 1:** Form  $k$  groups by partitioning optimally the cells. Any two cells belonging to the same group are distant from each other by at least distance  $D_{min}$ .

In [16],  $D_{min} = 3\sqrt{3}R$  and  $k = 9$ , where  $R$  is the radius of a cell  $C$ . In each group, any cell  $C$  belonging to a group  $G$  has exactly six cells at distance  $D_{min}$  from  $C$  in the same group  $G$ , because of the hexagonal form of cells. These cells are entitled *Locally-Optimal reuse Pattern of C* and designated by  $LOP(C)$ . Figure 2.7 shows the  $LOP$  of a cell  $C$ .

**Step 2:** Specify each carrier as primary or secondary. A carrier  $r$  is identified as primary carrier,

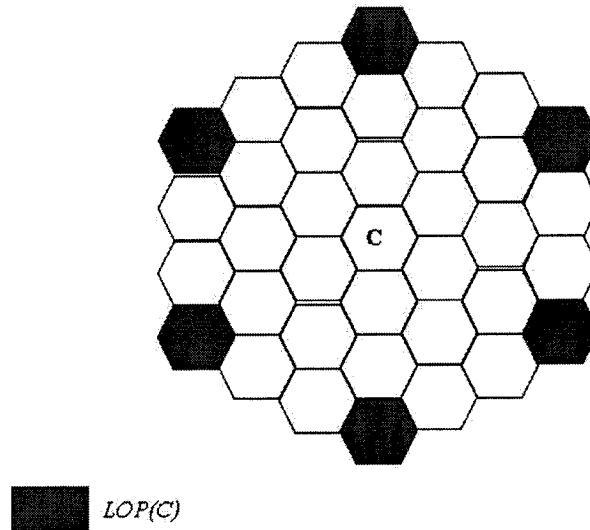


Figure 2.7: Identification of the Locally-Optimal reuse Pattern of C:  $LOP(C)$

if  $r$  is presently exploited by at least  $\lambda$  cells in  $LOP(C)$ .  $\lambda$  is a value that ranges between 0 and 7 as mentioned in [16]. In case  $r$  is presently exploited by less than  $\lambda$  cells in  $LOP(C)$ , carrier  $r$  is identified as secondary. Then the acquisition-priority and the release-priority are computed, depending on the status of carrier  $r$ , as follows:

1. *For primary carriers*

$$acquisition - priority(r, C) = \sum_{C' \in LOP(C)} weight(r, C')$$

where

$$weight(r, C') = \begin{cases} N & \text{if status}(r, C') = UC \ (N > |IN(c)|) \\ 1 & \text{if status}(r, C') = AC \\ 0 & \text{if status}(r, C') = IC \end{cases}$$

$$release - priority(r, C) = -acquisition - priority(r, C)$$

2. *For secondary carriers*

The priority functions of the Nanda-Goodman strategy [31] are reused.

**Step 3:** Depending on the function performed, the carrier with the highest acquisition-priority is acquired or the carrier with the highest release-priority is released.

## **2.3 Cell-Based Distributed Dynamic Channel Allocation Schemes**

The main future of cell-based schemes is that the assignment of a channel is achieved by the base station in the cell where the call is initiated. For that purpose, each base station is responsible for collecting the information about the presently free channels in its range. The communication of information between base stations permits the update of information

acquired at each one.

Cell-based distributed DCA schemes are considered among the optimal channel assignment techniques. The major disadvantage relative to these schemes is the huge number of information exchanged between base stations.

### 2.3.1 Local Packing Dynamic Distributed Channel Assignment

The Local Packing Dynamic Distributed Channel Assignment (LP-DDCA) algorithm was proposed by Li and Chao in [10]. It is a cell-based DDCA scheme. This algorithm uses an augmented channel occupancy (ACO) matrix exploited by each base station to satisfy a call request. The ACO matrix holds the pertinent local information, qualified as sufficient and necessary, required for the base station to accomplish a decision on channel assignment. We assume a mobile cellular network that contains  $N$  cells and  $M$  available channels. Each cell  $i$  has  $ni$  neighbors in its co-channel interference distance. As represented in Figure 2.8, the ACO matrix contains two basic columns and  $ni + 1$  rows. The first basic column is divided into  $M$  columns corresponding to the  $M$  channels. The second column contains the number of the present free channels ready for use by each of the  $ni + 1$  cells. The first row shows the channels used by the base station in cell  $i$  and the other  $ni$  rows denote the channel occupation for the base stations in the neighboring cells of  $i$ . When a base station receives a request from a mobile host in its cell  $i$ , it consult immediately its ACO matrix and look for an empty column, that denotes an idle channel ready to be assigned. Thus the base station holds the first encountered

Base Station number	Channel Number							Number of assignable channels
	1	2	3	4	5	...	M	
$i$			X			...		0
$i_1$		X				...		2
$i_2$	X			X		...		4
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	...	$\vdots$	$\vdots$
$i_n$					X	...		0

Figure 2.8: ACO Matrix at Base Station  $i$

idle channel. Each base station updates its *ACO* matrix based on the information collected from its neighboring base stations. In fact, when a channel occupation modification occurs in a cell, the corresponding base station communicates to its interfering cells the changes occurred so that each base station revises its local *ACO* matrix.

### **2.3.2 LP-DDCA with Adjacent Channel Interference Constraint**

To enhance the LP-DDCA algorithm, Li and Chao proposed a revised version in [11] of the previous algorithm that integrates the adjacent channel interference (*ACI*) constraint. In fact, additionally to the co-channel interference constraint, adjacent channel interference should be bounded in a mobile cellular network. *ACI* can be the cause of crosstalk, dropped calls, early handoff and so on that are the source of poorness in quality of service. The major source of such a problem in a mobile cellular network, is the differences of the distance between the mobile hosts and the base stations. The idea proposed by Li and Chao to improve the previous algorithm, is to force supplementary requirements when electing an idle channel from the *ACO* matrix. If we suppose  $C_{sep}$  the necessary channel separation between two channels to avoid *ACI*, then in the enhanced algorithm,  $C_{sep} - 1$  columns to the left and the same number of columns to the right of the elected channel are required to have empty entries in the first row of the *ACO* matrix. As soon as a mobile host wants to initiate a communication session, it sends a request to the base station in its cell, the base station seeks in the first row of the *ACO* matrix for a group of  $2C_{sep} - 1$  consequential unoccupied entries where the center column of the group is unoccupied. In the affirmative case, the channel is assigned to the requesting mobile host. In the negative case, the base station seeks in the first row of the *ACO* matrix for a group of  $2C_{sep} - 1$  consequential unoccupied entries where the center column of the group is occupied once. If successful, the base station verifies if the cell using the channel has supplementary unoccupied channels. In the successful case, a message is sent to the affiliated cell, requesting from the base station of that cell to shift the call using the required channel to a new channel. In the failed case, the call is blocked. In the case when the channel

separation is less than four, which is what is current in real networks, the simulations of the enhanced LP-DDCA [11] depict that the supplementary constraint has inconsiderable effect on the complexity of electing a free channel satisfying a request. Besides, the robustness of the enhanced LP-DDCA to *ACI* is due primordially to its capability to do channel reassignment to satisfy new calls reducing the number of blocked calls.

### 2.3.3 Locally Optimized Dynamic Assignment Strategy

The technique proposed in [44] by Zhang and Yum consists of using a *cost function* to evaluate each candidate channel. The *cost function* measures the probability of future call blocking. When a cell receives a call, it calculates the cost of use of each candidate channel. The channel having the lowest cost will be allocated. The purpose on reducing the cost is to minimize the channel reuse distance of a channel  $i$ . As a result, the same channel can be reused again in the nearest possible cells.

Let us consider a cell  $C$ .  $C(j)$ , where  $(j > 0)$ , denotes the  $j$ -th tier cells of  $C$  or simply all the cells  $j$  hops away from  $C$ . When a call reaches a cell  $CI$ , the algorithm considers the set of unused channels in the first and the second tier cells of  $CI$  as the set of candidate channels to be allocated. This set is denoted by  $CC(CI)$ . To determine the cost for using a channel  $i$  in  $CC(CI)$ , the algorithm proceeds in the following way:

**Step 1:** Compute the number of third tier cells of  $CI$  presently occupying channel  $i$ , this is denoted by the usage frequency  $U(i)$  of channel  $i$ .

**Step 2:** Identify  $U_{max}$  that depicts the set of channels having the maximum usage frequency  $U$ .

**Step 3:** Determine the cost of using  $i$ ,  $COST(i)$ ,  $i \in U_{max}$  using the following function:

$$COST(i) = \sum_{C2 \in F(i)} [D(C1, C2) - 3]$$

Where  $D(C1, C2)$  denotes the distance between the cells  $C1$  and  $C2$ , and  $F(i)$  denotes

the set of fourth and fifth tier cells of  $CI$  presently occupying channel  $i$ .

Finally, the channel with the lowest cost is allocated to the call received in  $CI$ .

## **2.4 Signal Strength Measurement-Based Distributed Dynamic Channel Allocation Schemes**

Signal Strength Measurement-Based Distributed DCA Schemes are also known as *DCA interference adaptation schemes*. In these techniques, to acquire a channel, a base station does not need the exchange of information with the other base stations in the network. It needs only local information. The basic purpose of this class of algorithm is to maximize the packing of channels.

### **2.4.1 Sequential Channel Search**

In [38] Goodman and Serizawa proposed the Sequential Channel Search (SCS) technique. SCS is considered as an elementary technique within the interference adaptation of dynamic channel allocation schemes. In this technique, a pair  $P$  of a mobile host and a base station explores channels in the same arrangement and then select the first unoccupied channel that have an adequate carrier-to-interference ratio ( $CIR$ ). SCS guarantees channel packing but results in many interruptions [38].

### **2.4.2 Minimum Signal-to-Noise Interference Ratio**

In [38], Goodman and Serizawa proposed the Minimum Signal-to-Noise Interference Ratio (MSIR) technique. In this technique, a base station looks for the channel having the lowest interference ratio in the uplink direction.

The purpose of MSIR is to assign unoccupied or slightly occupied channels to the incoming calls. This results in the reduction of the interruption probability of the total calls and the

decrease of the instability of the network.

### 2.4.3 Dynamic Channel Selection

In [36], Punt *et al* presented the Dynamic Channel Selection (DCS) technique. The main characteristic of DCA is that it is a completely distributed algorithm. It is designated for adjustable mobile cellular radio resource sharing. In this technique, it is assumed that mobiles have the capability to determine the amount of interference that they encounter in every channel. When a mobile station initiates a communication session, the algorithm executes the following two steps:

**Step 1:** First, a mobile station estimates the probability of interference. This value is computed by taking on consideration the availability of channels, the co-channel interference, the received signal power from base stations and several other parameters.

**Step 2:** The mobile station then picks the base station that reduces the value of the interference probability.

## 2.5 Biologically-Inspired Channel Allocation Protocols

The biologically-inspired channel allocation techniques [17, 26, 28] provides efficient solutions and heuristics to solve the channel allocation problem. Some of these solutions can be categorized under the *bio-computing* paradigm. In the literature, these techniques are classified depending on the basis of the algorithm, like the algorithms based on biological (such as cellular automata and neural networks), algorithms based on evolutionary (such as genetic algorithms), algorithms based on physical or natural phenomena (such as simulated annealing), algorithm based on the Ant Colony Optimization paradigm (such as the ANT algorithm) and many others.

Some well-known biologically-inspired dynamic channel allocation protocols are presented in the following part.

### 2.5.1 Simulated Annealing-Based Channel Allocation Protocols

The concept of Simulated Annealing (SA) was invented by S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi in [27] and by V. Cerny in [9]. It is a strong probabilistic heuristic approach for optimizing functions defined over complex systems. Moreover, it is a good way to find in a huge search space an appropriate approximation to the global optimum of a certain function. Its basic idea is inspired from statistical mechanics and consists on finding a similarity to the manner physical systems react in the existence of a heat bath.

The simulated annealing approach is different from the greedy algorithms and other elementary techniques. In fact, the previous mentioned algorithms, favor a new configuration of inferior cost and discard the states with expensive one. On the other hand, *SA* approves arrangements detaining greater cost from time to time with a probability fixed by the simulated "temperature" and thus avoiding local minima. In order to answer combinatorial optimization problems, *SA* is considered as being more adequate to fix heuristic algorithms than other approaches [9, 27]. Besides, several potentially conflicting targets in the same problem can be managed efficiently.

In [17], Kunz *et al.* presented a simulated annealing based approach to solve the channel allocation problem. The provided technique resembles to the treatment of a combinatorial optimization problem where the objective function is to avoid the interference while satisfying as much as possible the mobile hosts' requests. The proposed algorithm uses a configuration space  $\mathcal{S}$ , a cost function  $\mathcal{C}$  and the information of the neighborhood structure  $\mathcal{N}$ . To allocate  $m$  channels in a mobile cellular network composed of  $n$  cells, the traditional annealing process is considered and an adequate cooling schedule is performed following the next steps:

**Step 1:** An initial configuration space  $S_{start}$  and an initial temperature  $t_0$  are chosen. The initial configuration space is represented by a binary matrix  $(S_{ij})$  of dimension  $m \times n$  where:

$$S_{ij} = \begin{cases} 0 & \text{if channel } i \text{ is not used in radio cell } j \\ 1 & \text{if channel } i \text{ is used in radio cell } j \end{cases}$$

where  $i$  ranges from 1 to  $m$  and  $j$  ranges from 1 to  $n$ .

**Step 2:** A neighborhood structure  $N(S)$  of the present configuration space  $S$  is made by fulfilling the coming two transitions:

1. One channel  $i$  is used or dropped by one cell  $j$ .
2. One unused channel  $i_1$  replaces one unused channel  $i_2$  in one cell  $j$ .

**Step 3:** A new configuration space  $S'$  is established from the neighborhood structure  $N(s)$ .

**Step 4:** The cost function  $C(S)$  is computed by the following equation:

$$C(S) = C_{interference} + C_{traffic}$$

where  $C_{interference}$  depicts a penalty related to channel interference and  $C_{traffic}$  depicts the penalty related to the traffic violations.

**Step 5:** The acceptance probability is calculated with the following function:

$$\min\{1, \exp(-(C(S') - C(S))/t_k)\}.$$

In case the acceptance probability is in the range  $[0.7, 0.9]$ , then the new value  $S'$  is affected to  $S$ .

**Step 6:** In case the equilibrium has not been attained *go to Step 2*.

**Step 7:** In case there is no promise for a considerable enhancement in cost, the *final temperature* is supposed to be attained. **Stop**.

**Step 8:** Otherwise, the temperature level  $t_k$  is revised with regards to the following equation:

$$t_k = t_{k-1} \cdot \exp(-(\lambda t)/\sigma).$$

Then *Step 2* is processed again.

### 2.5.2 Neural Network-Based Dynamic Channel Allocation Protocol

In [28], Kunz proposed a channel allocation approach based on neural network. This approach prevents from channel interference and better the satisfaction of requests for channel acquisition.

In this algorithm, a neuron matrix is used. A neuron  $u_{ij}$  is utilized for each channel  $i$  at each base station  $j$ . Suppose that we have  $m$  channels. When a mobile host initiates a communication session, the algorithm executes the following steps to assign a channel satisfying the request:

**Step 1:** The neuron matrix  $(\{u_{ij}\})$  is filled with initial values.

**Step 2:** The interference matrix  $(\{interf(j, j')\})$  is calculated. if the base stations  $j, j'$  interfere then  $interfer(j, j') = 1$ , else  $interfer(j, j') = 0$ .

**Step 3:** Weights  $T_{ij,i',j'}$  and external input  $I_{ij}$  are computed with respect to the interference and channel demand as follows.

$$T_{ij,i',j'} = -A\delta_{i'j'}interf(j, j') - B\delta_{jj'}(\delta_{i'j'-1} + \delta_{i'j'+1}) - C\delta_{jj'} + D\delta_{i'j'}\delta_{jj'}$$

$$I_{ij} = Ctraf(j)$$

where A,B,C and D are nonnegative parameter values,  $\delta_{i'j'} = 1$ , and  $\delta_{ij} = 0$ .

**Step 4:** The iterative operation of the concerned energy function E is processed until it attains a stable value.

Assume that:

$$V_{ij} = (1 + \tanh(\lambda u_{ij}))/2,$$

where  $\lambda$  is a parameter.

The energy function is given by:

$$E = 0.5A \sum_{j=1}^n \sum_{i=1}^m V_{ij}V_{ij'} + B \sum_{j=1}^n \sum_{i=1}^{m-1} V_{ij}V_{i+1j} + 0.5C \sum_{j=1}^n (\sum_{i=1}^m V_{ij} - traf(j))^2 - D \sum_{j=1}^n \sum_{i=1}^m V_{ij}^2$$

**Step 5:** The final output  $V_{ij}$  of each neuron  $u_{ij}$  is verified if it is at its maximum or minimum. if the previous condition is verified, **Stop**. Otherwise, **go to step 3**.

After the execution of the algorithm the value  $V_{ij}$  of one neuron stores the maximum value or the minimum value. In case  $V_{ij}$  stores the maximum value, the channel  $i$  is allowed to be used at base station  $j$ . Otherwise, if  $V_{ij}$  stores the minimum value, the channel  $i$  is forbidden to be exploited by base station  $j$ .

### 2.5.3 Genetic-Based Dynamic Channel Allocation Protocol

The idea behind Genetic Algorithms (GA) is to use genetic concepts in order to resolve a given problem. Many important concepts related to the genetic paradigm are borrowed from biology. Such as chromosomes, reproduction, crossover and mutation. In genetic based techniques, a child chromosome improves its fitness in the new population by inheriting good

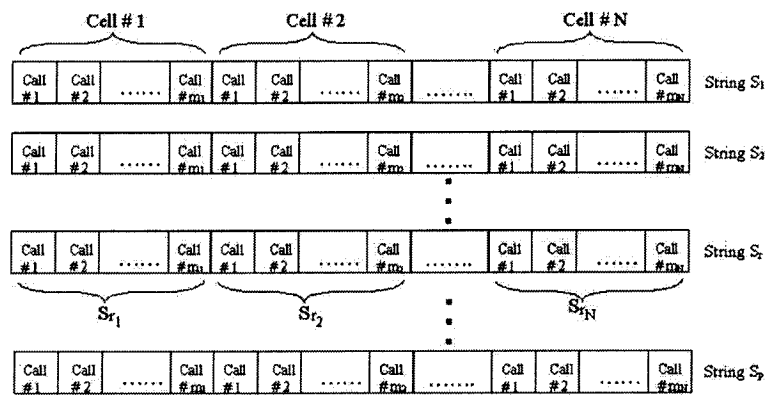


Figure 2.9: Structure of the Population Strings

characteristics from its parents.

In [26], J. S Kim *et al.* presented an efficient dynamic channel allocation technique based on the genetic paradigm. In their algorithm they assumed three constraints: the co-channel constraint (CCC), the adjacent-channel constraint (ACC) and the co-site constraint (CSC). The algorithm executes the following steps to affect  $m_i$  channels to the  $i$ -th cell.

**Step 1:** The size of the *population* is initialized. The structure of the *population* is shown in Figure 2.9. The channel allocation process is accomplished such that in each string  $S_r$ , the cell having the greatest number of calls channels is prior to hold channels. Then the one with the following greatest number of calls has the next priority, and so on, until each cell is affected frequencies. To assure the three mentioned above constraints, a minimal frequency interval  $\gamma$  should be guaranteed between any two frequencies affected to a cell.

**Step 2:** The energy function  $E_{S_r}$  and the fitness function  $F_{S_r}$  of each string  $S_r$  are calculated with the following equations:

$$E_{S_r} = E_{CSC_{S_r}} + E_{ACC_{S_r}} = (\text{energy for CSC}) + (\text{energy for ACC and CCC})$$

$$F_{S_r} = \frac{\frac{1}{E_{S_r}}}{\sum_{n=1}^P \frac{1}{E_{S_r}}}$$

**Step 3:** The roulette wheel slot of each string  $S_r$  is produced. The size of each one is proportional to the ratio of its fitness to the total amount of fitness in the population.

**Step 4:** In order to engender the next generation,  $P$  pairs of *strings* are selected and put into a mating pool. The process of selection considers each *string* and generates a random number between 0 and 1 for each selection. In case the generated number is in the range of the roulette wheel slot of the *string*, then it is selected.

**Step 5:** Perform the crossover operation between *strings* in the mating pool. Three crossover

techniques are known for *GA*: uniform crossover, one point crossover and two points crossover.

**Step 6:** Perform the mutation operation to seek a new search space. In the literature, five mutation techniques are known, all of them can be considered. While looking for a new search space, the algorithm prevents from being induced in a local minimum.

**Step 7:** The termination condition is reached if the maximum number of iterations is attained or if  $E_{S_r} = 0$ . In this case, the algorithm *stops*. Otherwise, it returns to **Step 2**.

The reason of setting the maximum number of iterations is that there is no insurance that the genetic algorithm will converge. Regardless, *GA* can attain an efficient global optimal solution very quickly compared to simulated annealing and neural networks approaches.

#### 2.5.4 ANTS Algorithm

The channel allocation problem is an NP-hard problem and very few number of solutions presented in the literature were considered as optimal. In [30], Maniezzo and Carbonaro used the Ant Colony Optimization (ACO) paradigm to provide an optimal solution to the channel allocation problem. The purpose of this technique is to reduce the overall interference in a considered cellular network using a metaheuristic algorithm.

With respect to Maniezzo and Carbonaro, the appellation ANTS of the algorithm is due to two things: first, the method is using basically the ANT System and second, the technique can be seen as an Approximate Non deterministic Tree-Search procedure.

In the algorithm, an ant is considered as an agent that contributes to calculate partial solutions to the channel allocation problem iteratively. A partial solution is considered as a state. At each iteration, an ant moves from a state to another. At the end of the iterations the optimal solution is obtained.

The algorithm proceeds using the following steps:

**Step 1:** Find a linear lower bound for the problem considered. The way to calculate the lower bound is explained in [30].

**Step 2:** For each ant  $k$ ,

– Repeatedly

- \* Calculate a set  $A_{\sigma}^k(\iota)$  of possible development solutions to the present state  $\iota$ .
- \* The probability  $P_{\iota\psi}^k$  of moving from state  $\iota$  to state  $\psi$  depends on two things: first the attractiveness  $\eta$  of the move which is calculated by some heuristic [23, 24]. This attractiveness shows the consequent convenience of the considered move.  $\eta_{\iota\psi}$  is computed as a lower bound to the cost of completing a solution comprising  $\psi$ . Second, the trail level  $\tau$  of the move, indicating the beneficence of performing that special move in the past. It indicates the a posteriori attraction of that move.
- \* select the state to move into, with probability:

$$P_{\iota\psi}^k = \begin{cases} \frac{\alpha \cdot \tau_{\iota\psi} + (1-\alpha) \cdot \eta_{\iota\psi}}{\sum_{(v\psi) \notin \text{tabu}_k} (\alpha \cdot \tau_{v\psi} + (1-\alpha) \cdot \eta_{v\psi})} & \text{if } (\iota\psi) \notin \text{tabu}_k \\ 0 & \text{otherwise.} \end{cases}$$

Parameter  $\alpha$  indicates the corresponding significance of the trail with regards to attractiveness, and  $\text{tabu}_k$  denotes the set of impossible moves for ant  $k$ .

- \* add the move selected to the set  $\text{tabu}_k$  of the  $k$ -th ant.

– until ant  $k$  has finished its solution

– execute a local optimization procedure to the obtained solution.

**Step 3:** update the trail for each ant. Trails update is achieved by augmenting the level of trails that were part of acceptable solutions, at the same time reducing the other ones. The following equation gives the way to perform this update:

$$\tau_{\iota\psi}(t) = \rho \tau_{\iota\psi}(t-1) + \Delta \tau_{\iota\psi}(t)$$

where  $\rho$  is a user-defined coefficient, and  $\Delta\tau_{i,\psi}$  gives the amount of beneficence of all ants that used move  $(i,\psi)$  to build their solution.

*Step 4:* Terminate if an optimal solution is reached, otherwise go to *Step 2*.

### 2.5.5 D satur

In [4], Borndfer *et al.* presented D satur with costs. The purpose from this technique is to find a realizable assignment of a set of channels inducing little cost. The cost of the different available combinations is registered in a *cost* matrix. The rows of the matrix indicates the carriers (links), and the columns indicates the channels in a given spectrum. When  $cost(r, i)$  in the matrix is at least as large as a specific value **BAD**, channel  $i$  is said to be *bad* for carrier  $r$ .

The algorithm proceeds in the following way:

*Step 1:* Mark all the entries relative to unavailable combinations of channels as

**BLOCKED**. The remaining entries are initialized to 0. Put each carrier  $r$  in the heap. Each carrier  $r$  is identified by a *key*. The *key* equals the number of blocked channels in  $r$ , or the number of bad channels multiplied by **BAD** added to the sum of the cost of all non-blocked, non-bad row entries of the matrix.

*Step 2:* Iteratively and as long as the heap contains carriers, the carrier  $rm$  with the maximum key is selected. The non-blocked channel  $i$  corresponding to the lowest cost value in the row  $cost(rm)$  of the matrix is assigned to  $rm$ . The cost matrix is updated due to the resulting costs from the frequency assignment. The keys of all the carriers still in the heap are also updated.

## 2.6 Algorithms Based on Cells Borrowing Mechanisms

In the following section, we present some well known algorithms based on the borrowing paradigm for dynamic channel allocation.

### 2.6.1 Simple Borrowing Algorithm and its Two Options

In [19], Engel and Peritsky proposed the basic algorithm which is one of the fundamental borrowing based algorithms. Besides, they proposed two other algorithms to improve the basic one.

#### 1. Basic Borrowing Algorithm:

In the basic borrowing algorithm, two classes of channels are specified:

- *Standard channels*: The standard channels for a cell  $i$  are only the channels used by the cell  $i$ .
- *Non-standard channels*: The non-standard channels are the channels that can be borrowed. When these channels are used in a cell, there is probability that they result in co-channel interference to the standard channels in other cells.

The algorithm tries to find the most suitable candidate channel among the non-standard channels that minimizes the probability of co-channel interference with standard channels in other cells. For this purpose, when a call is initiated and all the standard channels are consumed, the non-standard channels are considered. The algorithm finds the probability of blocking neighboring cells for each candidate for assignment, and chooses the candidate with the minimum probability.

#### 2. Second Algorithm:

To reduce the call blocking probability in a cell, it is important to keep the non-standard channels available. As a consequence, a call using a non-standard channel, can be shifted to a standard channel that was used by a dropped call. In case when more than one non-standard channels are assigned, the decision on which one of them should be freed is made in a way that decreases the maximum probability of eventual blockage.

#### 3. Third Algorithm:

The third algorithm consists of revising the way the standard and non-standard channels

are allocated, whenever it is possible, to allow other calls to be processed. Many feasible rearrangements can be accomplished. However, the purpose, as usual, is to decrease the maximum probability of eventual blockage.

### 2.6.2 Dynamic Channel Allocation Based on Borrowing from the Richest First Available

In [1], L. G. Anderson presented three algorithms to allocate channels in a mobile cellular network based on the borrowing concept. It is assumed in these algorithms that a set of channels is allocated to each cell as its nominal channels. In the same way, a set of cells is allocated to a channel as its *nominal* cells. When a mobile host initiates a communication session, the corresponding cell  $C$  tries to find a free channel among its *nominal* channels. In case there is no available nominal channel,  $C$  calculates for each one of its neighboring cells the number of channels available that it can borrow. In case all the neighboring channels are occupied, the call cannot be satisfied and thus it is blocked. However,  $C$  borrows a channel from the cell holding the greatest number of available channels for borrowing. The borrowed channel must be returned to its original cell as soon as  $C$  terminates from this channel.

The second algorithm is more elaborated. In fact, the author assumed that the nominal cell holding the fewest nominal channels available after the suggested borrowing as the worst case. The purpose from the proposed technique is to ensure that the borrowed channel maximizes the number of available nominal channels in the worst case among the corresponding interferable cells. When a communication session is initiated in a cell  $C$ , the algorithm proceeds in the following way:

**Step 1:** The cell  $C$  seeks for an available channel in its nominal set to satisfy the call.

**Step 2:** In case no nominal channel in  $C$  is available, all the channels belonging to each adjacent cell are checked one by one: First, each channel  $i$  is verified for its availability. Then, if this condition is satisfied, the algorithm examines the interferability of nominal cells in

case the channel  $i$  is borrowed and registers the minimum number of nominal channels  $min(i)$  in the interferable cells.

**Step 3:** The borrowed channel is the one that maximizes  $min(i)$ .

The third algorithm is less complex. In fact, when a mobile host wants to acquire a channel it picks directly the first available channel without seeking the optimization of the borrowing process. The algorithm starts by splitting all the channels into many sets. Then, each cell acquires a set of channels considered as its nominal channels with respect to the channel reuse interval. When a call is initiated and there is a need to borrow a channel from another cell, the search is performed in a sequential way to satisfy the call until an available channel is encountered or all the considered cells are visited and no channel is available. In the latter case, the call is blocked.

### 2.6.3 Dynamic Channel Allocation Algorithm Based on Dynamic Load Balancing Strategy

In [15], Das *et al.* presented an algorithm based on dynamic load balancing strategy. The basic idea of this technique is to use the dynamic load repartition to reallocate channels. It is assumed that all the cells in the mobile cellular network are categorized into *hot cells* and *cold cells*.

The algorithm uses the following parameters:  $d_c$  denotes the *degree of coldness* and it represents the ratio of the number of available channels to the number of initial channels allocated to the cell. The parameter  $h$  represents a fixed threshold of the mobile radio network and it is computed by the use of the Markov model. In case  $d_c$  is strictly greater than  $h$ , the cell is considered as a cold cell. Otherwise, it is considered as a hot cell. The state of a cell and its degree of coldness are altered with time.

Three types of Mobile Hosts (MH) are depicted in each cell in the mobile cellular network depending from the holding time and the distance from the base station. These types are:

- *new*: This state is given to a *MH* recently introduced in the present cell.
- *departing*: This state is given to a mobile host when it is near the boundary of the cell causing the attenuation of its Received Signal Strength (*RSS*).
- *other*: This state is given to a *MH* that is not classified as *new* or *departing*.

The selection of the channel to borrow depends widely from the state of the *MH*. In fact, when there is a need to borrow a channel, each *hot cell* calculates the number of *MHs* that are *departing* from it. The results are recorded in the variable *NumDepart*.

The presented algorithm considers the following function for borrower and lender pairs:

$$F(B, L) = f(d_c(L), D(B, L), H(B, L)),$$

where *B* depicts a Borrower, *L* depicts a Lender and  $d_c(L)$  gives the coldness of the lender.  $D(B, L)$  gives the value of the distance between the Borrower *B* and the Lender *L* in the cell.  $H(B, L)$  gives the number of hot co-channel cells corresponding to the cold lender cell *L*. These lender cells are also non co-channel cells of the borrower cell *B*. The lender is an elected cell among the cold cells, that maximizes the value of  $F(B, L)$ . When a cell wants to borrow a channel, the algorithm executes once the following steps in the borrower cell:

**Step 1:** The array *NumDepart* is determined from the hot cells.

**Step 2:** The adjacent cells of the borrower cell *B* are considered then the cold ones and the hot ones holding a value in *NumDepart* different from zero are nominated as possible lender cells.

**Step 3:** The possible lender cells *L* are sorted decreasingly using the function  $F(B, L)$ .

**Step 4:** Channels in each cell *C* in the above sorted list are borrowed until one of the following conditions is reached:

1. The function  $F(B, L)$  is not maximum, or

2. The number of borrowed channels is identical to the value of  $NumDepart$ .

The interference is prevented by locking each lent channel.

**Step 5:** In case there is still a need to borrow channels and the sorted list is not consumed, the algorithm recommences from **Step 4**. However, in case all the calls are satisfied and there is no need to borrow more channels, the algorithm *terminates*. In the remaining case, the algorithm continues the execution of the *following steps*.

**Step 6:** The function  $F(B, L)$  of each cold cell  $L$  in the frequency reuse pattern of  $B$  is computed except for those already considered above.

**Step 7:** A channel is borrowed from the cell  $L$  with the maximum  $F(B, L)$ . Then, this channel is locked and the function  $F(B, L)$  is estimated again.

**Step 8:** **Step 7** is reiterated till all the needed channels have been borrowed.

## 2.7 Mutual Exclusion Based Dynamic Channel Allocation Algorithms

In this section, four well known algorithms for dynamic channel allocation will be explored. These algorithms are based on the mutual exclusion paradigm.

### 2.7.1 The Prakash-Shivaratri-Singhal Algorithm

In [35], Prakash *et al.* presented their algorithm based on the mutual exclusion using the *timestamp* approach. The system model used by this technique is composed by a set of Mobile Base Stations (MBS) and Mobile Hosts (MH) connected by an entirely wireless network. The communication channels in the mobile cellular network are split into two sets of *spectrum*: the first is used for the communications between base stations and the second is used for the

communications between mobile hosts and base stations. The same algorithm is used to assign channels from the different sets.

Assume that  $Allocate_i$  denotes the set of allocated channels to a  $MBS$  in a cell  $C$ .  $Busy_i$  is a subset from  $Allocate_i$  that denotes the set of used channels in cell  $C$ .  $Transfer_i$  denotes the set of channels that can be transferred from the current  $MBS$  to other  $MBS$ 's. Initially, all these sets are empty. When there is a need for a channel for a  $MH$  or for a  $MBS$ , the algorithm is executed in the following way:

**Step 1:** The  $MBS_i$  computes the value of  $Allocate_i - Busy_i - Transfer_i$ . A result different from 0, means that some channels are still free and not yet assigned. In this case the  $MBS$  affects a free channel to satisfy the request.

**Step 2:** If the result is 0, i.e. all the channels are occupied. In this case, the  $MBS_i$  sends a *request* message to the neighboring mobile base stations with a *timestamp* and waits for a *reply* from all of them. Each *reply* message contains the  $Allocate$ ,  $Busy$  and  $Transfer$  sets of the corresponding  $MBS$ . When the  $MBS_i$  receives all the *reply* messages from the neighboring base stations, it takes the union of  $Allocate_i$  and the  $Allocate$  sets received in the *reply* messages, and puts the result in  $Interfere_i$ . After that, the  $MBS_i$  computes the number of free channels  $Free_i = Spectrum - Interfere_i$ . In case  $Free_i$  is vacant, the channel request is dropped.

**Step 3:** In case  $Free_i$  is filled,  $MBS_i$  picks a channel  $h$  from  $Free_i$  and sends *transfer(h)* messages to all its neighbors to ask them for the transfer of  $h$ . The neighboring cells can agree or disagree with this transfer. If all the received answers are *agreed* messages, the channel  $h$  is affected to satisfy the request. However, if at least one reply is a *refuse* message, the next channel in  $Free_i$  is picked and the same process is executed on the set  $Free_i$ , until a free channel is found and the request is satisfied, or all the channels in  $Free_i$  were tried. In this case, the communication request is dropped.

The above Dynamic Channel Allocation algorithm is characterized by the use of timestamps

to implement the mutual exclusion concept. The requests holding earlier timestamps are prior to acquire channels. Besides, the algorithm is characterized by the fact that the transferred channels to  $MBS_i$  are added to the set  $Allocate_i$  after the termination of the communication session. These channels are conserved until they are transferred again.

### 2.7.2 The Choy and Singh Algorithm

In [12], Choy and Singh presented another distributed Dynamic Channel Allocation algorithm based on mutual exclusion. This algorithm uses the dining philosophers algorithm. In this technique, a priority is set for each Mobile Base Station ( $MBS$ ). This priority is represented by the color value of the  $MBS$ . In case there is a conflict between two base stations, the channel is assigned to the one holding the larger priority. At start, a color is affected to each base station  $MBS_i$ . All the neighbors of  $MBS_i$  should hold different colors. When two base stations want to acquire the same channel, the channel is assigned to the one holding the higher color value.

At the beginning of the execution of the algorithm, the channels are split into  $g$  groups. Besides,  $g$  instances of the dining philosophers algorithm are executed separately. When a mobile host initiates a communication session, the corresponding  $MBS$  executes the following steps to satisfy the request:

**Step 1:** The  $MBS$  looks for a local unoccupied channel to be assigned to the  $MH$ . In case it is infeasible, **Step 2** is processed.

**Step 2:** The dining philosopher algorithm is executed to permit to the  $MBS$  to transfer channels. To ensure that, channels in the group are considered to find a feasible transfer. As a result, forthcoming requests can be satisfied at the same time.

### 2.7.3 The Cao and Singhal Algorithm

In [8], Cao and Singhal presented the adaptive distributed algorithm. It is similar to the Prakash algorithm with some important differences. At the beginning of the execution of the algorithm, each cell  $C_i$  holds a precise number of primary channels  $P_i$ . This technique outperforms the Prakash *et al.* algorithm since a cell can satisfy some requests using local channels without having to communicate with its neighbors. In case a cell has to borrow a channel, it should borrow it from its richest interference neighbor. This convention avoids the situation where a neighbor needs to exploit its primary channels just after lending them.

### 2.7.4 The Boukerche-Hong-Jacob Algorithm

In [6], Boukerche, Hong and Jacob presented an efficient distributed algorithm for dynamic channel/resource allocation (DDRA) based upon the mutual exclusion paradigm. Their algorithm adopts the co-channel group interference approach instead of the co-channel interference used in the previous techniques. The idea consists of dividing the set of channels into different groups, where the number of groups equals the number of base stations in a cluster. Based on the three-coloring algorithm, any two adjacent base stations should detain dissimilar groups.

In the DDRA algorithm, the mutual exclusion issue is achieved considering a variable *competition*. A base station holding a *competition* variable with larger value, is prior to acquire a group of channels when there is a conflict. In case the values of two competition variables belonging to two different base stations are similar, one base station is elected randomly to hold the considered group. When a mobile host initiates a communication session in a cell  $C_i$ , a  $BS_i$  executes the following steps to satisfy this request:

**Step 1:** The base station  $BS_i$  selects a group  $g_j$  from the list of its unvisited groups and sends a **REQUEST** message to all the  $BS$ s in its adjacent cells asking if it can hold the selected group.

*Step 2:* Base station  $BS_i$  receives a *REPLY* message from all its neighbors. If it receives at least one *REJECT* message from a neighbor, it cannot keep the group. Consequently, it increases its competition variable by one to improve its chance to hold the group in a future round. Then, the algorithm executes again *Step 1*. In case all the groups are occupied, the call is dropped by the base station  $BS_i$ .

*Step 3:* If all the answers are *AGREE* messages,  $BS_i$  looks for a free channel  $i$  in  $g_j$ . In case there is an available channel in  $g_j$ , the base station  $BS_i$  affects the channel  $i$  to satisfy the request and re-initializes the variable *competition* to zero. At the same time,  $BS_i$  sends a *BLOCK* message to all its adjacent cells for the acquired channel  $i$ . In case no free channel in  $g_j$  is available, the algorithm executes again *Step 1*.

## 2.8 Graph Coloring Based Dynamic Channel Allocation Algorithms

The channel allocation problem can be seen as a generalized list coloring problem. In the literature, some authors designed various protocols under this concept [21]. The basic idea of this class of algorithm is to allocate channels using graph theory. The idea is to assign different colors to any two adjacent cells or neighbors in the graph.

### 2.8.1 Deterministic Distributed Approach

In [21], Garg *et al.* presented the deterministic distributed approach. In addition to the use of graph coloring, this approach utilizes the mutual exclusion. At the beginning of the execution of the algorithm, the authors supposed an initial vertex coloring of the graph that acts as a priority scheme. They considered also  $R$  colors to be exploited. When multiple neighboring base stations demand simultaneously to acquire frequencies, the base station holding the smaller color is prior to select a channel first. A double doorway synchronization process is

used in purpose to prevent base stations holding high color from waiting for long time.

The system model is represented by a graph where a node  $v_i$  refers to a cell or base station, and a Color  $r_i$  refers to a frequency.  $Busy_i$  depicts the set of busy colors in node  $v_i$ .  $Free_i$  refers to the set of free colors in node  $v_i$ .  $l_{ij}$  denotes the latest synchronization message arrived from the neighbor  $v_j$ .

In the following, the basic steps of the execution of the algorithm are listed:

- Step 1:** A node  $v_i$  has to pass the first doorway which is similar to the fact of requesting a permission to enter after receiving a message  $\neq sync_1$  (that denotes a synchronization message) from all its neighbors. As a consequence, the node  $v_i$  waits until all  $l_{ij} \neq sync_1$  then sends a message  $sync_1$  to each one of its neighbors. This guarantees to  $v_i$  to cross the first doorway.
- Step 2:** Node  $v_i$  has to wait for crossing the second doorway. As a consequence, the node  $v_i$  waits until all  $l_{ij} \neq sync_2$  then sends a message  $sync_2$  to each one of its neighbors. This guarantees to  $v_i$  to cross the second doorway.
- Step 3:** After having passing both doorways, a node  $v_i$  has to wait to get the privilege  $pr$  from its neighbors that are also past both doorways.
- Step 4:** Node  $v_i$  first selects the set of frequencies  $S$  from  $Free_i$ . Then, it informs each one of its neighbors by sending them a message  $conf(S)$ . Finally, it waits until reception of an acknowledgement from all its neighbors.
- Step 5:** Node  $v_i$  updates the sets  $Busy_i$  and  $Free_i$  after reception of an acknowledgment from all its neighbors:  $Busy_i = Busy_i \cup S$  and  $Free_i = Free_i - S$ .
- Step 6:** Node  $v_i$  sends a message  $sync_3$  to each neighbor.

### 2.8.2 Randomized Distributed Approach

In [21], Garg *et al.* presented also the Randomized distributed approach. This approach affects colors to a node in a way that prevents conflicts with the neighbors. This technique admits that the list coloring is enough greater than the total number of colors required by a node. At the moment that a node  $v_i$  needs several colors, it selects randomly some colors from  $Free_i$ . Then a check process is performed with its neighbors and the colors are assigned in the following way:

**Step 1:** A node  $v_i$  selects randomly a set  $S$  of colors from  $Free_i$ . Update the sets  $Busy_i$  and  $Free_i$  in the following way:  $Busy_i = Busy_i \cup S$  and  $Free_i = Free_i - S$ .

**Step 2:** Node  $v_i$  sends to its neighboring cells a  $pick(S)$  message to verify if they are in conflict. Then it waits for a reply from each neighbor.

**Step 3:** Depending on the replies from the neighbors, if at least one neighbor answered “NO” for a channel  $i$  in  $S$  then the following updates are performed:

$$Busy_i = Busy_i - i$$

$$Free_i = Free_i \cup i$$

$$S = S - i$$

**Step 4:** Node  $v_i$  informs each one of its neighbors by sending a message  $conf(S)$ .

## 2.9 Fault-Tolerant Channel Allocation Algorithms

In a mobile cellular network, the exchange of information related to the usage of channels between the base stations of the neighboring cells is an important action taken by most of the techniques presented to solve the channel allocation problem. In some cases, a response message from a neighboring base station is absolutely required otherwise the algorithm cannot proceed its execution. Consequently, some base stations would wait indefinitely in case a base

station failure or a link failure between two neighboring base stations occurs. This results in the necessity to improve the performance of the channel allocation algorithms by integrating the concept of fault-tolerance in the generic dynamic channel allocation schemes.

### 2.9.1 Prakash, Shivaratri and Singhal's Channel Allocation Protocol

The Prakash, Shivaratri and Singhal's generic dynamic channel allocation (DCA) scheme was presented in a previous section. Their enhanced algorithm is presented in [35]. In the following we will demonstrate the behavior of the algorithm when a mobile host fails, the link between neighboring base stations fails or a base station fails.

- **Case 1:** Failure of a mobile host. When a mobile host initiates a communication session and then fails, the corresponding base station ( $BS_i$ ) assumes that this communication session has terminated and executes the termination actions by removing the occupied channel from the set  $Busy_i$  and returning it to the set  $Free_i$ .
- **Case 2:** Failure of a link between adjacent base stations. This failure is handled easily by the underlying network protocol. In fact, in case there is a direct link failure between two adjacent base stations, an alternative path can be found. In the worst case, there is a failure of all the wired links. In this case, the  $8kb/s$  control channel is still available.
- **Case 3:** Failure of a base station  $BS_i$ . This case needs more treatments. In fact, a timer is set by each base station to control the exchange of information. In case  $BS_j$  detects a timeout after sending a **REPLY** message, it considers that a **REPLY** message from  $BS_i$  has been received and thus  $Allocate_i$ ,  $Busy_i$  and  $Transfer_i$  are empty sets. In case  $BS_j$  detects a timeout after sending a **TRANSFER(k)** message, it considers that an **AGREED(k)** message has been received from the base station  $BS_i$ . When  $BS_j$  detects that  $BS_i$  has failed, it deletes  $BS_i$  from its list of neighbors, and in the future it will not send messages to this failed base station. As soon as  $BS_i$  recovers from a failure, it notifies all its neighboring cells. These cells, add  $BS_i$  to their neighboring list.

### 2.9.2 Cao and Singhal's Fault-Tolerant Channel Allocation Protocol

The Cao and Singhal's dynamic channel allocation algorithm presented in the previous section has been enhanced in [8] by integrating the fault-tolerance concept. In this algorithm, the three types of failures presented in the Prakash, Shivaratri and Singhalare algorithm (MH failure, link failure and BS failure) are considered in this scheme. The treatment of these failures is quit similar with some difference in the case of base station failure.

The cause of this difference is related to the distinct system models that have been considered. In fact, in the Prakash, Shivaratri and Singhal's model, each cell has six adjacent cells and any channel can be reused by a cell as long as it is not used by one of its neighbors. However, the system model in the Cao and Singhal's approach considers a co-channel reuse distance given by  $3\sqrt{3}R$  where  $R$  is the radius of a cell  $C$ . Consequently, one cell can have more than thirty neighboring cells. To insure the fault tolerance in this schema, the following property is introduced: if the same channel  $i$  is requested by two cells  $C_i$  and  $C_j$  in a cluster, then it exists at least one common cell for the previous cells that represents an interference-nominated cell of channel  $i$ . As a result, when a borrower sends a request, it can take a decision even though not all the response messages have been received from the interference neighboring cells. Thus, when a *BS* fails, the process of borrowing launched by the other *BSs* is not usually affected. This allows the channel allocation scheme to satisfy the concept of fault tolerance.

### 2.9.3 Yang, Jiang, Manivannan and Singhal's Fault-Tolerant Enhanced Channel Allocation Scheme

In [42], a fault-tolerant distributed dynamic channel allocation scheme for cellular networks is presented. This technique provides a useful way to tolerate the mobile service stations failure, link failure, and deals with network congestion. At the same time it permits an efficient reuse of channels.

The algorithm deals with a 3-cell cluster model where a channel cannot be used by more than one cell in a cluster of three mutually adjacent cells. Every cell has six neighbors where each one is identified by a unique id between 1 and 6 ( $nb_1, \dots, nb_6$ ). Five groups are formed from these neighboring cells:

- Group\_1:  $nb_1, nb_4$
- Group\_2:  $nb_2, nb_5$
- Group\_3:  $nb_3, nb_6$
- Group\_4:  $nb_1, nb_3, nb_5$
- Group\_5:  $nb_2, nb_4, nb_6$

All the channels are ordered such that the least order is assigned to the channel holding the smallest frequency and the largest order is assigned to the one having the greatest frequency.

When a communication is initiated in a cell  $C_i$ , the algorithm proceeds on the following way:

**Step 1:**  $C_i$  starts by seeking the set of the unused channels  $f$  that has been assigned to it.

**Step 2:** In case the set is not empty,  $C_i$  picks a channel  $f$  from this set with the highest order and it is allocated to the MH to satisfy the request. Then,  $i$  is added to the set  $Busy_i$  of the cell  $C_i$ .

**Step 3:** In the other case, when no free channel in  $C_i$  is found, a *request* message is sent to all its adjacent cells to borrow a channel (with the lowest order if possible) and a timer is initiated.

**Step 4:**  $C_i$  collects the replies. Two cases are then distinguished:

**Step 5:** In case all the neighbors answer the request,  $C_i$  looks for the channels that are not allocated neither to itself nor to its six neighbors. If at least one channel  $f$  satisfies the

previous condition,  $f$  is allocated to the call and added to the sets  $Allocate_i$  and  $Busy_i$ . Otherwise, when the computed set is empty,  $C_i$  seeks the set of the unused channels by itself and each one of its neighbors. If the set is not empty, a **transfer(f)** message is sent to the adjacent cells that acquire  $f$ . If all the concerned neighbors send an **agree** message, then  $C_i$  is allowed to use  $f$  which is added to  $Allocate_i$  and  $Busy_i$ . Besides,  $C_i$  sends a **release(f)** message to the concerned cells. However, if not all the agreed messages are received, a **keep(f)** message is sent to the cells that have sent an **agree(f)** message and another candidate channel is selected.

**Step 6:** In case the timer expires and some replies are not yet received, the number of *reply* messages is recorded and an appropriate action is taken according to this number to compute the set of available channel candidates to be borrowed.

**Step 7:** If this set is not empty, a channel  $f$  is picked and a **transfer(f)** message is sent to all the adjacent cells that hold  $f$ . However, if the set is vacant, the call is dropped.

Moreover, the algorithm uses two other kinds of sets:  $Granted_i(f)$  and  $Lent_i(f)$ .  $Granted_i(f)$  records the set of cells that have received an **agree(f)** message from  $C_i$ , while  $Lent_i(f)$  registers the set of cells that have sent a **release(f)** message to  $C_i$ .

Supposing a channel  $f$  is borrowed for a cell  $C_i$ , the sets  $Granted_i(f)$  and  $Lent_i(f)$  are set to empty.

In the proposed algorithm, the number of reject messages is largely minimized while trying to borrow a channel. In fact, a channel can be borrowed even when not all the neighbors replied to a request for borrow. Besides, a cell is allowed to lend the same channel to many borrowers at the same time providing that they are not adjacent. All this leads to an efficient utilization of the channels due to its fault tolerance side.

## **Chapter 3**

# **Design and Performance Evaluation of a QoS-Based Dynamic Channel Allocation Protocol for Wireless and Mobile Networks**

In recent years, we have witnessed a growing interest in the study of channel allocation and hand-off strategies for wireless networks to ensure continuous services that guarantee QoS to mobile users. To the best of our knowledge, most of the proposed channel allocation schemes as described in Chapter 2 do not take the QoS provisioning into account. In this chapter, we propose a distributed algorithm for dynamic channel allocation with an efficient adaptive channel reservation schema providing continuous QoS support. To acquire the low dropping rate, a proper number of channels in the congested cells is reserved for the handoff calls. This number of reserved channels is related to the wireless data traffic network. We present our QoS-Based dynamic channel allocation protocol, and its performance evaluation, and discuss our experimental results we have obtained using realistic scenarios.

To guarantee the QoS provisions, the proposed algorithm dynamically adjusts the number

of reserved channels for the *handoff* in terms of the traffic situation while prohibiting to a base station to be a host of a group of channels. Besides, in addition to its fast response time and its reduced denial rate under a very high system load, it ensures a very low dropping rate. Our algorithm is executed at each base station and the control of channel usage does not require a Mobile Switch Center (MSC) since the neighboring base stations cooperate together by exchanging the channel usage information and assign available channels at run time. In order to reduce the connection dropping rate, we propose to use an adaptive channel reservation mechanism.

### 3.1 System Model

The system model considered in this approach consists of a mobile cellular network constituted of a conglomeration of cell clusters. Every cluster assembles seven cells and each base station belonging to a cell processes a channel allocation mechanism. The set of channels in the mobile cellular system counts  $M$  channels that are partitioned equitably into three groups. Any base station can acquire a channel group as long as no one of its adjacent cells is holding this group. The three-coloring theorem permits an assignment of groups to base stations such that the previous condition is satisfied. Consequently, all the base stations in the mobile cellular network are able to acquire a group at the same time considering the mutual exclusion concept.

In our algorithm, we deal only with the adjacent channel interference since all the channels are orthogonal. Two base stations in two adjacent cells that accedes the same channel engender co-channel interference. In order to escape from this problem and permit channel reuse efficiently, a channel can be accessed simultaneously by two cells  $C_i$  and  $C_j$  as long as they are sufficiently distant. On purpose to avoid co-channel interference in our system model, channel groups are allocated simultaneously to two cells provided that they are not adjacent.

In the mobile cellular network, two categories of channels can be depicted: control chan-

nels and communication channels. *Forward* setup channels from MHs to BSs and *reverse* setup channels from BSs to MHs are parts from the control channels. The establishment of a communication session requires the collaboration between three components. First, a wireless channel is needed to provide a connection between the originator MH and its correspondent BS. Second, wired links are necessary between the source BS and the destination BS. Third, a wireless channel is deserved between the destination BS and the destination MH. Whenever a wireless channel cannot be acquired, the communication session cannot be established and hence it is dropped. At the end of a communication session, the occupied wireless channels are liberated for future use by other MHs.

Before we proceed further, we shall describe the original distributed dynamic channel allocation protocol proposed by Boukerche *et al.* [6].

### 3.2 Distributed Dynamic Channel Allocation Algorithm

In the distributed channel allocation algorithm, which we refer to as DDRA [6], all channels are partitioned into equal sized groups where the number of groups equals to three. As Figure.3.1 shows, our system model assumes that a cluster is constructed from seven cells.

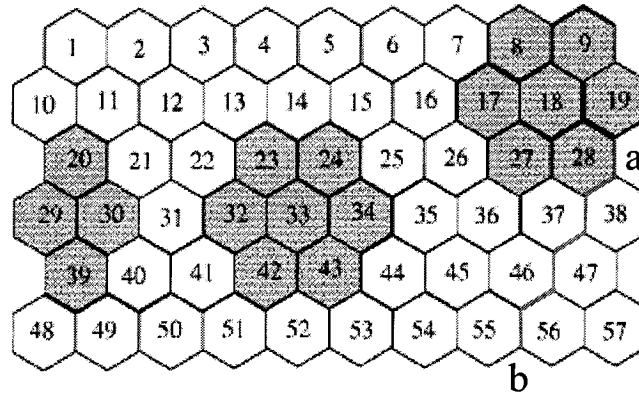


Figure 3.1: Cellular System

Any base station can acquire a channel group at any time as long as no one of its neighbors is already holding it. Since we use a critical section in our scheme to protect every channel group, only one base station in a cluster can acquire a channel group at a given time. In fact, a BS is able to acquire a channel group only after it gets permission from all its neighbors. The channel allocation algorithm can be run concurrently by all the base stations in the wireless network. As opposed to earlier dynamic channel assignment algorithms, in this scheme, there is no channel transfer between two base stations. Furthermore, this scheme permits the exchange of a lower number of messages between base stations, and thus decreasing the global overhead.

Suppose that  $MH_i$  requests a free channel from the base station  $B_i$ .  $B_i$  picks up a free group  $g_j$  which has not yet been visited by  $B_i$  and sends a *Request* message to all neighboring base stations. As long as it receives the *Reject* message from one neighbor,  $B_i$  cannot hold the group  $g_j$  and then tries another group. Upon receiving a permission from all of its neighbors,  $B_i$  seeks for a free channel  $Ch_f$  from the group  $g_j$ . If the group  $g_j$  has no free channel,  $B_i$  keeps trying group by group until it has found a free channel or visited all the available groups.

When  $B_j$  receives a *Request* message from its neighbor  $B_i$ , if  $B_j$  is holding the group  $g_j$ , then it sends a *Reject* message to  $B_i$ . If  $B_j$  has already requested the same group  $g_j$ , a competition scheme is implemented to determine which one,  $B_i$  or  $B_j$ , is the winner. If  $B_j$  is the winner, it transmits a *Reject* message to  $B_i$ . Otherwise,  $B_j$  sends an *Agree* message to  $B_i$ . If  $B_j$  is neither using nor requesting the group  $g_j$ ,  $B_j$  delivers an *Agree* message to  $B_i$ .

Upon receipt of a *Block* message for a channel  $Ch_f$ ,  $B_j$  records the channel  $Ch_f$  as blocked in its record.

Upon receipt of a *Free* message for a channel  $Ch_f$ ,  $B_j$  marks the channel  $Ch_f$  free in its record.

Upon receipt of a *Release* message from  $MH_i$ ,  $B_i$  broadcasts a *Free* message to all of its neighboring base stations.

Upon receipt of a *Free* message for a group  $g_j$  with a *Block* message or  $Ch_f$ ,  $B_j$  updates

its group usage table to reflect the new state.

### 3.2.1 Mutual Exclusion Paradigm

The DDRA algorithm runs on all base stations. Two neighboring base stations might probably request the same group  $g_j$  concurrently. To perform the mutual exclusion, a variable *competition* is employed to count the number of competitions with other base stations. In each competition, the base station whose *competition* is greater will be the winner and gets the group  $g_j$ . The looser will increase *competition* in order to enhance its competition capability in the next competition round. When the two competitors have the same *competition* number, the base station *ids* are used to select a leader which randomly chooses a winner from those competitors.

### 3.2.2 Storage Information for Allocation Channels

Each base station  $B_i$  has a simple two-dimensional table that contains the information about channel usage for itself and its neighbors. The size of the table is determined by the number of channels in the wireless communication system and the number of cells in a neighborhood. In our model a neighborhood consists of 7 cells, a channel-requesting cell and its neighbors. For example, in Figure 1, if cell 33 is using channel  $Ch_f$ , then the neighboring cells 23, 24, 32, 34, 42 and 43 cannot use  $Ch_f$ . Most mobile systems have 666 channels, including 42 control channels, so that there are 624 voice channels. In this case,  $N$ , the size of the table in the cell  $C_i$ , is about  $N = 624k$ , where  $k$  is the number of cells in a cluster bits. To run this channel assignment algorithm, a base station needs to maintain only the table and several variables. Only a partition of the table can be accessed by a base station.

### 3.3 QoS-based Distributed Dynamic Channel Allocation Algorithm

#### 3.3.1 A Timer for Channel Reservation

In our proposed algorithm, each base station adaptively reserves some channels for QoS-provisions through periodically checking the recent connection dropping rate. Therefore, each base station sets up a timer. If the predetermined amount of time elapses, the base station counts the statistical connection dropping rate. If the connection dropping rate is beyond an expected value, the base station increases the number of reservation channels. Otherwise, the base station decreases the number of reservation channels.

The timeout value of the timer is selected according to the traffic distribution and the empirical practices. If the traffic distribution is uniform, the timeout value may be bigger. If the traffic distribution is non-uniform and changes dramatically, the timeout value has to be smaller because the base station needs to be aware of the instant connection dropping rate.

#### 3.3.2 Prediction of the Moving Direction of MHs

In mobile cellular network, some cells are very congested. For these cells, it is impossible that their *BSs* reserve enough free channels to hold all the incoming *handoffs*. To lower the failures of such these incoming *handoffs*, it is necessary that *BSs* are aware of the moving direction of *MHs*. Some call requests from the *MH* which may move from a non-congested cell to a congested cell are rejected. For example, we assume three neighboring *BSs*:  $B_i$ ,  $B_j$  and  $B_k$ , where  $B_j$  is congested and the others are not.  $B_i$  and  $B_k$  may block call requests from the *MHs* which may enter  $B_j$  but accept call requests from the *MHs* which may enter those non-congested cells. Therefore, *BSs* have to be able to locate these *MHs* and predict the moving direction.

There are some practical technologies to satisfy the previous condition. If the global po-

sition system (*GPS*) is available, the *BS* may locate the *MH* and predict the moving direction of the *MH* based on the geographical information. An alternative technology that *BSs* can use is to cooperate with each other. A *BS* can obtain the distance to a *MH* through measuring the signal strength or the Time of Arrival (*ToA*) [7, 22]. Three neighboring *BSs* can locate the *MH* by computing the intersection of three circles. If *BSs* are able to locate *MHs*, they can estimate the *MH's* moving direction.

### 3.3.3 Dynamic Channel Reservation for *Handoffs*

The *handoff* is a major factor causing connection dropping rate (*CDR*). During a *handoff*, if the destination *BS* cannot assign a free channel to the incoming *handoff*, the ongoing call will be dropped. The more *handoffs* fail, the higher the *CDR* becomes. Reserving some channels in all cells is in favor of the success of *handoff* and the fall of *CDR*. However, it will waste some wireless bandwidth and make the connection blocking rate (*CBR*) higher because more new connection requests will be blocked due to the lack of channels. In order to enhance the QoS provision, an efficient scheme to reserve channels is indispensable for the substantial decline of *CDR* under the slight sacrifice of *CBR*. In terms of the expected QoS guarantees, our proposed algorithm pre-specifies a target value for each cell which is a system parameter and tries to keep the *CDR* below the target value. If a fixed number of channels are reserved or channel reservation is made solely depending on the prediction, it easily happens that a large amount of wireless channels are wasted or the *CDR* is always beyond the target value.

Therefore, our proposed scheme dynamically adjusts the amount of reserved channels according to the instant traffic situation. Each *BS* periodically calculates its recent average *CDR*. Once the average *CDR* is larger than the target value, the number of reserved channels is increased. Otherwise, the number of reserved channels is decreased. Through this way, the long-term *CDR* is around the pre-specified target value.

The target value  $P_{target}$  of *CDR* is independently selected by each *BS*. Each *BS* periodically calculates the ratio  $\eta$  of the number of new call requests to incoming *handoffs*. For those cells

which has the big  $\eta$ , a big target value  $P_{target}$  is used because of lots of new connection requests. If the target value  $P_{target}$  in these cells is small, the connection blocking rate is too high to be acceptable due to too much channel reservation for incoming *handoffs*. By contrast, a small target value  $P_{target}$  is employed in those cells with a small  $\eta$  since the number of incoming *handoffs* is relatively small.

Reserving channels for *handoffs* is an approach executed in the destination cell for decreasing the *CDR*. For those heavily congested cells, we also employ an approach in the source cells which are neighboring. As aforementioned, the *BS* has the capability of locating a *MH* and predicting a *MH*'s moving direction. If a new connection request from a *MH* which goes ahead to a congested cell (a hot cell), it will possibly be blocked by the *BS*. If the *MH* will enter a cold neighboring cell, the new call request is accepted by the *BS*. To implement this idea, a parameter  $\gamma\%$  is selected to determine the blocking possibility of a new connection call which is going to *handoff* to a heavily congested cell. A *BS* knows which neighboring *BSs* are heavily congested through two ways. Either the heavily congested *BSs* inform their neighbors by sending messages. Or, the source *BS* monitors the outgoing *handoffs*. The first method increases the message overhead and consumes an amount of wired bandwidth between *BSs*. While in the second method, the outgoing *handoffs* to those hot destination cells fail much more possibly due to the lack of channels. If the outgoing *CDR* for these *MHs*, which leads to a destination cell, is much higher than the average outgoing *CDR* for all six neighbor, the *BS* considers the destination cell is heavily congested. If more than one neighboring cells are heavily congested, the second method does not work well because their outgoing *CDRs* almost equal the average *CDR*. In this case, our proposed algorithm employs the first method.

## 3.4 Formal Description of the QoS-based Dynamic Channel Allocation Algorithm

In this section, we formally describe the proposed QoS-based dynamic channel allocation algorithm running on a base station  $B_i$  by means of pseudo-code:

**Case A.** Mobile host  $MH_i$  requests a free channel  $Ch_f$  from  $B_i$ .

Step A0.  $B_i$  initializes the group usage table;

and sets  $competition(B_i) \leftarrow 0$ .

Step A1. From its group usage table,  $B_i$  selects a group  $g_j$  which has not yet been visited.

Step A2. **If**  $B_i$  has visited all groups **then**

send "Wireless network is busy" to  $MH_i$  ;

**go to step 9.**

Step A3.  $g_j \leftarrow$  Block.

Step A4.  $B_i$  requests the group  $g_j$  from all of its neighbors with a timer.

Step A5. **Wait until**  $B_i$  has the *REPLY* messages from all its neighbors.

Step A6. **If**  $B_i$  cannot get a free group  $g_j$  **then**

$competition(B_i) \leftarrow competition(B_i) + 1$ ;

$g_j \leftarrow$  "Free";

**go to step 1.**

Step A7.  $B_i$  searches for a free channel  $Ch_f$  in  $g_j$  .

Step A8. **If**  $B_i$  finds  $Ch_f \in g_j$  **then**

$Ch_f \leftarrow$  "Block";  $g_j \leftarrow$  "Free";

$B_i$  sends the message "Make  $g_j$  free"

and "Block  $Ch_f$ " to all of its neighbors;

$competition(B_i) \leftarrow 0$ ;

**else**  $g_j \leftarrow$  "Free"; group usage table ( $g_i$ )  $\leftarrow$  "Visited";

**go to step 1.**

Step A9. Exit

**Case B.**  $B_i$  receives a request from  $B_j$  .

Step B1. **If** the requested  $g_j$  is being used at  $B_i$  **then** reply "Reject".

Step B2. **If**  $g_j$  is "Free" **then**

reply "Agree".

Change the owner of  $g_j$  to "Others".

Step B3. **If** the requested  $g_j$  is also requested by  $B_i$  **then**

**If**  $\text{competition}(B_j) > \text{competition}(B_i)$  **then**

reply "Agree" to the base station  $B_j$  ;

$B_i \leftarrow$  "False" to make "Fail" for  $B_i$ ;

**else If**  $\text{competition}(B_j) < \text{competition}(B_i)$  **then**

reply "Reject" to the base station  $B_j$  ;

$B_i \leftarrow$  "Agree" to make "Success" for  $B_i$ ;

**else If**  $\text{competition}(B_i) == \text{competition}(B_j)$  **then**

**If**  $\text{priority}(B_j) > \text{priority}(B_i)$  **then**

randomly select the winner for  $g_j$  ;

**If**  $B_i$  was selected **then**

send "Reject" to base station  $B_j$  ;

$B_i \leftarrow$  Agree;

**else**

Send "Agree" to base station  $B_j$  ;

$B_i \leftarrow$  "False".

**Case C.**  $B_i$  receives a new connection request requiring  $n$  channels from a MH.

Step C1.  $B_i$  invokes the algorithm in Case A to request  $n$  channels.

Step C2. **If**  $B_i$  gets  $n$  free channels **then**

The new connection is accepted.

**else**

The new connection is blocked.

**Case D.**  $B_i$  receives an incoming *handoff* requiring  $n$  channels.

Step D1. **If** the available reserved channels on  $B_i > n$

**then** The incoming *handoff* is accepted.

**else**

The incoming *handoff* is dropped.

Step D2. **If** the reserved channels  $<$  a specified value  $R_i$

**then**  $B_i$  invokes the algorithm in Case A to reserve available channels up to  $R_i$

**Case E.** The timer on  $B_i$  for channel reservation is timeout

or an incoming *handoff* is dropped.

Step E1. **If** the CDR > the target value  $P_{target}$  **then**

**If**  $R_i < R_{max}$  **then**

$$R_i = R_i + \pi;$$

Invoke the algorithm in Case A to request  $\pi$  channels for incoming *handoffs*,

Step E2. **If** the CDR < the target value  $P_{target}$  **then**

**If**  $R_i > R_{min}$  **then**

$$R_i = R_i - \pi;$$

Invoke the algorithm in Case G to release  $\pi$  channels

**Case F.**  $B_i$  receives a message for “Block  $Ch_f$ ” and “Free  $g_i$ ”.

Step F1. Channel  $Ch_f \leftarrow$  “Block”.

Step F2. Group  $g_i \leftarrow$  “Free”.

**Case G.**  $B_i$  receives a “Release  $Ch_f$ ” message from  $MH_k$ .

Step G1. Channel  $Ch_f \leftarrow$  “Free”.

Step G2. Broadcast “ $Ch_f$  is free” to all of  $B_i$ 's neighbors.

**Case H.**  $B_i$  receives a “Free  $Ch_f$ ” message from  $B_j$ .

Step H1. Channel  $Ch_f \leftarrow$  “Free”.

**Case I.**  $B_i$  receives a “Free  $g_i$ ” message from  $B_j$ .

Step I1. Group  $g_i \leftarrow$  “Free”.

**Case J.**  $B_i$  receives a “heavily congested” message from  $B_j$

Step J1. Block  $\gamma$  percent of connections from the MH which may lead to  $C_j$ .

**Case K.**  $B_i$  receives a “becoming cold” message from  $B_j$

Step K1. Stop blocking connections from the MH may lead to  $C_j$

## 3.5 Simulation Experiments

In this section, we shall present the simulation we have conducted to evaluate the performance of our QoS-based dynamic channel allocation protocol. In the course of our simulation, we have simulated a wireless network of 57 cells and we varied the number of channels between 50, 100, 200, 400 and 600 wireless channels using both *uniform* and *non-uniform* call

arrival distributions. With respect to the uniform distribution, we have assumed that each cell has the same channel demand, i.e., call arrivals to a mobile host are equal to  $\lambda$ , and the service time is equal to  $\mu$ . Whereas in the context of a non-uniform distribution, one third of the cells were supposed to be heavily loaded. The call arrival rate of the heavily loaded cells is considered to be as five times as that of the lightly loaded cells (cold cells). Two strategies have been established to decide upon the cells in order to choose the heavily loaded ones. The first strategy selects the evenly distributed cells over the region as heavily loaded cells, and the second strategy picks randomly cells as heavily loaded cells.

When two base stations communicate together, they need to exchange the following information: *Packet type*, *Source BS*, *Destination BS*, *Competition\_num* and *Channel\_num*. In our algorithm, five types of packets are considered. Consequently, 3 bits are enough to distinguish a packet. Thus, the size of a message is at least  $3 + 2\log(N_n) + \log(N_n + \log(N_c))$  bits where  $N_n$  is the number of neighboring cells and  $N_c$  is the number of channels in the system. The simulation parameters of our experiments are listed in Tab. 3.1. Among these parameters, the mean service time per communication session is estimated to 180ms. A Poisson process was applied to model the call duration that is approximated to 180ms, and a Normal process was employed to determine the call arrival rate which is set to  $\lambda$ . For a heavily loaded cell, the value of the call arrival rates is maintained to five times  $\lambda$ . Litter's Law<sup>1</sup> was adopted to permit the variation of the parameter  $\lambda$  in order to obtain certain percentage of the system load. Moreover, we assumed that the delay of the average one-way communication between two *BSs* is 0.03ms and that a packet needs 2ms to be processed. To evaluate the QoS performance, we choose a series of parameters to implement the adaptive channel reservation. We choose the target value of *CDR* to be 0.015, the system parameter  $\gamma$  to be 0.5,  $R_{min}$  to be 0,  $R_{max}$  to be 20,  $\pi$  to be 2 and the timeout threshold to be 15 minutes.

Simultaneously, we assume other parameters. For a *MH* under the low mobility, an inter-

---

<sup>1</sup>Little's Law allows one to compute the average number of jobs (calls) in a stable system given the mean arrival rate and mean service time [29]. This is converted to a percentage using total channels available and number of cells in a cluster.

Table 3.1: Simulation Parameters

Parameters	Value
Number of cells	57
Number of channels	50, 100, 200, 400, 600
Mean service time/session	180ms
Arrival rate in normal cell	$\lambda$
Arrival rate into a hot cell	$5\lambda$
Mean delay to processing a message	2.0ms
Mean message delay between BSs	0.03ms
Probability of low user mobility	0.15
Probability of high user mobility	0.85
Mean inter- <i>handoff</i> rate in a normal cell	6/60
Mean inter- <i>handoff</i> rate in a hot cell	6/180
Mean rate of change from normal state to hot state	1/1800 s
Mean rate of change from hot state to normal state	1/1800 s
Service time per data/video/voice connection	6/6/4 minutes
Probability of data/video/voice connection	0.2/0.2/0.6
Number of channels for each data/video/voice connection	2/4/1
The target CDR of the system	0.015
the system parameter $\gamma$	0.5
The minimum amount of reserved channels $R_{min}$	0
The maximum amount of reserved channels $R_{max}$	20
The step amount of reserved channels $\pi$	2
The timeout threshold of the timer	15 minutes

*handoff* takes place with probability 0.15. But for a *MH* under the high mobility, an inter-*handoff* happens with probability 0.85. Since the QoS is mainly required by the multimedia traffics: *data*, *video* and *voice*, we choose specific parameters for them. Data, video and voice have their probabilities respectively 0.2, 0.2 and 0.6 to be requested and demand respectively 2, 4 and 1 channels. Moreover, they have a mean service time of 6, 6 and 4 minutes respectively. Finally, the communication line between two base stations is estimated to have 1 Mbit/s of bandwidth.

A great number of test cases were considered to supply efficient experimental evaluation of our channel assignment strategy.

### 3.5.1 Performance Metrics

In order to establish the performance of our QoS-based channel allocation algorithm and compare it to the original DDRA [6] (which we refer to as generic DDRA), four performance metrics have been considered: *dropping rate*, *denial rate*, *acquisition time* and *message complexity*.

- The *dropping rate* denotes the percentage of ongoing calls being dropped during inter-handoffs. It is an important performance metric to measure Quality of Service: the less *dropping rate*, the better the QoS performance.
- The *denial rate* gives the amount of the failed requests and thus the blocked calls. It is agreed that the lower the denial rate, the efficient the channel selection algorithm.
- The *acquisition time* refers to the total time needed for a BS to allocate a channel to a MH. It measures the efficiency of the channel acquisition.
- The *message complexity* determines the number of messages exchanged in order to respond to a channel request.

### 3.5.2 Experimental Results

Figures. 3.2(a), 3.2(b), 3.2(c), 3.2(d) and 3.2(e) depict respectively the *dropping rate* of the generic DDRA (i.e., without the QoS) and the QoS-based DDRA. From these figures, we can infer that the more wireless channels the less the *dropping rate* regardless of generic DDRA or QoS-based DDRA. For the generic DDRA, the dropping rate varies around 25% with 50 wireless channels and about 17% with 100 wireless channels. It decreases with the increase of the number of channels to reach around 7% with 400 wireless channels and around 6% with 600 wireless channels. For the QoS-based DDRA, the dropping rate reaches its minimum at 5% with 50 wireless channels, 3% with 100 wireless channels and it goes even to 2% or less with 400 wireless channels and 600 wireless channels. Obviously, the performance *dropping rate* is improved largely by the QoS-based DDRA. In fact, it is reduced by up to 25% at a system load of 100% with 50 wireless channels. Simultaneously, the *dropping rate* of the QoS-based DDRA is almost stable as the system load increases but that of the generic DDRA increments dramatically with the increasing system load.

Figures. 3.3(a), 3.3(b), 3.3(c), 3.3(d) and 3.3(e) portray the denial rate of the generic DDRA and the QoS-based DDRA. Since the QoS-based DDRA reserves some wireless channels for inter-handoff, the denial rate becomes higher because the number of wireless channels available to the new call requests decreases. The average addition of denial rate is about 5% for 50 wireless channels and it increases to reach 12% with the increase of the number of wireless channels. In order to guarantee the QoS of ongoing calls, it is desirable to sacrifice a little denial rate for the new call requests. From these figures, the denial rate rises with the increasing system load and goes down as total wireless channels increments.

Figures. 3.4(a), 3.4(b), 3.4(c), 3.4(d) and 3.4(e) depict the acquisition time. We can conclude that the QoS-based DDRA has a larger acquisition time which is about 0.015s greater than that of the generic DDRA for a system load of 100% and 100 wireless channels. This conclusion is in line with what the QoS-based DDRA scheme assume, i.e. some wireless channels are reserved for inter-handoff. In fact, the decreased number of wireless channels that were

reserved for *inter-handoff* results in a larger time spent to search an available wireless channel for new call requests. The difference on the acquisition time decreases with the increase of the number of wireless channels to reach almost 0.005s with 600 wireless channels.

The last performance metric is message complexity. As we can see in Figures. 3.5(a), 3.5(b), 3.5(c), 3.5(d) and 3.5(e), our QoS-based dynamic channel allocation protocol has less message complexity. In fact, the procedure of reserving some wireless channels is running on the background. Wireless channels for an *inter-handoff* call can be acquired directly. Hence, the average of message complexity for the new calls and *inter-handoff* calls decreases. From our simulations, it is reduced by up to 18. From these figures we can see that the message complexity goes down as the system load increases. A base station allocates a free channel to a call through two search steps: first searches an available group and then searches a free channel. When the system load increases, it is more possible that neighbors of a base station are occupying channel groups. Hence, the base station just checks whether the channel groups are available or not. If all channel groups are occupied by its neighbors, it will drop the call without entering the channel searching.

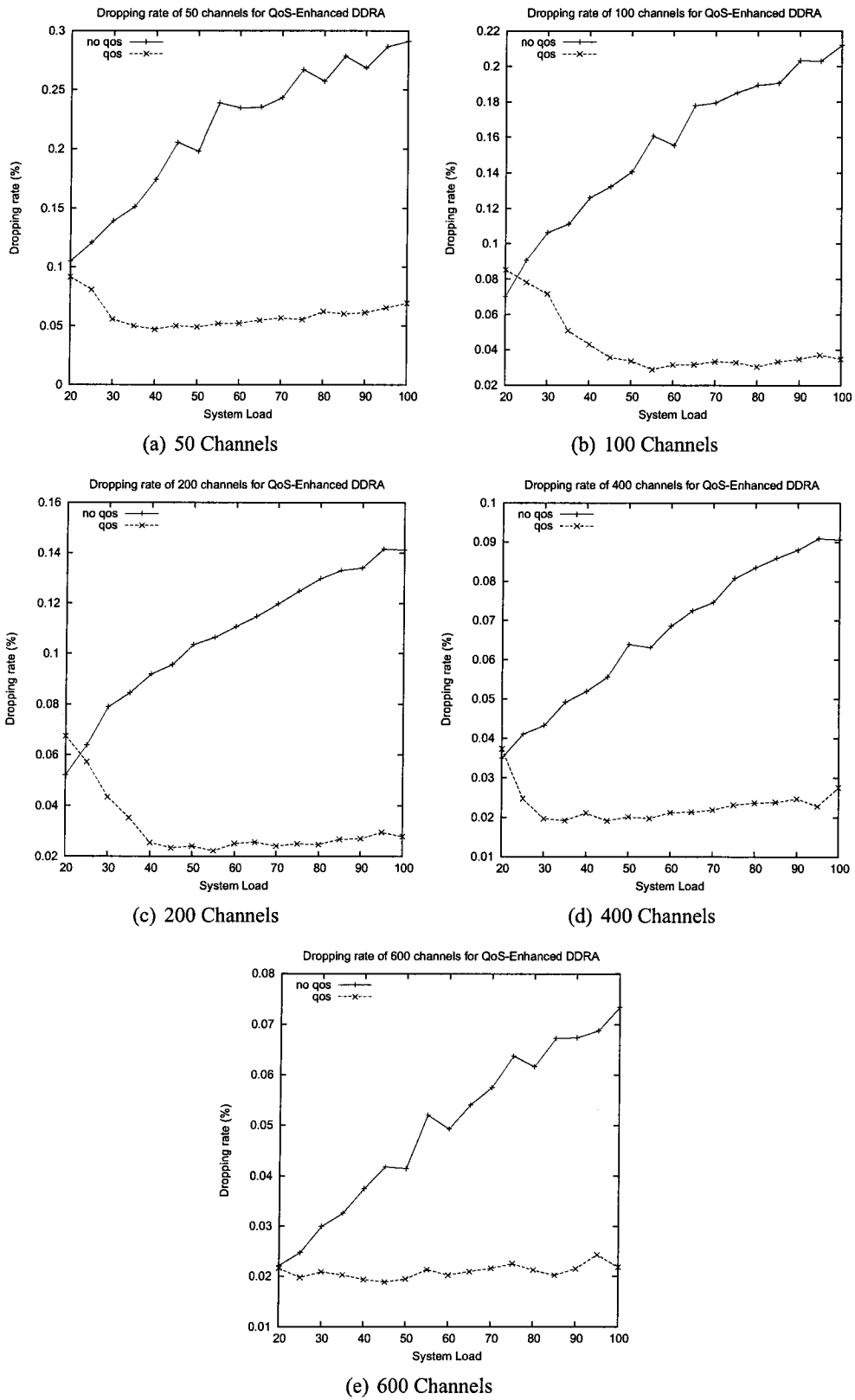


Figure 3.2: Dropping Rate

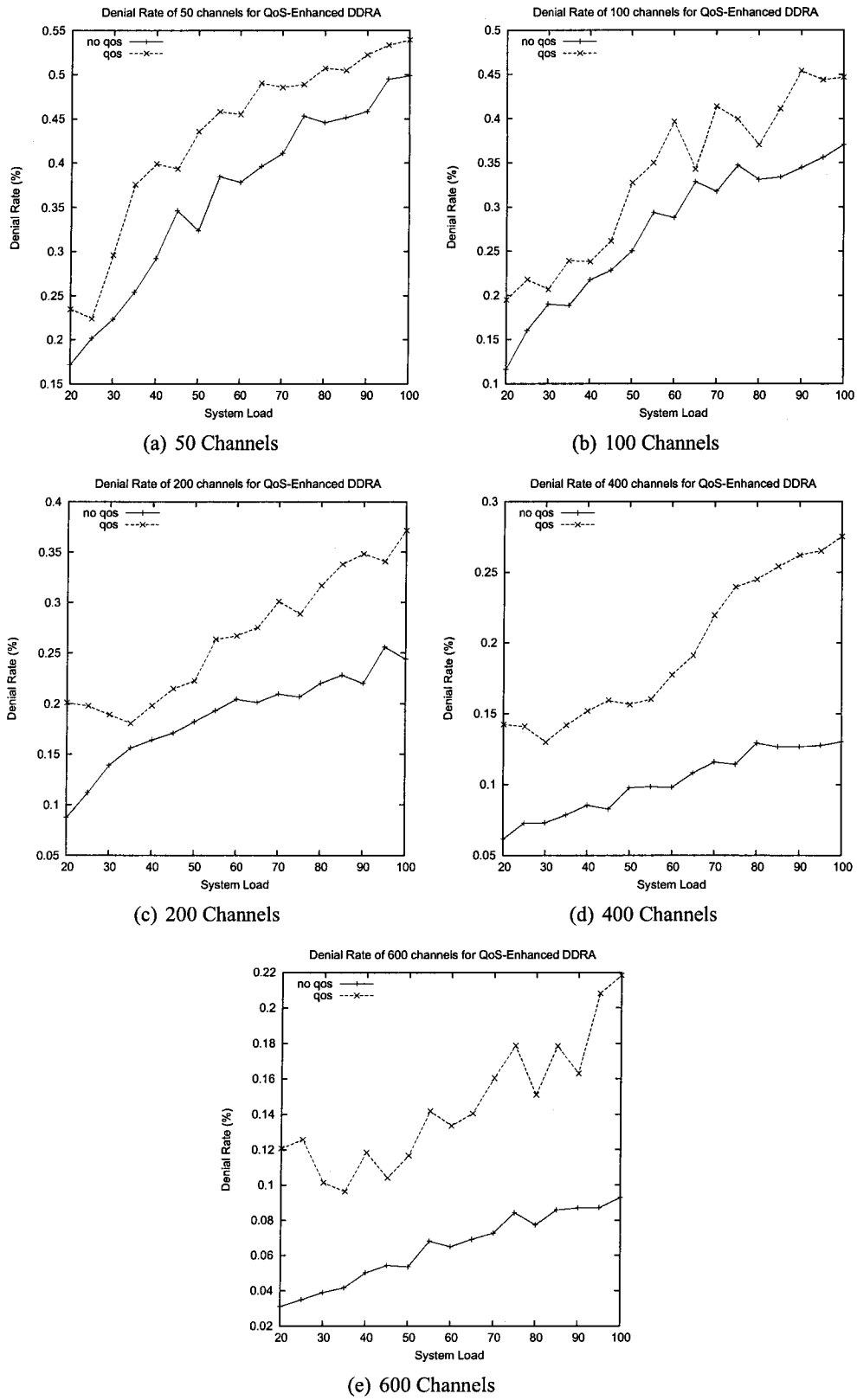


Figure 3.3: Denial Rate

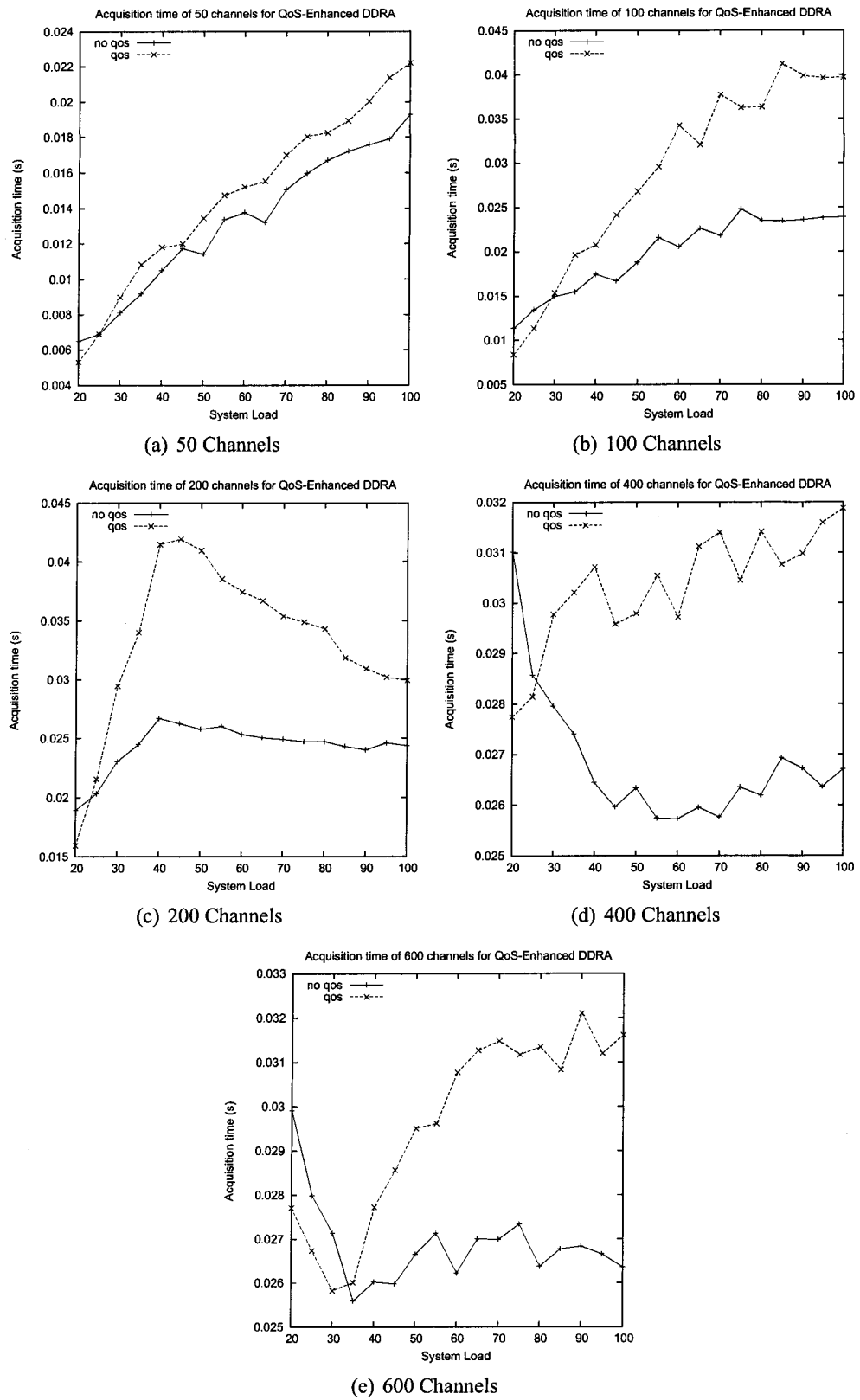
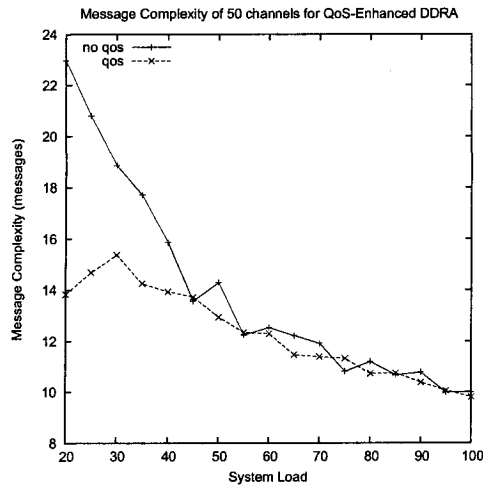
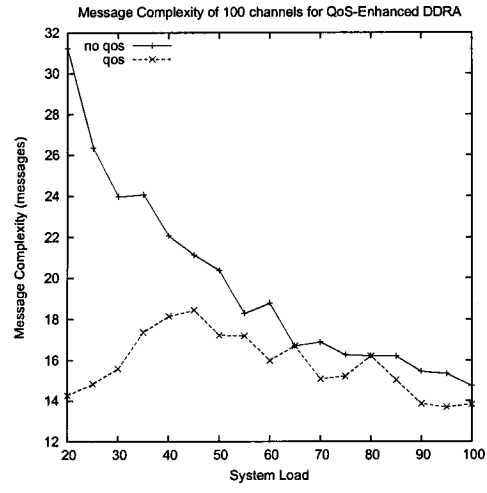


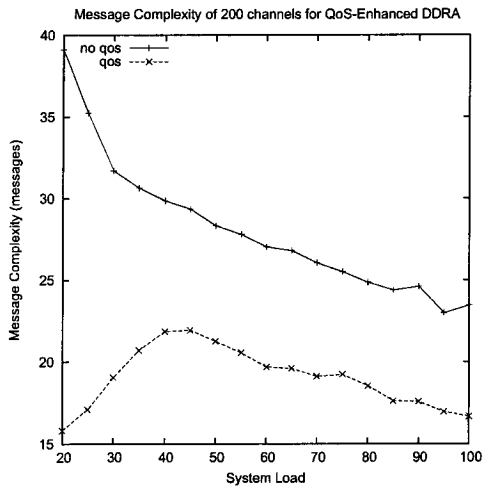
Figure 3.4: Acquisition Time



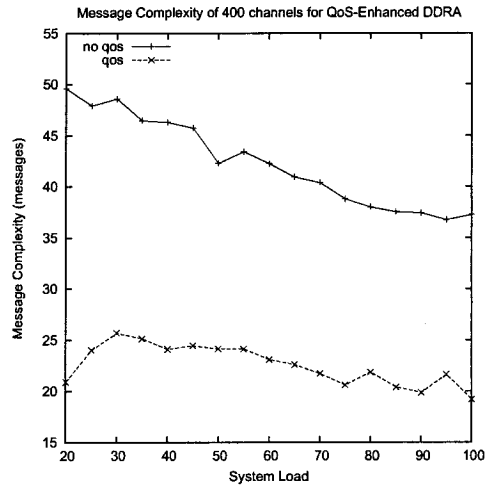
(a) 50 Channels



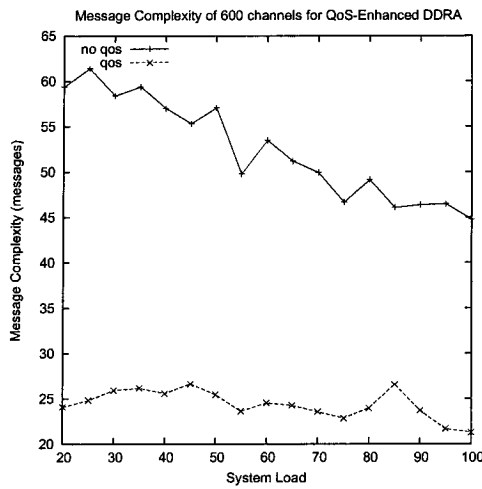
(b) 100 Channels



(c) 200 Channels



(d) 400 Channels



(e) 600 Channels

Figure 3.5: Message Complexity

## 3.6 Conclusion

In this chapter, we have proposed a QoS-based dynamic channel allocation protocol for wireless networks. To ensure and guarantee a continuous service to mobile users, our proposed scheme reserves some wireless channels for *inter-handoff*, while a base station around a hot-cell is able to monitor the mobile host's path. If a mobile host is trying to launch a new call and is heading to the neighboring hot-cell, the base station will block its call request.

We have presented our dynamic channel allocation protocol, and its simulation based performance evaluation with and without the QoS component. Our experimental results indicate clearly that our QoS-based channel allocation scheme exhibits a better performance results when compared to channel allocation scheme that doesn't support any QoS. We have observed that our scheme achieves a low dropping rate of only 3% that can be improved further but at the expenses of the denial rate. This is mainly due to the reduced wireless channels for the new call requests that are reserved for potential *inter-handoffs*.

Our protocol does not take into account the possibility that base stations or links may fail, which is the subject of the next chapter.

## **Chapter 4**

### **A Fault-Tolerant Dynamic Channel**

### **Allocation Protocol for Cellular Networks**

Distributed channel allocation schemes have received much attention because of their high reliability and scalability. In these schemes, a base station has to consult with its neighboring base stations in order to assign a channel to a call. If it cannot communicate with its neighbors, it fails in allocating a channel. However, it is a common phenomenon that a base station fails in communicating with its neighboring base stations due to some reasons, such as heavy traffic load. In this chapter, we propose a distributed fault-tolerant channel allocation schemes which can work well under the mobile host failures, base station failures and communication link failures. We prove its correctness. We also report our algorithm's performance with several channel systems using different types of call arrival pattern through comparing with a popular generic distributed algorithm for channel allocation DDRA [6].

## 4.1 Fault-Tolerant Distributed Dynamic Channel Allocation Algorithm

In addition to the mutual exclusion technology and the storage information for allocating channels concepts discussed in the previous chapter, a third concept is added for the proposed fault-tolerant protocol:

### The Timers of Waiting for the *Reply* message

To deal with base station or communication link failures, some timers are used. When  $B_i$  requests a group  $g_j$ , it delivers a *Request* message to all of its neighboring base stations.  $B_i$  cannot make the decision whether the group  $g_j$  can be held until it receives the reply message from all neighboring base stations. If a neighboring base station or a communication link fails,  $B_i$  cannot get the reply message from this neighboring base station. To avoid deadlock,  $B_i$  sets a timer for each request-reply exchange.

The selection of the timeout value is important for the system performance. If the timeout value is too small,  $B_i$  considers that some failures happened and makes the decision too early. If its neighbor  $B_j$  is holding the group  $g_j$ , the *Reject* message sent by  $B_j$  spends the time beyond the timeout because of the traffic congestion. But,  $B_i$  has made the decision to hold the group  $g_j$ . Therefore, the co-group interference or co-channel interference is caused. On the other hand, if the timeout period is too large,  $B_i$  may spend a long time on assigning a channel to a *handoff* call. But, this *handoff* may be dropped due to the long delay. Generally, the timeout value of the timers is up to the applications or is empirically selected. There are some good possible approaches. For example, two timeout values are set. A small timeout period is for a channel group  $g_j$ . If the waiting time is beyond this small timeout value,  $B_i$  considers the channel group  $g_j$  has been held by its neighbors and tries next group. However, if all the received reply messages are *Agree*,  $B_i$  caches the status and does further operation later. A large timeout value is used to decide whether some neighboring base station or communication

link have failed.

Because of these timers, some outdated messages may be created. Some reply messages arriving at  $B_i$  after  $B_i$  has already decided to hold the requested group  $g_j$  are outdated messages. They must be identified and discarded. Otherwise, the fault-tolerant algorithm cannot work well. To identify outdated messages, each message is timestamped using Lamport's clock. When  $B_i$  receives a reply message, it compares the timestamp with that of the *Request* message. If the timestamp of the reply message is smaller, it is an outdated message. If the timestamp of the reply message is larger but  $B_i$  has finished the decision about the corresponding channel group, it is also an outdated message.

## 4.2 Fault Tolerance

Our proposed channel allocation scheme based upon the generic dynamic channel allocation scheme (see section 3.2) requires the information exchange between two *BSs* or between a *MH* and its affiliated *BS*. In a mobile cellular network, the *MHs*, the wireless links between *MHs* and *BSs*, the *BSs* and the communication links between two *BSs* are prone to fail [8, 35]. The channel assignment algorithm is required to tolerate any failure. Otherwise, deadlock would happen due to waiting for the response forever. Hence, we elaborate how the algorithm can tolerate failures.

### 4.2.1 MH Failures and Wireless Link Failures

Generally speaking, due to the tough conditions the failure probability of *MHs* is much higher than that of *BSs* and communications links. For a normal *MH*, when it terminates a communication session it sends a *Release Ch<sub>f</sub>* message to the *BS*. If in the middle of a communication session a *MH* failure or a wireless link failure happens, the session is terminated naturally. The corresponding channel *Ch<sub>f</sub>* occupied by the *MH* is not in use any more. Once the *BS* detects such a failure, it applies the same operations as the completion of a commu-

nication session. The  $BS$  first frees the channel  $Ch_f$  and then broadcasts “ $Ch_f$  is free” to all neighboring  $BS$ s.

### 4.2.2 BS failures

When  $BS_i$  of cell  $C_i$  fails, all the communication sessions between  $BS_i$  and all  $MH$ s within this cell are terminated. That is to say, any channel is not in use in cell  $C_i$  and no co-channel interference exists between  $BS_i$  and its neighboring  $BS$ s. However, its neighboring  $BS$ s cannot detect that  $BS_i$  has failed and still request a channel group  $g_j$  from  $BS_i$ . We assume  $BS_j$  of a neighboring cell  $C_j$  sends a request message to  $BS_i$  and is expecting a response, then  $BS_j$  would wait indefinitely. This will affect the performance of the channel allocation algorithm. In order to tolerate this failure, we employ two timeout thresholds: a small one and a big one. The small threshold is a predetermined amount of time for requesting a channel group. The big threshold is a predetermined amount of time for detecting the  $BS_i$  fails. The usage of the two thresholds is discussed in the last section. When the big threshold has elapsed and the response message for the corresponding request message has been received,  $BS_j$  assumes that  $BS_i$  has failed and  $BS_j$  has received an  $AGREE(g_j)$  message from  $BS_i$ . Once  $BS_j$  has detected that  $BS_i$  has failed,  $BS_j$  removes  $BS_i$  from its list of neighbors. The request messages for a channel group are not sent to  $BS_i$  in the future.

Once  $BS_i$  recovers from a failure, it has to reconstruct its group usage table. Simultaneously,  $BS_i$  broadcasts a *recovery* message to all its neighboring  $BS$ s. Its neighboring  $BS$ s add  $BS_i$  to their list of neighbors. Their next response message will be sent to  $BS_i$ .

### 4.2.3 Communication Link Failures

It is more complicated to tolerate the communication link failures between two neighboring  $BS$ s. The communication link failures and BS failures have the same behaviors:  $BS_j$  is not able to receive the response message from its neighbor  $BS_i$ . The handling operations for the two types of failures are completely different. If  $BS_j$  misinterprets a link failure as a BS

failure, the co-channel interference perhaps might happen. Fortunately, the underlying protocol such as the network layer can distinguish the two types of failures. For the link failure, the underlying layer tries to route the *Request* and *Reply* messages along other links. In the worst case, a whole cellular network is partitioned into two parts if all wired links between the two parts fails. According to some policy, a BS with a smaller BS number has a higher priority to keep on working and some selected base stations beside the split line are forced to fail and stop requesting any channels. For example, in Figure.3.1 there is a split line  $ab$ . We implement a policy that for three neighboring BSs separated by the split line, the BS with a smaller BS number keeps on working and the BS with a bigger BS number is forced to fail. Hereby, those cells along  $ab$ , 28, 37, 46 and 55, keep on working. But other cells along  $ab$ , 38, 47 and 56, are forced to fail and stop requesting any channels. In the specific case in which all wired links between  $BS_i$  and all its neighboring BSs fail,  $BS_i$  is forced to fail and stop requesting any channels, whereas its neighbors are not affected. The toleration operations and recovery implementation for the forced failures are similar to those in the BS failures.

### **4.3 Formal Description of the Fault-Tolerant Dynamic Channel Allocation Algorithm**

In this section, we formally describe the proposed fault-tolerant algorithm running on a base station  $B_i$  by means of pseudo-code:

**Case A.** Mobile host  $MH_i$  requests a free channel  $Ch_f$  from  $B_i$ .

Step A0.  $B_i$  initializes the group usage table;  
and sets  $\text{competition}(B_i) \leftarrow 0$ .

Step A1. From its group usage table,  $B_i$  selects a group  $g_j$  which has not yet been visited.

Step A2. **If**  $B_i$  has visited all groups **then**  
send Wireless network is busy to  $MH_i$  ;  
**go to Step 12.**

Step A3.  $g_j \leftarrow$  Block.

Step A4.  $B_i$  requests the group  $g_j$  from all its neighbors and sets two timers: one has a small time-out threshold and the other has a big time-out threshold.

Step A5. **If** the underlying network protocol detects any communication link to all neighboring BS fails **then**

**go to Case G.**

Step A6. **If** after the amount of time predetermined by the small time-out threshold has elapsed some REPLY messages has been received **then**

$B_i$  temporarily assumes the group  $g_j$  is busy.

$g_j \leftarrow$  "Free";

**go to Step 1.**

Step A7. **If** after the amount of time predetermined by the big time-out threshold has elapsed some REPLY messages has been received **then**

$B_i$  assumes to receive the "AGREE" messages from the corresponding BSs and removes them from its neighbor list.

Step A8. **Wait until**  $B_i$  has the REPLY messages from all of its neighbors.

Step A9. **If**  $B_i$  cannot get a free group  $g_j$  **then**

$\text{competition}(B_i) \leftarrow \text{competition}(B_i) + 1;$

$g_j \leftarrow$  "Free";

**go to Step 1.**

Step A10.  $B_i$  searches for a free channel  $Ch_f$  in  $g_j$ .

Step A11. **If**  $B_i$  finds  $Ch_f \in g_j$  **then**

$Ch_f \leftarrow$  "Block";  $g_j \leftarrow$  "Free";

$B_i$  sends the message "Make  $g_j$  free"

and "Block  $Ch_f$ " to all of its neighbors;

$\text{competition}(B_i) \leftarrow 0;$

**else**  $g_j \leftarrow$  "Free"; group usage table ( $g_j$ )  $\leftarrow$  "Visited";

**go to Step 1.**

Step A12. Exit

**Case B.**  $B_i$  receives a request from  $B_j$ .

Step B1. **If** the requested  $g_j$  is being used at  $B_i$  **then** reply "Reject".

Step B2. **If**  $g_j$  is "Free" **then** reply "Agree".

Change the owner of  $g_j$  to "Others".

Step B4. **If** the requested  $g_j$  is also requested by  $B_i$  **then**

**If**  $\text{competition}(B_j) > \text{competition}(B_i)$  **then**

reply "Agree" to the base station  $B_j$  ;

$B_i \leftarrow$  "False" to make "Fail" for  $B_i$ ;

**else If**  $\text{competition}(B_j) < \text{competition}(B_i)$  **then**

reply "Reject" to the base station  $B_j$  ;

$B_i \leftarrow$  "Agree" to make "Success" for  $B_i$ ;

**else If**  $\text{competition}(B_j) == \text{competition}(B_i)$  **then**

**If**  $\text{priority}(B_j) > \text{priority}(B_i)$  **then**

randomly select the winner for  $g_j$  ;

**If**  $B_i$  was selected **then**

send "Reject" to base station  $B_j$  ;

$B_i \leftarrow$  Agree;

**else**

Send "Agree" to base station  $B_j$  ;

$B_i \leftarrow$  "False".

**Case C.**  $B_i$  receives a message for "Block  $Ch_f$ " and "Free  $g_i$ ".

Step C1. Channel  $Ch_f \leftarrow$  "Block".

Step C2. Group  $g_i \leftarrow$  "Free".

**Case D.**  $B_i$  receives a "Release  $Ch_f$ " message from  $MH_k$ .

Step D1. Channel  $Ch_f \leftarrow$  "Free".

Step D2. Broadcast " $Ch_f$  is free" to all of  $B_i$ 's neighbors.

**Case E.**  $B_i$  receives a "Free  $Ch_f$ " message from  $B_j$  .

Step E1. Channel  $Ch_f \leftarrow$  "Free".

**Case F.**  $B_i$  receives a "Free  $g_i$ " message from  $B_j$  .

Step F1. Group  $g_i \leftarrow$  "Free".

**Case G.** The communication links between  $B_i$  and all of its neighboring BSs fail.

Step G1. send a "Release" message to all MHs in its cell.

Step G2.  $B_i$  stops.

**Case H.**  $B_i$  receives a "Recovery" message from  $B_j$ .

Step H1. add  $B_j$  to its neighbor list.

## 4.4 Illustrative Examples

In this section, some illustrative examples are presented explaining the steps performed by the proposed algorithm under each one of the following four cases: (i) *No failures*, (ii) *Mobile Host failure and wireless link failure*, (iii) *Base Station failure* and (iv) *Communication link failure*.

Let us consider the mobile cellular network given in Figure.3.1. In the following examples, we assume that a mobile host in *cell\_33* requests a free channel.

- **Case 1: No Failures**

- A mobile host  $MH_1$  in *cell\_33* requests a free channel  $Ch_f$  from its  $BS_{33}$ .
- $BS_{33}$  initializes its group usage table and its competition variable ( $competition(BS_{33}) \leftarrow 0$ ).
- From its group usage table,  $BS_{33}$  selects the first group  $g_1$  which has not yet been visited.  $BS_{33}$  blocks  $g_1$  ( $g_1 \leftarrow \text{Block}$ ).
- $BS_{33}$  requests the group  $g_1$  from all its neighbors and sets two timers: one has a small time-out threshold and the other has a big time-out threshold.
- $BS_{33}$  **Waits until** it receives the REPLY messages from all its neighbors. We assume that  $BS_{33}$  receives all REPLY messages from its neighboring  $BS$ s before the amount of time predetermined by the small time-out threshold elapses and all the replies are AGREE messages.
- $BS_{33}$  searches for a free channel  $Ch_f$  in the group  $g_1$ .
- We assume that  $B_{33}$  did not find a free channel  $Ch_f$  in the group  $g_1$ . Consequently, it frees  $g_1$  ( $g_1 \leftarrow \text{Free}$ ) and marks  $g_1$  in its group usage table as visited ( $g_1 \leftarrow \text{Visited}$ ).

- $B_{.33}$  selects  $g_{.2}$  which is the next group in its group usage table not visited yet and blocks it ( $g_{.2} \leftarrow \text{Block}$ ).
- $B_{.33}$  requests the group  $g_{.2}$  from all its neighbors and sets two timers: one has a small time-out threshold and the other has a big time-out threshold.
- $B_{.33}$  **Waits until** it collects the REPLY messages from all its neighbors. We assume that  $B_{.33}$  receives all REPLY messages from its neighboring  $BS$ s before the amount of time predetermined by the small time-out threshold elapses and all the replies are AGREE messages.
- $B_{.33}$  searches for a free channel  $Ch_f$  in the group  $g_{.2}$ .
- We assume that  $B_{.33}$  found a free channel  $Ch_{f,1}$  in the group  $g_{.2}$ , it assigns this channel to  $MH_{.1}$  ( $Ch_{f,1} \leftarrow \text{“Block”}$ ) and frees the group  $g_{.2}$  ( $g_{.2} \leftarrow \text{“Free”}$ ).  $B_{.33}$  sends the messages “Make  $g_{.2}$  free” and “Block  $Ch_{f,1}$ ” to all its neighbors and re-initializes the variable competition ( $\text{competition}(B_{.33}) \leftarrow 0$ ).

- **Case 2: Mobile Host Failure or Wireless Link Failure**

- Let us assume that a channel  $Ch_{f,1}$  is assigned to the requesting mobile host  $MH_{.1}$  in  $cell_{.33}$  following the steps cited in the previous subsection “No failures”.
- We also assume that in the middle of a communication session a MH failure or a wireless link failure happens.  $BS_{.33}$  detects this failure and applies the same operations as the completion of a communication session.
  - \*  $BS_{.33}$  first frees the channel  $Ch_{f,1}$ .
  - \*  $BS_{.33}$  broadcasts  $Ch_{f,1}$  is free to all neighboring  $BS$ s.

- **Case 3: Base Station Failure**

In this case, we will assume that the base station  $BS_{.42}$  fails.

- A mobile host  $MH_{.1}$  in  $cell_{.33}$  requests a free channel  $Ch_f$  from its  $BS_{.33}$ .
- $BS_{.33}$  initializes its group usage table and its competition variable ( $\text{competition}(BS_{.33}) \leftarrow 0$ ).

- From its group usage table, *BS* \_33 selects the first group *g*\_1 which has not yet been visited. *BS* \_33 blocks *g*\_1 ( $g_1 \leftarrow \text{Block}$ ).
- *BS* \_33 requests the group *g*\_1 from all its neighbors and sets two timers: one has a small time-out threshold and the other has a big time-out threshold.
- *BS* \_33 waits for the REPLY messages from its neighboring *BS* s.
- *BS* \_33 receives all the REPLY messages from its neighboring *BS* s except from the base station *BS* \_42 and the amount of time predetermined by the small time-out threshold has elapsed.
- *BS* \_33 temporarily assumes the group *g*\_1 is busy and frees *g*\_1 ( $g_1 \leftarrow \text{“Free”}$ ).
- From its group usage table, *BS* \_33 selects the next group *g*\_2 which has not been visited yet. *BS* \_33 blocks *g*\_2 ( $g_2 \leftarrow \text{Block}$ ).
- *BS* \_33 requests the group *g*\_2 from all its neighbors and sets two timers: one has a small time-out threshold and the other has a big time-out threshold.
- *BS* \_33 waits for the REPLY messages from its neighboring *BS* s.
- *BS* \_33 receives all the REPLY messages from its neighboring *BS* s except from the base station *BS* \_42 and the previous amount of time predetermined by the big time-out threshold has elapsed.
- *B*\_33 assumes that it has received an “AGREE” message from the base station *BS* \_42 and removes it from its list of neighbors. We assume also that the replies received are AGREE messages.
- *B*\_33 searches for a free channel  $Ch_f$  in the group *g*\_2.
- We assume that *B*\_33 found a free channel  $Ch_{f,1}$  in the group *g*\_2, it assigns this channel to *MH*\_1 ( $Ch_{f,1} \leftarrow \text{“Block”}$ ) and frees the group *g*\_2 ( $g_2 \leftarrow \text{“Free”}$ ). *B*\_33 sends the messages “Make *g*\_2 free” and “Block  $Ch_{f,1}$ ” to all of its neighbors and re-initializes the variable competition ( $\text{competition}(B_{33}) \leftarrow 0$ ).

- **Case 4: Communication Link Failure**

In this case we assume that the wired link between the base stations *B*\_33 and *B*\_34 fails.

- A mobile host  $MH_1$  in  $cell_{33}$  requests a free channel  $Ch_f$  from its  $BS_{33}$ .
- $BS_{33}$  initializes its group usage table and its competition variable  
(competition( $BS_{33}$ )  $\leftarrow$  0).
- From its group usage table,  $BS_{33}$  selects the first group  $g_1$  which has not been visited yet.  $BS_{33}$  blocks  $g_1$  ( $g_1 \leftarrow$  Block).
- $BS_{33}$  requests the group  $g_1$  from all its neighbors and sets two timers: one has a small time-out threshold and the other has a big time-out threshold.
- The underlying network protocol detects that the communication link between the base stations  $BS_{33}$  and  $BS_{34}$  has failed and tries to route the Request and Reply messages along other links.
- $BS_{33}$  routes its request message to  $BS_{34}$  through  $BS_{24}$ .
- $BS_{34}$  routes its REPLY message to  $BS_{33}$  through  $BS_{24}$ .
- $BS_{33}$  **Waits until** it receives the REPLY messages from all its neighbors. We assume that  $BS_{33}$  receives all REPLY messages from its neighboring  $BS$ s before the amount of time predetermined by the small time-out threshold elapses and all the replies are AGREE messages.
- $BS_{33}$  searches for a free channel  $Ch_f$  in the group  $g_1$ .
- We assume that  $BS_{33}$  found a free channel  $Ch_{f,1}$  in the group  $g_1$ , it assigns this channel to  $MH_1$  ( $Ch_{f,1} \leftarrow$  “Block”) and frees the group  $g_1$  ( $g_1 \leftarrow$  “Free”).  $BS_{33}$  sends the messages “Make  $g_1$  free” and “Block  $Ch_{f,1}$ ” to all of its neighbors and re-initializes the variable competition (competition( $BS_{33}$ ) $\leftarrow$  0).

## 4.5 Simulation Experiments

To gauge the performance of our proposed algorithm, we simulated a wireless network of 57 cells through designing a discrete-event mode [41]. In the experiments, the number of channels is 400 and two environments, *uniform* and *non-uniform* call arrival distributions, are

Table 4.1: Simulation Parameters

Parameters	Value
Number of cells	57
Number of channels	100, 200, 400, 600
Mean service time/session	180ms
Arrival rate in normal cell	$\lambda$
Arrival rate into a hot cell	$5\lambda$
Mean delay to processing a message	2.0ms
Mean message delay between BSs	0.03ms
Number of BS failures	0, 1, 5, 10
Number of communication link failures	0, 1, 6, 12
BSs along a split line	28, 37, 38, 46 47, 55, 56

employed. As for the uniform distribution, we assume that each cell has the same channel demand, i.e., call arrivals to a mobile host are equal to  $\lambda$ , and the service time are equal to  $\mu$ . As regards the non-uniform distribution, we assume that one third of the cells are heavily loaded. Heavily loaded cells have call arrival rate as five times as that of the lightly loaded cells (cold cells). In order to select the heavily loaded cells, we settled with two strategies: one is that heavily loaded cells are evenly distributed over the region, and the other is that any cell can be selected as a heavily loaded cell.

The messages exchanged between the base stations are constructed from the information: *Packet type*, *Source BS*, *Destination BS*, *Competition\_num* and *Channel\_num*. There are five types of packets.<sup>1</sup> Tab. 4.1 illustrated the simulation parameters of our experiments. We assume that the mean service time per communication session is 180ms. The call durations are modeled by a Poisson process with mean 180ms, and the call arrival rates by an exponential process with an arrival rate  $\lambda$ . The call arrival rates for heavily loaded cell is kept to equal to

<sup>1</sup>Therefore, 3 bits are sufficient to identify a packet. Thus, the size of a message is at least  $3 + 2\log(N_n) + \log(N_n + \log(N_c))$  bits, where  $N_n$  is the number of neighboring cells and  $N_c$  is the number of channels in the system. Hence, we need at least 32 bits if we use 400 channels.

5 $\lambda$ . The parameter  $\lambda$  is varied to achieve a certain system load percentage, calculated using Litter's Law.<sup>2</sup> The average one-way communication delay between two BSs is 0.03ms and it takes 2ms to process a packet.

To evaluate the fault-tolerant performance of our proposed algorithms two types of failures were simulated: BS failures and link failures. When we evaluate a category of failure, the corresponding parameter is set to non-zero but others are zero. For BS failures, zero, one, five or ten BSs are respectively rendered inoperable. These BSs do not respond to any messages sent by neighbors, and any calls raised by MH in these cells are blocked. Similarly for link failures, zero, one, six or twelve links between neighboring BSs are cut. If a BS must reach one of its neighbor and the direct communication link fails, it is assumed that the messages may be routed through other non-failed neighbors although there is a penalty of the message transmission duration, as long as that neighbor is indirectly reachable. However, if the whole wireless network is partitioned into two separated parts, the algorithm counts the sum of BSs' number beside the split line in the two parts. Those BSs which have the smaller sum are forced to fail and block the MH's new calls.

We assume that the communication line has 1 Mbit/s bandwidth between two BSs. Experimental evaluation of channel assignment strategies requires not only an implementation but also the examination of a large number of test cases.

#### 4.5.1 Performance Metrics

To evaluate the performance of an algorithm for channel allocation, three performance metrics are measured for the general DDRA algorithm and the FT-DDRA algorithm and are defined as follow:

- The *denial rate* represents the percentage of blocked calls and it usually indicates the efficiency of the channel selection algorithm.

---

<sup>2</sup>Little's Law helps in computing the average number of jobs (calls) in a stable system given the mean arrival rate and the mean service time [29]. This is converted to a percentage using total available channels and number of cells in a cluster.

- The *acquisition time* indicates the time needed for the BS to satisfy a channel request. It scales the channel acquisition efficiency.
- The *message complexity* measures the number of messages exchanged per process to satisfy a channel request.

### 4.5.2 Experimental Results

In order to quantify the effect of failures, various failures are simulated with different numbers of wireless channels (100, 200, 400 and 600). For base stations, there are four scenarios: 0 BS failure, 1 BS failure, 5 BS failures and 10 BS failures. As regards link failures, there are two cases. In the first case, some local links fail but BSs can still reach each other through routing around other available links. In the other case, the whole wireless network is partitioned into two independent parts which are completely separated. In the former case, 0 link failure, 1 link failure, 6 link failures and 12 link failures are implemented. In the latter case, a specific split line is simulated.

#### Base Station Failures

As shown in the Figures. 4.1(a), 4.1(b), 4.1(c) and 4.1(d), the number of BS failures have a dramatic effect on denial rate. For low numbers of BS failures, the system load has a noticeable affect with increasing system load causing greater denial rates. However, for larger numbers of BS failures, the denial rate is dominated by calls blocked due to BS's unavailability. If we investigate the cause of denial rate, we find two main direct factors: all blocked calls raised at the failed BSs and some calls blocked at the BSs which have some failed neighbors. The FT-DDRA algorithm improves the performance through affecting the former case.

It is demonstrated in the figures of denial rate. The FT-DDRA algorithm largely improves the denial rate over the generic DDRA algorithm at all system loads and numbers of channels. The denial rate of FT-DDRA is reduced by 70 percent. For example, for 400 wireless channels, when 10 BSs fail the denial rate of generic DDRA fluctuates around 90 percent but that of FT-

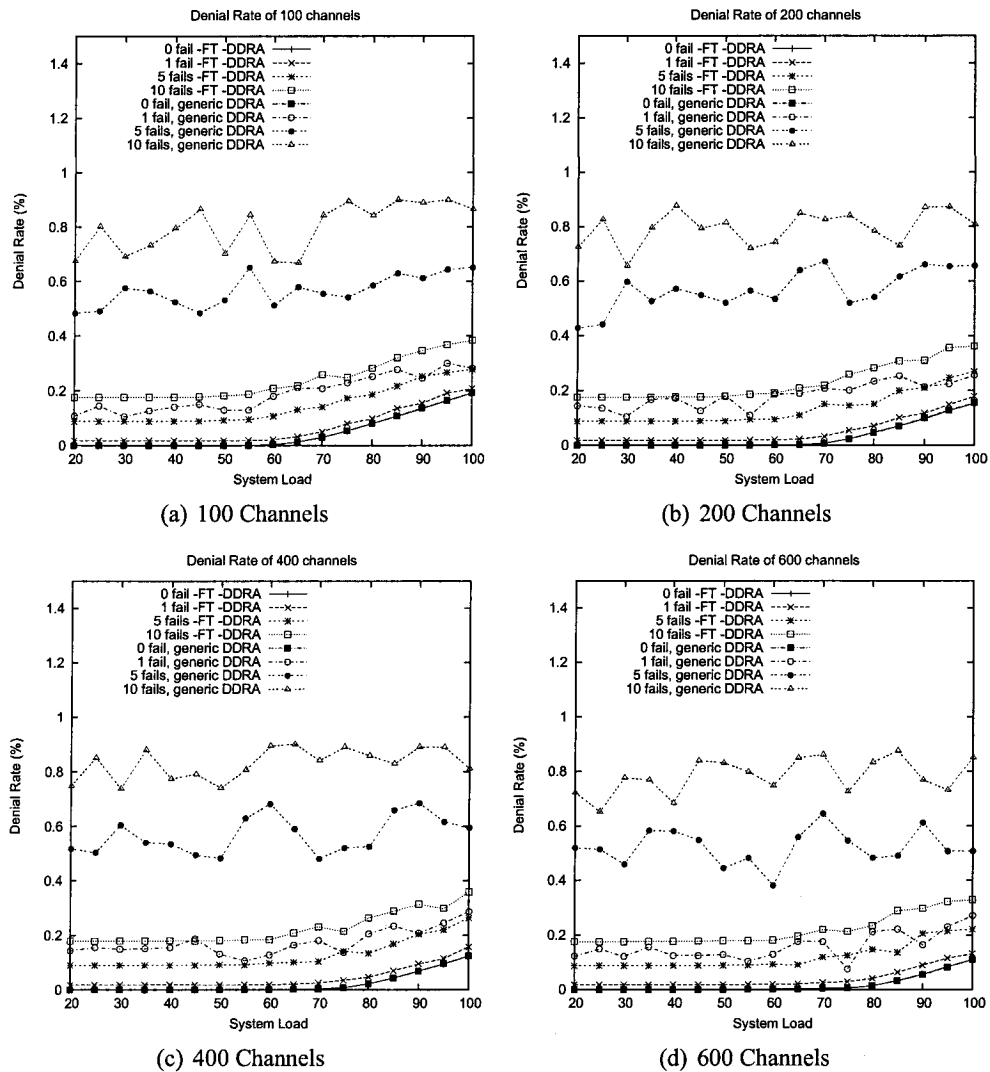


Figure 4.1: Denial Rate with BS Failures

DDRA is around 18 percent. As aforementioned, in generic DDRA the BS cannot allocate channels to the call request if it has one or more failed BSs. In the FT-DDRA, only the failed BSs cannot assign channels and block the new call requests however their neighboring BSs still keep working. The theory and our simulations undoubtedly proves the utility of FT-DDRA.

From Figures. 4.2(a), 4.2(b), 4.2(c) and 4.2(d) we see that acquisition time is largely unaffected by BS failures as there are no new messages added. Moreover, the FT-DDRA has the acquisition time almost equal to that of the generic DDRA. The acquisition time increases

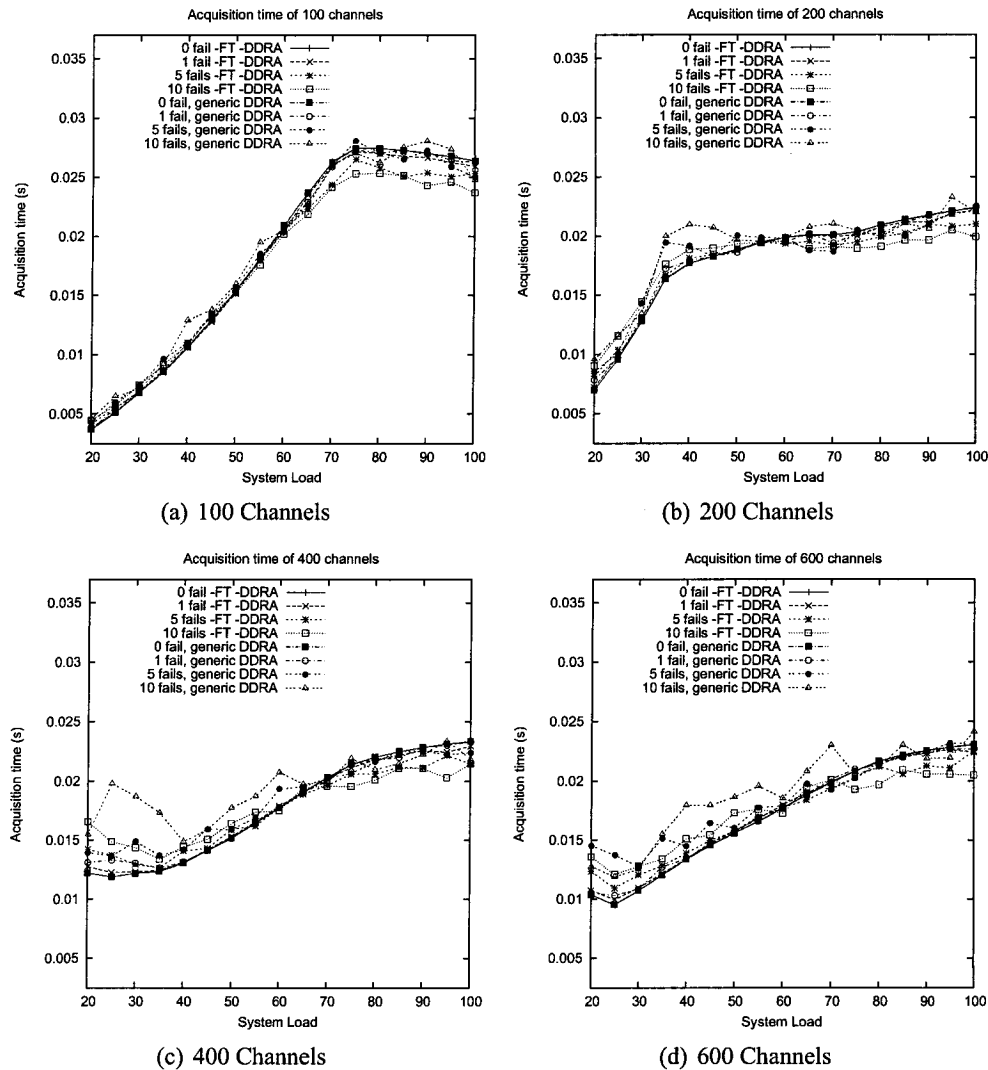


Figure 4.2: Acquisition Time with BS Failures

from 0.012s to 0.023s as the system load increments for 400 wireless channels as an example. This range is acceptable for the real application. Therefore, we can infer that the FT-DDRA scales well and maintains a certain level of quality despite the increase of BS failures.

Message complexity rises slightly as failure rates increase in the FT-DDRA over the generic DDRA. This is because message the complexity is measured for successful calls. The FT-DDRA algorithm allocates channels to many of the difficult calls through using some extra message to negotiate with failed neighbors, but the generic DDRA algorithm only directly

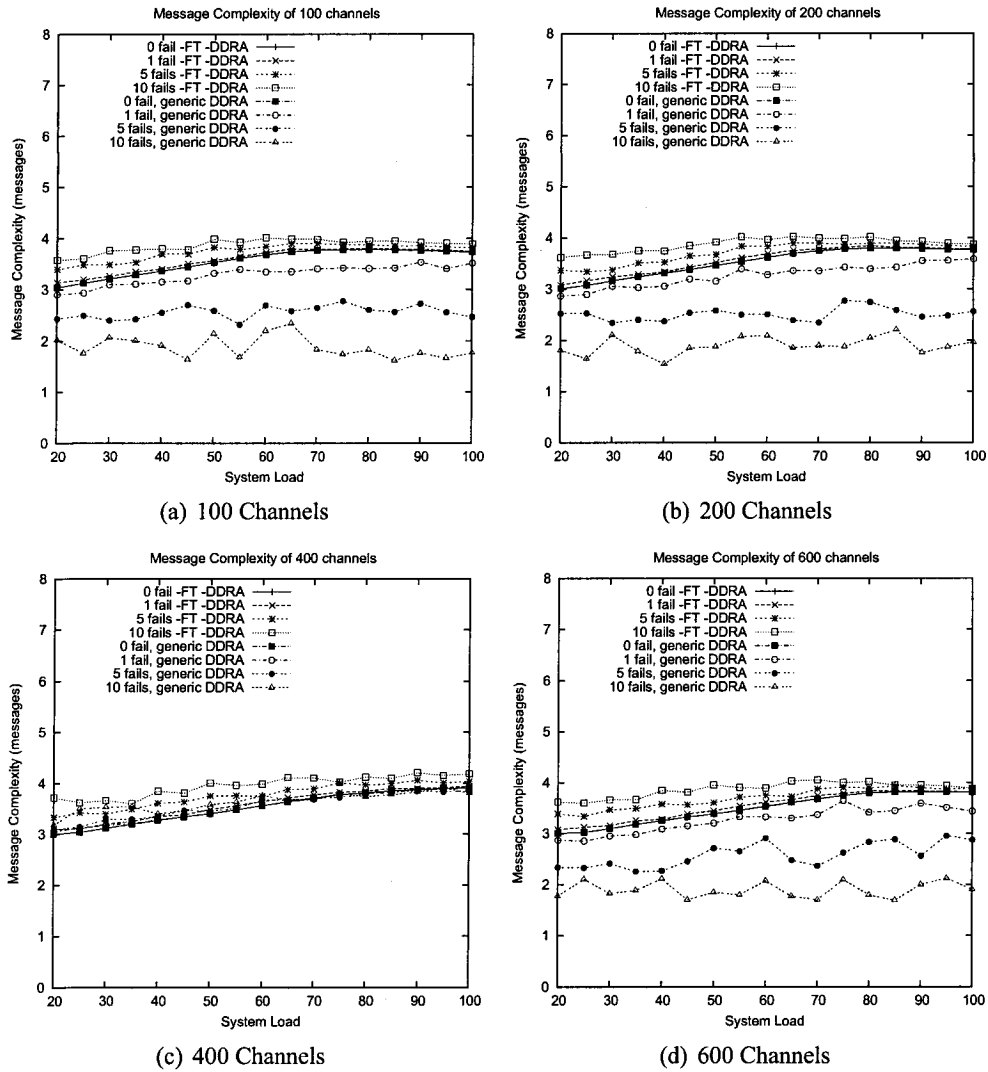


Figure 4.3: Message Complexity with BS Failures

blocks those difficult calls which requires negotiation with failed neighbors. Figures.4.3(a), 4.3(b), 4.3(c) and 4.3(d) depict the message complexity of the genetic DDRA and FT-DDRA respectively. The message complexity of the FT-DDRA is 0.4 greater than that of the genetic DDRA under 10 BS failures and with 400 wireless channels. Therefore, it is desirable to improve largely the performance of denial rate through the penalty of some message complexity.

### Link Failures

The results shown in Figures. 4.4(a), 4.4(b), 4.4(c) and 4.4(d) concerning the denial rate, in Figures. 4.5(a), 4.5(b), 4.5(c) and 4.5(d) concerning the acquisition time and in Figures. 4.6(a), 4.6(b), 4.6(c) and 4.6(d) concerning the message complexity indicate that the number of link failures have little or no effect on the performance of the system in the generic DDRA or FT-DDRA. This is in line with what the algorithm assumes, i.e. that failed links can be routed around.

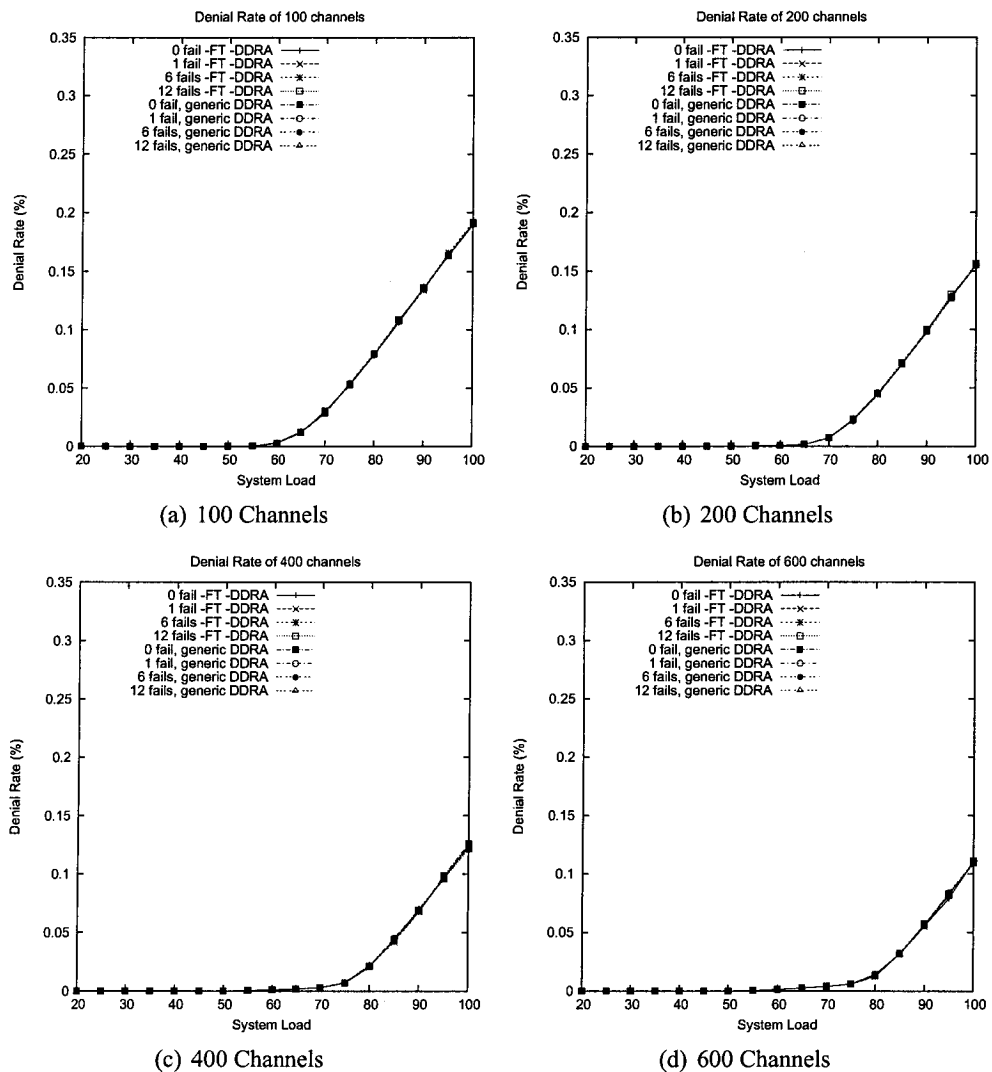


Figure 4.4: Denial Rate with Link Failures

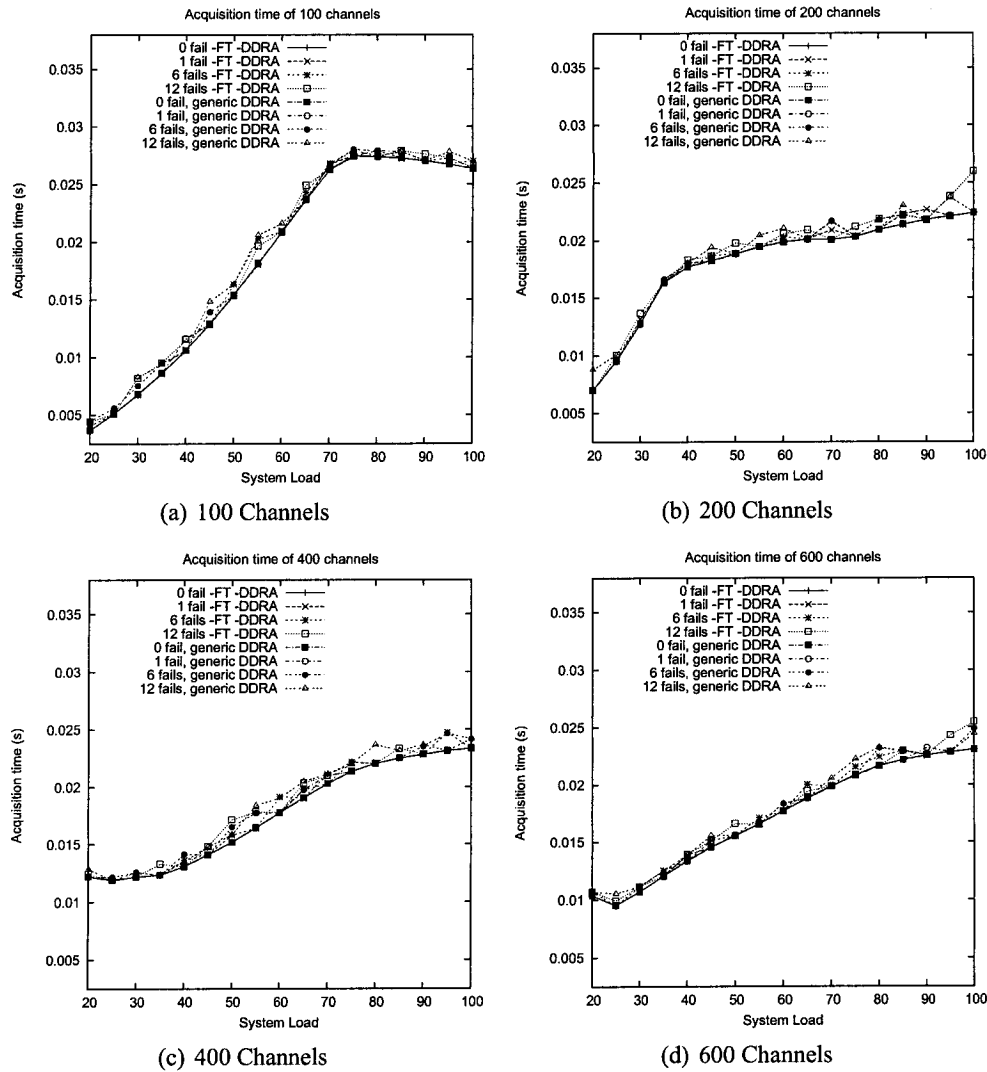


Figure 4.5: Acquisition Time with Link Failures

Figure. 4.7(a), Figure. 4.7(b) and Figure. 4.7(c) portray the performance in the case that the whole wireless network is partitioned into two separated parts. Because the FT-DDRA implements some policies to keep some BSs beside the split line working, the denial rate becomes much better from 27% to 5%. However, the FT-DDRA has a little penalty of acquisition time and message complexity. The acquisition time rises by 0.001s and the message complexity increases 0.03. It also proves the utility of the FT-DDRA for link failures.

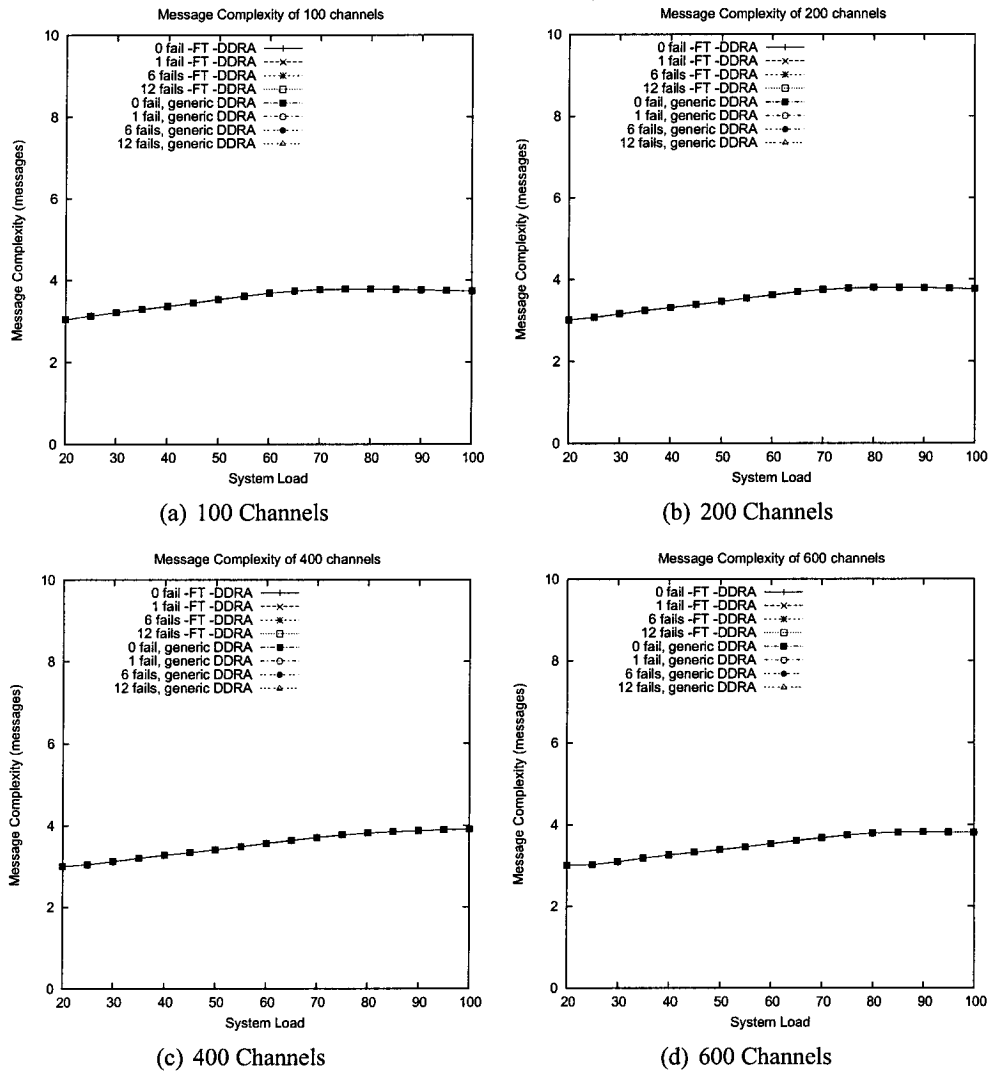
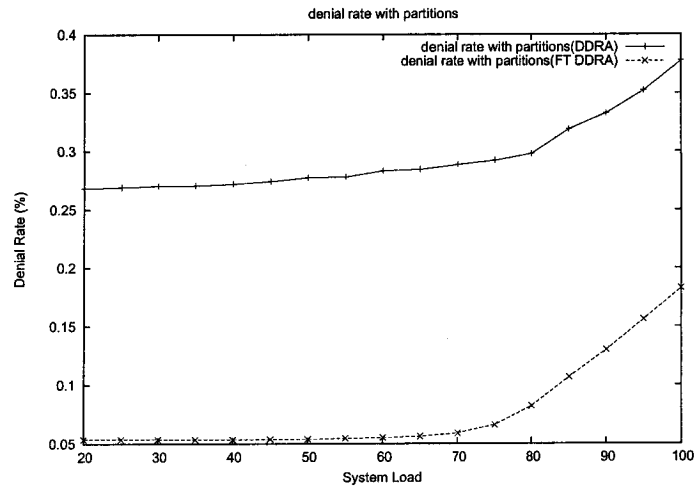
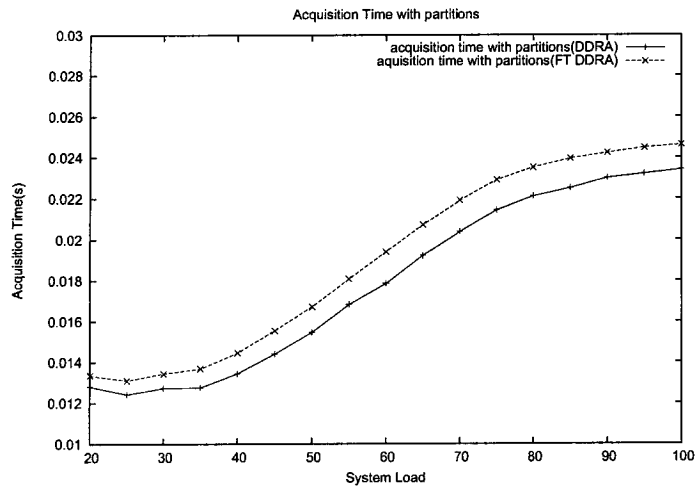


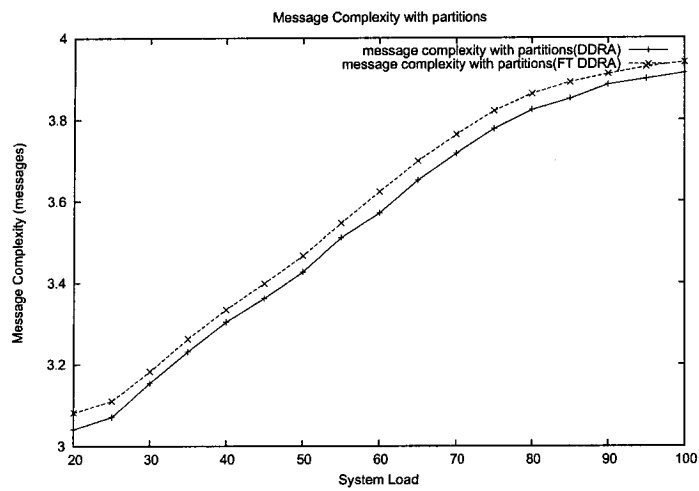
Figure 4.6: Message Complexity with Link Failures



(a) Denial Rate



(b) Acquisition Time



(c) Message Complexity

Figure 4.7: Partition Split with 400 Channels

## 4.6 Conclusion

Based on the generic DDRA algorithm, we proposed a fault-tolerant DDRA scheme. In the real world, it is common that some base stations fail or that some communication links are congested because of the heavy call traffic. Our proposed fault-tolerant DDRA algorithm handles three categories of failures: mobile host failures, base station failures and communication link failures.

When some failures occur, the BS can identify the category of failures by setting time-out thresholds and acquiring some information from the underlying network layer. After detecting some failures, the base station can still assign channels to the new call request under the case that it does not receive any response from some neighbors.

Through a series of simulations, we can see the FT-DDRA improves largely the performance of the denial rate with a little penalty of message complexity. For example for 400 wireless channels, the denial rate is reduced by 70 percent under 10 BS failures in the FT-DDRA. However, the message complexity increases by 0.4 under 10 BS failures. With respect to the acquisition time, the FT-DDRA does not affect it. To the general link failures, the generic DDRA and the FT-DDRA have almost the same performance. With respect to the special case that a whole wireless network is partitioned into two separated parts, the FT-DDRA improves largely the denial rate.

## **Chapter 5**

### **QoS and Fault-Tolerant Based**

### **Distributed Dynamic Channel Allocation**

### **Protocol for Cellular Networks**

Our earlier studies on dynamic channel allocation protocols based upon the mutual exclusion paradigm have shown promising results. Though, very little data have been reported on how to integrate the quality of service (QoS) and fault tolerant components within these schemes. These two components are vital to the success of the deployment of future generation of wireless networks. Mobile users are not expected to have their connections cut off because of a base station or a link failure and require QoS guarantees on their connections. In this chapter, we present an efficient QoS and fault tolerant based protocol for dynamic channel allocation using the mutual exclusion paradigm. Its main feature is its ability to tolerate the base station failure, mobile host failure and communication link failure, as well as reducing the connection dropping rate. QoS provisions are guaranteed in our protocol. We present our QoS and fault-tolerant protocol, and report on its performance evaluation using an extensive set of simulation experiments.

## 5.1 Introduction

The connection dropping rate caused by unsuccessful *handoff*'s might be quite high in the congested cells, and thereby, the QoS can be seriously affected. In this chapter, we show how one can integrate QoS and fault-tolerant mechanisms to enhance further the resource management for wireless networks. Thus, we present an efficient QoS and fault tolerant based protocol for dynamic channel allocation using the mutual exclusion paradigm. We present our protocol and prove its performance with exhaustive experiments. Our algorithm accommodates dynamically the number of reserved channels for the *handoff* depending from the traffic situation in purpose to insure the QoS provisions. Besides, it has many characteristics. In fact, it is able to tolerate base station failures, mobile host failures and communication link failures. In addition, it decreases the connection dropping rate due to its adaptive channel reservation technique, which engender the guarantee of QoS provisions to wireless multimedia applications.

The system model concerning this part is similar to the system model explained in Chapter 3. Also, since our algorithm integrates all the features of the QoS and fault-tolerant protocols we will show how they are integrated using a formal description, then we will report on the simulation experiments and evaluate its performance.

## 5.2 Formal Description of the QoS and Fault-Tolerant Based Dynamic Channel Allocation Algorithm

In this section, we formally describe our proposed QoS and fault-tolerant based algorithm for cellular networks running on a base station  $B_i$  by means of a pseudo-code:

**Case A.** Mobile host  $MH_j$  requests a free channel  $Ch_f$  from  $B_i$ .

Step A0.  $B_i$  initializes the group usage table;

and sets  $\text{competition}(B_i) \leftarrow 0$ .

Step A1. From its group usage table,  $B_i$  selects a group  $g_j$  which has not been visited yet.

Step A2. **If**  $B_i$  has visited all groups **then**

send Wireless network is busy to  $MH_i$  ;

**go to Step 12.**

Step A3.  $g_j \leftarrow$  Block.

Step A4.  $B_i$  requests the group  $g_j$  from all of its neighbors and sets two timers:

one has a small time-out threshold and the other has a big time-out threshold.

Step A5. **If** the underlying network protocol detects any

communication link to all neighboring BS fails **then**

**go to Case G.**

Step A6. **If** after the amount of time predetermined by the small time-out threshold has elapsed

some REPLY messages has been received

**then**

$B_i$  temporarily assumes the group  $g_j$  is busy.

$g_j \leftarrow$  "Free";

**go to Step 1.**

Step A7. **If** after the amount of time predetermined by the large

time-out threshold has elapsed some REPLY messages has been received

**then**

$B_i$  assumes to receive the "AGREE" messages from the

corresponding BSs and removes them from its neighbor list.

Step A8. **Wait until**  $B_i$  has the REPLY messages from all of its neighbors.

Step A9. **If**  $B_i$  cannot get a free group  $g_j$  **then**

$\text{competition}(B_i) \leftarrow \text{competition}(B_i) + 1$ ;

$g_j \leftarrow$  "Free";

**go to Step 1.**

Step A10.  $B_i$  searches for a free channel  $Ch_f$  in  $g_j$ .

Step A11. **If**  $B_i$  finds  $Ch_f \in g_j$  **then**

$Ch_f \leftarrow$  "Block";  $g_j \leftarrow$  "Free";

$B_i$  sends the message "Make  $g_j$  free"

and "Block  $Ch_f$ " to all its neighbors;

$\text{competition}(B_i) \leftarrow 0$ ;

**else**  $g_j \leftarrow$  "Free"; group usage table ( $g_i$ )  $\leftarrow$  "Visited";

**go to Step 1.**

Step A12. Exit

**Case B.**  $B_i$  receives a request from  $B_j$ .

Step B1. **If** the requested  $g_j$  is being used at  $B_i$

**then** reply “Reject”.

Step B2. **If**  $g_j$  is “Free” **then** reply “Agree”.

    Change the owner of  $g_j$  to “Others”.

Step B4. **If** the requested  $g_j$  is also requested by  $B_i$  **then**

**If**  $\text{competition}(B_j) > \text{competition}(B_i)$  **then**

        reply “Agree” to the base station  $B_j$  ;

$B_i \leftarrow$  “False” to make “Fail” for  $B_i$ ;

**else If**  $\text{competition}(B_j) < \text{competition}(B_i)$  **then**

        reply “Reject” to the base station  $B_j$  ;

$B_i \leftarrow$  “Agree” to make “Success” for  $B_i$ ;

**else If**  $\text{competition}(B_i) == \text{competition}(B_j)$  **then**

**If**  $\text{priority}(B_j) > \text{priority}(B_i)$  **then**

            randomly select the winner for  $g_j$  ;

**If**  $B_i$  was selected **then**

            send “Reject” to base station  $B_j$  ;

$B_i \leftarrow$  Agree;

**else**

            Send “Agree” to base station  $B_j$  ;

$B_i \leftarrow$  “False”.

**Case C.**  $B_i$  receives a new connection request requiring  $n$  channels from an MH.

Step C1.  $B_i$  invokes the algorithm in Case A to request  $n$  channels.

Step C2. **If**  $B_i$  gets  $n$  free channels **then**

    The new connection is accepted.

**else**

    The new connection is blocked.

**Case D.**  $B_i$  receives an incoming *handoff* requiring  $n$  channels.

Step D1. **If** the available reserved channels on  $B_i > n$  **then**

    The incoming *handoff* is accepted.

**else**

The incoming *handoff* is dropped.

Step D2. **If** the reserved channels < a specified value  $R_i$  **then**

$B_i$  invokes the algorithm in Case A to reserve available channels up to  $R_i$

**Case E.** The timer on  $B_i$  for channel reservation is timeout or

an incoming *handoff* is dropped.

Step E1. **If** the CDR > the target value  $P_{target}$  **then**

**If**  $R_i < R_{max}$  **then**

$$R_i = R_i + \pi,$$

Invoke the algorithm in Case A to request  $\pi$  channels for incoming *handoffs*,

Step E2. **If** the CDR < the target value  $P_{target}$  **then**

**If**  $R_i > R_{min}$  **then**

$$R_i = R_i - \pi$$

Invoke the algorithm in Case G to release  $\pi$  channels

**Case F.**  $B_i$  receives a message for “Block  $Ch_f$ ” and “Free  $g_i$ ”.

Step F1. Channel  $Ch_f \leftarrow$  “Block”.

Step F2. Group  $g_i \leftarrow$  “Free”.

**Case G.**  $B_i$  receives a “Release  $Ch_f$ ” message from  $MH_k$ .

Step G1. Channel  $Ch_f \leftarrow$  “Free”.

Step G2. Broadcast “ $Ch_f$  is free” to all of  $B_i$ 's neighbors.

**Case H.**  $B_i$  receives a “Free  $Ch_f$ ” message from  $B_j$ .

Step H1. Channel  $Ch_f \leftarrow$  “Free”.

**Case I.**  $B_i$  receives a “Free  $g_i$ ” message from  $B_j$ .

Step I1. Group  $g_i \leftarrow$  “Free”.

**Case J.** The communication links between  $B_i$  and all its neighboring BSs fail.

Step J1. send a “Release” message to all MHs in its cell.

Step J2.  $B_i$  stops.

**Case K.**  $B_i$  receives a “Recovery” message from  $B_j$ .

Step K1. add  $B_j$  to its neighbor list.

**Case L.**  $B_i$  receives a “heavily congested” message from  $B_j$

Step L1. Block  $\gamma$  percent of connections from the MH which may go to Cell  $C_j$ .

**Case M.**  $B_i$  receives a “becoming cold” message from  $B_j$

Step M1. Stop blocking connections from the MH may go to  $C_j$

## 5.3 Simulation Experiments

To evaluate the performance of our QoS and fault-tolerant based channel allocation protocol, we choose a wireless network of 57 cells through designing a discrete-event simulator [41]. We considered 400 wireless channels in our experiments and we dealt with two kinds of environments depending on the distributions of the call arrivals: *uniform* and *non-uniform* environments. For uniform distribution, the call arrival rate to mobile hosts in a cell is equal to  $\lambda$  and the service time is equal to  $\mu$ , and this rate is the same for all the cells in the mobile cellular network. In this case, the cells are known as cold cells. However, for the non-uniform distribution, the third of the cells are supposed to be heavily loaded. This means that the call arrival rate at these cells, is five times greater than the load at the cold cells. These cells are known as hot cells. Two simple strategies have been used in our algorithm to choose the heavily loaded cells: the first considers heavily loaded cells uniformly distributed over the area, and the second assumes that a heavily loaded cell can be any cell picked up from the wireless network.

The structure of the information messages exchanged between the base stations is as follows: *Packet type*, *Source BS*, *Destination BS*, *Competition\_num* and *Channel\_num*. In our protocol we have only five types of packets. From Table. 5.1, we can depict the simulation parameters utilized in our experiments. The mean service time per communication session is fixed to 180ms. We used a Poisson process to model the call durations that were approximated to 180ms, and we used an exponential process to settle the call arrival rates  $\lambda$ . For heavily loaded cells (hot cells) the call arrival rate was set to  $5\lambda$ . Little’s Law.<sup>1</sup> provided for us a way to obtain a certain system load percentage when varying the parameter  $\lambda$ . Besides, we

<sup>1</sup>Little’s Law allows one to calculate the average number of jobs (calls) in a stable system when the mean arrival rate and mean service time are given[29]. This is converted to a percentage using total channels available and number of cells in a cluster.

Table 5.1: Simulation Parameters

Parameters	Value
Number of cells	57
Number of channels	100, 400, 600
Mean service time/session	180ms
Arrival rate in normal cell	$\lambda$
Arrival rate into a hot cell	$5\lambda$
Mean delay to processing a message	2.0ms
Mean message delay between BSs	0.03ms
Number of BS failures	0, 1, 5, 10
Number of communication link failures	0, 1, 6, 12
Probability of low user mobility	0.15
Probability of high user mobility	0.85
Mean inter- <i>handoff</i> rate in a normal cell	6/60
Mean inter- <i>handoff</i> rate in a hot cell	6/180
hot/normal and normal/hot states	1/1800 s
Service time per data/video/voice connection	6/6/4 min
Probability of data/video/voice connection	0.2/0.2/0.6
Number of channels for each data/video/voice connection	2/4/1
The target CDR of the system	0.015
the system parameter $\gamma$	0.5
The minimum amount of reserved channels $R_{min}$	0
The maximum amount of reserved channels $R_{max}$	20
The step amount of reserved channels $\pi$	2
The timeout threshold of the timer	15 min

considered the average delay of a one-way communication between two BSs as 0.03ms and that a packet requires 2ms to be processed.

In order to establish the fault-tolerant performance side of our proposed algorithm we simulated two types of failures: BS failures and link failures and we varied the number of wireless channels (100, 400 and 600). When one of the two types of failures is evaluated its correspondent parameter is set to non-zero and the non needed parameters are set to zero. We considered four scenarios in the case of BS failures: 0, 1, 5 and 10 BSs are failed and simulated respectively. When these BSs fail, they cannot handle the MH's requests and also they cannot reply to the messages received from the BSs in the neighboring cells. For the link failures we considered also four scenarios: 0, 1, 6 and 12 links between neighboring BSs are cut. In case a direct communication link between a BS and its neighbor is cut and this BS should communicate an information to its neighbor, we considered that it is possible to route the messages over the safe links until they reach their destination. It is obvious that this will cause a delay on the message transmission duration due to the indirect path to the destination. In case the link failures result in a separation of the whole wireless network into two parts, the algorithm computes the sum of BSs' number beside the split line in the two sides and forces the failure of the base stations having the smaller sum. Thus the new calls covered by the failed base stations will be certainly blocked.

To depict the QoS performance side of our algorithm, a series of parameters has been selected to implement the adaptive channel reservation: the target value of CDR is set to 0.015, the system parameter  $\gamma$  is set to 0.5,  $R_{min}$  is set to 0,  $R_{max}$  is set to 20,  $\pi$  is set to 2 and the timeout threshold is set to 15 minutes. At the same time many other parameters have been assumed. For example, under the low mobility of a MH, an *inter-handoff* happens with probability 0.15. However, for a MH under the high mobility, an *inter-handoff* occurs with probability 0.85. In addition, since the QoS is largely recommended by the multimedia traffics (*data*, *video* and *voice*), we considered some parameters for them: the request probabilities for data, video and voice are respectively 0.2, 0.2 and 0.6, the channel demands are respectively

2, 4 and 1 and the mean service time are respectively 6, 6 and 4 minutes.

Moreover, we supposed that the communication line has 1 Mbit/s bandwidth between two BSs. To estimate the performance of our algorithm, we examined many test cases in our experimental evaluation and we took on consideration the following performance metrics.

- The *denial rate* gives the percentage of blocked calls and in general it reflects the efficiency of the channel selection algorithm.
- The *dropping rate* depicts the percentage of ongoing calls being dropped during inter-handoffs.
- The *acquisition time* gives the time needed for a BS to satisfy a channel request. It measures the channel acquisition efficiency.
- The *message complexity* determines the number of messages exchanged per process that permits the satisfaction of a channel request.

### Experimental Results

Figures. 5.1(a), 5.1(b), 5.1(c), 5.2(a), 5.2(b) and 5.2(c) give the *dropping rate* of the generic DDRA and the QoS and fault-tolerant based DDRA with BS failures and link failures respectively. From these figures, regardless of BS failures or link failures or the number of failures or the system load or the number of wireless channels we can deduce that the QoS and fault-tolerant based DDRA guarantees lower *dropping rate* that goes until 2% and is maintained at this value. Compared with the QoS and fault-tolerant based DDRA, the generic DDRA cannot guarantee the quality of service of the inter-handoff calls. For the generic DDRA, as the system load increments and the failures rises, the *dropping rate* is going up. For example, with 400 wireless channels, when the system load is 100%, and with 10 BS failures the *dropping rate* of the generic DDRA is around 50% and with 12 link failures the *dropping rate* is about 9% . However, as for the QoS and fault-tolerant based DDRA, the

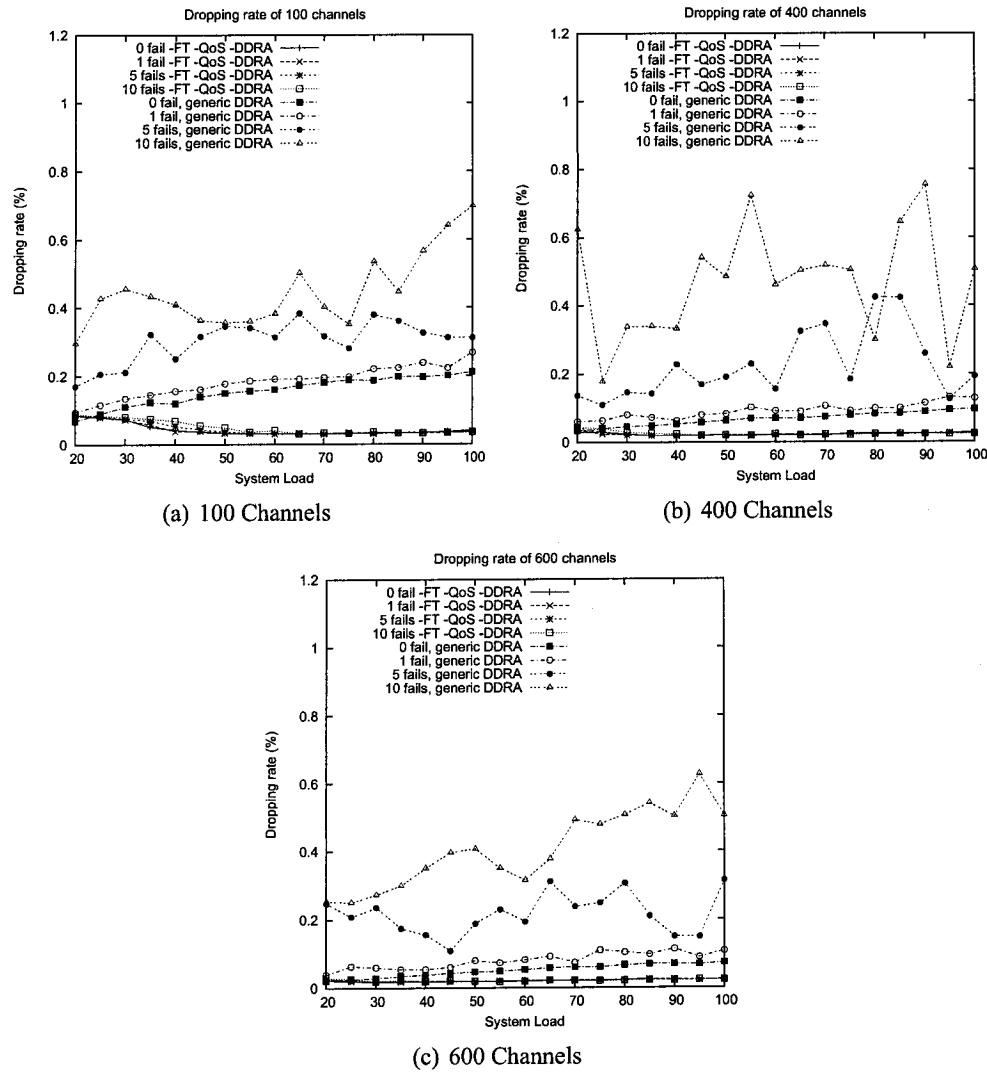


Figure 5.1: Dropping Rate with BS Failures

*dropping rate* always keeps about 2%. Hence, we can see from these graphs that our protocol improves greatly the *dropping rate*.

Figures. 5.3(a), 5.3(b), 5.3(c), 5.4(a), 5.4(b) and 5.4(c) present the *denial rate* of the generic DDRA and our QoS and fault-tolerant protocol with BS failures and link failures respectively. Since our algorithm supports the QoS feature, some wireless channels are reserved for inter-handoff, causing the reduction of available wireless channels to the new call requests. With no BS failures, the *denial rate* of our protocol is 20% higher than that of the generic

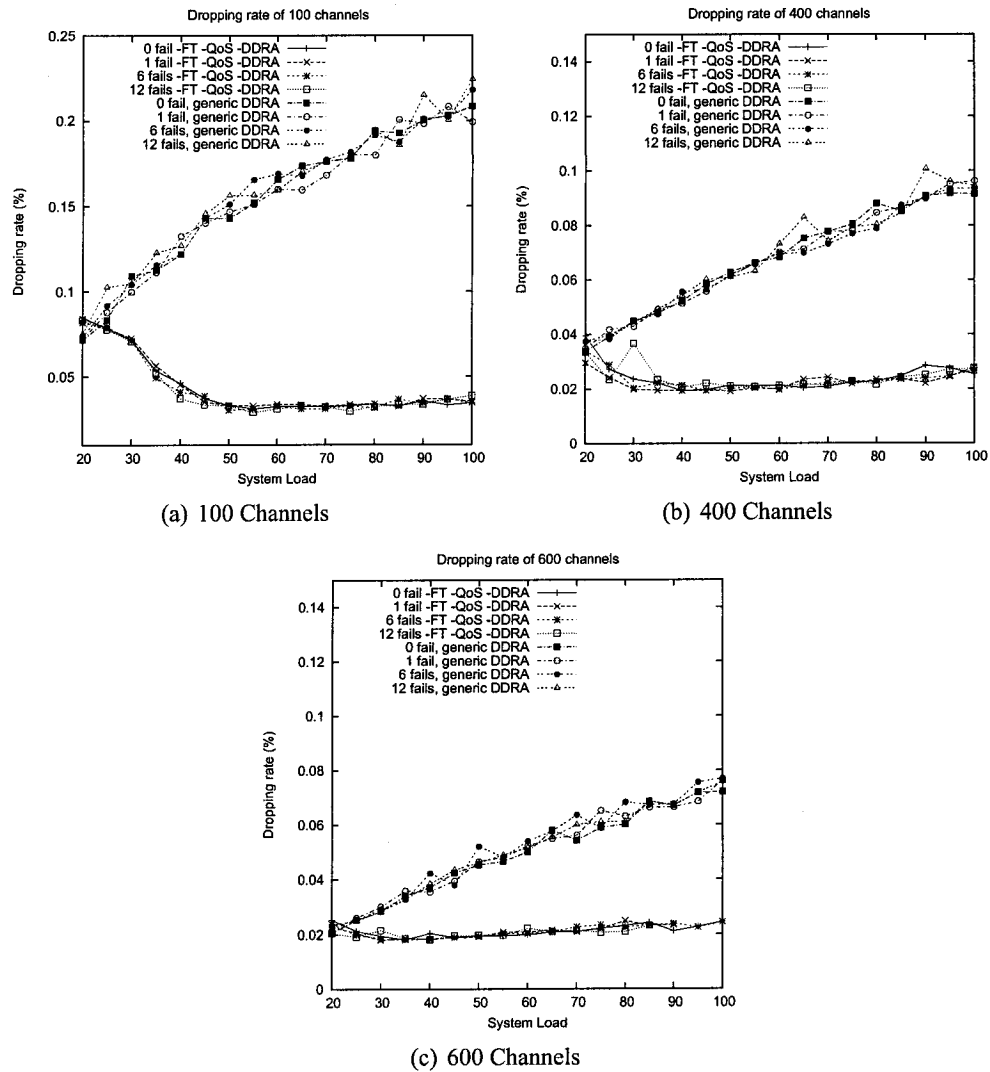


Figure 5.2: Dropping Rate with Link Failures

DDRA when the system load is 100%. With one BS failure, the *denial rate* of our protocol is almost equal to that of the generic DDRA. With ten BS failures, the *denial rate* of our scheme is 30% whereas that of the generic DDRA is 80%. Hence, our algorithm outperforms the generic DDRA when the number of BS failures increases by reducing considerably the *denial rate*. For link failures, our protocol has higher *denial rate* due to the channel reservation for the inter-handoff calls. We can further see that the *denial rate* keeps stable along the system load.

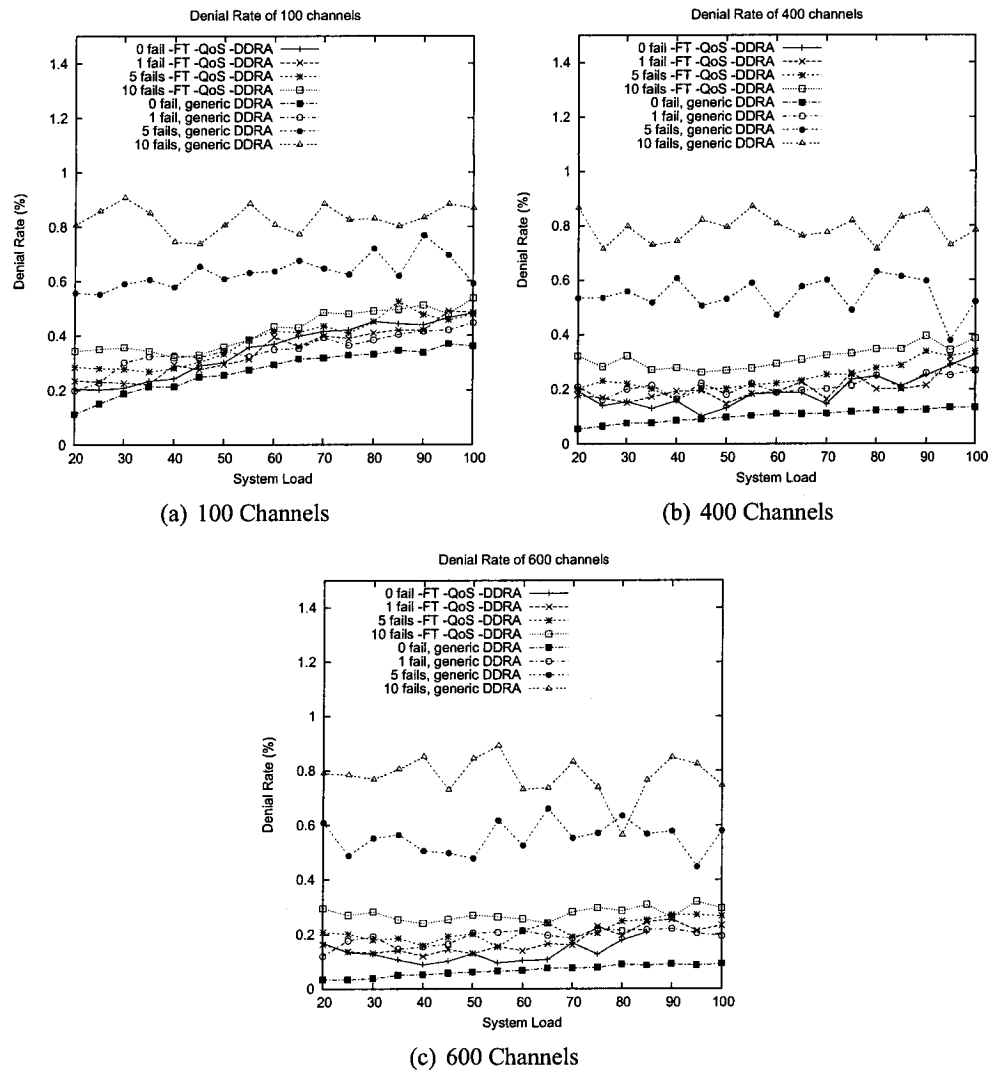


Figure 5.3: Denial Rate with BS Failures

Figures 5.5(a), 5.5(b), 5.5(c), 5.6(a), 5.6(b) and 5.6(c) show the experiment results related to the *acquisition time* metric. We can deduce that our QoS and fault-tolerant based scheme has higher *acquisition time* which is about 0.005s more than that of the generic DDRA when the system load is 60% and with 400 wireless channels for example. This is because some wireless channels are reserved for inter-handoffs. Thus more time is spent to find an available wireless channel because of the reduced wireless channels for new call requests.

The final performance metric is the *message complexity*. Figures 5.7(a), 5.7(b) and 5.7(c)

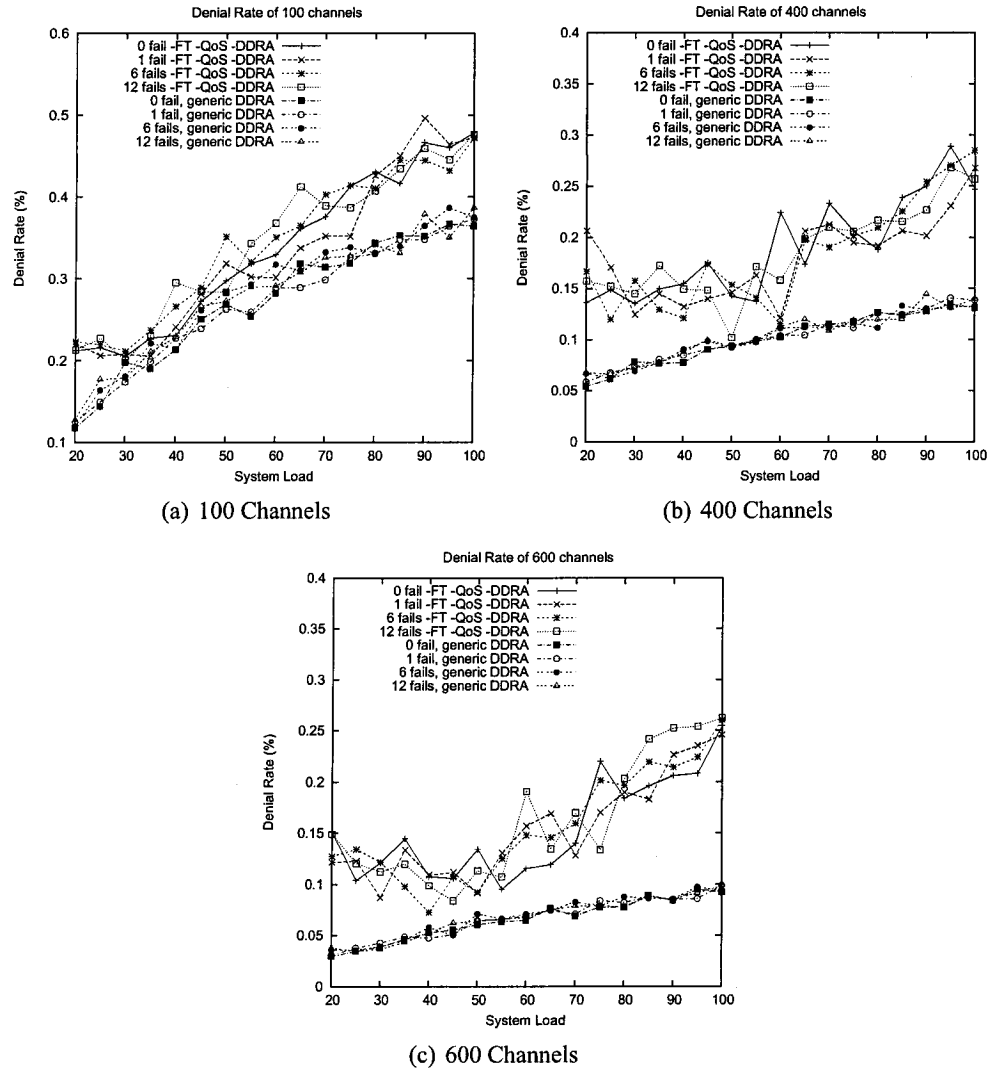


Figure 5.4: Denial Rate with Link Failures

show that for base station failure the *message complexity* of our algorithm fluctuates depending from the number of failed base stations. For example, with 400 wireless channels and for 10 base station failures, the message complexity of the generic DDRA when the system load is 100% is 4 messages, whereas it is 20 messages for the QoS and fault-tolerant based algorithm. The reason of this higher number of messages is mainly due to the fact that *message complexity* is measured for successful calls, and since our algorithm provides the fault tolerance feature, some extra messages are used to negotiate with failed neighbors in order to

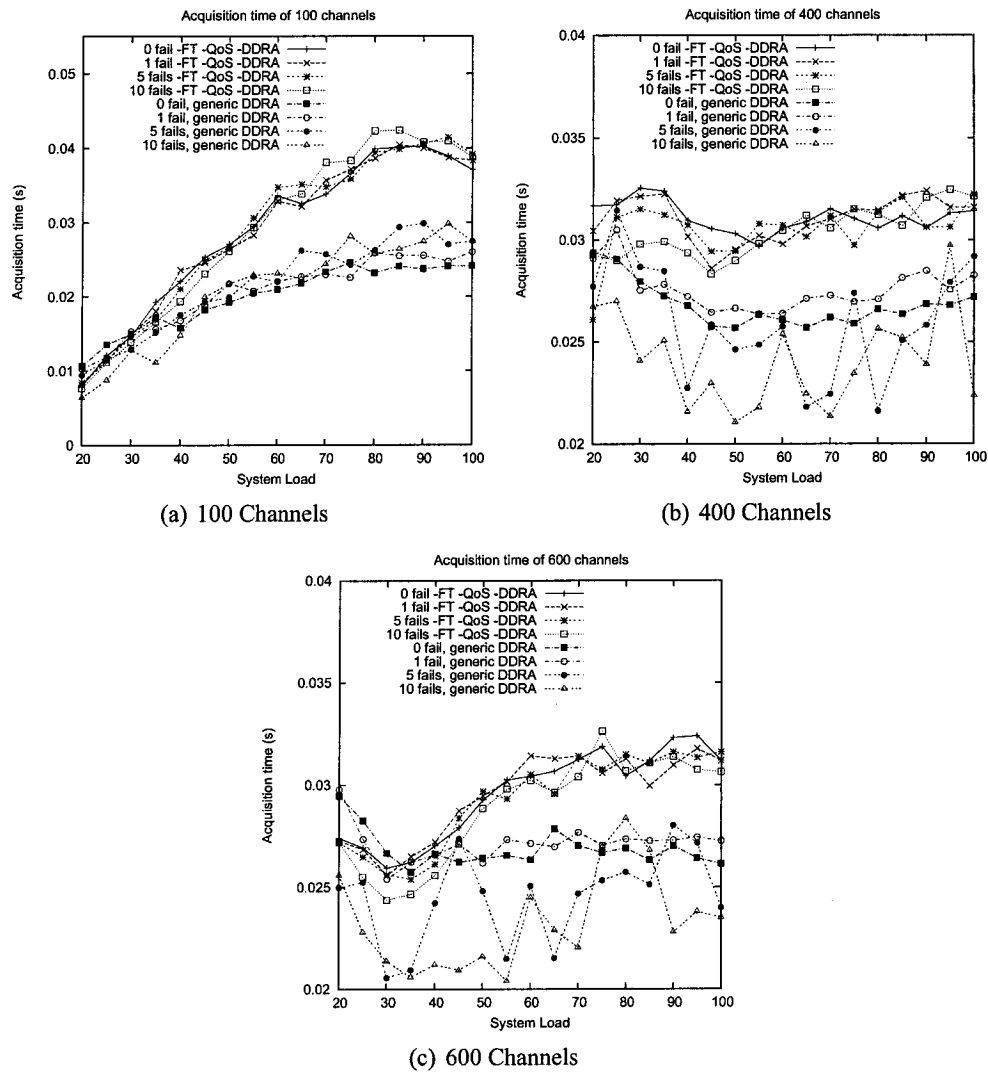


Figure 5.5: Acquisition Time with BS Failures

satisfy the maximum number of calls. Whereas when the number of BS failure is 0 or 1, the *message complexity* of our protocol is less than that of the generic DDRA. The main reason behind this is due to the fact that fewer extra messages are used for handling BS failures.

Figures. 5.8(a), 5.8(b) and 5.8(c) show that the *message complexity* with link failures for the QoS and fault-tolerant based algorithm is less than the generic DDRA independently from the number of link failures. This is mainly due to the procedure of reserving some wireless channels for *inter-handoffs* required to guarantee the QoS characteristic. Thus, the wireless

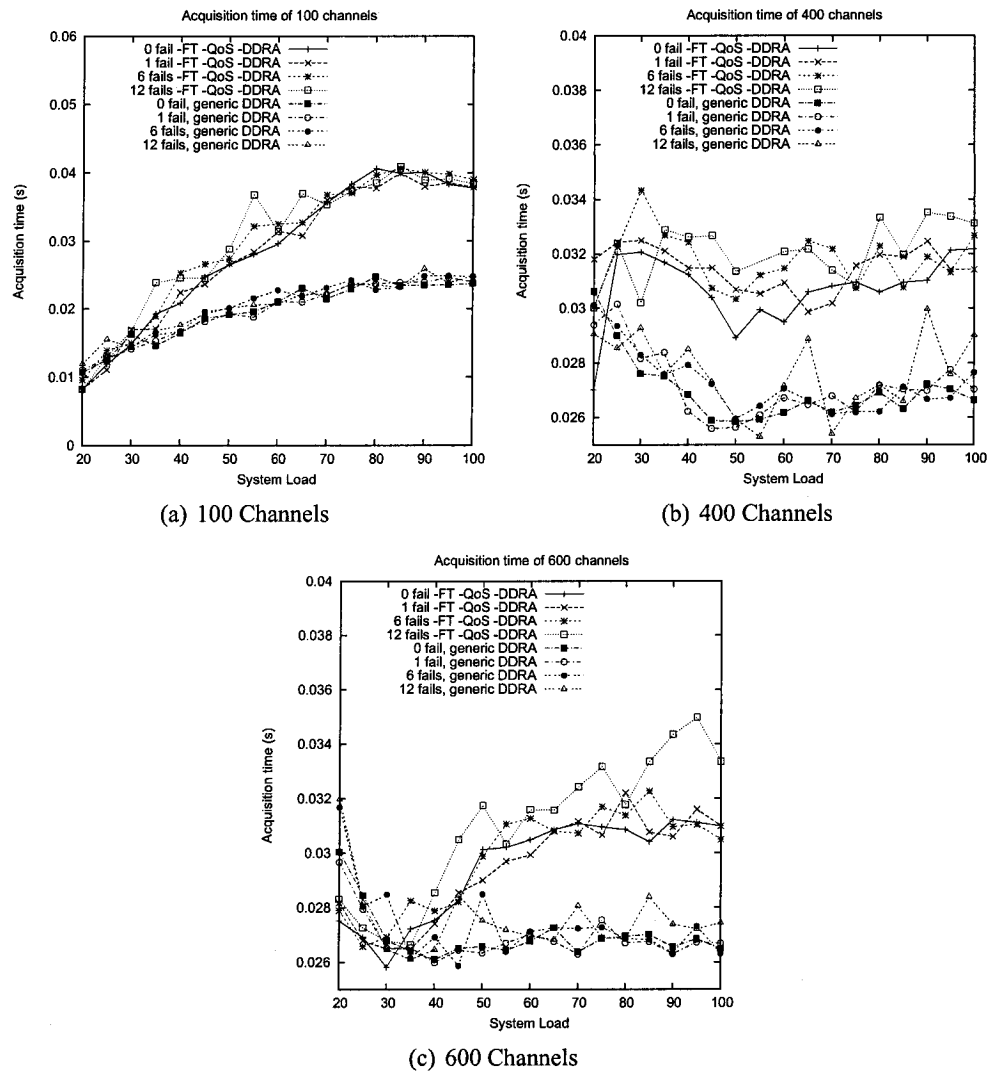


Figure 5.6: Acquisition Time with Link Failures

channels for an inter-handoff call can be acquired directly.

Besides, we can depict from these figures that the *message complexity* goes down as the system load increases. This is because, when the system load increases, it is more possible that neighbors of a base station were occupying channel groups, and thus, the base station has the information concerning its neighbors. In this case, the base station just verifies if the channel groups are available or not. In the negative case, the BS will drop the call without exchanging more messages.

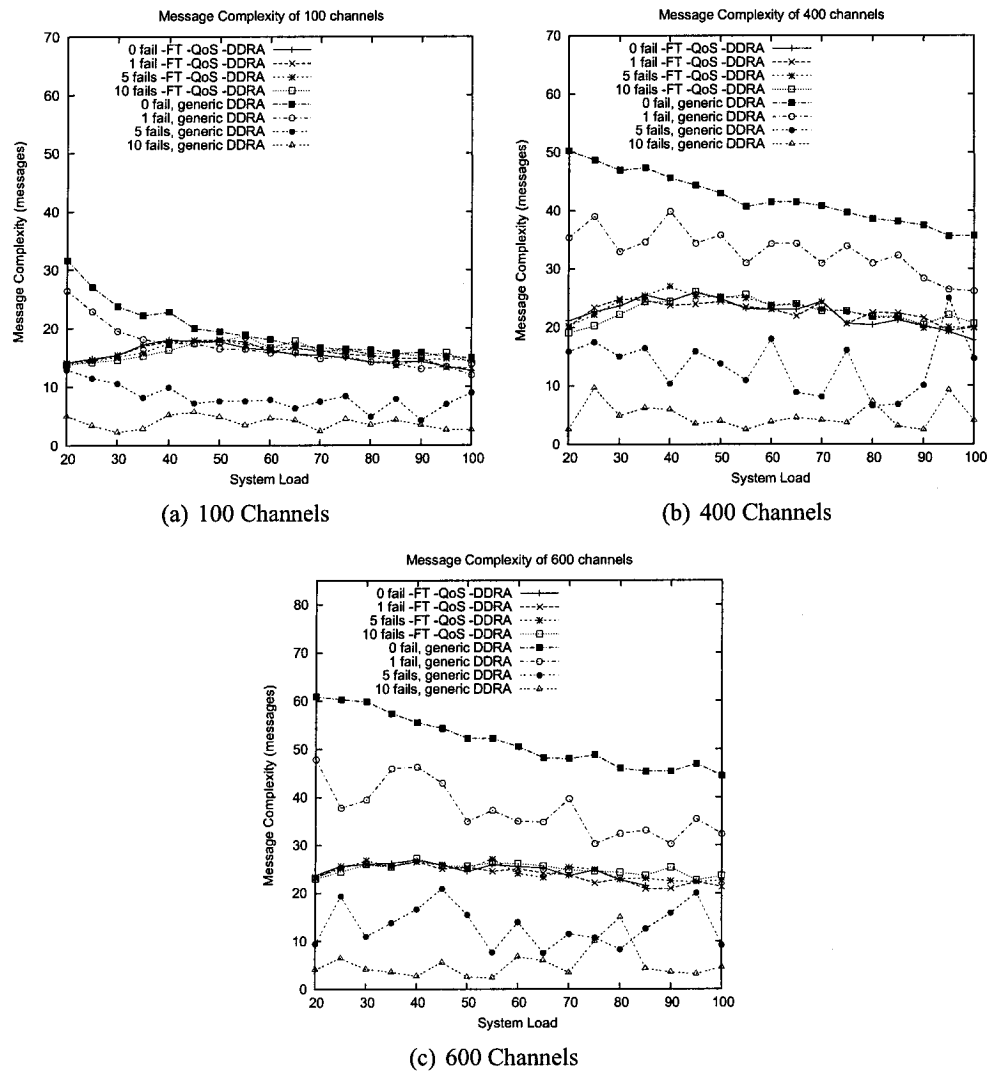
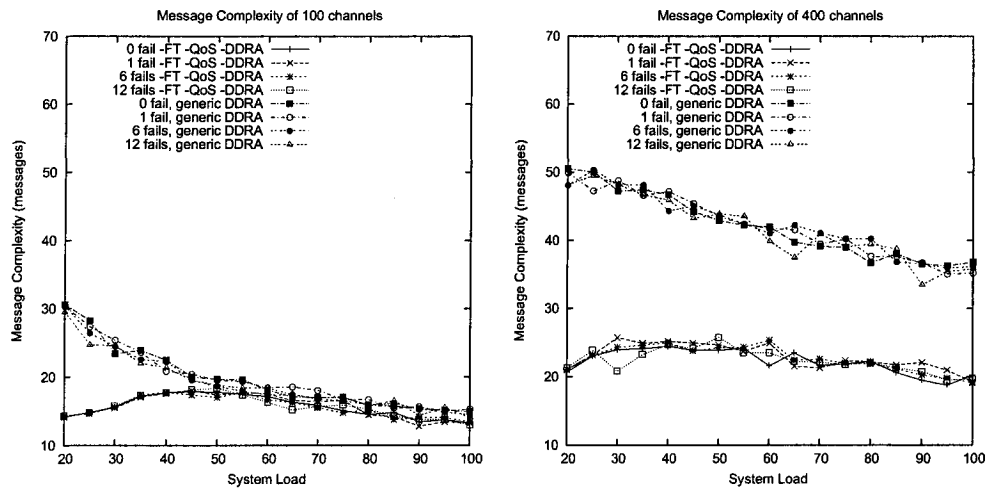


Figure 5.7: Message Complexity with BS Failures

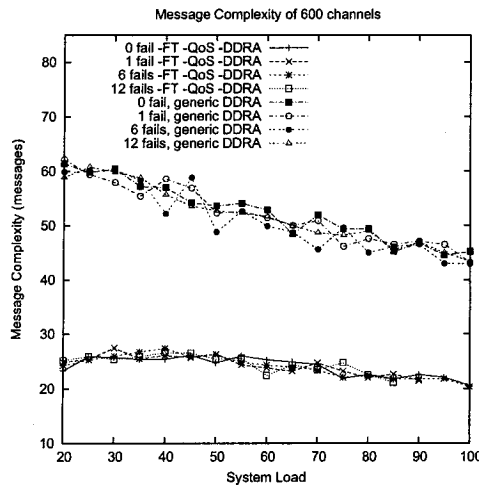
## 5.4 Conclusion

In this chapter, we have proposed an efficient QoS and fault tolerant based distributed channel allocation protocol for cellular networks. We have presented our protocol, and have discussed our approach on how to integrate the QoS and fault tolerant components within the dynamic channel allocation protocol. We have also reported on its simulation based performance using an extensive set of experiments. Our results indicate clearly that our protocol



(a) 100 Channels

(b) 400 Channels



(c) 600 Channels

Figure 5.8: Message Complexity with Link Failures

achieves a low dropping rate and helps to reduce significantly the phone call denial rate.

# Chapter 6

## Conclusion

The research on wireless and mobile cellular networks has been growing exponentially in recent years. In fact, due to the increasing demands for mobile telephone services, many research have been focusing upon the problem of channel allocation. In the previously proposed protocols on channel allocation, only few of them considered the QoS and the fault tolerant aspects while assigning available channels to mobile users. The lack on doing such improvements, can have a dramatic effect on the performance of mobile cellular networks since many calls will be blocked or dropped.

In this chapter, we will summarize our contributions to the dynamic channel allocation field and outline possible directions for future research.

### 6.1 Summary of Contributions

In this thesis we focused on the problem of channel allocation based on mutual exclusion. First, we presented a classification model that groups many proposed channel allocation protocols based upon their characteristics. Then, we presented our contributions and reported the performance of our algorithms using an extensive set of simulation experiments.

The contributions of this thesis are as follow:

1. *A comprehensive classification of channel allocation protocols for wireless and mobile*

*networks.*

2. *A QoS based dynamic channel allocation algorithm:* In order to reduce the connection dropping rate and guarantee the QoS provisions, an adaptive channel reservation mechanism is proposed. Our algorithm dynamically adjusts the number of reserved channels for the *handoff* with respect to the traffic situation. Our results indicate that our QoS based dynamic channel allocation protocol performed significantly better with dropping rate.
3. *A Fault tolerant based dynamic channel allocation algorithm:* We also proposed a distributed fault-tolerant channel allocation scheme which can work well under the mobile host failures, base station failures and communication link failures. Our results indicate that our protocol outperforms largely the generic channel allocation algorithm by reducing the denial rate up to 70 percent.
4. *A QoS and fault tolerant based dynamic channel allocation algorithm:* We integrated the quality of service (QoS) and fault tolerant components within the mutual exclusion based dynamic channel allocation algorithm. These two components are utile to the success of channel allocation protocols and the deployment of future generation of wireless networks. Our results have shown clearly that our protocol achieves a low dropping rate and minimizes considerably the phone call denial rate.

## 6.2 Future Works

We can identify several directions for further research:

- We will perform an analytical study and present an analytical perspective of our work and discuss further the performance metrics used to evaluate our QoS based dynamic channel allocation algorithm using an analytical model.

- We will extend our work to support ad-hoc networks and propose a channel allocation algorithm working in an ad-hoc environment where the links between base stations are wireless.

## References

- [1] L. G. Anderson. A Simulation study of some dynamic channel assignment algorithms in a high capacity mobile telecommunications system. *IEEE Transactions on Communications*, 21(11):1294–1301, November 1973.
- [2] A. 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 Transactions on Vehicular Technology*, 44:14–23, February 1995.
- [3] R. Beck and H. Panzer. Strategies for Handover and Dynamic Channel Allocation in Micro-Cellular Mobile Radio Telephone Systems. *IEEE Vehicular Technology Society*, 1:178–185, 1989.
- [4] R. Borndfer, A. Eisenblter, M. Grschel, and A. Martin. Frequency Assignment in Cellular Phone Networks. *Annals of Operational Research*, pages 73–93, 1998.
- [5] A. Boukerche. *Algorithms and Protocols for Wireless and Mobile Systems*. 2005.
- [6] A. Boukerche, S. Hong, and T. Jacob. A distributed Algorithm for Dynamic Channel Allocation. *Mobile Networks and Applications*, 7(2):115 – 126, April 2002.
- [7] J. Caffery and G. Stuber. Subscriber Location in CDMA Cellular Networks. *IEEE Transactions on Vehicular Technology*, 47(2):406–416, 1998.
- [8] G. Cao and M.Singhal. An Adaptive Distributed Channel Allocation Strategy for Mobile Cellular Networks. *J. Parallel and Dist. Comput*, 60(4):451–473, 2000.
- [9] V. Cerny. A thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm. *Journal of Optimization Theory and Applications*, 45:41–51, 1985.
- [10] I. Chih-Lin and P. Chao. Local Packing - Distributed Dynamic Channel Allocation at Cellular Base Station. *IEEE Global Telecommunication Conference (GLOBECOM)*, 1993.
- [11] I. Chih-Lin and P. Chao. Distributed Dynamic Channel Allocation Algorithms with Adjacent Channel Constraints. *Private Land Mobile Radio Communications (PIMRC)*,

- pages 69–75, 1994.
- [12] M. Choy and K. A. Singh. Efficient Distributed Algorithms for Dynamic Channel Assignment. *Proc. of 7th IEEE Intl Symp. on Personal, Indoor and Mobile Radio Communication*, 1996.
- [13] D. Cox and D. Reudink. Dynamic Channel Assignment in Two Dimension Large-scale Mobile Radio Systems. *Bell System Technical Journal*, 51:1611–1628, 1972.
- [14] D. Cox and D. Reudink. Increasing Channel Occupancy in Large Scale Mobile Radio Systems: Dynamic Channel Reassignment. *IEEE Transactions On Vehicular Technology*, 22:218–222, 1973.
- [15] K. S. Das, S. K. Sen, and R. Jayaram. A Dynamic Load Balancing Strategy for Channel Assignment Using Selective Borrowing in Cellular Mobile Environment. *ACM/Baltzer J. on Wireless Networks*, 3(5):333–347, October 1997.
- [16] X. Dong and H. Lai. An Efficient Priority-Based Dynamic Channel Allocation Strategy for Mobile Cellular Networks. *INFOCOM*, pages 892–899, 1997.
- [17] M. Duque-Anton, D. Kunz, and B. Ruber. Channel Assignment for Cellular Radio Using Simulated Annealing. *IEEE Transactions on Vehicular Technology*, 42(1):14–21, February 1993.
- [18] S. M. Elnoubi, R. Singh, and S. C. Gupta. A new Frequency Channel Assignment Algorithm in High Capacity Mobile Communication Systems. *IEEE Transaction on Vehicular Technology*, 31(3):125–131, August 1982.
- [19] J. Engel and M. Peritsky. Statistically-Optimum Dynamic Server Assignment in Systems with Interfering Servers. *IEEE Transactions on Vehicular Technology*, 22:203–209, 1973.
- [20] H. G. Forman and J. Zahorjan. The Challenges of Mobile Computing. *Computer*, 27(4):38–47, 1994.
- [21] N. Garg, M. Papatriantafilou, and P. Tsigas. Distributed List Coloring: How to Dynamically Allocate Frequencies to Mobile Base Stations. *Proc. of the 8th IEEE Symp. on*

- Parallel and Dist. Processing (SPDP'96)*, page 18, October 1996.
- [22] D. J. Gibson. *Mobile Communications Handbook*. Springer-Verlag New York Inc., Secaucus, NJ, USA, 1999.
- [23] F. Glover. Tabu Search - Part I. *ORSA Journal on Computing*, 1, pages 190–206, 1989.
- [24] F. Glover. Tabu Search - Part II. *ORSA Journal on Computing*, 2, pages 4–32, 1990.
- [25] T. Kahwa and N. Georganas. A Hybrid Channel Assignment Scheme in Large Scale Cellular-Structured Mobile Communication Systems. *IEEE Transactions on Communications*, 26:432–438, 1978.
- [26] J.-S. Kim, S. Park, P. Dowd, and N. Nasrabadi. Channel Assignment in Cellular Radio using Genetic Algorithms. *Wireless Personal Communications*, 3(3):273–286, 1996.
- [27] S. Kirkpatrick, C. Gelatt, and J. Vecchi. Optimization by Simulated Annealing. *Science*, (4598), May 1983.
- [28] D. Kunz. Channel Assignment for Cellular Radio Using Neural Networks. *IEEE Transactions on Parallel and Vehicular Technology*, 40(1):188–193, February 1991.
- [29] J. Little. A Proof of the Queuing Formula  $L = W$ . *Operations Research*, 9:383–387, 1961.
- [30] V. Maniezzo and A. Carbonaro. An ANTS Heuristic for the Frequency Assignment Problem. *Future Generation Computer Systems*, pages 927–935, 2000.
- [31] S. Nanda and J. Goodman. Dynamic Resource Acquisition: Distributed Carrier Allocation for TDMA Cellular Systems. *IEEE Global Telecommunication Conference (GLOBECOM)*, pages 883–889, 1991.
- [32] S.-H. Oh and D.-W. Tcha. Prioritized Channel Assignment in a cellular Radio Network. *IEEE Transactions On Communications*, 40:1259–1269, 1992.
- [33] K. Okada and F. Kubota. On Dynamic Channel Assignment in Cellular Mobile Radio Systems. *Proc. IEEE Int. Symp. on Circuits and Sys.*, 2:938–941, 1991.
- [34] K. Okada and F. Kubota. On Dynamic Channel Assignment Strategies in Cellular Mobile Radio Systems. *IEICE Trans. Fundamentals*, 75:1634–1641, 1992.

- 
- [35] R. Prakash, N. G. Shivaratri, and M. Singhal. Distributed Dynamic Fault-Tolerant Channel Allocation for Mobile Computing. *IEEE Transactions on Vehicular Technology*, 48(6):1874–1888, November 1999.
- [36] J. B. Punt and D. Sparreboom. Mathematical Models for the Analysis of Dynamic Channel Selection or Indoor Mobile Wireless Communications Systems. *PIMRC*, E6.5:1081–1085, 1994.
- [37] M. Serizawa and D. Goodman. Instability and Deadlock of Distributed Dynamic Channel Allocation. *Proc. 43rd IEEE VIC*, pages 528–531, 1993.
- [38] M. Serizawa and D. Goodman. Instability and Deadlock of Distributed Dynamic Channel Allocation. *Proc IEEE Vehicular Technology Society*, 43:528–531, 1993.
- [39] J. Sin and N. Georganas. A Simulation Study of a Hybrid Channel Assignment Scheme for Cellular Land-Mobile Radio Systems with Erlang-C Service. *IEEE Transactions on Communications*, 9:143–147, 1981.
- [40] R. Singh, S. M. Elnoubi, and C. Gupta. A New Frequency Channel Assignment Algorithm in High Capacity Mobile Communication System. *IEEE Transactions on Vehicular Technology*, 31, 1982.
- [41] A. Thesen and E. L. Travis. *Simulation for Decision Making*. PWS Publishing Co., Boston, MA, USA, 1991.
- [42] J. Yang, Q. Jiang, D. Manivannan, and M. Singhal. A Fault-Tolerant Distributed Channel Allocation Scheme for Cellular Networks. *IEEE Transactions On Computers*, 54(5):616–628, May 2005.
- [43] W. Yue. Analytical Methods to Calculate the Performance of a Cellular Mobile Radio Communication System with Hybrid Channel Assignment. *IEEE Transactions on Vehicular Technology*, 40:453–459, 1991.
- [44] M. Zhang and T. Yum. Comparisons of Channel-Assignment Strategies in Cellular Mobile Telephone Systems. *IEEE Transaction on Vehicular Technology*, 38(4):211–215, November 1989.

- 
- [45] M. Zhang and T.-S. Yum. The Non-Uniform Compact Pattern Allocation Algorithm for Cellular Mobile Systems. *IEEE Transactions On Vehicular Technology*, 40:387–391, 1991.