EFFICIENT ALGORITHMS AND FRAMEWORK FOR BANDWIDTH ALLOCATION, QUALITY-OF-SERVICE PROVISIONING AND LOCATION MANAGEMENT IN MOBILE WIRELESS COMPUTING

DISSERTATION

To be presented to the Graduate Council of
University of North Texas in Partial
Fulfilment of the Requirements

For the Degree of

DOCTOR OF PHILOSOPHY

By

Sanjoy Kumar Sen, M.S.
Denton, Texas
December, 1997

Sen, Sanjoy Kumar, Efficient Algorithms and Framework for Bandwidth Allocation, Quality-of-Service Provisioning and Location Management in Mobile Wireless Computing Doctor of Philosophy (Computer Science), December 1997, 163 pp, 9 tables, 47 illustrations, bibliography, 55 titles.

The fusion of computers and communications has promised to herald the age of information super-highway over high speed communication networks where the ultimate goal is to enable a multitude of users at any place, access information from anywhere and at any time. This, in a nutshell, is the goal envisioned by the Personal Communication Services (PCS) and Xerox's ubiquitous computing. In view of the remarkable growth of the mobile communication users in the last few years, the radio frequency spectrum allocated by the FCC (Federal Communications Commission) to this service is still very limited and the usable bandwidth is by far much less than the expected demand, particularly in view of the emergence of the next generation wireless multimedia applications like video-on-demand, WWW browsing, traveler information systems etc. Proper management of available spectrum is necessary not only to accommodate these high bandwidth applications, but also to alleviate problems due to sudden explosion of traffic in so called *hot cells*.

In this dissertation, we have developed simple load balancing techniques to cope with the problem of tele-traffic overloads in one or more hot cells in the system. The objective is to ease out the high channel demand in hot cells by borrowing channels from suitable cold cells and by proper assignment (or, re-assignment) of the channels among the users. We also investigate possible ways of improving system capacity by rescheduling bandwidth in case of wireless multimedia traffic. In our proposed scheme, traffic using multiple channels releases one or more channels to increase the carried traffic or throughput in the system. Two orthogonal QoS parameters, called *carried traffic* and *bandwidth degradation*, are identified and a cost function describing the

total revenue earned by the system from a bandwidth degradation and call admission policy, is formulated. A channel sharing scheme is proposed for co-existing real-time and non-real-time traffic and analyzed using a Markov modulated Poisson process (MMPP) based queueing model.

The *location management problem* in mobile computing deals with the problem of a combined management of *location updates* and *paging* in the network, both of which consume scarce network resources like bandwidth, CPU cycles etc. An easily implementable location update scheme is developed which considers per-user mobility pattern on top of the conventional location area based approach and computes an *update strategy* for each user by minimizing the average location management cost. The cost optimization problem is elegantly solved using a genetic algorithm.

EFFICIENT ALGORITHMS AND FRAMEWORK FOR BANDWIDTH ALLOCATION, QUALITY-OF-SERVICE PROVISIONING AND LOCATION MANAGEMENT IN MOBILE WIRELESS COMPUTING

DISSERTATION

To be presented to the Graduate Council of
University of North Texas in Partial
Fulfilment of the Requirements

For the Degree of

DOCTOR OF PHILOSOPHY

By

Sanjoy Kumar Sen, M.S.
Denton, Texas
December, 1997

ACKNOWLEDGMENTS

I am grateful to my advisor Dr. Sajal K. Das for his encouragement, support and guidance at every stage of the work which constitutes this dissertation. I would like to take this opportunity to thank my committee members, Dr. Roy T. Jacob, Dr. Stephen Tate, Dr. Neal Brand from UNT and Mr. Kalyan Basu from Northern Telecom, for their review and suggestions for the improvement of this dissertation. I would also like to thank the Department of Computer Sciences at UNT for providing both computing and financial support during my tenure as a student.

Special thanks to my colleagues and friends at UNT, Amiya Bhattacharya, Rajeev Jayaram and Naveen Kakani, for the many valuable discussions and suggestions at various stages of this dissertation. I am grateful to Prof. Bhabani Sinha of Indian Statistical Institute for making me believe that I can accomplish further as a student. I acknowledge the financial support from Texas Advanced Technology Program grant and from Northern Telecom (Nortel), Richardson, Texas. My internship with the Wireless Systems Engineering Department at Nortel provided me with valuable insights for improving the contents of this dissertation.

This task would not have been possible without the constant support and faith of my parents and my sister, Pallabi. Their "presence", love and encouragement have been a constant source of sustainance for me over the past few years.

Sanjoy Kumar Sen
University of North Texas
December, 1997

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF NOTATIONS[1]

CHAPTER 2

S/I	signal to noise ratio
CP	compact pattern
N_{CP}	number of cells in CP
D_{cc}	cell distance between two co-channel cells
R_{cell}	cell radius
BS	Base Station
MSC	Mobile Switching Center

CHAPTER 3

C	fixed number of channels allocated to a cell
d_c	degree of coldness of a cell
d_c^{avg}	average degree of coldness of all cells
h	threhold parameter for a cell to become hot
r_p	width of peripheral region of a cell; used for user classification.
τ_{new}, α_d	timer values used by user classfication algorithm
RSS	received signal strength
H(L,B)	number of hot co-channel cells of cell L which are non-cochannel cells of cell B
CP	compact pattern

[1]Notations listed under each chapter may be referred in subsequent chapters but not repeated in their lists.

CHAPTER 3 (contd.)

R_{CP}	cell radius of compact pattern CP
D(L,B)	cell distance between cells L and B
X	number of channels required by a hot cell
p	cell perimeter
K_d	average mobile user density in a cell
δ	mesg. delay between BS and MSC
CC	set of cochannel cells of a cell
NCC	set of non-cochannel cells of a cell
H_{CC}	set of hot cochannel cells of a cell
H_{NCC}	set of hot non-cochannel cells of a cell
N_h	number of hot cells in the compact pattern
p_f (p'_f)	forward transition probability from cold (hot) states of cell in Markov chain
p_r (p'_r)	reverse transition probability from cold (hot) states of cell in Markov chain
λ_i	average arrival rate for class i $(1 \leq i \leq 4)$ traffic demand
μ	average service rate of the calls
λ'	average rate of channel borrow demand
P_{block}	call blocking probability
p_h	probability of a call being hot
Π_i	steady state probability of state i of a Markov chain

CHAPTER 4

'Ring'	a hexagonal structure containing at least one hot cell
'Peripheral Ring'	a hexagonal structure containing no hot cell
ring	generic name which includes 'Ring' and 'Peripheral Ring'
d_{hs}	diameter of a hot spot
H_n	hot spot whose outermost 'Ring' is ring n
N_n	number of cells in H_n
l_{i+1}	number of channels that a ring $i+1$ cell can lend to adjacent ring i cells
f_{d1}, f_{d2}, f_{d3}	fractional demand from adjacent cells (atmost 3) in next outer ring
CS	cold safe
CU	cold unsafe
CSS	cold semisafe
λ	average call arrival rate
μ	average service rate of calls
μ'	probability of satisfying the channel demand for an entire hotspot
λ''	average call arrival rate from all cells
μ''	average service rate for calls from all cells
sd	density of hot cells in a hot spot

CHAPTER 5

D_{bw}	a bandwidth degradation policy
K	number of traffic classes
α_i	proportion of class i calls in the call mix
$t_{c,i}$	number of admitted class i calls
$C_{t,i}$	revenue generated by each admitted class i call
$t_{d,i}$	number of degraded class i call
$C_{d,i}$	cost of degrading a class i call
Φ	effective revenue earned by the system using degradation and admission control
$\mathcal{D}$	demand vector for various traffic classes
$\mathcal{A}$	allocation vector for various classes of traffic
Γ	vector containing the fractions of incoming calls of various classes admitted in degraded mode
λ_R	average call arrival rate for real-time calls
λ_{NR}	average call arrival rate for non-real-time calls
μ_R	average service rate for real-time calls
μ_{NR}	average service rate for non-real-time calls
λ_{PNR}	average packet arrival rate for non-real-time calls
μ_{PNR}	average packet service rate for non-real-time calls
L_{avg}	average queue length

CHAPTER 6

G	network with the location areas (LA) as nodes
N	number of LA's
$\Gamma_G(i)$	neighboring node set of node i in G
u_i	binary decision variable denoting "update" or "no-update" for a user in LA i
$\mathcal{S}_u$	update strategy for a user
$\mathcal{S}_p$	paging strategy followed to locate the user
$P_{i,j}$	user transition probability from LA i to j
T_i	user's sojourn time in LA i
λ_p	rate of call arrival triggering paging
r	probability of a call arrival in a slot
P_i	transition probability out of LA i
τ	length of a time slot
$C_p(i)$	paging cost of LA i
$C_u(i,j)$	update cost for a transition from LA i to j
t_f	time slot at which first call arrives during the sojourn in LA i
$C_p^0(i)$	Average cost of paging the user for the first call in non-update LA i
U	set of update LA's for the user
$\bar{U}$	set of non-update LA's for the user
$LMC^{(1)}$	average cost per slot in reporting areas
$LMC^{(0)}$	average cost per slot in non-reporting areas
LMC	average total location management cost

CHAPTER 1

INTRODUCTION

Similar to the revolution brought about by microprocessors and later by multiprocessing technology in the last two decades, this decade is beginning to experience the fruits of yet another technological revolution, called *wireless communication* and *mobile computing*. The fusion of computer and communication technologies has promised to herald the age of information super-highway over high speed communication wire-line and wireless networks. The ultimate goal is to enable a multitude of users at any place access information from anywhere at any time. Thus, the desire for ubiquitous access to information is expected to characterize entirely new kinds of information systems and technology as we move into the 21st century. The rapidly emerging wireless communications systems based on radio and infrared transmissions, the advent of such technologies as cellular mobile telephony, personal communications systems (PCS), wireless PBXs, wireless LANs (local area networks), and the promise of broadband ISDN through ATM networks prelude a not-so-distant future realization of this goal. The driving force behind the field of mobile computing is also due to the commercial availability of hand-held, portable (e.g. laptop, palm-top) computers and personal digital assistants (PDAs) such as Apple Newton MessagePad and PARC Tab. This field has the potential to dramatically change the society as workers become untethered from their information sources and communication mechanisms.

1.1. **What is Mobile Computing ?**

Wireless or cellular mobile computing has three essential components – communication, mobility and portability – which distinguish it from its wire-line counterpart (see Figure 1.1). The issues and challenges in mobile computing belong to one of

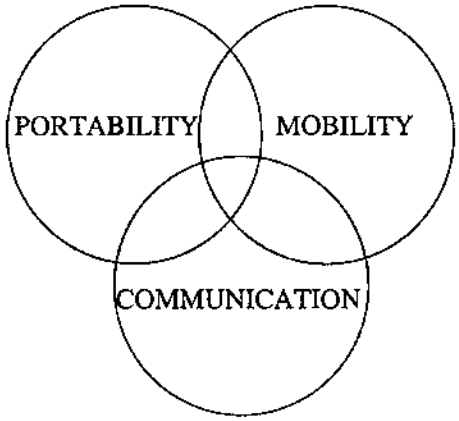


Figure 1.1: A Venn Diagram for Mobile Computing

these three components which are not necessarily independent. Wireless communication aspects deal with efficient bandwidth management and allocation to mobile applications attempting to solve the fundamental problem in wireless domain – that of low bandwidth availability. The mobility related issues are location management for the mobile user, data management, disconnections due to unreliable radio links and transmission security. The portability issues are concerned with the design of lightweight terminals with limited storage capacity and power consumption and also effective user interface design for them. Note that, designing portable computers for wireless applications entails consideration of many mobility related issues like disconnections due to unreliable links, as well as communication aspects like low bandwidth availability. Also, there are significant overlaps between mobility and radio communication domains, and as we will see later, one cannot be considered independent of the other.

In this dissertation, the portability issues of mobile computing are not being dealt with. The focus is mainly on the other two areas – *radio communication* and *mobility* management. The communication aspects have not been considered from a radio frequency (RF) engineer's point of view, but have been used to the extent required

for developing an algorithmic framework for managing the radio frequency resource, which is the underlying thread of continuity for this dissertation work.

1.2. Issues and Challenges in Mobile Computing

1.2.1. Bandwidth Management

In spite of the tremendous growth of the mobile communication users, the frequency spectrum allocated by the FCC (Federal Communications Commission) to this service in USA is still very limited. In particular, the alloted 824-894 MHz spectrum can only accommodate 832 40-KHz communication channels and 42 control channels. Although the 1850-1990 MHz band has recently been auctioned to the *personal communication services* (PCS) by the FCC, the amount of usable bandwidth is by far much less than the expected demand, which becomes even more prominent with the emergence of and the need to support the next generation wireless multimedia applications for roving users with portable computers. Examples include video-on-demand, news-on-demand, fax, e-mail, WWW browsing and traveler information systems. Proper management of available spectrum is necessary not only to accommodate these high bandwidth applications, but also to alleviate problems due to sudden explosion of traffic in so called hot cells.

Bandwidth management deals with the efficient allocation of the scarcely available radio frequency (RF) spectrum among the various wireless applications run by the mobile users in the system. The limitation of the frequency spectrum was felt about one and half decade back leading to the concept of cellular architecture [Mac79] as a collection of geometric areas called *cells* (typically, hexagonal-shaped), each serviced through an RF transceiver by a *base-station* (BS). A number of cells (or BS's) are linked to a *mobile switching center* (MSC) which also acts as a gateway of the cellular network to the existing wire-line networks like PSTN, ISDN, LAN/WAN or the Internet. A base station communicates with the mobile stations (or users) through wireless links, and with the MSC's through wire-line links. The cellular architecture

is shown in Figure 1.2.

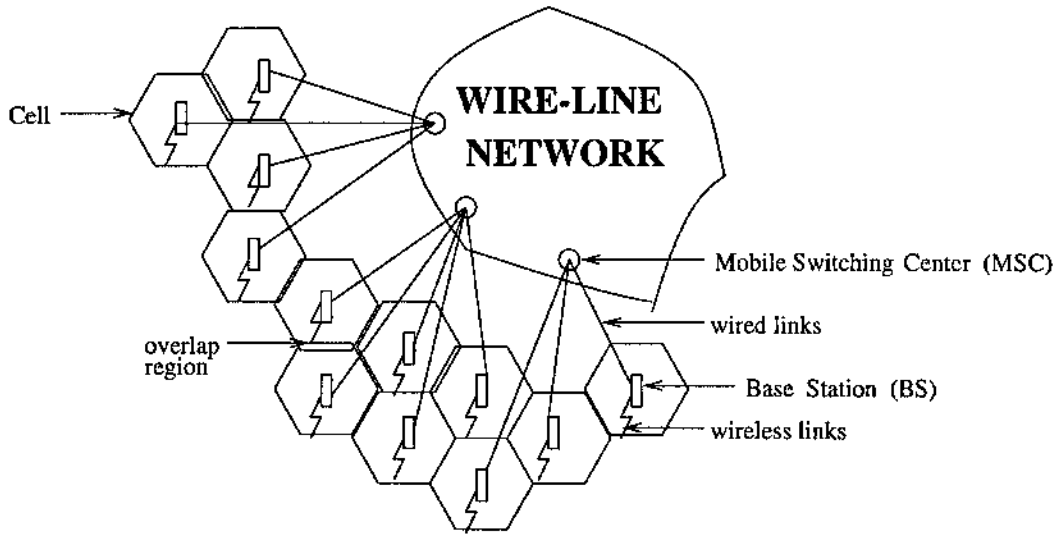


Figure 1.2: System model of a cellular mobile architecture

At the heart of bandwidth management in mobile computing is the channel assignment problem which deals with how to allocate wireless channels to the cells with a goal to maximize the frequency reuse. To this end, many channel assignment strategies have been proposed over the past few years. There is an associated problem with channel assignment which is motivated by the non-uniformity of user traffic in different cells which may lead to a gross load imbalance in the system. For example, there might be a traffic jam in the downtown area of a big city in the late afternoon, and the home-bound drivers are just too eager to call home using their cellular phones. This is typically known as the *hot cell* or *hot region* problem.

1.2.2. Mobility Management

There are several fundamental issues involved in mobility management. They are outlined here.

- *Location Management* is associated with the user mobility. As an user moves from one area (e.g. cell, or a subnetwork) to another, his location information need to be *updated* properly so that in case he receives a call, the system should be able to *page* him and track him down within a finite amount of time. Both update and paging necessitates exchange of messages to and from the system as well as some amount of computation and storage of user location data. These are expensive resources from the system's perspective as millions of roving users will be trying to use them simultaneously. Hence, an efficient location management scheme should be able to keep track of user movements with minimal consumption of of both wire-line and wireless resources. Various such schemes, seeking to maintain accurate location informations of the mobile users by efficient utilization of the system resources like wireless and wire-line bandwidth, buffer space, or CPU cycles, have been proposed. A common way to implement location management is to partition the entire geographical area into *zones* or *location areas (LA)* of cells, and maintain accurate informations about the current zone of each user. In case of an incoming call, the current zone is paged for the user.

- *Data Management* deals with location dependent data such as vehicular traffic, weather, local yellow pages, data like "where is the nearest gas station", which the mobile user may need to access frequently and quickly with the least possible consumption of wireless bandwidth and battery power.

- *Disconnections* may be either due to communication failures or deliberately executed by the mobile user. Disconnected operation is a mode that enables a mobile client to continue its operation during temporary suspension of wireless connection. The more autonomous the mobile equipment is, the better it can tolerate network disconnectivity.

- *Security* is also very important issue in mobile computing because radio transmission is more prone to eavesdropping and fraud than wire-line networks. Use of *personal identity number* (PIN), *subscriber identity module* (SIM) and encryption of the wireless data are some of the commonly used techniques.

1.2.3. Portability Management

Traditionally desktop computer/terminal design has been liberal with respect to dimension, power requirements, cabling, and heat dissipation. In contrast, we need exactly converse features in mobile terminals. Some of the major challenges faced in designing a mobile terminal are:

- Terminal should be lightweight, portable, and with a long battery life.
- Power consumption of the terminal should be low.
- Terminal can have only limited storage capacity.
- Minimal user interface is provided on the terminal.
- Cost of the terminal should be low.

1.3. Contributions of this Dissertation

In this dissertation, efficient resource utilization techniques are proposed in two important areas of wireless mobile computing, namely, *bandwidth management* and *location management*. Our contributions are summarized below.

1. **Bandwidth Management:**

   In this dissertation, we propose to use simple load balancing techniques in cellular mobile environment to cope with the problem of teletraffic overloads which happens when there is suddenly a great demand in one or more hot cells in the system. A cell is classified as 'hot' if the number of available channels is less

than a certain threshold. The primary objective of our approach is to ease out the high channel demand in hot cells by borrowing channels from suitable cold cells and by proper assignment (or, re-assignment) of the available channels among the users. Assuming a fixed channel assignment scheme is available to start with, we have proposed a centralized and a distributed channel borrowing schemes, and a channel assignment strategy as part of our load balancing algorithm. Performance of our algorithm is analyzed using simple birth and death Markov models and detailed simulation experiments. Comparisons with other existing schemes show significant improvement of performance in case of a highly loaded system.

We also propose another load balancing scheme to cope with the problem of traffic overloads in a hot spot which is a region of adjacent hot cells. A hot spot is conceived as a stack of hexagonal 'Rings', each containing at least one hot cell. In our load balancing approach, a hot cell in 'Ring i' borrows channels from its adjacent cells in 'Ring i+1' with the help of a channel *demand graph*, to ease out its high channel demand. This structured lending mechanism decreases excessive co-channel interference and borrowing conflicts, which are prevented through channel locking in other schemes. A probabilistic analysis model for our system incorporating load balancing is proposed and extensive simulation experiments are conducted to compare our scheme with other existing schemes.

While load balancing strategies attempt to eliminate teletraffic imbalances in the system by resource (channel) migration, we investigate possible ways of improving system capacity by rescheduling bandwidth in case of wireless multimedia communication systems, in which certain classes of traffic use multiple channels for a single high bandwidth application. In our proposed scheme, traffic using multiple channels releases one or more channels to increase the carried traffic or capacity of the system, thereby undergoing *graceful degradation*. A framework for modeling QoS degradation strategies for real-time and non-real-

time multi-media traffic is proposed. Two orthogonal QoS parameters, called *carried traffic* and *bandwidth degradation*, are characterized and a function describing the total revenue earned by the system from a *bandwidth degradation policy* is formulated. This cost function model is then extended to incorporate a *call admission policy.* Detailed simulation experiments are conducted to validate the proposed model. In most systems, the real-time traffic has preemptive priority over the non-real-time traffic. As more real-time traffic vies for bandwidth, the already existing non-real-time traffic are preempted and buffered for future rescheduling which is dynamically adjusted against bandwidth availability. A novel *channel sharing scheme* is proposed for co-existing real-time and non-real-time traffic and analyzed using a Markov modulated Poisson process (MMPP) based queueing model. A suitable QoS metric called the *average queue length* for non-real-time traffic, is derived from the model which is also validated by simulation experiments.

For wireless systems requiring contiguous spectrum allocation to high bandwidth traffic, a new approach called *bandwidth compaction* (similar to memory compaction in operating systems) is proposed for efficient utilization of the available spectrum.

2. **Location Management:**

An optimal and easily implementable location update scheme is proposed which considers per-user mobility pattern on top of the conventional zone (LA) based approach. Most of the practical cellular mobile systems partition a geographical region into location areas (LAs) and users are made to update on entering a new LA. The main drawback of this scheme is that it does not consider the individual user mobility and call arrival patterns. Combining per-user mobility and call arrival patterns with the LA-based approach, we have proposed an optimal update strategy which determines whether or not a user updates in each LA.

The update strategy minimizes the average location management cost derived from a user-specific mobility model and call generation pattern. Thus, the user updates in certain preselected location areas called *reporting areas.* The location management cost optimization problem is elegantly solved using a *genetic algorithm.* Detailed simulation experiments are conducted to capture the effects of mobility and call-arrival patterns on the location update strategy. The experimental results clearly demonstrate that for low user residing probability in LAs, low call arrival rate and high update cost, skipping location updates in several LAs leads to minimization of the overall location management cost.

1.4. Chapter Organization

The rest of this dissertation is organized as follows. Chapter 2 gives an overview of the various bandwidth allocation schemes with or without load balancing considerations. Our load balancing strategies for hot cell problem are detailed in Chapter 3, which also includes performance analysis using probabilistic models and simulation. A load balancing scheme for a cluster of hot cells (or a hot region) is described in Chapter 4. Chapter 5 introduces the Quality-of-Service issues for the next generation wireless multimedia applications. Our QoS provisioning algorithms, using bandwidth degradation and call admission policies for both real-time and non-real-time calls as well as a novel approach called bandwidth compaction, are also detailed in the same chapter. In Chapter 6, we develop a new selective location update scheme for individual user using the conventional location area infratsructure in already deployed wireless systems. The scheme proposes to optimize both wireline and wireless resources and is implemented using a genetic algorithm. Chapter 7 concludes this dissertation with a summary and directions of future research.

CHAPTER 2

THE CHANNEL ASSIGNMENT PROBLEM

In spite of the tremendous growth of the mobile communication users, the frequency spectrum allocated by the FCC (Federal Communications Commission) to this service in USA is still very limited. In particular, the alloted 824-894 MHz spectrum can only accommodate 832 40-KHz communication channels and 42 control channels. Although the 1850-1990 MHz band has recently been auctioned to the *personal communication services* (PCS) by the FCC, the amount of usable bandwidth is by far much less than the expected demand, which becomes even more prominent with the emergence of and the need to support the next generation wireless multimedia applications for roving users with portable computers. Examples include video-on-demand, news-on-demand, fax, e-mail, WWW browsing and traveler information systems. Proper management of available spectrum is necessary not only to accommodate these high bandwidth applications, but also to alleviate problems due to sudden explosion of traffic in the so called hot cells. In a practical wireless system, a *channel* is defined to be a fixed block of communication medium such as ⟨time slot, carrier frequency⟩ tuple in the narrow band TDMA systems or simply a fixed block of radio frequency bandwidth as in FDMA systems.

Since frequency channels are a scarce resource in a cellular mobile system, many schemes have been proposed to assign frequencies to the cells with a goal to maximize the frequency reuse. They can be broadly classified as fixed or static [Mac79, ESG82, ZY89], dynamic [CR73, ZY89] and flexible or hybrid [TI88, ZY89] assignment schemes.

In the *fixed assignment* (FA) schemes, a set of channels are permanently allocated to each cell, which can be reused in another cell, sufficiently distant, such that the co-channel interference is tolerable. Such a pair of cells is called *co-channel* cells.

In one type of FA scheme, clusters of cells, called *compact patterns*, are formed by finding the shortest distance between two co-channel cells [Mac79]. Each cell within a compact pattern is assigned a different set of frequencies. The advantage of an FA scheme is its simplicity, but the disadvantage is that if the number of calls exceeds the number of channels assigned to a cell, the excess calls are blocked. Variations of FA generally use *channel borrowing* methods [ESG82, TJ91], in which a channel is borrowed from one of the neighboring cells in case of blocked calls, provided that it does not interfere with the existing calls.

In *dynamic assignment* schemes, there is a global pool of channels from where channels are allocated on demand. A channel assignment cost function is computed and the channel with the minimum cost is assigned. For example, a channel can be selected for allocation from the global pool if it allows the minimum number of cells in which that channel will be locked. *Flexible channel assignment* (FCA) schemes combine the concepts of both fixed and dynamic strategies, whereby there is a fixed set of channels for each cell, but channels are also allocated from a global pool in case of shortage.

In this chapter, the *compact pattern* based fixed assignment scheme is first introduced. Various borrowing strategies as variations of this scheme are also discussed. Then we describe two major schemes proposed in the literature to include load balancing as one of the major criteria in the assignment strategies.

2.1. Fixed Assignment Scheme Using Compact Pattern

In the fixed assignment scheme, a set of frequencies is statically allocated to each cell and the same frequencies are reused in another cell sufficiently far apart such that the co-channel interference is negligible. The minimum distance at which the frequency can be reused with no interference is called the *co-channel reuse distance.* The *signal-to-interference ratio* is an index of channel interference. The objective is to maximize the reuse of the assigned frequencies under the constraints of co-channel

interference. The scheme is sketched below [Mac79].

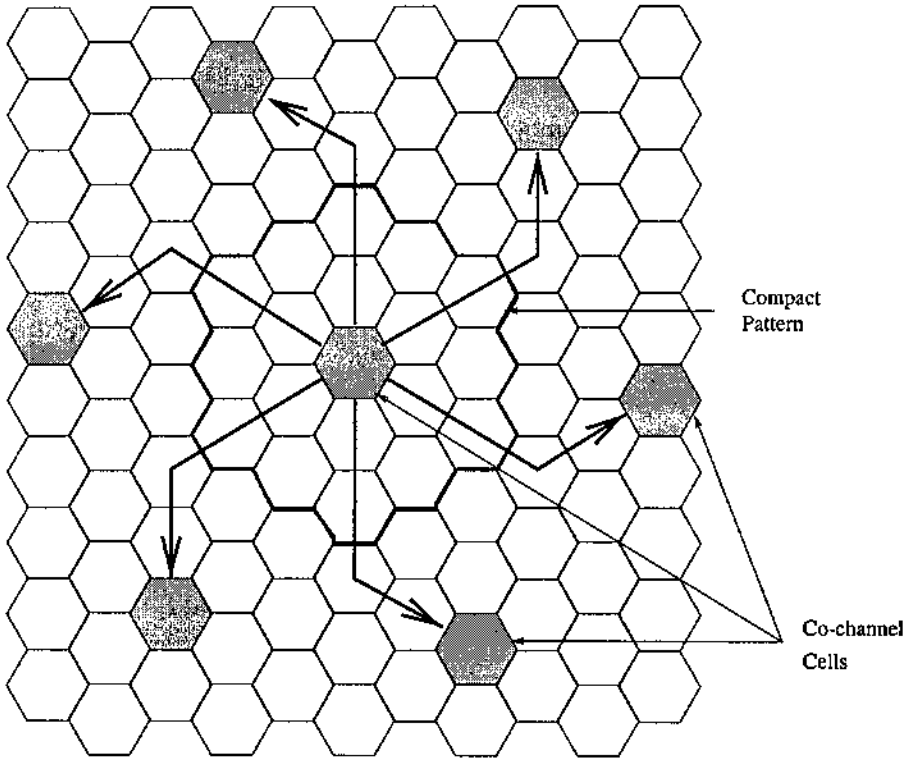


Figure 2.1: Determination of co-channel cells

Two parameters i and j, called *shift parameters*, are first determined This is illustrated in Figure 2.1. Starting from any cell, move i cells along any one of the six emanating chains of hexagons, turn clock (or counter-clock)-wise by sixty degrees and then move j cells along the chain. The destination cell is the nearest *co-channel* cell of the originating cell. For each cell, there are two sets of six nearest co-channel cells, depending on the clockwise and anti-clockwise moves. By repeating this pattern, clusters of cells are formed in which each cell is assigned a different set of frequency channels. Such a cluster is called a *compact pattern.*

The number of cells in a compact pattern is given by $N_{CP} = i^2 + ij + j^2$, which

determines the number of different channel sets to be formed [Mac79]. If D_{CC} is the Eucledian distance between two nearest co-channel cells and R_{cell} is the cell radius, the *co-channel reuse ratio* is defined as D_{CC}/R_{cell}. Now $\frac{D_{CC}}{R_{cell}} = \sqrt{3N_{CP}}$ for hexagonal cells [Lee96]. The signal-to-interference ratio, $S/I \approx \frac{(D_{CC}/R_{cell})^4}{6}$, in most practical systems. If the minimal allowable S/I is known, D_{CC}/R_{cell} can be easily determined and hence the size of the compact pattern in terms of i and j.

Various graph coloring techniques have also been used to solve the frequency assignment problem optimally, i.e. to maximize reuse of the available frequency channels. An alternative approach using simulated annealing has been suggested recently [DKR93]. Although the fixed assignment schemes perform well under heavy traffic conditions, one major drawback is that if the number of calls exceeds the channel set for the cell, the excess calls are blocked until channels become available. To cope with this situation, strategies have been proposed to borrow channels from other neighboring cells [ESG82, TJ91].

2.2. Borrowing Strategies

- **Simple Borrowing:** This variant of the fixed assignment scheme proposes to borrow a channel from neighboring cells provided it does not interfere with the existing calls. The borrowed channel is locked in those co-channel cells of the lender, which are non-co-channel cells of the borrower. Since channels are locked, system performance suffers under heavy traffic conditions.

- **Hybrid Borrowing:** In this variant, the fixed channel set assigned to a cell is divided into two groups, one for local use only and the other for lending channels to neighboring cells on demand. The number of channels in each group is determined apriori depending on the history of traffic conditions.

- **Channel Ordering:** This is an extension of the hybrid scheme, where the number of channels in the two groups can vary dynamically depending on traffic

conditions. Each channel is ordered such that the first channel has the highest priority of being locally used, and the last channel has the highest priority of being borrowed. The ordering may change according to the traffic pattern. A released higher order channel is relocated to an ongoing call in a lower order channel, so as to reduce locking of the borrowable channels.

2.3. Frequency Assignment Schemes with Load Balancing

There is an associated problem with channel assignment which is motivated by the non-uniformity of user traffic in different cells which may lead to a gross load imbalance in the system. For example, there might be a traffic jam in the downtown area of a big city in the late afternoon, and the home-bound drivers are just too eager to call home using their cellular phones. This is typically known as the *hot cell* or *hot region* problem. It has been seen in already deployed systems that, in highly congested areas of a modern city, there can be up to 40% call blockade due to channel (carrier) unavailability. Under these circumstances, load balancing or sharing should be an intrinsic part of the channel assignment schemes. The schemes described next attempt to alleviate this non-uniformity of traffic demand which might affect the system performance to a significant extent.

2.3.1. Directed Retry

The *directed retry* scheme due to Eklundh [Eklun86] assumes that the neighboring cells overlap and some of the users in the overlapping region are able to hear transmitters from the neighboring cells almost as well as in their own cell. If there is a channel request from a subscriber and there is no free channel, then the subscriber is requested to check for the signal strength of the transmitters in the neighboring cells. If a channel from a neighboring cell with adequate signal strength is found, the call is set up using that channel. If no such channel is found, the call attempt fails.

To alleviate this drawback, Karlsson and Eklundh [KE89] proposed to incorporate

load sharing by treating subscribers differently based on whether they are able to hear more than one transmitter. Whenever the base station finds more than a certain number of channels occupied, it requests the users to check for the quality of the channels in the neighboring cells. If some of the users report that they are able to receive transmission from neighboring cells adequately well, a search for free channels begins in those cells and an attempt is made to move as many subscribers to those cells as possible. There is no concept of borrowing channels from neighboring cells but subscribers are simply moved from one cell to another by the process of *hand-off* [1]. If no subscriber finds an adequate channel to setup or switch a call to, the base station tries to find a free channel in the original cell or let the call proceed as usual.

Although this load sharing scheme increases the number of potential channels to a certain extent, the main disadvantages are the increased number of hand-offs and the co-channel interference. Also since a user has to be in the bordering regions of neighboring cells in order to be a potential candidate for a hand-off, it puts a severe constraint on the efficacy of the algorithm to share load. The bordering region of two cells can be very small which reduces the probability that a sufficient number of users can be found in those regions to carry the load over to the neighboring cells in case of a drastic increase of the channel demand in a cell, as might happen in the so called hot cells. Some of the neighboring cells may themselves be hot, in which case it may not be a good idea to transfer load between them.

2.3.2. Channel Borrowing without Locking (CBWL)

In the CBWL scheme, Jiang and Rappaport [JR94] proposed to use channel borrowing when the set of channels in a cell gets exhausted, but to use them under reduced transmission power. This is done to avoid interference with the other co-channel cells of the lender using the same frequency. Channels can be borrowed only from

[1] A user, communicating via a channel in a cell, finds channel quality very poor and switches over to a new channel.

adjacent cells in an orderly fashion. The set of channels in a particular cell is divided into seven groups. One group is exclusively for the users in that cell, while each of the six other groups caters for channel requests from one of the neighboring cells. This structured lending mechanism decreases excessive co-channel interference and borrowing conflicts, which are prevented through channel locking in other schemes. If the number of channels in a channel-group gets exhausted, a subscriber using one of the channels can be switched to an idle channel in another group, thereby freeing up one in the occupied group. Since borrowing channels are transmitted at low power, not all users (within range) are capable of receiving them. If such a user finds all the channels occupied, an ordinary user using regular channel can handover its channel to the former while itself switching to a borrowed channel if available. This particular variation is called CBWL with channel rearrangements or CBWL/CR.

The CBWL scheme has some advantages over the dynamic and flexible assignment, because channel utilization is increased without channel locking. But one serious drawback of the reduced power transmission strategy is that not all users are in the right zone all the time for borrowing channels if the need arises. The CBWL/CR attempts to solve this by channel reassignments thereby increasing the number of intra-cellular hand-offs. Also since only a fraction of the channels in all the neighboring cells is available for borrowing, this coupled with the previous drawback can seriously affect the performance in case of hot cells. For example, if there exists a cluster in which a hot cell is surrounded by six other hot cells, then the CBWL scheme performs very poorly for the hot cell in the center, since no channel is available for borrowing. Additionally, the limitation on the number of channels available for borrowing places severe restriction on the system performance if at least some of the neighboring cells are hot as well.

The above mentioned problems in the directed retry with load balancing or CBWL scheme are magnified when the channel demand is very high, e.g. in a hot cell. The load balancing schemes which we are going to propose in the next two chapters

overcome these problems when the channel demand is prohibitively high in some cells and comparatively low in others. The experimental results show that in an overloaded cellular environment with a large number of hot cells, our scheme exhibits a significant performance improvement over the other existing schemes in terms of reduction in the call blocking probability.

2.4. Summary

In this chapter, various schemes for the channel assignment problem are introduced. The problem of high traffic demand in certain cells at certain parts of the day, leading to the formation of hot cell or hot spot is described. A few schemes which lead to load balancing or sharing in such a cellular mobile environment are summarized from the existing literature. Some specific drawbacks of these schemes, which motivated the load balancing schemes proposed in the next two chapters of this dissertation, are also discussed.

CHAPTER 3

LOAD BALANCING STRATEGIES FOR THE HOT CELL PROBLEM

While the motivation behind any assignment strategy is the better utilization of the available frequency spectrum with the consequent reduction of the call blocking probability in each cell, not much attention [Eklun86, KE89, JR94] has been paid on the problem of non-uniformity in traffic demand in different cells which may lead to a gross imbalance in the system performance. As in the example discussed earlier, a situation might arise where there is a traffic jam in the downtown area of a big city after a big football game and the mobile users are just too eager to call home, making a few cells heavily loaded. It is desirable that the system is able to cope with such traffic overloads in certain cells. We will designate those cells as '*hot*', in which the traffic demand exceeds a certain *threshold* value at any time instant. Otherwise, they will be denoted as '*cold*'. In the next section, we shall give a more formal definition of hotness and coldness of a cell.

Fixed assignment schemes, by themselves, are unable to handle the *hot cell* problem [JR94] as the number of channels assigned to each cell cannot be changed, although they usually perform better under heavy traffic conditions than dynamic schemes. Dynamic assignment schemes are expected to cope better with traffic overloads to a certain extent, but on high demands the computational overheads deceive the purpose of the scheme. Flexible schemes will face the same problem as they are basically reduced to dynamic schemes on high channel demand, presumably after the fixed channel sets get exhausted.

Two schemes – directed retry with load balancing [KE89] and channel borrowing without locking [JR94] – brings load sharing into consideration while assigning channels. Their relative merits and demerits have been discussed in Chapter 2. The problems in the directed retry or CBWL schemes are magnified in the presence of a

hot cell. Our load balancing scheme proposes to alleviate these problems when the channel demand is prohibitively high in some cells and comparatively low in others. The experimental results show that in an overloaded cellular environment with a large number of hot cells, our scheme exhibits a significant performance improvement over the other existing schemes in terms of reduction in call blocking probability.

In this chapter, we propose and analyze a centralized and a distributed load balancing scheme for cellular mobile architecture. Such implementations form a novel approach for load balancing in the channel assignment problem in mobile computing. Each of these schemes consists of two parts – (*i*) a *resource migration* (channel borrowing) algorithm and (*ii*) a *resource allocation* (channel assignment) algorithm. Our load balancing architecture is described in Section 3.1. In Section 3.2, the new load balancing schemes, a centralized and a distributed version of the same basic scheme, are described in detail. The proposed scheme is analyzed using a Markov model and some numerical results are given in Section 3.3. Section 3.4 describes the simulation experiments in detail. The performances of the centralized and distributed versions of our load balancing scheme are also compared in the same section. A comparison of the new scheme with two of the existing ones, namely, directed retry and CBWL, is laid down in Section 3.5.

3.1. Load Balancing Architecture

The high level architecture for our load balancing scheme is as in Figure 1.2. A given geographical area consists of a number of hexagonal cells, each served by a base station. The base station and the mobile users communicate through wireless links and form small localized wireless networks. A group of cells are again served by a mobile switching center (MSC), each of which also acts as a gateway for the wireless networks to the base stations. The MSC's are connected through fixed wire-line networks.

Each cell is statically allocated a fixed set of C channels according to a compact

pattern based fixed assignment scheme. Let us now classify the cells and also the mobile users in a cell.

3.1.1. Cell Classification

A cell can be classified as hot or cold according to the value of its *degree of coldness*, which is defined as

$$d_c = \frac{number\ of\ available\ channels}{total\ number\ of\ channels} = \frac{number\ of\ available\ channels}{C} \tag{3.1}$$

If $d_c \leq h$, where $h > 0$ is a fixed *threshold* parameter, the particular cell is *hot*, otherwise it is *cold*. Typical values of h are 0.2, 0.25 etc., and determined by the average call arrival and termination rates, and also by channel borrowing rates from other cells. The usefulness of the parameter h is to keep a subset of channels available so that even when a cell reaches the hot state, an originating call need not be blocked. When a cell reaches a 'hot' state, it merely serves as a warning that the available resource (i.e. channel) in that cell has reached a critical low point and migration of resources is necessary to mitigate the pressure which might arise due to a sudden traffic explosion.

3.1.2. User Classification in a Cell

The mobile users in a cell are classified as one of three types – *new*, *departing* or *others*.

- A user is *new* if it is in the current cell for a period less than τ_{new} time units.
- A *departing* user is one who is within the shaded region bordering an hexagonal cell A as shown in Figure 3.1, and receiving a steadily diminishing signal strength from the base station of A for the last α_d time units, where $\alpha_d < \tau_{new}$. The width, r_p, of the shaded region determines the probability of finding an user in that region.

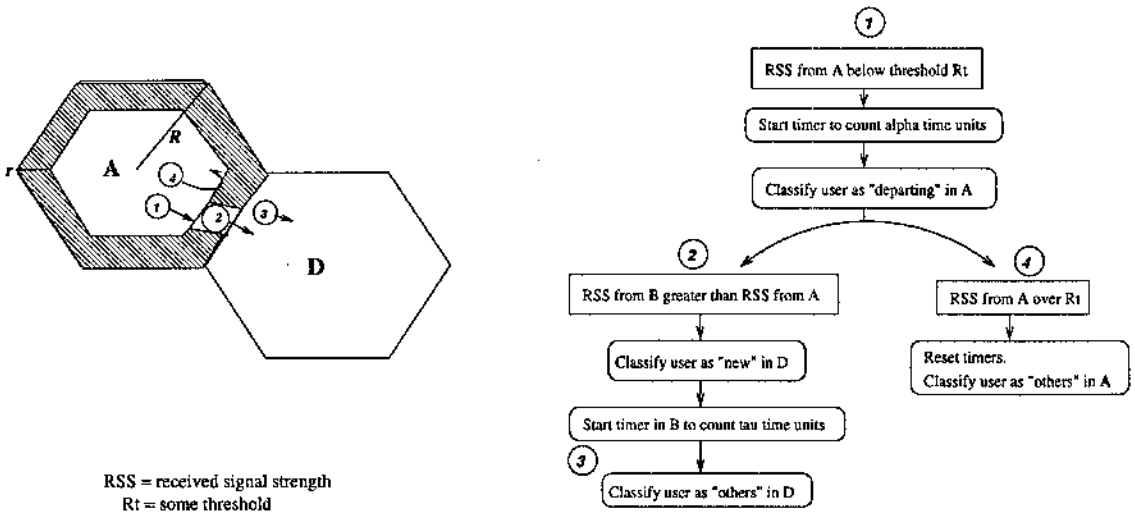


Figure 3.1: Classification of users in a cell

- A user who is neither 'new' nor 'departing' will be classified as *others*.

The class of a user is determined as follows by the base station (BS) controlling it. The BS periodically monitors the quality of the received signal strength (RSS) from each user through special control channels. Whenever a user enters a new cell, BS designates it as a 'new' user and starts a timer for τ_{new} time units. The user remains 'new' if its state does not change over this period of time.

If the RSS from a mobile user is less than a certain threshold value, it means that the user is within one of the shaded peripheral regions in the boundary of a cell (refer to Fig. 3.1). If a 'new' or 'others' type of user is within one of the peripheral regions and its signal strength begins to diminish, the BS starts another timer for α_d time units. If the signal strength continues to decrease for the next α_d time units as well, the user is classified as 'departing'. If within this time period, the signal strength stops diminishing, the counter is reset and the user's state remains unchanged.

As defined earlier, a user who is neither 'new' nor 'departing' is classified as 'others'. A 'new' user who remains in that state for τ_{new} time units, is converted to 'others' at the end of the period. A 'departing' user will change its status to 'others'

if the RSS from the user stops decreasing for over three time periods (of duration α) as monitored by the BS. An user can never be converted from the 'departing' or 'others' class to the 'new' class.

3.2. Load Balancing with Selective Borrowing (LBSB) – A New Scheme

In this section, first our channel borrowing strategy is described in detail. This includes (*i*) derivation of a lender cell selection criteria, and (*ii*) computation of certain channel borrowing parameters, such as the number of channels to borrow, the width of the peripheral region, and the *destination* cell of a 'departing' user. Then a centralized and a distributed implementation of the channel borrowing algorithm are described. The performances of these two implementations are compared based on the number of messages exchanged and the running times (speedup).

Dynamic load balancing is commonly used in processor scheduling in distributed systems to achieve better processor utilization [WLR93]. Such a scheme can be either centralized or decentralized. In *centralized* schemes, there is one central server running the load balancing algorithm either periodically or on demand from certain number of processors. When a processor (or group) becomes overloaded, it sends a message to the server to initiate load balancing in the demand-driven scheme. In the periodic scheme, processors have to wait for the server to initiate load balancing. The problem with centralized schemes is that much depends on the central server and maintaining continuous status information of the processors may lead to enormous update of traffic as well as computational overheads.

In *decentralized* (or distributed) schemes, each processor is capable of running the load balancing algorithm whenever it gets overloaded. After the necessary task migrations, each processor updates its individual status and obtains the status of other processors in the system through polling. In such schemes, a lot of overhead is incurred from the large number of messages needed to exchange status information by the processors. Also each processor should be capable of running the load balancing

algorithm when required.

In this chapter, we propose and analyze a centralized and a distributed load balancing scheme for cellular mobile architecture. Each of these schemes consists of two parts – (*i*) a *resource migration* (channel borrowing) algorithm and (*ii*) a *resource allocation* (channel assignment) algorithm. In the centralized scheme, the resource migration algorithm runs centrally in an MSC server, while in the distributed scheme, a distributed resource migration algorithm runs in all the base stations. It is the resource migration algorithm which distinguishes the centralized scheme from the distributed one. The resource allocation algorithm, on the other hand, is common to both the schemes and runs locally in each base station whenever there is a resource demand.

The centralized resource migration algorithm is run periodically by a server residing in the MSC in charge of a group of cells. This means that this scheme is applied only to a few cells implying that the load on the central server would not be too high. It is the task of the base stations to update the server about the current degree of coldness of the corresponding cells. On the other hand, the distributed resource migration algorithm is triggered by a base station as soon as its cell reaches the hot state. Here the base stations exchange messages among themselves concerning the degree of coldness, cell distance etc. and all these messages are routed through the MSC. Hence the MSC, in the distributed load balancing scheme, acts mostly as a basic router.

3.2.1. Basic Idea of Channel Borrowing in LBSB

The underlying idea is the migration of channels from the cold cells to the hot ones. A cold cell is not allowed to borrow channels from any other cell. Similarly, a hot cell cannot lend any channel to another cell. The privilege of borrowing channels is strictly limited to hot cells and the 'departing' users in such a cell will have the highest priority of using the borrowed channels. Also a certain fixed number of channels need

to be borrowed to relieve pressure from hot cells. When a cell is hot and channel migration is needed, channels are borrowed from some cold cells and stored in the available channel set of the borrower cell as *borrowed channels*. How these available channels are assigned to the users in a cell to maximize resource utilization is the problem of resource allocation. Under suitable conditions, borrowed channels are re-assigned to 'departing' users in the cell and the local channels which they were using are returned to the available channel set. Thus the channel set of the hot cell is replenished.

When a channel is borrowed from a cold cell, in order to avoid co-channel interference with the borrower cell, the borrowed channel has to be locked not only in the lender cell but also in its co-channel cells which are non-co-channel cells to the borrower. This group of cells might in turn include some hot cells where locking channels will be detrimental to our purpose of load balancing. We counter this drawback as follows.

1. Each cell will still have a limited number of available channels $(h \cdot C)$ when it reaches the hot state, so total call blockade will not result immediately. But conditions can become precarious soon.

2. Since a channel is borrowed in most cases by a 'departing' user from a hot cell, the duration of borrowing (and hence, locking) for a channel is expected to be low. Two cases may arise:

   - **Departing user going towards a cold cell:** In this case, the departing user usually borrows a channel from the destination cold cell and this channel being local to the destination cell, gets unlocked as soon as the user does a hand-off. This is a form of *soft hand-off* scheme and also a new channel is not required for the incoming user.

   - **Departing user going towards a hot cell:** In this case, the departing user borrows a channel from a particular cold cell satisfying some borrowing

conditions which are detailed in the next subsection. This 'departing' user soon becomes a 'new' user in the destination hot cell, where the available channel set is continuously being replenished by the local channels released by 'departing' users as they are re-assigned borrowed channels (Type 2 channel re-assignment). Some of these local channels are re-assigned to the 'new' or 'others' users carrying borrowed channels in order to release them (Type 1 channel re-assignment).

The reason why borrowed channels are assigned to 'departing' users with the highest priority in a hot cell is that they have the highest probability of crossing over to another cell and releasing the borrowed channels.

3.2.2. Parameter Computations

Channels are borrowed by a hot cell only from *suitable* cold cells within the compact pattern as discussed in Chapter 2. There are three parameters which determine the suitability of a cold cell as the potential lender, L.

- **Coldness:** The ratio of the number of available channels in a cell L to the total number of channels allocated to L determines the degree of coldness, $d_c(L)$, of that cell. It is a desirable property of any lender cell, the colder the cell the better. The coldest cell is analogous to the most lightly loaded processor in a distributed computing system and most often it is the best choice to migrate tasks to. Certainly, in our load balancing schemes this is not the sole criterion behind the determination of an appropriate lender.

- **Nearness:** This parameter is given by the cell-distance $D(B, L)$ between the borrower (hot) cell B and the lender cell L. It is desirable to have the lending cell as close to the borrowing cell as possible (the immediate neighbors being the most preferred) to decrease the message latency in the wire-line network.

- **Hot cell channel blockade:** This is another important parameter affecting the choice of the lender cell. The co-channel cells of the lender might contain certain number of hot cells where the borrowed channel has to be locked in order to avoid co-channel interference with the borrower cell, B. It is desirable to keep the number of such channel locking as low as possible in hot cells where channels are already a scarce resource. The number of hot co-channel cells of the lender cold cell, which are non-co-channel cells of the borrower, is denoted by $H(B, L)$.

Selection Criteria for Borrower and Lender Cells

For a user in a hot cell, a channel is borrowed from a cold cell such that the state of the cold cell is not altered. This means that after lending a channel to another cell, the degree of coldness, d_c, for that cell should not be equal to h, i.e. the cold cell should not become hot when a channel is reduced from its available channel set. We call this as the *basic borrowing criterion.*

Let B be a hot cell (borrower) which needs to borrow channels from cold cells. The set of cold cells in the compact pattern CP with B as the center cell (henceforth we will refer it as the compact pattern of B), are all probable candidates for borrowing a channel from. Let L be a probable candidate cell in CP for lending a channel. We select that cold cell as the lender whose parameters maximize the value of the following function

$$F(B,L) = \frac{d_c(L)}{\frac{D(B,L)}{R_{CP}} \cdot (\frac{1+H(B,L)}{7})}$$

Thus, the objective is to find a lender cell with a high degree of coldness, $d_c(L)$, close to the borrower cell, i.e. low $D(B, L)$, and having a low value of $H(B, L)$, i.e. fewer hot cell channel blockade. Here R_{CP} denotes the radius of the compact pattern in terms of cell distance which means $1 \leq D(B, L) \leq R_{CP}$. Also $0 \leq H(B, L) \leq 6$ holds for hexagonal cellular geometry. Hence, the factors R_{CP} and 7 in the denominator are used for normalization purposes.

Destination Cells for 'departing' Users

A 'departing' user listens to the transmissions of all the nearby base stations. The RSS from the current base station will be the highest. The next best signal strength is received from the closest neighboring cell of the user, towards which it is probably heading. We call it the *destination cell* for the user. A 'departing' user updates the base station about its current destination cell.

Thus, for each 'departing' user, the base station keeps track of which destination cell it is heading to. The six neighboring cells of any particular hot cell, B, are sequentially numbered from 1 to 6. The number of 'departing' users in B heading towards the ith neighboring cell is stored in an array, $NumDepart[i]$.

A borrowed channel will be re-assigned to a 'departing' user with the highest priority. To make this scheme useful, we follow a *directed borrowing* strategy. This is an extension of the directed hand-off strategy described in [KE89], where the users in the overlapping regions are handed over to the neighboring cells. In our scheme, the channels borrowed from the ith neighboring cell are re-assigned to the 'departing' users heading towards that cell. This is useful because of the following reasons:

1. This is a form of *soft hand-off* [1] scheme. The mobile user does not have to re-tune to a new channel after hand-off, as it can continue using the same channel in the new cell.

2. When a channel is borrowed by the user from its destination cell, it is assumed that the user will reach there in no time and release all the locked channels. So the channel locking time in most cases will be low.

3. The cell-distance of any destination cell from the borrower cell is 1, which is optimal.

[1] user acquiring a new channel before the current channel becomes unusable.

Number of channels to borrow

Let us define a parameter called the *average degree of coldness* of a cell, d_c^{avg}, which is computed by the MSC as the arithmetic mean of the d_c's of all of its cells over a certain period of time. In general, d_c^{avg} is the degree of coldness which every hot cell tries to achieve through channel migration from cold cells. Therefore, we can estimate the number of channels a hot cell needs to borrow, assuming that the number of available channels in the cell is $h \cdot C$.

Let X be the required number of channels to be borrowed. This leads to an increase in the number of available channels by the same margin. Hence,
$d_c^{avg} = \frac{h \cdot C + X}{C}$, yielding
$X = \lceil C \cdot (d_c^{avg} - h) \rceil$.

Of course, we have not specified yet how to estimate h, the threshold value of the degree of coldness of a cell where it turns hot from cold. To summarize, a hot cell needs to borrow $X = \lceil C \cdot (d_c^{avg} - h) \rceil$ number of channels from other cold cells.

Width of Threshold Region for 'departing' Users

As depicted in Figure 3.1, let r_p be the width of the shaded region on the boundary of a hexagonal cell with radius R. Assuming $r_p << R$, the area of the shaded region can be approximated by $p \cdot r_p$, where p is the cell perimeter. If the call originating rate is assumed to follow a uniform spatial distribution within a cell, then the number of 'departing' users making calls is given by $K_d \cdot p \cdot r_p$, where K_d is the *density* of mobile users (in a cell) making calls. If we confine the use of borrowed channels to the set of 'departing' users only, then $K_d \cdot p \cdot r_p \geq X$. Thus, a lower bound on r_p is given as $r_p \geq \frac{C(d_c^{avg} - h)}{K_d p}$.

3.2.3. A Centralized Channel Borrowing Algorithm

The MSC periodically sends a message to each cell x in the region requesting two parameters, namely, $d_c(x)$ and K_d. If there is any hot cell in the region, determined

by the values of $d_c(x)$'s, the MSC needs to run the channel borrowing algorithm. As an initialization step, the MSC computes the following parameters.

The parameter, H, can be computed for each cell once all the d_c's are known followed by the computation of d_c^{avg}. The value of h is computed once and for all at the MSC due to (i) the small range of variation of h, and (ii) the computation intensive nature of evaluating h. The values of the global parameters like C (fixed number of channels initially assigned to each cell), p (cell perimeter), d_c^{avg} and h and the obtained local parameter K_d are used to estimate the value of r_p (width of the threshold region) for each cell x, which is then conveyed to x by the MSC. Each hot cell B uses this parameter to compute the array, *NumDepart*, which stores the number of users departing from B and entering the neighboring cell. The value of X is computed at the MSC using the parameters d_c^{avg}, h and C.

The centralized channel borrowing algorithm, which runs at the MSC once for each hot cell, is outlined below.

Step 1: Send a message to the hot cell requesting the array *NumDepart*. Receive *NumDepart* from the hot cell.

Step 2: Select those neighboring cells of the borrower cell B as the probable lender cells, which are cold and and for which there are non-zero *NumDepart* entries. Order the candidate cells according to the decreasing values of the function $F(B, L)$ for each probable lender cell L.

Step 3: For each cell i in the listed order, continue borrowing channels until either the basic borrowing criterion is violated, or the number of borrowed channels $=$ *NumDepart*$[i]$. Lock each lended channel in the lender and its co-channel cells which are non-co-channel with B in order to avoid interference.

Step 4: Repeat Step 3 until either (i) the required number of channels are borrowed, or (ii) the list of ordered cells is exhausted. Terminate for Case (i).

Step 5: Compute the function $F(B, L)$ for all cold cells L in the compact pattern of B except those already considered in Steps 2 to 4.

Step 6: Borrow a channel from the cell L with the maximum value of $F(B, L)$. Lock the channel in L and its co-channel cells which are non-co-channel with B. Get the new values of d_c and recompute function F for each of these cells (since d_c is going to change). Repeat Step 6 until the required number of channels are borrowed.

Performance Analysis

The performance of the centralized algorithm is analyzed with the help of two metrics: the number of messages exchanged during one iteration and the running time. We assume static links between all base stations and the MSC, with a uniform message delay time of δ units. Also message exchanges between MSC and the base stations are concurrent. Let there be a total of N cells in the system. The messages and the corresponding delay times for one iteration of the centralized channel borrowing algorithm are enumerated in Table 3.1. Here X is the number of channels to be borrowed by a hot cell.

The 'Lend Channel' message in Table 3.1 is composed of three independent messages: (*i*) channel request from the MSC to the selected lender cell, (*ii*) a channel id and updated d_c value from the lender cell to the MSC, and (*iii*) the channel id conveyed to the borrower cell. Since a total of X channels are borrowed by a hot cell, each message is transmitted X times. Also each channel borrowing is accompanied by locking of the same channel in some of the co-channel cells of the lender cell. We consider the worst case scenario where the channel has to be locked in all the six cells. After X channels are borrowed, the borrower cell base station conveys the new value of its d_c (the degree of coldness) to the MSC which accounts for the last message. From Table 3.1, the total number of messages exchanged in one iteration of the centralized channel borrowing algorithm is given as,

Table 3.1: Messages during a iteration of centralized channel borrowing

When	Message	Between	Number of messages	Message delay
Initialization	request d_c, K_d	MSC→BS's	N	δ
time	send d_c, K_d	BS's→MSC	N	δ
Run time	send r_p and request *NumDepart*	MSC→hot BS	1	δ
(number of	send *NumDepart*	hot BS→MSC	1	δ
iterations =	Lend Channel			
number of hot	(i) request channel	MSC→lender BS	X	$X\delta$
cells, N_h)	(ii) channel id, d_c	lender BS→MSC	X	$X\delta$
	(iii) channel id	MSC→borrower BS	X	$X\delta$
	Co-channel locking			
	(a) request locking	MSC→Co-channel		
		cells of lender cell	6X	$X\delta$
	(b) send d_c	Co-channel cells of		
		lender cell→MSC	6X	$X\delta$
	send d_c	Borrower BS→MSC	1	δ

$M_{central} = 2N + 15X + 3,$

and the total message delay is $(5 + 5X)\delta$.

The running time of the algorithm is dominated by the delays from message exchanges. In many cases, the algorithm waits for parameter values from the base stations, which are required for further computations. These message delays are many orders of magnitude larger than the running times of individual steps of the algorithm without message passing. Hence, we can assume that the complexity of the algorithm is determined entirely by these delays, which is shown, for each message transmission, in the last column of Table 3.1. Assuming that there are N_h hot cells in the system at the time the algorithm is run (implying that all the run time messages are transmitted N_h times), the running time of the algorithm is given as,

$t_{central} = 2\delta + (5X\delta + 3\delta)N_h = 2\delta + (5X + 3)\delta N_h.$

3.2.4. A Distributed Channel Borrowing Algorithm

Each base station is capable of running the channel borrowing algorithm when its cell reaches the hot state. We assume that each cell knows the set NCC of its non-co-channel cells, the set of cells forming its compact pattern CP ($\subseteq NCC$), as well as the set CC of its co-channel cells. For the implementation of the channel borrowing algorithm, each cell needs to maintain three cell parameters – (i) its own degree of coldness, d_c, (ii) the set H_{NCC} of its hot non-co-channel cells, and (iii) the set H_{CC} of its hot co-channel cells. Whenever there is a channel allocation, blockade or release in the cell, the value of d_c is updated. We assume that every cell in the system maintains its H_{CC} independent of the running of the load balancing algorithm. This means that whenever a cell changes its state, it informs all its co-channel cells. The set H_{NCC} is computed by the hot cell B at the initialization phase of the channel borrowing algorithm. When the channel borrowing algorithm is running in B, if one of its non-co-channel cells changes state, it immediately informs B to update its H_{NCC}.

As mentioned earlier, there are three parameters – d_c^{avg}, h and C – which are used globally by all the cells. Out of them, we assume that C (fixed number of channels initially assigned to each cell) is known to all the cells. The computation of these parameters for the distributed channel borrowing algorithm is as follows:

- **Computation of d_c^{avg}:** A newly formed hot cell initiates the d_c^{avg} computation before starting the channel borrowing algorithm, in order to have the latest value of the parameter which it needs for computing X and r_p. It broadcasts a message to all other cells inquiring about their d_c values and then computes d_c^{avg} with the received information. Thus the newly formed hot cell knows the state (hot or cold) of all other cells in the system.

- **Computation of h:** The value of h is computed once and for all at the MSC due to (i) the small range of variation of h, and (ii) the computation intensive

nature of evaluating h.

Channel Borrowing Algorithm

This algorithm is run by the base station server as soon as the corresponding cell becomes hot. The activities of the probable or selected lender cells will also be mentioned as we describe the algorithm.

Initialization Steps:

1. Send messages to all other cells in the system inquiring about their d_c's. Compute d_c^{avg} and H_{NCC} from the received informations.
2. With the help of the known global parameters – C, d_c^{avg} and h – and known local parameter K_d (spatial density of mobile users in the cell), the width of the threshold region, r_p, is estimated.
3. With the help of r_p and the user classification algorithm, the array $NumDepart$ is computed.
4. The value of X is computed using the parameters C, d_c^{avg} and h.

Main Algorithm:

Step 1: Send messages to the cold neighboring cells L for which $Numdepart[L] > 0$, requesting the computed value of the function $F(B, L)$. Three pieces of information are sent to the probable lender L in the request message – (i) the set NCC (ii) the set H_{NCC} and (iii) $D(B, L) = 1$.

Cell L computes the number of its hot co-channel cells $H(B, L)$ which are non-co-channel to B by comparing the received H_{NCC} with its own H_{CC}. Then L computes the function $F(B, L)$ and sends it to B.

Step 2: The set L of neighboring cold cells in Step 1 are ordered according to decreasing values of the received $F(B, L)$ and then selected according to the listed order for channel borrowing. The selected cell computes the set of its co-channel cells which are non-co-channel with B by comparing the received set NCC with its own set CC.

Step 3: Channels are borrowed from the jth selected cell until either the basic borrowing criterion is violated, or the number of borrowed channels = $NumDepart[j]$. Upon each lending, the lender cell instructs its co-channel cells which are non-co-channel with the borrower cell (computed in Step 2) to lock the lended channel. Repeat Step 3 until either (i) the required number X of channels are borrowed, or (ii) the list of cells are exhausted. Terminate for case (i).

Step 4: Send a message to each cell L' in its compact pattern excluding the neighboring set of cells mentioned in Step 1, requesting the computed value of $F(B, L')$ if L' is in the cold state. The parameters required for this computation, namely, the cell distance $D(B, L')$ and the set H_{NCC}, and also the set NCC are conveyed in this message. Note that, only the cold cells in the compact pattern of B respond to this message.

Step 5: Select the cell, L', with $min\{F(B, L')\}$ for channel borrowing, if the basic borrowing criterion is not violated.

The selected cell L' computes the set of its co-channel cells which are non-co-channel to B by comparing the received NCC from B with its own CC. These cells are instructed to lock the borrowed channel.

Repeat Steps 4 and 5 until the required number of channels is borrowed.

In Step 4, a particular hot cell B asks for the recomputed values of the function F from all the cells although channel borrowing changes F only in the lender and some of its co-channel cells where a channel is locked. This is because, at any time, multiple

number of hot cells might be running their channel borrowing algorithm leading to the change of F in many other cells. Hence, at every step of the algorithm global information of the current status of each cold cell is required.

Performance Analysis

The performance of the algorithm is analyzed with the help of two metrics: (i) the number of messages exchanged during one iteration with one hot cell, and (ii) the running time. As in the centralized case, we assume static links between a base station and the MSC with a uniform message delay time of δ units. We consider a group of N cells under a single MSC. The BS-BS communication can only take place through their parent MSC, with a uniform message delay of 2δ. Message exchanges between pairs of base stations can be concurrent. Hence, a base station can broadcast a message to all other base stations in time 2δ. The messages and the corresponding delay times for one iteration of the distributed channel borrowing algorithm are given in Table 3.2.

Table 3.2: Messages during an iteration of distributed channel borrowing

When	Message	Between	Worst case number of messages	Worst case message delay
Initialization time	request d_c	hot BS→other BS's	N-1	2δ
	send d_c	other BS's→hot BS	N-1	2δ
Run time (run concurrently in all hot cells)	request F	hot BS→other BS's	$X(\|CP\|-1)$	$2\delta X$
	receive F	other BS's→hot BS	$X(\|CP\|-1)$	$2\delta X$
	lend channel:			
	(i) request channel	borrower BS→lender BS	X	$2\delta X$
	(ii) receive channel id	borrower BS←lender BS	X	$2\delta X$
	co-channel locking:			
	(i) request locking	lender BS→co-channel cells	6X	$2\delta X$
	(ii) acknowledgment	lender BS←co-channel cells	6X	$2\delta X$

As discussed in the previous section, a hot cell requests for computed values of

the function F from all other cells in its compact pattern (of size, $|CP|$) after each channel borrowing. Since X channels are borrowed, the number of messages sent out requesting F is $X(|CP|-1)$. Only the cold cells will respond to this message; in the worst case, all the cells in CP might be cold, implying that the worst case number of messages received is also $X(|CP|-1)$. The lend channel message comprises of a "channel request" message sent out to the selected lender and an "acknowledgment" with the lended channel id, from the borrower. Each channel lending is accompanied by locking the same channel in those co-channel cells of the lender which are non-co-channel to the borrower. Again we consider the worst case scenario where the channel has to be locked in all six co-channel cells of the lender. From Table 3.2, the number of messages exchanged in one iteration of the algorithm is given as,

$M_{distribute} = 2(N-1) + 2(|CP|+6)X.$

Note that the message complexity is a function of the compact pattern size – the larger the size of the compact pattern, the greater is the number of messages exchanged.

The running time of the algorithm is dominated by the delays from message exchanges. In many cases, the algorithm waits for parameter values from other base stations, which are required for further computations. These message delays are many orders of magnitude larger than the running times of individual steps of the algorithm without message passing. Hence, we can assume that the complexity of the algorithm is determined entirely by these delays, which is shown, for each message transmission, in the last column of Table 3.2. Each of the messages during initialization is concurrently sent or received only once. Hence the time delay is 2δ for each message type. Each of the messages for the main algorithm during run time is exchanged concurrently between multiple BS's, but X times in all, resulting in a delay of $2\delta X$ for each type of message. Hence the running time of the algorithm is given as,

$t_{distribute} = 4\delta + 12\delta X.$

3.2.5. Comparison Between Centralized and Distributed Borrowing

The distributed channel borrowing algorithm is run concurrently in all the hot cells in the system at any point of time, while the centralized scheme is run periodically at the MSC. Hence, a system running the distributed scheme is more sensitive to any load imbalance and seeks to rectify it by triggering off the load balancing algorithm immediately at the point of imbalance, namely, a hot cell. The effectiveness of the centralized scheme can be severely limited by the choice of a suitable period to run the algorithm. The period should be dynamically varied, with shorter periods ideal in case of frequent load variation and longer periods suitable for a more stable system.

However, there are some problems associated with the distributed channel borrowing scheme which is typical of the concurrent nature of execution of the algorithm. At any time, all the hot cells in the system will be running the channel borrowing algorithm. This might lead to simultaneous channel requests from multiple hot cells to the same cold cell. A way to arbitrate between these requests is to assign priority to each request and satisfy them according to the assigned priorities. The hotter the cell, the higher the priority assigned to a channel request from that cell. Another problem typical of the distributed environment as discussed in the previous section is, every time a channel is borrowed, the borrower cell needs to know the current value of the function F of all the other cells in the system (this is a global knowledge). The value of F may be changed for any cell when a channel is borrowed from that cell by another hot cell running the channel borrowing algorithm concurrently. This is different from the centralized scheme where the MSC needs to know the changed value of F of a limited number of cells, namely, the lender and its co-channel cells (this is local knowledge in some sense).

The message and time complexities of the two schemes are compared below.

- **Number of messages:** The number of messages exchanged in one iteration of the centralized channel borrowing algorithm is given by, $M_{central} = 2N+15X+3$. Note that the message complexity is independent of the compact pattern size.

The message complexity for the distributed algorithm is given by, $M_{distrbute} = 2(N-1) + 2(|CP|+6)X$.

Hence, $M_{distribute} > M_{central}$ if $|CP| > \frac{3}{2}(1 + \frac{1.67}{X})$. For $X \gg 2$, $M_{distribute} > M_{central}$ if $|CP| > \frac{3}{2}$, which is true for all practical purpose. Therefore, we conclude that the message complexity is higher in the distributed scheme. However, we will see from experiments that the ratio $\frac{M_{distribute}}{M_{central}}$ is constant and independent of N_h.

- **Speedup:** The running time for the centralized channel borrowing algorithm is $t_{central} = 2\delta + (5X+3)\delta N_h$, where N_h is the number of hot cells in the system. The centralized algorithm acts sequentially for each hot cell of the system. The distributed algorithm has time complexity $t_{distribute} = 4\delta + 12\delta X$. Hence, the speedup of the distributed scheme over the centralized is given as,

$$S_p = \frac{t_{central}}{t_{distribute}} = \frac{\{2 + (3+5X)N_h\}\delta}{(4+12X)\delta} = \frac{2 + (3+5X)N_h}{4+12X} \tag{3.2}$$

Under the assumption that $X \gg 1$, the above expression can be approximated to, $S_p \approx \frac{5}{12}N_h$. Thus in our system model with N_h hot cells, a speedup of about $0.5N_h$ of the distributed scheme over the centralized scheme is expected. We will verify this later from our simulation experiments.

3.2.6. Channel Assignment Strategy

The way to assign channels to the users in a cell is now described. The set of available channels in a hot cell can be divided into two classes: channels local to the cell, and borrowed channels. Clearly, cold cells contain only local channels.

The channel demands arising in a hot cell can be divided into four priority classes which are enumerated below in the order of decreasing priority. Also described are the types of users who generate such channel requests or demands. The proposed channel assignment algorithm uses these demand classes to prioritize channel requests.

Channel Demand Classes

- **Class 1 demands:** These are the channel requests generated by the users crossing over from the neighboring cells and are also called *hand-off* requests. To make sure that an ongoing call is not disrupted, this class gets the highest priority for channel assignment.

- **Class 2 demands:** These are channel requests made by the originating calls. Demands of this class gets the next higher priority after Class 1 demands.

- **Class 3 demands:** These are *Type 1 channel reassignment* requests which are not generated by a mobile user, but generated internally by a base station function which continuously monitors the state of channel assignments to the users in the cell. A 'new' or 'others' type of user communicating through a borrowed channel is reassigned a local channel by the base station if the local channel is not used to satisfy a Class 1 or Class 2 demand.

- **Class 4 demands:** These are *Type 2 channel reassignment* requests which are also internally generated. A 'departing' user communicating through a local channel is reassigned a borrowed channel by the base station, if the borrowed channel is not used to satisfy a Class 1 or Class 2 demand.

In a cold cell x, channel demands are of Classes 1 and 2 only, because there is no concept of borrowed channels. However, there will be cases when a hand-off user to cell x from a neighboring hot cell, communicating through a channel borrowed from x, will be assigned the same channel, thereby releasing channel locking in other co-channel cells. This is nothing but a hand-off scenario where the incoming user is assigned the same frequency channel that he was using in the previous cell.

Channel Assignment Algorithm

At any instant, there can be more than one simultaneous channel requests to the base station. Each such request must fall in one of the above four demand classes for a hot cell, or in one of the first two classes for a cold cell. The channel requests are prioritized according to the class they belong to. In case of multiple requests of the same class, they are selected in any random order by the channel assignment algorithm, which is shown as a flowchart in Figure 3.2.

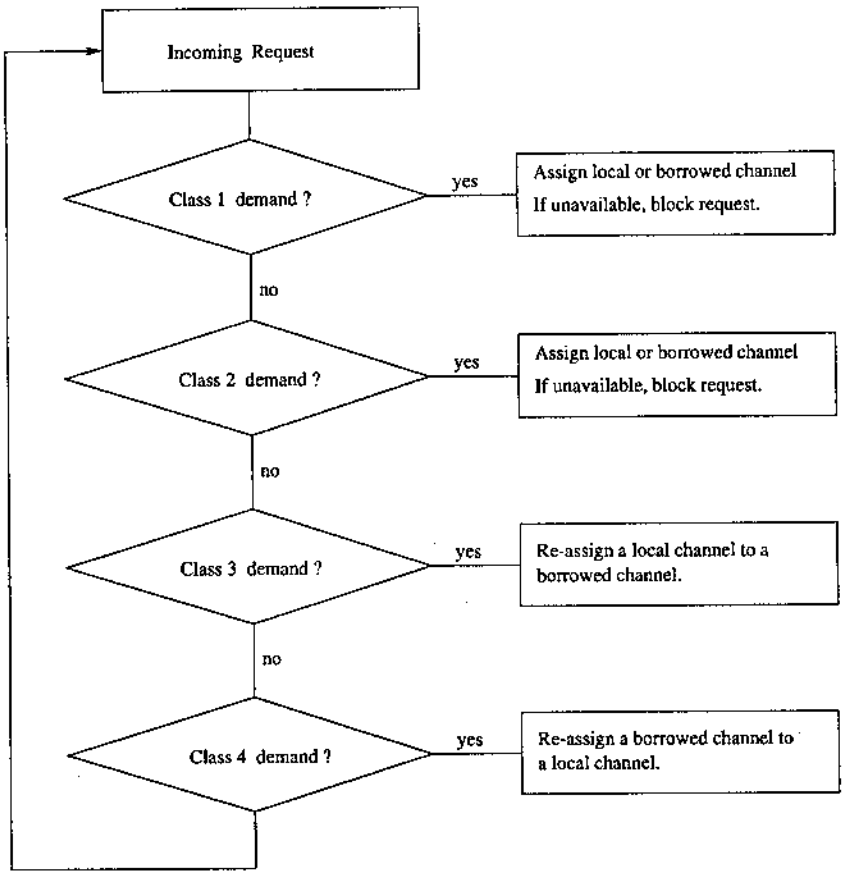


Figure 3.2: Channel allocation algorithm for the users in a cell

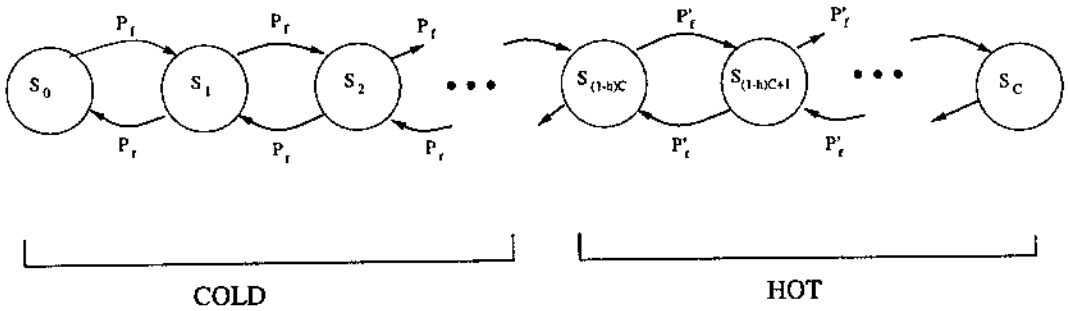


Figure 3.3: Markov model for a cell

3.3. Markov Model of a Cell

In this section, we derive a discrete Markov model of a cell x in the system. This cell is in state S_i, if i is the number of occupied channels in the channel set of that cell. By channel occupation, we mean that either the channel is being used for a local call in the same cell or it is borrowed by another hot cell. The Markov chain model is shown in Figure 3.3.

If the cell is in any one of the states S_i, for $i < \lceil (1-h)C \rceil$, then it is a cold cell. From now on, wherever we use $(1-h)C$, it is interpreted as $\lceil (1-h)C \rceil$ and hC as $\lfloor hC \rfloor$. When the cell enters the state $S_{(1-h)C}$ from $S_{(1-h)C-1}$, it becomes a hot cell from a cold one. The cell remains in the hot state as long as it is in one of the states S_i, where $i \geq (1-h)C$. The state transition probabilities are different for the system depending on whether it is in the hot state or in cold state. For a cell in cold (respectively, hot) state, the forward transition probability is p_f (respectively, p'_f) and the reverse transition probability is p_r (respectively, p'_r).

In our channel assignment algorithm within each cell, a channel request in a hot cell can be classified as one of the four classes and a request in a cold cell as one of the first two classes. In deriving our Markov model, we assume that a Class i (for $1 \leq i \leq 4$) channel demand is a Poisson process with parameter λ_i. The aggregate channel demand process in a hot cell is a superposition of these four Poisson processes,

and hence, by an well-known queuing theoretic result [NEL96], it itself is a Poisson process with rate $\sum_{i=1}^{4} \lambda_i$. Similarly, the aggregate channel demand process in a cold cell is the superposition of only Class 1 and Class 2 channel demand processes, and is itself a Poisson process with parameter $\sum_{i=1}^{2} \lambda_i$.

Apart from the local channel requests in each cell, we need to model the global channel requests arising when a hot cell requests for channel borrowing from a cold cell. It has been shown in [JR94] that modeling the channel borrow demands from other cells as a Poisson process leads to good analytical results. Hence, in our analytical model, the channel borrow demand is also modeled as a Poisson process with parameter λ'. Since, in our load balancing scheme, a cold cell lends one channel at a time, this channel lending process suitably models the resource migration phase of our load balancing strategy. With these assumptions, let us now compute the transition probabilities of the Markov model of a cell.

Let p_f define the probability that the cold cell goes from state S_i to S_{i+1}, where $0 \leq i \leq (1-h)C - 1$. This can occur in any of the two ways:

Case(1): There is a new channel request in the cell.

Case(2): A channel is lended on demand from another hot cell.

A new channel request in a cold cell is equivalent to a Class 1 type channel demand with probability of arrival λ_1 or a Class 2 channel demand with probability λ_2. Moreover a Class 2 demand is satisfied only when there is no Class 1 demand. The probability of a channel borrow demand from a remote hot cell is λ', and this is satisfied only when there are no Class 1 or Class 2 demands. Considering these facts, the forward transition probability for a cell in the cold state is given as

$$p_f = \lambda_1 + (1 - \lambda_1)\lambda_2 + (1 - \lambda_1 - \lambda_2)\lambda' \tag{3.3}$$

Let the call holding time in a cell be an exponentially distributed random variable with mean $\frac{1}{\mu}$. The reverse transition probability, p_r, for a cold cell, is the probability

of a local call terminating or a lended channel release. The probability of a lended channel release is equal to the probability that a call terminates on lended channel. Since the state of the cell is cold, a channel borrow demand will always be satisfied. Hence, the probability of lending a channel is equal to the probability of a channel borrow demand from a hot cell. This gives,

$$\begin{aligned} p_r &= \text{prob(local call termination)} + \text{prob(lended channel release)} \\ &= \text{prob(local call termination)} + \\ &\quad \text{prob(ch. lending)}\cdot\text{prob(call termination on lended channel)} \\ &= \mu + (1-\mu)\lambda'\mu \end{aligned}$$

We do not distinguish between an ongoing call in local or borrowed channel as far as call termination is concerned.

Next we compute the forward transition probability p'_f for a cell in the hot state. Since borrowing is not allowed from a hot cell, this can be triggered by any one of Class 1, Class 2 or Class 3 channel demands. The channel is assigned according to the priority classes. Hence

$$p'_f = \lambda_1 + (1-\lambda_1)\lambda_2 + (1-\lambda_1-\lambda_2)\lambda_3 \tag{3.4}$$

A reverse transition by a cell in the hot state will occur when there is a local call termination or a Type 2 channel reassignment. The probability of a Type 2 channel reassignment (Class 4 demand) is λ_4. Hence, the reverse transition probability p'_r is given as

$$p'_r = \mu + (1-\mu)\lambda_4 \tag{3.5}$$

Let P_i be the limiting (or steady state) probability of the state S_i. Let $l = \frac{p_f}{p_r}$ and $l' = \frac{p'_f}{p'_r}$. Solving the state equations for this birth and death Markov chain, we have

$$P_i = \begin{cases} l^i P_0 & : \quad 0 \le i \le (1-h)C \\ l^{(1-h)C} l'^{(i-(1-h)C)} P_0 & : \quad (1-h)C < i \le C \end{cases}$$

where,

$$P_0 = \frac{1}{\frac{1-l^{(1-h)C+1}}{1-l} + l^{(1-h)C} l' \{\frac{1-l'^{hC}}{1-l'}\}}.$$

The probability of a cell being hot is

$$p_h = \sum_{i=(1-h)C}^{C} P_i \tag{3.6}$$

$$= l^{(1-h)C}\{\frac{1-l'^{(hC+1)}}{1-l'}\}P_0 \tag{3.7}$$

Also the call blocking probability in a cell is given by the limiting probability of the state S_C, where all the channels in the cell are occupied.
Hence, the call blocking probability is given by

$$P_{block} = l^{(1-h)C} l'^{hC} P_0 \tag{3.8}$$

3.3.1. Estimation of Threshold, h

The probability of a cell being hot, p_h, is given as

$$p_h = \frac{l^{(1-h)C}\frac{1-l'^{(hC+1)}}{1-l'}}{\frac{1-l^{(1-h)C+1}}{1-l} + l^{(1-h)C} l' \frac{1-l'^{hC}}{1-l'}} \tag{3.9}$$

The value of h (the threshold for the degree of coldness of a cell below which it will be hot) is one of the factors determining the above probability, p_h. Other parameters determining h are λ_i's, $1 \leq i \leq 4$, λ', μ and C. Given these parameters, the problem is to estimate the value of h such that any channel borrowing will lead to channel blockade in not more than b $(0 < b < 6)$ hot cells with a very high probability, p. Let us describe a two step method to estimate the parameter h. In the first step, we describe a procedure to estimate p_h. In the second step, using this value of p_h and the other given parameters, Equation (3.9) is solved for the value of h.

We estimate the value of p_h such that any channel borrowing will lead to channel blockade in no more than b $(0 < b < 6)$ hot cells with probability p. The number

of hot cells among the co-channel cells of a cold lender cell will follow the binomial distribution with probability p_h. Each set of co-channel cells of a particular cell consists of six cells. Hence, the probability that i cells among the six co-channel cells are hot, is given by $\binom{6}{i}p_h^i(1-p_h)^{6-i}$. For any channel borrowing scheme to have a channel-locking with probability p in less than b hot cells, $0 < b < 6$, the following equation must be satisfied:

$$\sum_{i=0}^{b}\binom{6}{i}p_h^i(1-p_h)^{6-i} = p \tag{3.10}$$

A solution to Equation (3.10) gives an estimate for the probability, p_h, of a cell being hot.

3.3.2. Experimental Results

For given values of the parameters $\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda', \mu$ and C, the values of p_h are estimated for b varying from 1 to 5 and with $p = 0.98$. Then the value of h is estimated by solving the non-linear Equation (3.9) for each estimate of p_h. The results are shown in Table 3.3 for two sets of values for λ_1 and λ_2, with $\lambda_3 = \lambda_4 = 0.05, \mu = 0.7, \lambda' = 0.3$ and $C = 20$.

Table 3.3: Estimated h for various b, λ_1, λ_2

b	$\lambda_1 = 0.5, \lambda_2 = 0.4$		$\lambda_1 = 0.6, \lambda_2 = 0.3$	
	p_h	h	p_h	h
1	0.04	0.00	0.04	0.00
2	0.10	0.15	0.10	0.09
3	0.20	0.28	0.20	0.20
4	0.34	0.44	0.34	0.34
5	0.52	0.60	0.52	0.51

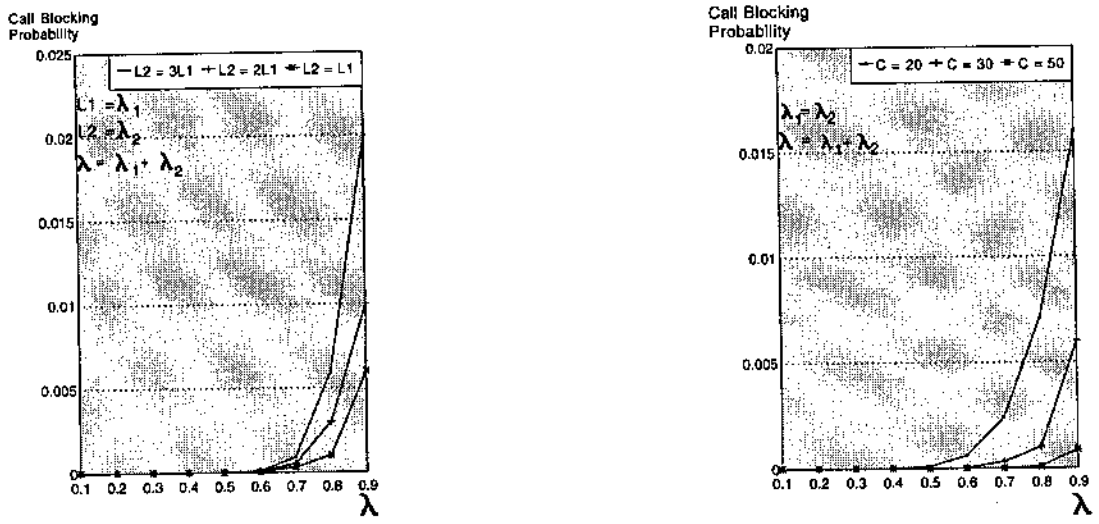


Figure 3.4: Variation of call blocking probability with λ and C

From Table 3.3, it is seen that as b increases, the estimated value of the probability of a cell being hot, p_h, increases. This is expected because only a corresponding increase in the probability of a cell becoming hot will lead to an increase in the number of its hot co-channel cells. The third and fifth columns of Table 3.3 show the estimated values of h solving the non-linear Equation (3.9), from where it can be seen that the values of the threshold, h, where a cell becomes hot increases almost proportionately to the value of p_h. If the value of h increases, we can declare a cell hot with less number of channels blocked than it was before. This implies that the probability of a cell becoming hot, p_h, increases with increasing h, and serves as a sanity check for our estimate of h.

The call blocking probability, P_{block}, from our analytical model is shown as a function of call arrival rate λ in Figure 3.4. Figure 3.4(a) depicts the variation for various ratios of the arrival rate λ_1 of Class 1 (hand-off) demands to the arrival rate λ_2 of Class 2 (originating call) demands. Here call blocking implies blocking of both originating and hand-off requests. As expected, the call blocking probability increases with an increase in the call arrival rate in all cases. It is also observed that

as the ratio of λ_2 to λ_1 (or vice versa) increases from one (we considered the cases $\lambda_2 = 2\lambda_1$ and $\lambda_2 = 3\lambda_1$), the call blocking probability also increases. This leads to a very interesting conclusion: *the call blocking probability is minimum when the distributions of originating and hand-off calls tend to be equal and increases with the amount of imbalance between them.*

Figure 3.4(b) depicts the variation of call blocking probability with the number of channels C initially allocated to each cell. As C is increased, the call blocking probability reduces for a given call arrival rate.

3.4. Simulation Experiments

A sequential simulation for the proposed channel-borrowing and channel-assignment algorithms is implemented. The problem domain naturally lends itself to parallel simulation or simulation using multiple threads since there are a lot of concurrency and global resource management issues in the system. However, we have implemented a sequential simulation algorithm since this would suffice for us to test our algorithms and would also simplify the design of the simulator.

3.4.1. Simulation Parameters

Modeling Received Signal Strength (RSS): In order to classify the users correctly we need to model the role of the signal strength received by the base station from the user. Since actual signal strengths cannot be generated, we overlay on each cell a grid of size 100×100. A user position within a cell is given by a pair of co-ordinates (x, y) in this grid. When we update a user's position within a cell, we change its co-ordinates. A user is modeled as *new*, *departing* or *others* according to the user-classification algorithm. A fixed set of co-ordinates defines the peripheral shaded region of a cell shown in Figure 3.1.

Modeling User Mobility: We model user mobility pattern as a simple random walk on a grid. This implies that a user at a particular co-ordinate (x, y) in a cell has equal probability of moving to one of the four neighboring co-ordinates $(x-1, y), (x+1, y), (x, y-1)$ or $(x, y+1)$.

Call Origination and Termination: Call arrival in a cell is programmed as a Poisson process with inter-arrival time exponentially distributed with mean $\frac{1}{\lambda}$. The call holding time is programmed as an exponentially distributed random variable with mean $\frac{1}{\mu}$.

3.4.2. Performance Results

The main metric used to evaluate the performance of our load balancing algorithm and compare it with other existing schemes is the call blocking probability. The impact of varying various system parameters like (i) the threshold h, (ii) the number of channels C initially allocated to each cell, and (iii) the size of the compact pattern, on the performance of our load balancing scheme are observed to determine the stability of our algorithm. The results from our experiments are presented below. In carrying out these experiments, we used the distributed channel borrowing algorithm. In terms of functionality, both the centralized and distributed algorithms are the same and the particular algorithm used does not affect the results.

Impact of threshold, h, on the call blocking probability, P_{block}

This experiment used the number of channels C = 100 per cell and a total of N = 100 cells in the system. As expected and also as observed from the results of our analytical model, the call blocking probability increases with the call arrival rate. The similarity in the trend of variations of the call blocking probability with λ from the simulation experiments (Figure 3.5), with that from the analytical results (see Figure 3.4) validates our analytical model.

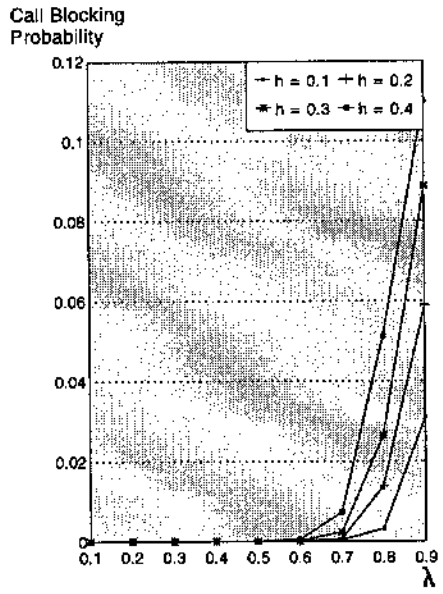


Figure 3.5: Call blocking probability vs arrival rate for various h

It is also observed that the call blocking probability increases with increasing h for heavy traffic load, λ. Increasing the threshold h implies that we are accommodating more hot cells in the system, which thereby reduces the number of cold cells from which channels can be borrowed. This decrease in channel borrowing provision increases the probability of call blockade in the existing hot cells under heavy traffic load.

Impact of the number of fixed channels, C, on call blocking probability

Figure 3.6 shows the variations of the call blocking probability with λ for three values of C, the number of channels initially allocated to each cell under the fixed assignment scheme. It is observed that for small value of C (say $C = 20$), the blocking probability is as high as 0.22 for $\lambda = 0.9$. As C is increased, the blocking probability decreases under heavy traffic load (i.e., $\lambda > 0.5$). With $C = 40$, the blocking probability is as low as 0.03 for the same value of λ. We again observe a similar trend of variation in

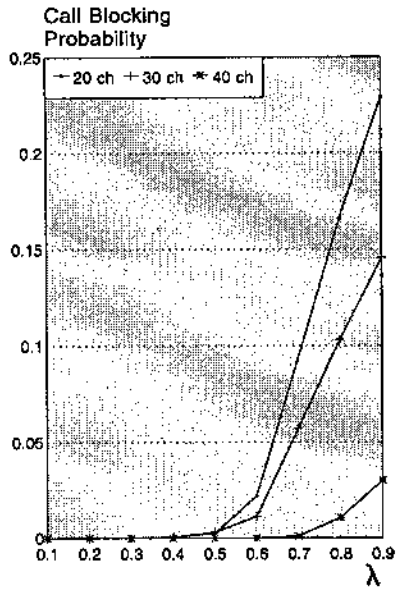


Figure 3.6: Call blocking probability vs arrival rate for various C

our performance metric from simulation with that from the analytical model.

Impact of the size of the compact pattern on the call blocking probability

A change in the shift parameters affects the compact pattern size. We ran the simulation for different tuples of shift parameters, i and j, where $i = j$. This would also test if the algorithm behaved differently when its co-channel cells were different. This is important to know because channels are locked in the co-channel cells when borrowing. Figure 3.7 shows the experimental results. The experiment was conducted with $h = 0.2$, and $N = 1000$ cells in the system. The values of i (or j) are shown along the horizontal axis. We observe that the call blocking probability decreases with an increase in the shift parameters because in that case, the compact pattern size increases, which in turn implies that the hot cells will find more cold cells to borrow channels from.

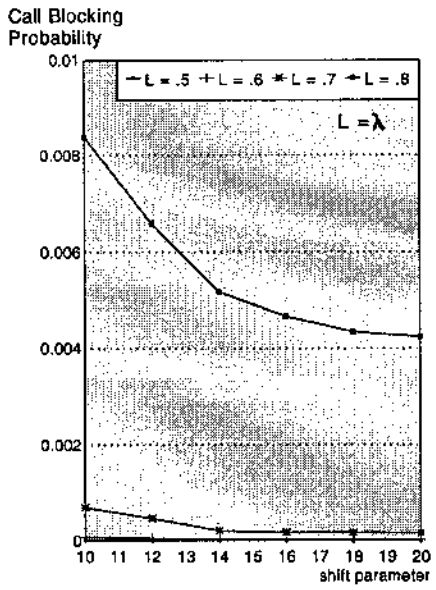
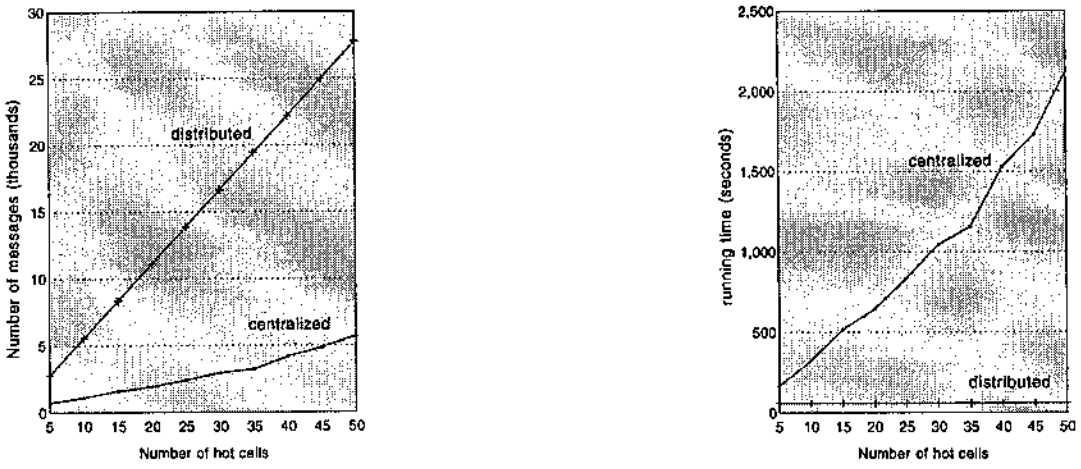


Figure 3.7: Call blocking probability vs compact pattern sizes for various λ

3.4.3. Comparison of Centralized and Distributed Schemes from Simulation

Figure 3.8 compares the performance of the centralized and distributed schemes with increasing number of hot cells. The comparison metrics are the number of messages (M) exchanged and the running times (t) in sec. While computing the running times, we assumed the message delay $\delta = 1$ sec. The values of M and t obtained from the experiments are also shown in Table 3.4. We define a new metric called the message ratio as $M_r = \frac{M_{central}}{M_{distributed}}$, analogous to the speedup defined as $S_p = \frac{t_{central}}{t_{distributed}}$.

Figure 3.8(b) shows that, as the number of hot cells (N_h) is increased, the running time of the distributed scheme remains approximately constant. The algorithm works concurrently in all the hot cells and since the total message delay (which dominates the running time) is independent of the number of hot cells this result is expected. The running time of the centralized scheme, on the other hand, increases with a progressively higher rate as N_h is increased. A speedup of over $0.4N_h$ is observed in all cases, as shown in Table 3.4. We estimated a speedup of around $0.4N_h$ from the



(a) (b)

Figure 3.8: Message Complexity and Running Times of Centralized and Distributed Schemes

analytical model of our system. From the results in Table 3.4, the speedup increases from $0.56N_h$ to $0.72N_h$ as the number of hot cells (N_h) is increased from 5 to 50.

In case of the number of messages, we observe that the message ratio, M_r, hovers around 0.20 with a slight decrease to 0.17 as N_h is varied from 5 to 50. This implies that there are approximately five times as many messages exchanged in the distributed scheme as in the centralized scheme, and this remains more or less constant with varying number of hot cells. This also validates the results from our analytical model. The above analysis concludes that in a region with large number of hot cells, the distributed scheme will be more suitable to use.

3.5. Comparison of LBSB scheme with Directed Retry and CBWL

The main disadvantage of the directed retry (with load balancing) scheme [KE89] is that it shares load between two cells depending on the number of users in the overlap region. If the number of such users is few, proper load balancing is not achieved. In our LBSB (load balancing with selective borrowing) scheme, on the other hand,

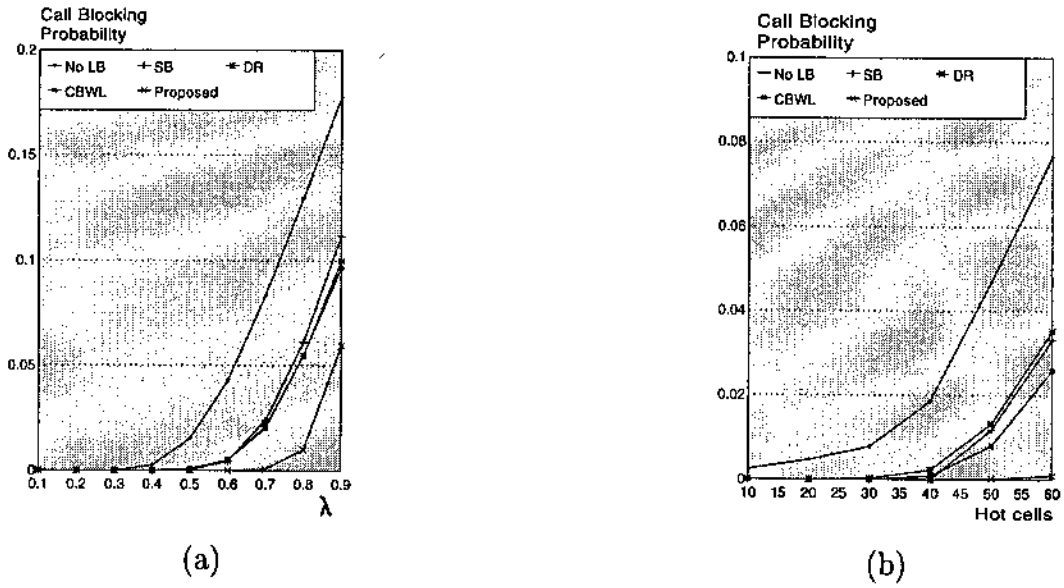
Table 3.4: Speed up (S_p)and message ratio (M_r) for the distributed scheme with varying N_h (# of hot cells) from simulation

N_h	$t_{central}$	$t_{distributed}$	$\mathbf{S_p}$	$M_{central}$	$M_{distributed}$	$\mathbf{M_r}$
5	163.3	58.3	2.8	705	2775	0.25
10	323.8	58.7	5.5	1110	5550	0.20
15	516.9	58.8	8.8	1596	8325	0.19
20	644.4	58.6	10.9	1920	11100	0.17
25	835.8	58.5	14.2	2406	13875	0.17
30	1045.0	58.8	17.7	2945	16650	0.17
35	1158.3	58.7	19.7	3230	19425	0.16
40	1527.7	58.7	26.0	4174	22200	0.18
45	1749.1	58.3	30.0	4845	24975	0.19
50	2125.6	58.6	36.2	5691	27750	0.20

a fixed number of channels is always transferred between multiple underloaded cells and an overloaded one. This achieves almost perfect load balancing, because not only the overloaded cell gets the necessary number of channels, but also the increase in load (in the form of decreasing number of channels) is shared evenly by multiple underloaded cells. Another problem with the directed retry scheme is that channels may be shared with a neighboring cell which is hot. Since channel borrowing from a hot cell is not allowed in our scheme, this problem will not occur.

The channel borrowing without locking (CBWL) scheme [JR94] performs poorly for certain clusters of hot cells. Consider, for example, a cluster where a hot cell has six hot neighbors. Then, in case the channel sets of these six cells get exhausted, the inner hot cell is going to starve as channels are allowed to be borrowed only from the neighboring cells. Our LBSB scheme performs equally well for all types of hot cell

distribution because channels are allowed to be borrowed from any suitable cold cell in the compact pattern. Another drawback of CBWL is its limited scope of usage of the borrowed channels (due to low power transmission). This problem is also absent in our scheme.



(a) (b)

Figure 3.9: Comparison of our scheme with others

Figure 3.9 compares the performance of our scheme with the fixed assignment (no load balancing), simple borrowing (SB), directed retry (DR) and CBWL strategies, using the call blocking probability as the metric. The total number of cells in the system is $N = 100$ in all cases. The first set of graphs in Figure 3.9(a) shows the variation of call blocking probability (P_{block}) with λ with $N_h = 40$ hot cells. It is observed that for moderate call arrival rate ($\lambda = 0.5$), our scheme performs the best followed by simple borrowing, directed retry, CBWL and no load balancing (fixed assignment) schemes in that order as shown in Table 3.5. As λ increases, CBWL outperforms both directed retry and simple borrowing but the blocking probability is still the least for our proposed load balancing scheme. In fact, our proposed scheme outperforms all the other schemes under heavy traffic load ($\lambda \geq 0.5$), while the performance of all schemes are neck to neck under moderate and low loads ($\lambda < 0.5$).

Table 3.5: A comparison of blocking probabilities for various channel assignment schemes with and without load balancing

Schemes for Channel Assignment	$\lambda = 0.5$	$\lambda = 0.9$	Number of hot cells, $N_h = 40$	Number of hot cells, $N_h = 60$
Fixed Assignment (No load balancing)	0.015304 (5)	0.177662 (5)	0.018584 (5)	0.076577 (5)
Simple Borrowing (SB)	0.000229 (2)	0.111761 (4)	0.0 (1)	0.033140 (3)
Directed Retry (DR)	0.000485 (3)	0.100101 (3)	0.002266 (4)	0.035151 (4)
Channel Borrowing Without Locking (CBWL)	0.000986 (4)	0.096539 (2)	0.000889 (3)	0.025806 (2)
Proposed scheme (LBSB)	0.000012 (1)	0.059354 (1)	0.0 (1)	0.000965 (1)

Table 3.5 shows the actual data for $\lambda = 0.5$ and 0.9. A number in the parentheses gives the rank of that scheme in terms of achieving a low call blocking probability.

In the second set of graphs in Figure 3.9(b), the call blocking probabilities for each scheme are plotted against the number of hot cells in the system. It is observed that with $N_h = 40$ hot cells in the system, our scheme and the simple borrowing perform equally well, followed by CBWL, directed retry and fixed assignment in that order as shown in Table 3.5. As $N_h > 40$, the CBWL scheme outperforms simple borrowing which continues to perform better than directed retry. The distributed LBSB scheme outperforms all others in all cases. Also as the number of hot cells in the system increases, the proposed scheme shows the least variation among all other schemes in terms of call blocking probability (Figure 3.9(b)). These results show the efficacy of our load balancing strategy to mitigate the load imbalances in the system and to improve the system performance.

3.6. Summary

In this chapter, we have proposed dynamic load balancing strategies for the hot cell problem in cellular mobile environment, which can be implemented in a centralized or distributed manner. These strategies constitute two main parts. In the first part, a channel borrowing strategy is proposed where channels are borrowed by a hot cell from suitable cold cells. The suitability of a cold cell as a lender is determined by an optimization function constituting three cell parameters – coldness, nearness and hot cell channel blockade. In the second part, a channel assignment strategy is proposed where the assignment is done on the basis of different priority classes in which the user demands are classified. The relative merits and demerits of the centralized and distributed schemes are discussed both quantitatively and qualitatively.

A Markov model for an individual cell is also proposed and expressions for the probability of a cell being hot and the call blocking probability in a hot cell are derived. One of the important parameters of our model is the threshold h, below which a cell is classified as hot. A method to estimate the value of h is described and the variations of call blocking probabilities with the call arrival rate are noted from the analytical model, which are compared later with similar results from simulation experiments. Exhaustive simulation is also carried out to compare the centralized and distributed schemes. From these results, we conclude that in a region with a large number of hot cells, the distributed scheme performs better. Simulation experiments showed a significant improvement of our scheme in system performance as compared to the fixed channel assignment, simple borrowing and two existing load balancing strategies like directed retry and CBWL.

CHAPTER 4

STRUCTURED LOAD BALANCING FOR A HOT REGION

One of the disadvantages of the schemes proposed in Chapter 3 is the large number of message exchanges among base stations or between base stations and MSC's during a run of the load balancing algorithm. Also, we do not have a definite bound on the number of cells in which the borrowed channel has to be locked. Our objective in this chapter is to propose another load balancing scheme based on structured channel borrowing mechanism between adjacent cells, which is not as computation (message) intensive as our previous schemes, yet achieves almost perfect load balancing and eliminates the drawbacks of both the directed retry and the CBWL schemes, i.e. performs equally well under all types of traffic conditions and load distribution in the hot cells of a region.

The proposed dynamic load balancing scheme employs channel borrowing in order to cope up with the problem of teletraffic overloads in a region of adjacent hot cells called a *hot spot.* Designating a particular cell within the hot spot as the *center cell,* hot spot can be conceived as a stack of hexagonal 'Rings' around the center cell, where each 'Ring' consists of at least one hot cell among others like cold safe, cold unsafe and cold semi-safe cells. A hot spot will be called *complete* if all the cells within it are hot. Otherwise it is *incomplete.* In our load balancing approach, a hot cell in 'Ring i' borrows channels from its adjacent cells in 'Ring i+1' to ease out its high channel demand. This structured lending mechanism decreases excessive co-channel interference and borrowing conflicts, which are generally prevented through channel locking in other schemes. Also the number of channels to be borrowed by each cell will be predetermined by its class and its position within the hot spot. By using a simple and efficient construction of a *demand graph,* unused channels are migrated from the cold cells within or in the periphery of the hot spot to the hot cells constituting the

hot spot.

Assuming a fixed channel assignment scheme is available to start with, we have proposed a channel migration scheme through borrowing between cells of adjacent rings such that all the hot cells are provided with the required number of channels to cope up with their teletraffic overloads. A discrete Markov model for a cell in our system incorporating load balancing is also proposed and another similar model is developed to capture the evolution of the hot spot region. Extensive simulation experiments are carried out to evaluate the performance of our scheme.

The rest of this chapter is organized as follows. The classification of cells and regions are described in Section 4.1. The structured load balancing schemes for complete and incomplete hot spots are laid down in Section 4.2. In Section 4.3, the scheme is analyzed using a Markov model and detailed simulation experiments are described in Section 4.4.

4.1. Classification of Cells and Regions

As discussed in Chapter 3, a cell is classified as hot or cold according to its *degree of coldness* defined as

$$d_c = \frac{number\ of\ available\ channels}{C} \tag{4.1}$$

where C is the fixed number of channels allocated to that cell.

If $d_c \leq h$ where $h > 0$ is a fixed *threshold* parameter, the particular cell is *hot*, otherwise it is *cold*. Typical values of h are 0.2, 0.25 etc., and determined by the average call arrival and termination rates and also by channel borrowing rates from other cells. The usefulness of the parameter h is to keep a subset of channels available so that even when a cell reaches the hot state, an originating call need not necessarily be blocked. With these distinctions between hot and cold cells, let us now define a hot spot region.

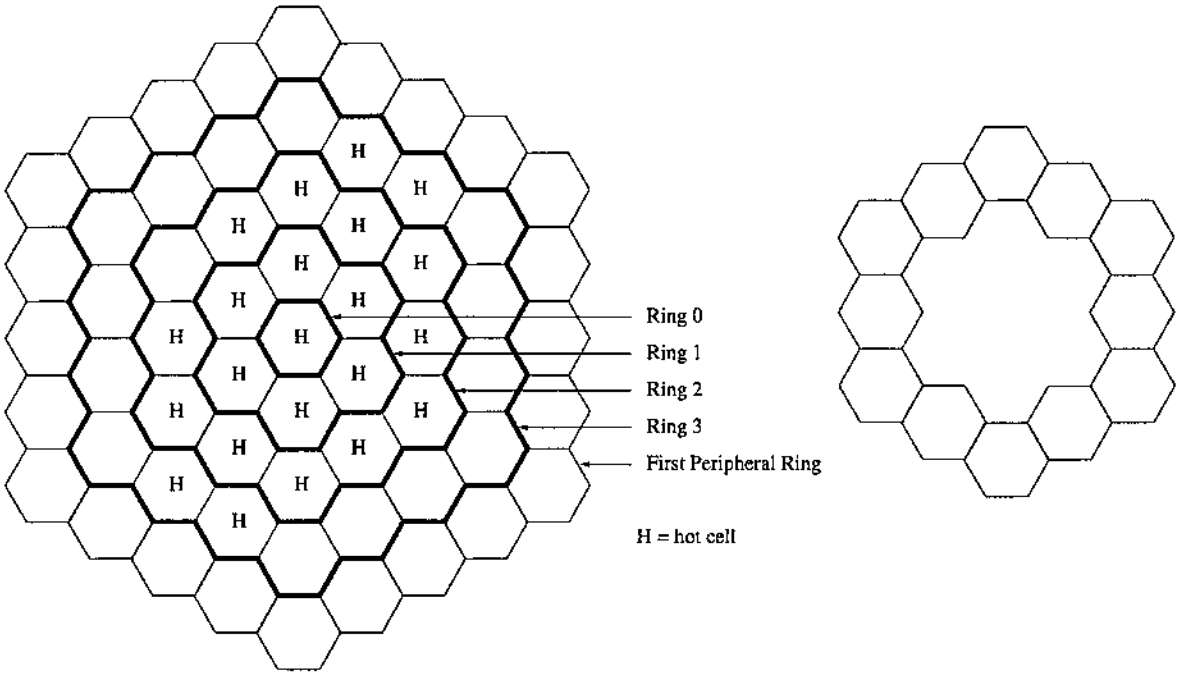


Figure 4.1: A hot spot region shown as a collection of hexagonal rings

4.1.1. Definition of a Hot Spot and Ring

Two cells are said to be *adjacent* if they have a common edge. For example, a hexagonal cell has six adjacent cells. A set S of hot cells (marked as H) is said to form a *hot spot* if any cell in S is adjacent to at least another cell in S. Figure 4.1 depicts an example of a hot spot in cellular mobile environment.

Let us now introduce the concept of a 'Ring'. We first select a *center cell* for the hot spot. A preferable way to make the selection is to compute the diameter, d_{hs}, of the hot spot which can be defined as the maximum cell distance between two cells in the hot spot. The cell lying at a distance of $\lceil \frac{d_{hs}}{2} \rceil$ from either end of the two farthest cells yielding the diameter, is selected as the center cell.

Now consider the center cell and the hexagonal rings of cells around it. If the center cell itself is hot, we call it 'Ring 0'. Otherwise the ring of cells nearest to the center cell containing at least one hot cell is denoted as 'Ring 0'. We define 'Ring i'

($i > 0$) as the ring of cells containing at least one hot cell, at a cell distance i from 'Ring 0' and further away from the center cell. Note that, if a hot spot consists of n 'Rings', then all the 'Rings' have to be contiguous by definition.

In the example of Figure 4.1, there are four such 'Rings' – numbered as 0, 1, 2 and 3 – consisting of 1, 6, 11 and 4 hot cells, respectively. The first ring of cells outside the hot spot containing all cold cells is called the 'First Peripheral Ring'. The next such ring is the 'Second Peripheral Ring', and so on. So we distinguish between two types of hexagonal structures, one type containing at least one hot cell called 'Rings' and the other type containing no hot cell called 'Peripheral Rings'. While referring to both of them, we use the generic term *ring*. Henceforth, the center cell will be called ring 0 and a 'Ring' (or a 'Peripheral Ring') at a cell distance i from the center will be called ring i.

For hexagonal geometry, the number of cells in ring i is 1 for $i = 0$ and $6i$, otherwise. Let H_n denote a hot spot whose outermost 'Ring' is ring n. Therefore, H_0 is equivalent to a single hot cell and the total number of cells in H_n is given by

$$N_n = 3n(n+1) + 1. \tag{4.2}$$

A hot spot, H_n, is said to be *complete*, if it contains N_n hot cells. Otherwise it is defined as *incomplete*. For a complete hot spot, 'Ring 0' always happens to be the center cell. Also, the number of cells in the 'First Peripheral Ring' of a complete hot spot H_n is $6(n+1)$.

4.2. A Structured Load Balancing Scheme

The underlying idea behind this scheme is the migration of channels between cells through channel borrowing mechanism, in order to satisfy the channel demand of overloaded (hot) cells. Channel migration takes place between a borrower and a lender cell. Unlike our previous schemes, the borrower (hot) cell does not have the opportunity to select its lender from among all the cold cells in its compact pattern. A

hot cell in 'Ring i' of a complete hot spot can borrow channels only from its adjacent cells in 'Ring i+1' or the 'First Peripheral Ring' if 'Ring i' is the outermost 'Ring' of the hot spot. This structured borrowing mechanism reduces the amount of co-channel interference between the borrower cell and the co-channel cells of the lender using the borrowed channel. Thus, in at most two cells the borrowed channel needs to be locked. Also a certain fixed number, X, of channels is needed by each hot cell to relieve the excess traffic demand. The estimation for X is similar to the method used in Chapter 3.

A hot cell in 'Ring i' should borrow sufficient number of channels so that not only can it satisfy its own requirement X, but also cater for the channel demand from adjacent hot cells in 'Ring i-1'. In this way, each 'Ring' of cells caters for the channel demand of overloaded cells in the next inner 'Ring' as well as within itself, borrowing channels from the immediate outer 'Ring'. The hot cells in the last 'Ring' of the hot spot borrow channels from the cold cells in the 'First Peripheral Ring'. Here we make the following assumptions:

- The base station transmitter of each cell has the capability of transmitting any of the frequencies of the available spectrum. A channel borrow implies locking the same in the lender cell transmitter and unlocking it in the borrower cell transmitter.

- Due to the structured borrowing mechanism used in our load balancing scheme, the borrowed channel needs to be locked in at the most two cells. This will not significantly affect the system performance even under heavy load. To avoid channel locking, a borrowed channel may be used under reduced transmission power as in CBWL [JR94].

- Only local channels of a cell are lended on demand to adjacent cells in the next inner ring. After borrowing channels from adjacent cells, a ring i cell reassigns the borrowed channels (by intra-cellular handoff) to some of the users to release

sufficient number of local channels and meets the demands of the ring i-1 cells.

- All cells in the 'First Peripheral Ring' are able to provide the required number of channels to the hot spot without exhausting their channel set or becoming hot themselves. If this is not true, channels are borrowed from multiple 'Peripheral Rings' and the algorithm adopted is the same as in the more general case of an incomplete hot spot.

For the general case of an incomplete hot spot (having cold cells along with hot ones), a cold cell is further classified as *cold safe, cold unsafe* and *cold semi-safe.* Cold unsafe, cold semi-safe or hot cells in ring i need to borrow channels from adjacent cells in ring i+1, the number of channels borrowed being different for different classes. A cold safe cell does not need any channel borrowing. A *demand graph* is constructed describing the number of channels required by the cells in a ring from their neighbors in the next outer ring. It can be shown that the following channel borrowing algorithm for a complete hot spot is equivalent to the demand graph approach in the general case.

4.2.1. Channel Borrowing for a Complete Hot Spot

Starting from the center cell, all the cells along any of the six emanating chains of hexagons, will be referred to *corner cells* (see Figure 4.2). Each ring, except ring 0, will then contain six corner cells. The cells between two corner cells in a ring are called *non-corner* cells. For example, ring 1 contains no non-corner cells, ring 2 contains 6, ring 3 contains 12 and so on.

Every ring is a repetitive pattern of a corner cell and a fixed number of non-corner cells constituting what will be called a *cell array.* Ring 1 is composed of six such arrays, where each array consists of only a single corner cell. Ring 2 is again composed of six arrays, each consisting of a corner and a non-corner cell. In general, ring i is composed of six cell arrays, each of which consists of a single corner cell and

$(i-1)$ non-corner cells. Our load balancing scheme for a complete hot spot does not differentiate between corner (or non-corner) cells of different arrays in a ring. When developing the algorithm for the more general case of an incomplete hot spot, we will use a different convention of addressing the cells which will be described later.

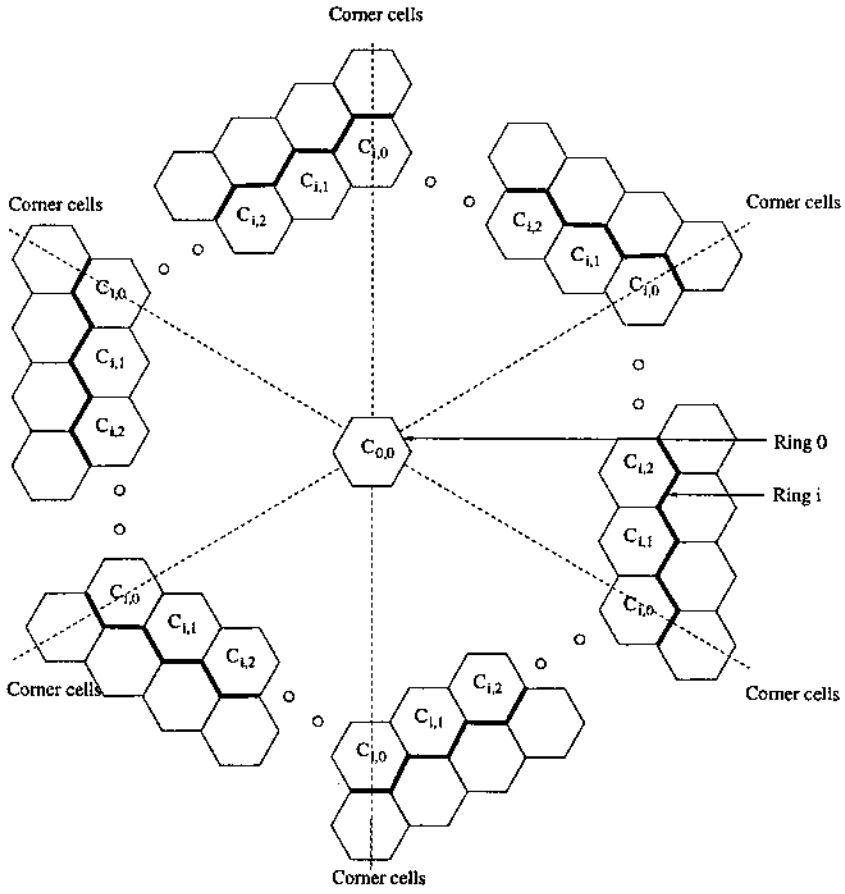


Figure 4.2: Co-ordinates of a cell within a hot spot

All the corner cells in a particular ring of a complete hot spot will have the same co-ordinate, $(i, 0)$. Moving in the anti-clockwise direction along ring i from one corner cell to the next, the co-ordinate of the jth non-corner cell in a cell array, will be (i, j), where $1 \leq j \leq i-1$. The co-ordinates repeat for the other cell arrays. Hence, in a complete hot spot H_n, the co-ordinate of any cell (except the center cell) is given by

(i, j), where $1 \leq i \leq n$ and $0 \leq j \leq i-1$. The co-ordinate $(0, 0)$ is assigned to the center cell. A cell with co-ordinate (i, j) will be denoted by $C_{i,j}$. See Figure 4.2 for illustration.

Enumeration of Channel Borrow Demand

We consider a complete hot spot H_n, a hexagonal region of cells containing $(n+1)$ 'Rings', such that there is no cold cell in any of the 'Rings' within H_n. Figure 4.3 shows a complete H_4.

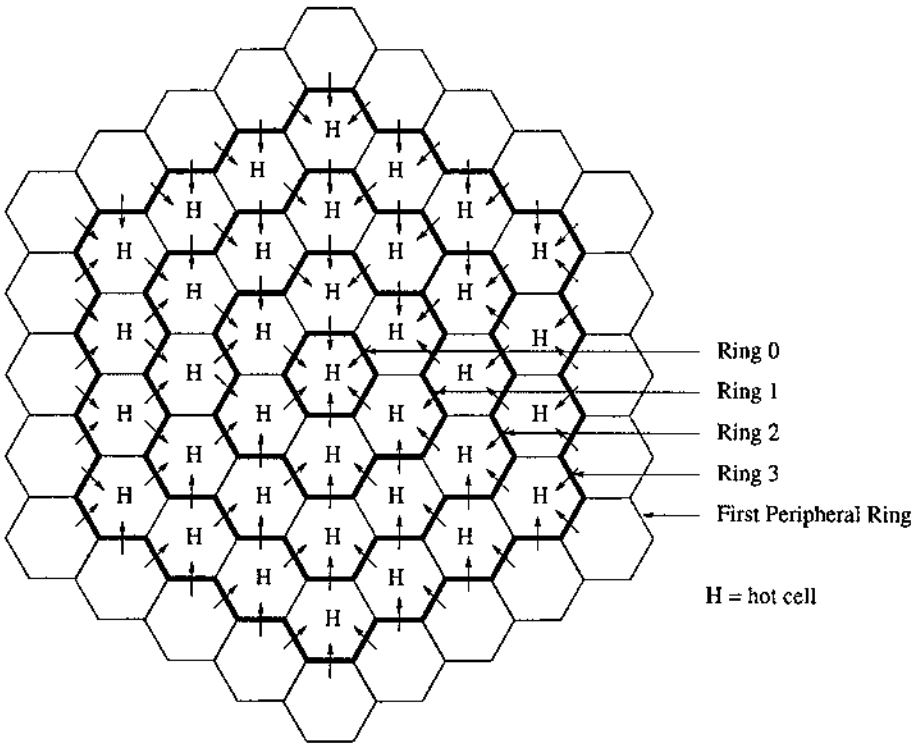


Figure 4.3: Channel lending in the complete hot spot, H_4

The total channel demand of all the cells from 'Ring 0' to 'Ring i' (for $0 \leq i \leq n$) of H_n is $\{3i(i+1)+1\}X$ which, according to our channel borrowing protocol must be provided by the $6(i+1)$ cells in ring i+1. If each cell in ring i+1 can lend l_{i+1}

channels to adjacent hot cells in ring i, then

$$l_{i+1} = \frac{\{3i(i+1)+1\}X}{6(i+1)}. \tag{4.3}$$

Let us now derive the general expressions for channel demand, i.e., the number of channels to be borrowed by a cell in 'Ring i' of a complete hot spot from its adjacent cells in ring i+1. For this purpose, let us consider the example of Figure 4.3 and trace the channel bargain (lending-borrowing) between the cells, starting from 'Ring 0' to 'Ring 3' and the 'First Peripheral Ring'.

Putting $i = 0$ in Equation (4.3), we obtain $l_1 = \frac{X}{6}$, implying that each of the six cells in 'Ring 1' lends $\frac{X}{6}$ channels to the center cell thereby satisfying its own demand of X channels. Each of the 'Ring 1' cells require X channels themselves. So each cell will borrow $\frac{X}{6} + X = \frac{7X}{6}$ channels from adjacent cells in 'Ring 2'. We make use of the following two simple facts.

Fact 1. *Any corner cell in ring i has three adjacent cells in ring i+1 and any non-corner cell has two adjacent cells in ring i+1.*

Fact 2. *Any corner cell in ring i can lend channels to its only adjacent corner cell in ring i-1, while any non-corner cell in ring i can lend channels to its two adjacent cells in ring i-1.*

Since all the cells in 'Ring 1' are corner cells, each of them can borrow channels from three adjacent cells in 'Ring 2'.

Putting $i = 1$ in Equation (4.3), we obtain $l_2 = \frac{7X}{12}$. Each cell in 'Ring 2' can lend $\frac{7X}{12}$ channels to adjacent cells in 'Ring 1'. A borrower cell $C_{1,0}$ in 'Ring 1' requires $\frac{7X}{6}$ channels, and it has three adjacent lenders in 'Ring 2' such as one corner cell $C_{2,0}$ and two non-corner cells (one belonging to the same cell array as $C_{2,0}$ and another to an adjacent array). Cell $C_{2,0}$ lends all it can (i.e., $\frac{7X}{12}$ channels) to the only borrower cell $C_{1,0}$, while the rest of the channel demand $\frac{7X}{6} - \frac{7X}{12} = \frac{7X}{12}$ of $C_{1,0}$ is satisfied by equal contributions from the other two non-corner lender cells. Since all the cells in 'Ring

1' are corner cells, the same distribution is applicable to all of them. We propose the following convention of channel borrowing for the corner cells in every 'Ring'.

Proposition 1. *The cell $C_{i,0}$ will borrow the whole of what its adjacent corner cell $C_{i+1,0}$ in ring i+1 has to offer, and its remaining channel demand will be satisfied by equal contributions from its two other adjacent non-corner cells in ring i+1.*

Now each cell of 'Ring 2' requires X channels for itself, thus borrowing $\frac{7X}{12} + X = \frac{19X}{12}$ channels from its adjacent cells in 'Ring 3'. Let us now find out what each cell in 'Ring 3' has to offer. Putting $i = 2$ in Equation (4.3), we obtain $l_3 = \frac{19X}{18}$. Proceeding in a way similar to the case for $i = 1$, it can be shown that a corner cell $C_{2,0}$ in 'Ring 2' will borrow l_3 channels from its adjacent corner cell $C_{3,0}$ in 'Ring 3' and $\frac{l_3}{4}$ channels each from its two adjacent non-corner cells, one of which ($C_{3,1}$) belongs to the same cell array as $C_{3,0}$.

Note that $C_{3,1}$ has also another adjacent cell ($C_{2,1}$) in 'Ring 2' to which $C_{3,1}$ lends its remaining $(\frac{3}{4})l_3$. The cell $C_{2,1}$ requires another $(\frac{3}{4})l_3$ channels to fulfil its demand of $\frac{19X}{12} = \frac{3}{2}l_3$ channels, which it borrows from its other adjacent cell $C_{3,2}$ in 'Ring 3'.

Next, a 'Ring 3' cell array has one corner and two non-corner cells. This is the last 'Ring' of our hot spot, and its cells borrow channels from the adjacent cold cells of the 'First Peripheral Ring'. According to Equation (4.3), each cell in the 'First Peripheral Ring' lends $l_4 = \frac{37X}{24}$ channels. Proceeding similarly, a corner cell $C_{3,0}$ in 'Ring 3' borrows l_4 channels from a corner cell in the 'First Peripheral Ring' and $(\frac{1}{6})l_4$ channels from two adjacent non-corner cells in the same ring. The non-corner cell $C_{3,1}$ borrows $(\frac{5}{6})l_4$ and $(\frac{3}{6})l_4$ channels from its adjacent cold cells $C_{4,1}$ and $C_{4,2}$ respectively. The other non-corner cell $C_{3,2}$ borrows $(\frac{3}{6})l_4$ and $(\frac{5}{6})l_4$ channels from the adjacent cold cells $C_{4,2}$ and $C_{4,3}$ respectively.

For a complete hot spot, it is easy to generalize the expressions for the number of channels borrowed by a hot cell in 'Ring i' from adjacent cells in 'Ring i+1'. Recall that $l_{i+1} = \frac{\{3i(i+1)+1\}X}{6(i+1)}$ is the number of channels that each cell in 'Ring i+1' can lend to its adjacent cells in 'Ring i'.

Lemma 1. *(a) A corner cell $C_{i,0}$ in ring i will borrow l_{i+1} channels from its adjacent corner cell $C_{i+1,0}$ and $\frac{l_{i+1}}{2i}$ channels from each of its other two adjacent non-corner cells in ring $i+1$.*

(b) A non-corner cell $C_{i,j}$ in ring i will borrow $(1-\frac{2j-1}{2i})l_{i+1}$ and $(\frac{2j+1}{2i})l_{i+1}$ channels from its adjacent non-corner cells $C_{i+1,j}$ and $C_{i+1,j+1}$ respectively.

From the viewpoint of a lender cell, the generalized expressions for the number of channels lended is as follows:

Lemma 2. *(a) A corner cell $C_{i,0}$ in ring i will lend all of its l_i lendable channels to its adjacent corner cell $C_{i-1,0}$ in ring i-1.*

(b) A non-corner cell $C_{i,j}$ will lend $\frac{2j-1}{2(i-1)}l_i$ and $\{1-\frac{2j-1}{2(i-1)}\}l_i$ channels to the adjacent cells $C_{i-1,j-1}$ and $C_{i-1,j}$ in ring i-1.

The channel borrowing algorithm is sketched below. For a complete hot spot H_n, each cold cell in the 'First Peripheral Ring' lends l_{n+1} channels to the adjacent cells in 'Ring n', each of which, in turn, retains X channels for its own use and lends l_n channels to the adjacent cells in 'Ring n-1'. This continues until 'Ring 0' is reached. When the algorithm terminates, the number of available channels in all the cells within the hot spot will be increased exactly by X. If all cells in the 'First Peripheral Ring' are not capable of lending the required l_{n+1} channels, then channels are borrowed from the subsequent 'Peripheral Rings' using the method described in the next section.

4.2.2. Channel Borrowing for an Incomplete Hot Spot

In this section, we further classify a cold cell into three subclasses, which will lead to a general channel borrowing algorithm for cells in an incomplete hot spot. This channel borrowing algorithm can also be used for the complete hot spot as well as its 'Peripheral Rings'.

Let us fix any one of the six emanating chains of corner cells as the *reference chain*. The co-ordinate of such a reference corner cell is $(i, 0)$ if it belongs to ring i. Moving in the anti-clockwise direction along ring i from the corner cell $C_{i,0}$, the co-ordinate of the jth cell will be (i, j). Here we distinguish between the different cell arrays since the same set of co-ordinates cannot be assigned to the cells within different cell arrays.

Classification of Cold Cells

We will classify a cold cell into three groups – *cold safe*, *cold semi-safe* and *cold unsafe* – according to the demands of the adjacent cell(s) of the next inner ring and the number of channels available within the cell, denoted as N_{avail}. The definitions will be different for each class depending on whether the cell is a corner or a non-corner cell.

Consider first a cold corner cell $C_{i,j}$ in ring i, having a channel demand $f_{d1}X$, where f_{d1} is a fraction given as w or w' (Figures 4.4(vii)-(x)) depending on whether its adjacent cell in ring i-1 is hot or cold. From the previous section, $w = \frac{3i(i-1)+1}{6i}$. Let w' represent the demand from a cold cell, i.e. w' will assume different values for different classes of cold cells.

On the other hand, let the channel demands for a non-corner cell in ring i be denoted as $f_{d2}X$ and $f_{d3}X$. Then f_{d2} is represented by a or a' depending on whether the demanding cell is hot or cold (a' will assume different values for the different classes of coldness). Similarly, f_{d3} can assume the values b or b' as in Figures 4.4(i)-(vi). Here $a = \frac{(2j-1)\{3i(i-1)+1\}}{12i(i-1)}$ and $b = \{1 - \frac{2j-1}{2(i-1)}\}\{\frac{3i(i-1)+1}{6i}\}$.

The criteria for cell classification are given in Table 4.1. The intuition behind this classification and the channel lending/borrowing protocol for each class of cells are described below.

1. A corner cell is termed *cold unsafe* if its channel availability falls below the average, i.e., it does not have enough channels to satisfy the demand $f_{d1}X$ without

itself becoming hot. Such a cell borrows $f_{d1}X$ channels from its three adjacent neighbors in the next outer ring and lends them entirely to its neighboring cell in the immediate inner ring.

2. A corner cell is termed *cold safe* if it has sufficient number of channels to satisfy the demand of its adjacent cell(s) without itself changing state. Hence a cold safe cell does not need to borrow any channel.

3. A non-corner cell is classified as *cold unsafe* if its channel availability is less than average or it does not have enough available channels to cater for the minimum of the two demands $f_{d2}X$ and $f_{d3}X$ without itself becoming hot. Such a cell borrows $(f_{d2} + f_{d3})X$ channels from its adjacent cells in the next outer ring and lends the whole of it, without retaining any for itself.

4. A non-corner cell is in the *cold semi-safe* state if its channel availability is more than average and it has enough available channels to satisfy the maximum demand out of $f_{d2}X$ and $f_{d3}X$, but does not have enough to satisfy both. Such a cell will lend channels from its available channel set to the adjacent cell with the minimum channel demand, i.e., $min(f_{d2}X, f_{d3}X)$. This kind of lending strategy guarantees that the available channel set will not be reduced to the critical value hC in the worst case, after channels are lended.

   To cater for the channel demand of the other adjacent cell(s), a cold semi-safe cell acts like a cold unsafe cell, i.e. it borrows the requisite channels from adjacent cells in the next outer ring only to lend them to the demanding cell.

5. A non-corner cell is termed *cold safe* if it has sufficient number of channels to satisfy the demand of its adjacent cell(s) without itself changing state. A cold safe cell does not need to borrow any channel.

Table 4.1: Cell Classification for Incomplete Hot Spot

Cell position	*Class*	*Condition*
corner	cold safe	$(N_{avail} \geq d_c^{avg}C)$ *and* $(N_{avail} - f_{d1}X > hC)$
	cold semi-safe	*does not exist*
	cold unsafe	$(N_{avail} < d_c^{avg}C)$ *or* $(N_{avail} - f_{d1}X \leq hC)$
	hot	$N_{avail} \leq hC$
non-corner	cold safe	$(N_{avail} \geq d_c^{avg}C)$ *and* $(N_{avail} - (f_{d2} + f_{d3})X > hC)$
	cold semi-safe	$(N_{avail} \geq d_c^{avg}C)$ *and* $(N_{avail} - \{max(f_{d2}, f_{d3})\}X \geq hC)$ *and* $(N_{avail} - (f_{d2} + f_{d3})X \leq hC)$
	cold unsafe	$(N_{avail} < d_c^{avg}C)$ *or* $(N_{avail} - \{min(f_{d2}, f_{d3})\}X \leq hC)$
	hot	$N_{avail} \leq hC$

Modification of Channel Demands

Figure 4.4 shows the channel demand of a ring i cold cell from adjacent cell(s) in ring i+1. This demand is a function of the classification of the cold cell as well as the demand from its neighboring cell(s) in ring i-1. Figure 4.4(i) shows the case for a complete hot spot, where both the ring i-1 cells are hot. Thus, the coefficients a, b, c and d can be computed exactly the same way as shown in the previous section. For the other classes of cells in ring i, our objective is to determine the coefficients c', d' etc. in Figures 4.4(ii)-(vi) in terms of the known coefficients and the modified channel demands a' and b' from ring i-1 cells, whenever applicable.

In the case of corner cells, Figure 4.4(vii) shows the situation for a complete hot spot where the ring i-1 cell is hot. The coefficients w, e, f and g then are determined, and our objective is to determine the modified demands e', f', g' etc. (refer to Figures 4.4(viii)-(x)) in terms of these known parameters and the modified demand w', whenever applicable.

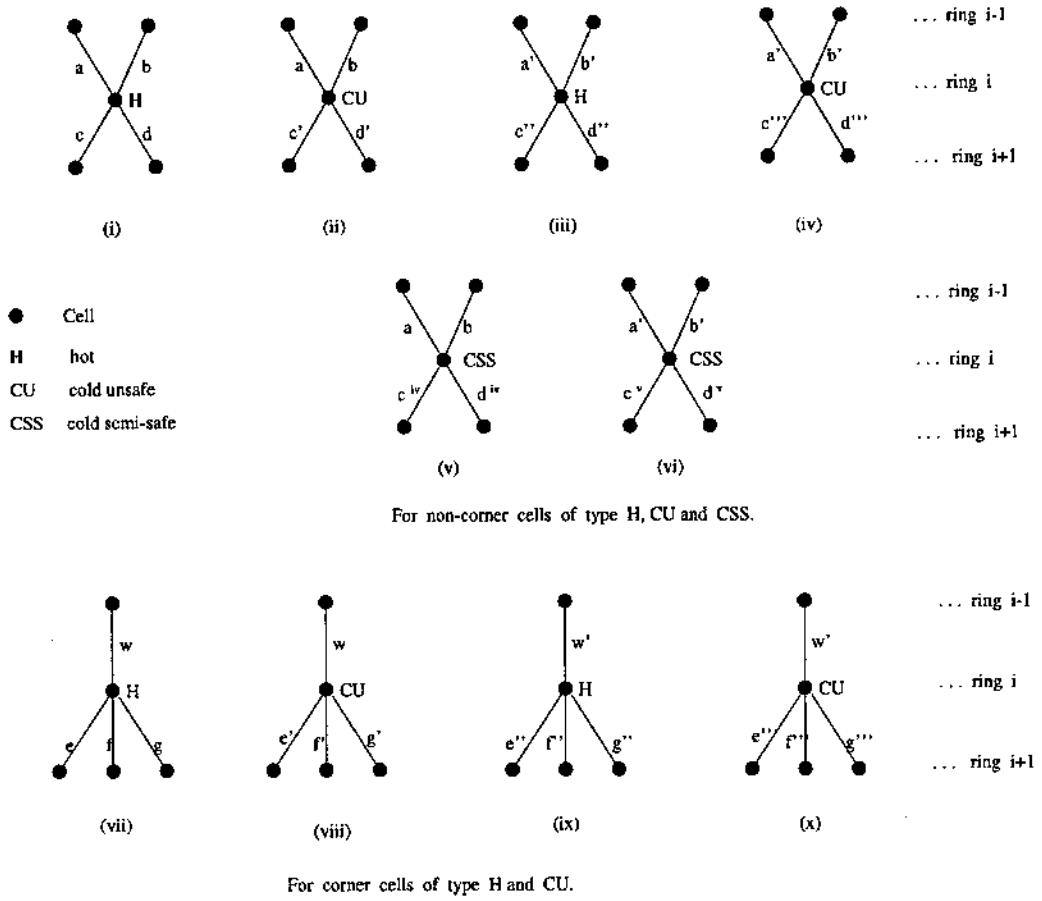


Figure 4.4: Channel demand graphs for corner and non-corner cells

Non-Corner Cells of Types Hot, Cold Unsafe and Cold Semi-safe

The channel demand equation for a hot non-corner cell in ring i whose adjacent cells in ring i-1 are also hot (see Figure 4.4(i)) is given by

$$cX + dX - aX - bX = X \tag{4.4}$$

The channel demand equation for a cold unsafe non-corner cell in ring i, whose adjacent cells in ring i-1 are hot (Figure 4.4(ii)) is given by,

$$c'X + d'X - aX - bX = 0 \tag{4.5}$$

Rewriting Equation (4.4) as

$$c(1-\frac{1}{c+d})X + d(1-\frac{1}{c+d})X - aX - bX = 0, \tag{4.6}$$

we get $c' = c(1-\frac{1}{c+d})$ and $d' = d(1-\frac{1}{c+d})$.

The channel demand equation for a cold semi-safe, non-corner cell in ring i whose adjacent cells in ring i-1 are also hot (Figure 4.4(v)) is given by

$$c^{iv}X + d^{iv}X - \{max(a,b)\}X = 0 \tag{4.7}$$

Rewriting Equation (4.4) as

$$c(1-\frac{1+min(a,b)}{c+d})X + d(1-\frac{1+min(a,b)}{c+d})X - \{max(a,b)\}X = 0 \tag{4.8}$$

we get $c^{iv} = c(1-\frac{1+min(a,b)}{c+d})$ and $d^{iv} = d(1-\frac{1+min(a,b)}{c+d})$.

Proceeding in a similar way, it can be shown that

$$\begin{aligned}
c'' &= c(1-\tfrac{(a+b)-(a'+b')}{c+d}) & d'' &= d(1-\tfrac{(a+b)-(a'+b')}{c+d}) \\
c''' &= c(1-\tfrac{1+(a+b)-(a'+b')}{c+d}), & d''' &= d(1-\tfrac{1+(a+b)-(a'+b')}{c+d}) \\
c^{v} &= c(1-\tfrac{(a+b)-(a'+b')}{c+d}-\tfrac{1+min(a',b')}{c+d}), & d^{v} &= d(1-\tfrac{(a+b)-(a'+b')}{c+d}-\tfrac{1+min(a',b')}{c+d}).
\end{aligned}$$

Corner Cells of Types Hot and Cold Unsafe

The channel demand equation for a hot corner cell in ring i whose adjacent cells in ring i-1 are also hot is given as,

$$eX + fX + gX - hX = X \tag{4.9}$$

Proceeding in a similar way as in non-corner cells leads to Figure 4.4(vii)-(x). It can be then be shown that

$$\begin{aligned}
e' &= e(1-\tfrac{1}{e+f+g}), & f' &= f(1-\tfrac{1}{e+f+g}), & g' &= g(1-\tfrac{1}{e+f+g}) \\
e'' &= e(1-\tfrac{h-h'}{e+f+g}), & f'' &= f(1-\tfrac{h-h'}{e+f+g}), & g'' &= g(1-\tfrac{h-h'}{e+f+g}) \\
e''' &= e(1-\tfrac{1+h-h'}{e+f+g}), & f''' &= f(1-\tfrac{1+h-h'}{e+f+g}), & g''' &= g(1-\tfrac{1+h-h'}{e+f+g}).
\end{aligned}$$

4.2.3. Channel Demand Graph for Hot Spot

Initially, a channel demand graph is constructed based on the channel demand and class of each cell in the hot spot and its 'Peripheral Rings'. A *demand graph* is a layered graph with the uppermost layer representing the 'Ring 0' of the hot spot, while each subsequent layer consists of nodes representing cells in the next outer ring, until we reach a 'Peripheral Ring' consisting only of cold safe cells which form the lower most layer. Therefore, the demand graph spans all the 'Rings' of the hot spot as well at least one of its 'Peripheral Rings'. Let ring i correspond to layer j of the demand graph. Then for a node u in layer j and a node v in layer j+1, there exists an edge (u,v) if there is channel demand from the ring j cell, to the ring j+1 cell. The edge weight is given by $\frac{demand(u,v)}{X}$. Hence the lower most layer of the demand graph consists only of cells in the cold safe state with no channel demand.

The preceding section shows that given the channel demands towards a cell in ring i in an incomplete hot spot, we can find exactly how the ring i cells' own channel demands from adjacent ring i+1 cells are modified with respect to the same demands if it belonged to a complete hot spot. Since we know the expressions for these channel demands in a complete hot spot, the modified channel demand for a ring i cell in an incomplete hot spot from adjacent ring i+1 cells can be derived in terms of these demands in a complete hot spot and the demand(s) from the ring i-1 cell(s) in our incomplete hot spot. Construction of the demand graph involves computing these modified demands for the cells in a layer by layer fashion, starting from the uppermost layer.

Let the uppermost layer of the graph consist of node(s) representing the hot cell(s) of 'Ring 0'. Let us assume for the time being that it is the center cell itself. The demands from each of its six adjacent cells in ring 1 is $\frac{X}{6}$. Depending on this demand and the channel availability of these cells, they are classified as hot, cold safe, cold unsafe or cold semi-safe, and thus form the second layer of nodes in the demand graph. The demands of the ring 1 cells from ring 2 cells can now be computed, which lead to

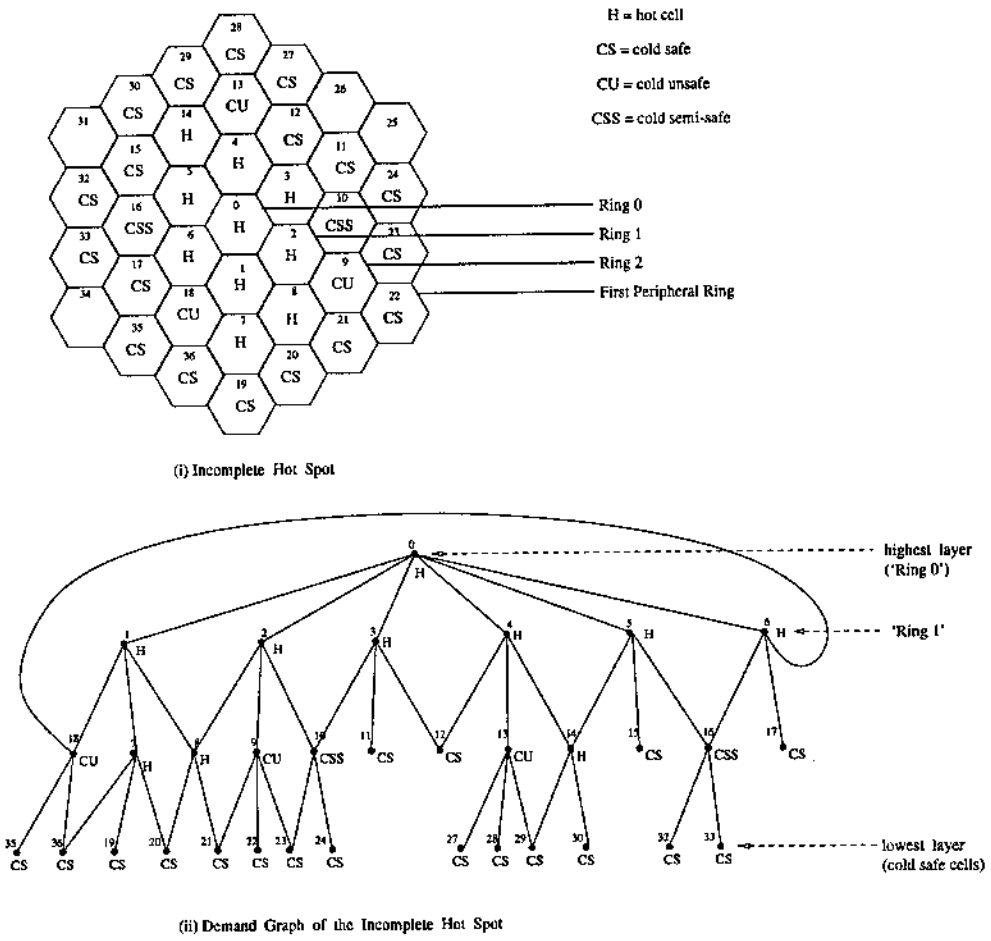


Figure 4.5: Channel demand graph for an incomplete hot spot

the classification of the ring 2 cells. In this way, the demand graph is constructed in a top down fashion. The construction of the demand graph terminates when a ring consisting only of cold safe cells is reached. An example of an incomplete hot spot and its demand graph is shown in Figure 4.5. The edge weights are not shown in this figure.

Now consider the case when the center cell does not constitute the uppermost layer of the demand graph, i.e. the center cell is not hot. Then the uppermost layer consists of node(s) representing the hot cells(s) of the ring j (say) nearest to the center cell and containing at least one hot cell. To compute the channel demand of

the hot cells in ring j from the adjacent cells in ring j+1, we proceed as in the previous subsection and for non-corner hot cells in ring j, we obtain

$$c^{vi} = c(1 - \frac{a+b}{c+d}), \quad d^{vi} = d(1 - \frac{a+b}{c+d}). \tag{4.10}$$

Similarly for corner hot cells in ring j, we obtain

$$e^{iv} = e(1 - \frac{w}{e+f+g}), \quad f^{iv} = f(1 - \frac{w}{e+f+g}), \quad g^{iv} = g(1 - \frac{w}{e+f+g}). \tag{4.11}$$

Let us now sketch the channel borrowing algorithm. After the channel demand graph for the incomplete hot spot is constructed, the channel borrowing algorithm works in a bottom up fashion, starting from the lower most layer of the demand graph. These cold safe cells lend the necessary amount of channels to adjacent nodes in the next higher layer to satisfy the edge demands. This brings all the cells in this layer to the cold safe state thereby allowing them to lend channels to nodes in the next higher layer, and so on. The algorithm continues till the uppermost layer of the demand graph is reached. Thus, all the cells in the hot spot will have their channel demands fulfilled and will be in the cold state.

4.3. Performance Modeling

We develop two discrete time Markov models, one for a complete hot spot and the other for a cell within the hot spot. In fact, the Markov model for a cell is developed first and some of the analytical results obtained in this model are then used to capture the evolution of a complete hot spot. Although a complete hot spot is assumed to simplify the analysis, our analysis be easily extended to the case of an incomplete hot spot.

4.3.1. Markov Chain Model of a Cell

We develop a Markov chain model of a cell in a complete hot spot, to capture the channel availability pattern in that cell with respect to discrete time intervals. Figure

4.6 shows the Markov chain model for such a cell. Let the stochastic process describing this Markov chain take up the discrete set of values $\{n : 0 \leq n \leq C\}$, where n denotes the number of available channels in the cell. The process is said to be in state S_i if $n = i$. Let us assume that the call arrival process in the cell is Poisson with rate λ. Let the call termination process be also Poisson with parameter μ. These are valid assumptions so far as the normal telephone calls are concerned.

If the cell is in any one of the states S_i, for $0 \leq i \leq \lceil hC \rceil$, then it is a hot cell, where C is the total number of channels assigned to the cell and h is the threshold parameter. From now on, wherever we use hC, it should be interpreted as $\lceil hC \rceil$. When the cell enters the state S_{hC+1} from S_{hC}, it becomes a cold cell from a hot one. The cell remains in the cold state as long as it is in one of the states S_i, where $hC + 1 \leq i \leq C$.

Our cell is in state S_i if the number of available channels is i, i.e. the number of channels in use is $C - i$. Hence the cell will make a state transition from S_i to S_{i+1} whenever any one of $C - i$ ongoing simultaneous calls terminate. This implies that the forward state transition probability from S_i to S_{i+1}, is given as $(C-i)\mu$. Some of these forward transition probabilities are shown in Figure 4.6. The reverse transition will take place whenever there is a new call arrival, implying that the probability is given as λ. Also, there can be a discrete increase of X (the number of channels to be retained by a hot cell) in the number of available channels of the cell in hot state when the load balancing algorithm runs. We assume that this increase in the number of channels can take place with a constant probability μ' from any of the states S_i, $0 \leq i \leq hC$, which accounts for the forward arcs from S_i to S_{i+X} in Figure 4.6. In the next subsection, we will describe a method to estimate the value of μ'.

Let us now compute the steady state probability Π_i for state S_i of the Markov chain. It is evident from Figure 4.6 that the balance equations are not identical for all the states. We can actually partition the whole chain into four subchains and derive the limiting probabilities of the states within each subchain individually, in terms of

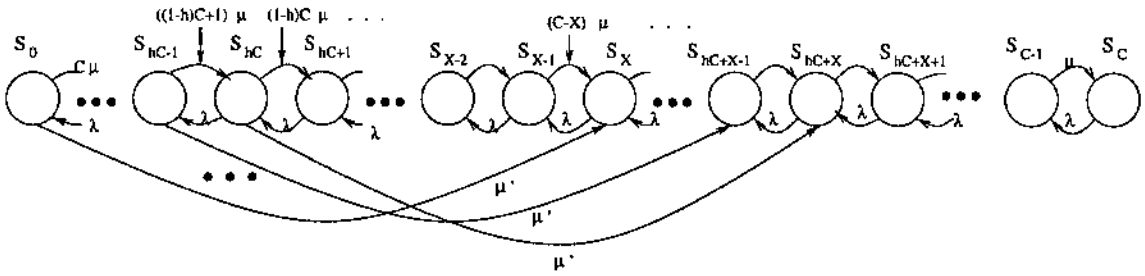


Figure 4.6: Markov model for a cell in a complete hot spot

the limiting probability Π_0 of the state S_0. The subchains are:

Subchain 1 : consists of the states S_i for $0 \leq i \leq hC$,

Subchain 2 : consists of S_i, for $hC + 1 \leq i \leq X - 1$,

Subchain 3 : consists of S_i, for $X \leq i \leq hC + X$,

Subchain 4 : consists of rest of the states.

Let us now write down the balance equation for a state S_i in Subchain 1. The boundary cases for the recursive balance equation are

$$\Pi_i = \begin{cases} \Pi_0 & : \quad i = 0 \\ (\frac{C\mu+\mu'}{\lambda})\Pi_0 & : \quad i = 1 \end{cases} \tag{4.12}$$

and the balance equation for S_i is

$$\Pi_{i-1}(C - i + 1)\mu + \Pi_{i+1}\lambda = \Pi_i((C - i)\mu + \mu' + \lambda). \tag{4.13}$$

To solve for the closed form expression for Π_i, we apply Geometric transform (G-transform) to Equation (4.13). The G-transform of Π_i is given as, $G(\Pi_i) = G(z) = \sum_{i=0}^{\infty} \Pi_i z^i$. Also, by the shifting and scaling properties of G-transform, we have, $G(\Pi_{i+r}) = z^{-r}(G(z) - \sum_{j=0}^{r-1} \Pi_j z^j)$ and $G(i\Pi_i) = z\frac{dG(z)}{dz} = zG'(z)$, respectively. Using the initial conditions of Π_i given in Equation (4.12), the following linear first order differential equation is solved for $G(z)$.

$$G'(z) + \frac{\{Cz^2 - (C + \rho' + \rho)z + \rho}{\mu z^2(1 - z)} G(z) = \frac{\rho\Pi_0}{z^2} \tag{4.14}$$

where $\rho = \frac{\lambda}{\mu}$ and $\rho' = \frac{\mu'}{\mu}$.

The expression for the steady state probability for Subchain 1 is obtained by taking the inverse transform of $G(z)$ as

$$\Pi_i = \frac{\Pi_0}{(C-i)!} \sum_{i'=0}^{C} \binom{C}{i'} \sum_{j=0}^{\rho'+i'} (-1)^{i+i'-j} \prod_{k=1}^{j} \frac{(\rho' + i' - k + 1)}{\rho} \prod_{m=1}^{C-i} (i' - j - m + 1) \quad (4.15)$$

where $1 \leq i \leq hC$.

Equation (4.13) can be used to derive the steady state probability Π_{hC+1} of S_{hC+1} in terms of Π_{hC}, which in turn is given by Equation (4.15). Let $\Pi_{hC+1} = \eta_1 \Pi_{hC}$. The values of Π_{hC} and Π_{hC+1} are the boundary cases for the recursive balance equation for Π_i in case of Subchain 2, which is given below.

$$\Pi_{i-1}(C - i + 1)\mu + \Pi_{i+1}\lambda = \Pi_i(\lambda + (C - i)\mu). \quad (4.16)$$

Proceeding in a similar manner as in the case of Subchain 1, we obtain

$$\Pi_i = \Pi_0[A \cdot \{(\eta_1 - 1)\rho - C\} + \{\sum_{k=1}^{i} \prod_{j=1}^{k} \frac{(C + j - k)}{\rho} - \sum_{k=1}^{i-1} \prod_{j=1}^{k} \frac{(C + j - k - 1)}{\rho}\}] \quad (4.17)$$

where, $hC + 1 \leq i \leq X - 1$ and

$$A = \frac{1}{\rho}\{1 + \sum_{k=1}^{i-1} \prod_{j=1}^{k} \frac{(C + j - k - 1)}{\rho}\} - \frac{1}{e^{\rho}} \sum_{j \in Z} \frac{1}{j!} \sum_{i' \in Z - \{0\}} \frac{\rho^{i'+j}}{i'i'!} \binom{i'}{k} (-1)^{i'-k} + \frac{1}{e^{\rho}} \sum_{j' \in Z} \frac{\rho^{j'}}{j'!} \sum_{i'' \in Z} \frac{1}{2i'' + 1} \binom{2i'' + 1}{k'} (-2)^{2(i''+1)-k'} \quad (4.18)$$

Here Z is the set of positive integers and the variables i', j, i'', j' will assume certain integer values from Z such that the following two conditions are satisfied.

(1) i' and j are integers satisfying the equation $k = i + i' + j - C$ for $0 \leq k \leq i'$.

(2) i'' and j' are integers satisfying the equation $k' = C + 2i'' - j' - i + 1$ for $0 \leq k' \leq 2i'' + 1$.

Henceforth, wherever A will be used, it is assumed that A is given as in Equation (4.18) satisfying the above two conditions.

Using Equation (4.16), the boundary cases for the recursive solution for the steady state probabilities of Subchain 3 can be derived as in the case of Subchain 2. Let $\Pi_X = \eta_2 \Pi_{X-1}$. The recursive balance equation for Π_i in Subchain 3 is given as

$$\Pi_{i-1}(C-i+1)\mu + \Pi_{i-X}\mu' + \Pi_{i+1}\lambda = \Pi_i((C-i)\mu + \lambda). \tag{4.19}$$

Here we note that Π_{i-X}, for $X \leq i \leq hC + X$, are the steady state probabilities of Subchain 1. Hence the G-transform of Π_{i-X} is equal to the G-transform of Π_i for $0 \leq i \leq hC$, which is derived earlier. Let A_l denote the same expression as A with the constant C replaced by the variable l. Substituting $G(\Pi_i)|_{0 \leq i \leq hC}$ for $G(\Pi_{i-X})|_{X \leq i \leq hC+X}$ and proceeding similar to the cases of Subchains 1 and 2, we obtain

$$\Pi_i = \Pi_0[A(\eta_2\rho - C) + \{\sum_{k=1}^{i}\prod_{j=1}^{k}\frac{(C+j-k)}{\rho} - \sum_{k=1}^{i-1}\prod_{j=1}^{k}\frac{(C+j-k-1)}{\rho}\} + \rho_1\sum_{l=0}^{\infty}\frac{\eta_3 A_l}{l!}] \tag{4.20}$$

where $X \leq i \leq hC + X$ and

$$\eta_3 = \sum_{i'=0}^{C}\binom{C}{i'}\sum_{j=0}^{\rho'+i'}\prod_{k=1}^{j}\frac{(\rho'+i'-k'+1)}{\rho}\prod_{m=1}^{l}(i'-j-m+1)(-1)^{C+i'-j-l}. \tag{4.21}$$

Let the base cases for the recursive solution for the steady state probabilities of Subchain 4 be Π_{hC+X} (derived from Equation (4.20)) and $\Pi_{hC+X+1} = \eta_4 \Pi_{hC+X}$ (derived from Equation (4.19)). Then the steady state probability Π_i for Subchain 4 will have the same expression as that for Subchain 2 with η_4 replacing η_1. Hence

$$\Pi_i = \Pi_0[A \cdot \{(\eta_4 - 1)\rho - C\} + \{\sum_{k=1}^{i}\prod_{j=1}^{k}\frac{(C+j-k)}{\rho} - \sum_{k=1}^{i-1}\prod_{j=1}^{k}\frac{(C+j-k-1)}{\rho}\}] \tag{4.22}$$

for $hC + X + 1 \leq i \leq C$.

Using Equations (4.15), (4.17), (4.20), (4.22) and the fact that $\sum_{i=0}^{C}\Pi_i = 1$, we first derive the expression for Π_0. Expressions for the steady state probabilities Π_i, where $i > 0$, are then derived in terms of Π_0 using Equations (4.15), (4.17), (4.20), (4.22). Two important performance metrics for our load balancing algorithm are the

probability of call blockade in a cell and the probability of a cell being hot. The steady state probability Π_0 gives the call blocking probability of a cell.

The probability of a cell being hot is given by

$$\begin{aligned} p_h &= \sum_{i=0}^{hC} \Pi_i \\ &= \Pi_0[1 + \sum_{i=1}^{hC} \tfrac{1}{(C-i)!} \sum_{i'=0}^{C} \binom{C}{i'} \sum_{j=0}^{\rho'+i'} (-1)^{i+i'-j} \prod_{k=1}^{j} \tfrac{(\rho'+i'-k+1)}{\rho} \prod_{m=1}^{C-i} (i'-j-m+1)]. \end{aligned}$$

4.3.2. Estimation of the probability μ'

Let us consider a cell in 'Ring j' of a complete hot spot. By our channel borrowing strategy, μ' is the probability that sufficient number of channels are available in the system excluding the part of the hot spot from 'Ring 0' to 'Ring j' to meet the channel demand, $D_{hs} = \{3j(j+1)+1\}X$. Let H_j denote the set of cells forming 'Rings' 0 to j and $\bar{H}_j$ is complement set. A cell will only be able to lend channels if it is in the cold safe state. If $N_{avail}(i)$ be the number of available channels in cell i of the system, this implies that the actual number of channels available in the system for lending purposes is $A = \sum_{i \in \bar{H}_j} N_{avail}(i) - |\bar{H}_j|(hC+1)$. Thus,

$$\begin{aligned} \mu' &= \text{Prob}[A \geq D_{hs}] \\ &= \text{Prob}[\text{total number of available channels} \geq D_{hs} + |\bar{H}_j|(hC+1)]. \end{aligned}$$

To compute μ', let us consider the evolution of the entire system in the time period between two successive calls of the load balancing algorithm. Assume that the system contains M cells.

By our previous assumption, the call arrival and termination processes in each cell i are Poisson and denoted as λ_i'' and μ_i'', respectively. Then, using results from queuing theory, the call arrival and termination processes for the entire system are Poisson, and the rates are given by $\lambda'' = \sum_{i=1}^{M} \lambda_i''$ and $\mu'' = \sum_{i=1}^{M} \mu_i''$ respectively. Our system can then be modeled as an $M/M/k/k$ queuing system with state $S_i = i$, where $0 \leq i \leq MC$ is the total number of available channels in the system (see Figure 4.7). From known queuing theoretic results, the steady state probability of S_i for such a

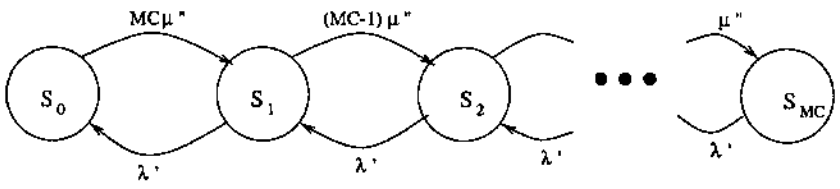


Figure 4.7: Markov model of the system between two runs of the load balancing algorithm

birth and death Markov chain is given by,

$$\Pi_i = \frac{(\frac{\lambda''}{\mu''})^i}{i!}\Pi_0 \tag{4.23}$$

where $\Pi_0 = [\sum_{l=0}^{MC} \frac{(\frac{\lambda''}{\mu''})^l}{l!}]^{-1}$.

Thus, the expression for the transition probability μ' can be derived as

$$\mu' = Prob[total\ number\ of\ available\ channels \geq D_{hs} + |\bar{H}_j|(hC+1)] \tag{4.24}$$

$$= 1 - \sum_{i=0}^{D_{hs}+|\bar{H}_j|(hC+1)-1} \Pi_i \tag{4.25}$$

where Π_i is given by Equation (4.23).

4.4. Simulation Experiments

Simulation experiments are carried out emulating a real time cellular mobile environment in an urban area. For example, the downtown area of the city is chosen as the hot spot and the suburbs comprise the outer rings of cells. A considerable reduction in the blocking probability of the system with load balancing is observed as compared to the system without load balancing. Also the performance of our scheme is compared with CBWL (channel borrowing without load balancing) scheme proposed in the literature. In our simulation model, call arrivals and terminations are modeled as Poisson processes with rates λ and μ, respectively, and 'time' is equivalent to the number of iterations. A fixed channel assignment scheme with an initial allocation

of C channels per cell is assumed. The hot spot density and the number of rings comprising the hot spot are variable.

4.4.1. Impact of size and density of hot spot

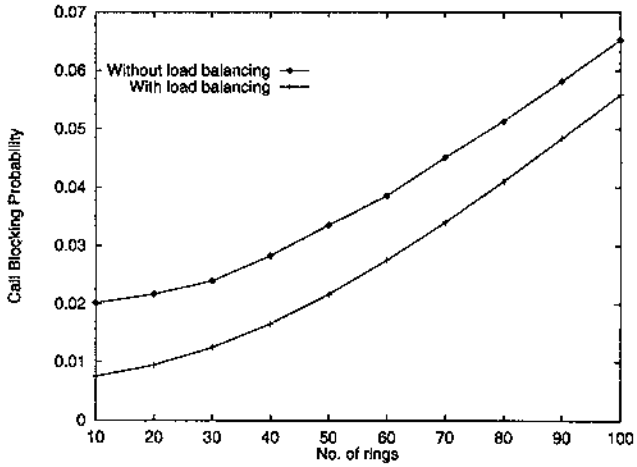


Figure 4.8: Blocking probability vs. size of the hot spot

The goal of this experiment is to measure the stability of our load balancing scheme under most severe tele-traffic demands. For this we define a parameter called, *spot density* (sd), varying which we can control the number of hot cells in the hot spot. For example, sd = 0.1 gives only 3 hot cells in a 4 'Ring' hot spot, while sd = 0.5 gives 16 and sd =0.9 gives 32 hot cells in the same hot spot. The size of the hot spot is varied by the number of rings that comprise hot cells. For this particular run, the spot density was fixed at sd = 0.5 and the size was varied. The total size of the system was 100 rings.

It is observed from Figure 4.8 that the percentage reduction of blocked calls attains a maximum of 62.50% with a very small hot spot (number of rings = 10), and a minimum of 14.38% with almost the entire system being a hot spot (number of rings = 90). Hence, under very high tele-traffic demand, there is still a 15% improvement in system performance with the introduction of load balancing.

4.4.2. Impact of call arrival rate

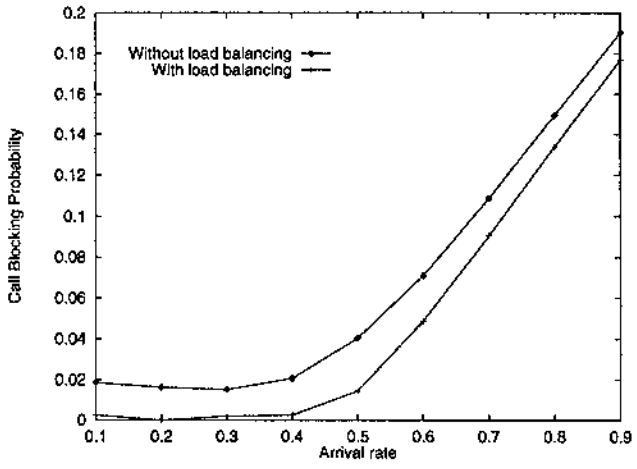


Figure 4.9: Blocking probability for various call arrival rates

This experiment evaluates the performance of the load balancing strategy under various call generation rates. With a low call generation rate of $\lambda = 0.1$, the average percentage reduction in the number of blocked calls is 85.9%. With a very high call generation rate of $\lambda = 0.9$, the performance of the algorithm suffers, but still an improvement of about 7% is observed as compared to the system without load balancing.

4.4.3. Comparison with CBWL

The proposed load balancing scheme is compared with the CBWL scheme [JR94]. In Chapter 3, we observed that CBWL outperforms every other existing load balancing scheme under moderate tele-traffic load, while under heavy load only LBSB (Load Balancing with Selective Borrowing) performs better. Hence CBWL is chosen as the most suitable candidate for comparison with our new load balancing scheme.

Figure 4.10 compares the performance of our scheme with CBWL with respect to the call blocking probability for various call arrival rates. The results show that our

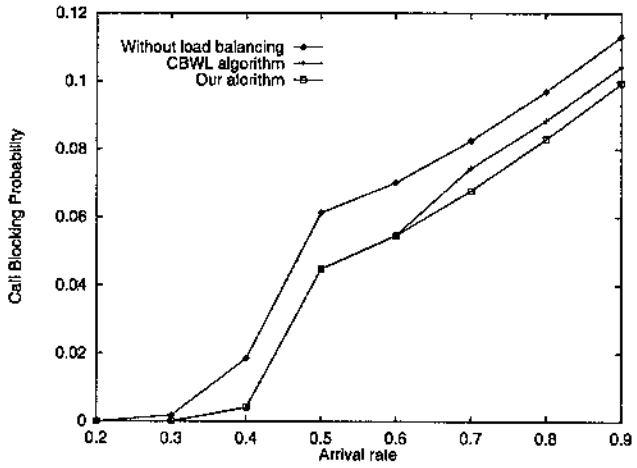


Figure 4.10: Comparison of our scheme with CBWL and no load balancing

scheme performs better than CBWL, both for moderate and high tele-traffic loads in the system. This is expected because in case of a dense hot spot, the interior cells tend to starve in the CBWL channel assignment scheme. Our scheme adopts a layered (structured) approach of channel migration from the exterior cold cells to the interior cells of the hot spot region. Performance improvements over a scheme without load balancing are 12.0% for our scheme and 7% for CBWL for a high arrival rate of 0.9, whereas for an arrival rate of 0.3 the improvements are 95.3% for our schéme and 98.7% for CBWL.

4.5. Summary

In this chapter, we propose a load balancing strategy for the tele-traffic hot spot problem in cellular networks. A hot spot is viewed as a stack of hexagonal rings of cells and is termed complete if all the cells within it are hot. We first propose a load balancing scheme for a complete hot spot, which is later extended to the general case of incomplete hot spots. Load balancing is achieved by a structured borrowing mechanism whereby a hot cell can borrow a fixed number of channels (depending

on its relative position within the hot spot) only from adjacent cells in the next outer ring. In this way, unused channels are migrated into the hot spot from its peripheral rings. The structured borrowing mechanism also reduces the amount of interference between the borrower cell and the co-channel cells of the lender. Detailed analytical modeling of the system with our load balancing scheme captured certain useful performance metrics like call blocking probability, the probability of a cell being hot and the evolution of the hot spot size. Exhaustive simulation experiments are carried out proving that our load balancing algorithm is robust under severe load conditions. Also, comparison of our scheme with the CBWL strategy demonstrates that under moderate and even very high load conditions, a performance improvement of as high as 12% in terms of call blockade is achievable with our load balancing scheme.

CHAPTER 5

QUALITY-OF-SERVICE BASED RESOURCE MANAGEMENT FOR WIRELESS MULTI-MEDIA

Over the last decade there has been a rapid growth of wireless communication technology. Voice communication over wireless links using cellular phones has matured and become a significant feature of communication today. Alongside, portable computing devices such as notebook computers and personal digital assistants (PDA) have emerged – as a result of which such applications as electronic mail and calendar/diary programs are being provided to mobile or roving users. Observing this trend, it can be predicted that the next generation of traffic in high-speed wireless networks will be mostly generated by personal multimedia applications including fax, video-on-demand, news-on-demand, WWW browsing, and traveler information systems. For multimedia traffic (voice, video, and data) to be supported successfully, it is necessary to provide *quality-of-service* (QoS) guarantees between the end-systems.

QoS means that the multimedia traffic should get predictable service from resources in the communication system. Typical resources are CPU time (for the communication software to execute) and network bandwidth. The communication software also gives an acceptable end-to-end delay and maximum delay jitter, i.e., maximum allowed variance in the arrival of data at the destination. In most cases, QoS requirements are specified by the 3-tuple $\langle bandwidth, delay, reliability \rangle$. The QoS provisioning problem for multimedia traffic in non-wireless networks such as broadband wire-line networks (e.g., B-ISDN) has been extensively studied. For a good introduction of the relevant issues, refer to [Kur93]. The ongoing work mainly concentrates on the problems of bandwidth management and switch-based scheduling to provide deterministic guarantees on end-to-end delay, throughput and packet losses.

However, the two major differences [NSA96] between wire-line and wireless networks are in *link characteristics* and *mobility*. The broadband wire-line network transmission links are characterized by high transmission rates (in the order of Gbps) and very low error rates. In contrast, wireless links have a much smaller transmission rate (Kbps-Mbps) and a much higher error rate. The most recent private wide area wireless data networks such as ARDIS or Mobitex offer a channel rate of about 8Kbps to 2Mbps and similar local area wireless networks such as Motorola's Altair-II offer about 6 Mbps [NT95]. Additionally, wireless links experience losses due to multi-path dispersion and Rayleigh fading. The second major difference between the two networks is the user mobility. In wire-line networks, the user-network-interface (UNI) remains fixed throughout the duration of a connection whereas the UNI in a wireless environment keeps on changing throughout the connection. Therefore, it is necessary to re-design or revise existing QoS provisioning techniques for wireless networks.

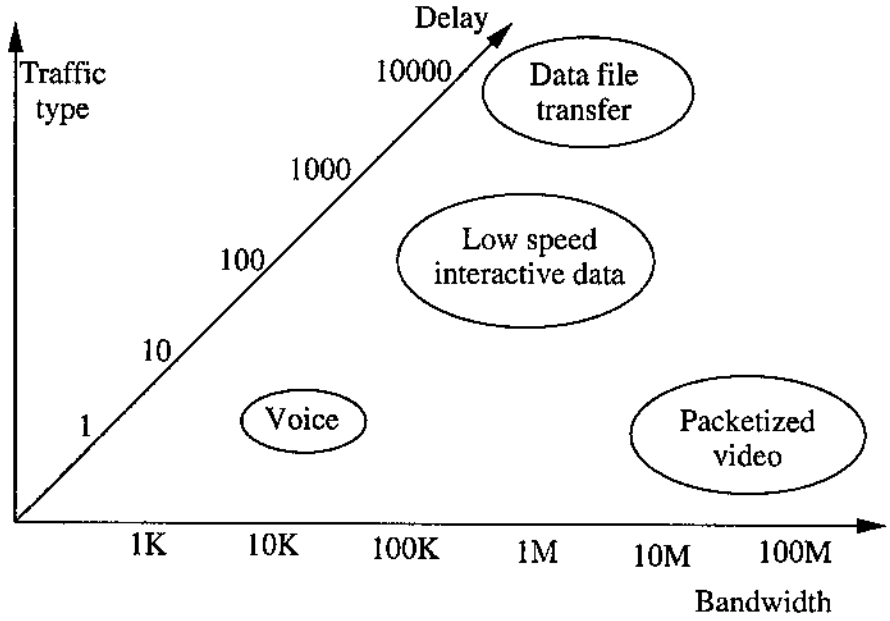


Figure 5.1: Characteristics of multimedia traffic

Figure 5.1 shows the characteristics of traffic types in wireless networks, in terms of

the bandwidth usage and typical tolerable delay. Since the traffic varies significantly within a wide range of parameters, guaranteeing QoS becomes even more challenging. From this viewpoint, multimedia traffic can be broadly classified as *real-time* and *non-real-time* [AN95]. Real-time traffic (e.g., video and voice) is highly delay sensitive, while non-real-time traffic (e.g., TCP packets and text data transfers) can tolerate large delays. Existing public data networks such as cellular digital packet data (CDPD), general packet radio service (GPRS), and high speed circuit switched data (HSCSD) utilize the unused voice capacity to support low-priority, non-real-time data. In case of scarcity of available bandwidth, the transmitted data packets are buffered or suitable flow control techniques are used leading to an increase in transmission delay.

In spite of the recent auction of 1850-2000 MHz band by the FCC for *personal communication services* (PCS) users, bandwidth is still the major bottleneck in most real-time multimedia services. Such services can substantially differ in bandwidth requirements, e.g. 9.6 Kbps for voice service and 76.8 Kbps for video. Most of the earlier research on wireless bandwidth allocation concentrated on the problem of optimizing frequency reuse, and hence the carried traffic, for only one class of service (i.e., voice). For a multi-class wireless service provider, the carried traffic in the system for each class is to be considered individually.

5.1. **Related Work**

Recently, some work has been proposed for providing better QoS guarantees for multimedia traffic in wireless cellular networks [OKS96, AN94, AN95, NSA96, Sur96a, Sur96b, PTP94, RP96]. Rappaport and Purzynski [RP96] have developed analytical models for a cellular mobile environment consisting of mixed platform types with different classes of channel and resource requirements. Different platform types are pedestrian, autos etc. where a mobile terminal is installed. Different call types will require different types and amounts of resources. The authors have considered two

broad cases – with hand-off queuing and without. In both cases, hand-off calls are given priority over ordinary calls with the former given a certain cut-off resource priority over the latter. Also, quotas for the usage of each type of resource is implemented. The problem is mapped into multi-dimensional Markov chains, with permissible states determined by the resource usage constraints. The underlying driving processes of the model are call arrivals, call completions, hand-off arrivals into and departures from a cell. Numerical algorithms are devised to solve for the steady state probabilities. Various performance measures like carried traffic, blocking and forced termination probabilities for each platform and call type are numerically computed from the analytical models.

The carried traffic in a wireless network can be increased by the graceful degradation of some or all of the existing services in the system [Sur96a, Sur96b]. Seal and Singh [Sur96b] have identified two QoS parameters namely, *graceful degradation of service* and *guarantee of seamless service.* Graceful degradation of service refers to reducing allocated bandwidth to the existing calls. The quality of each connection deteriorates as data is discarded by the base station transmitter to adjust to the reduced allocated bandwidth. It is highly possible that discarding of data results in the loss of some critical portions of data which may not be recoverable. With the help of user-supplied *loss profiles* which tell the system the user-preferred way to lose data, bandwidth usage of applications that can sustain loss is degraded in situations where user demands exceed the network's capacity to satisfy them. A new transport sub-layer called, loss profile transport sub-layer (LPTSL), is proposed to implement loss profiles by selectively discarding data out of special applications like a compressed video stream. This is implemented as a library of *discarding functions* which discard data in various manner (e.g., clustered loss, uniform loss etc.), and the user chooses the most appropriate function according to his needs. The algorithms have been incorporated in the Multi-stream Protocol (MSP) and MPEG-2 transport systems.

Based on the minimum requirement criteria provided by the user, Oliveira, Kim and Suda [OKS96] proposed a bandwidth reservation algorithm for guaranteeing QoS to multimedia traffic. Two main types of multimedia traffic was considered – real-time and non-real-time. For real-time traffic, the call is admitted only if the requested bandwidth can be reserved in the call-originating cell and all its neighbors. For a non-real-time call, the requested bandwidth is reserved only in the originating cell. Various approaches for bandwidth reservation for real-time traffic in neighboring cells were suggested. The amount of bandwidth reserved is either a function of the number of real-time calls or the requested bandwidth of all real-time connections in the cell. Detailed simulation experiments were performed to compare this scheme with two other variants. In one variant, the incoming real-time call is accepted if the requested bandwidth is available. If not, the algorithm attempts to allocate the minimum required bandwidth (provided by the call) only if it is a hand-off request. In the second variant, if this minimum required bandwidth is not available, bandwidth is "stolen" from the ongoing non-real-time calls and allocated to the hand-off request. Although this scheme guarantees QoS, the main drawbacks are: (*i*) bandwidth is reserved redundantly, since the user moves only to one of the six neighboring cells (assuming hexagonal cell geometry), and (*ii*) the stringent call admission procedure might not admit many real-time requests in a highly overloaded system.

Acampora and Naghshineh [AN94] proposed a virtual connection tree (a cluster of base stations) approach which is used for call routing, call admission and resource allocation. This concept is used to reduce the call set-up and routing load on the network call processor such that a large number of mobile connections can be supported. A *virtual connection tree* is a collection of base stations and wire-line network switching nodes and links, with the root being a fixed switching node. For setting up a call for a mobile terminal, the call processor allocates two sets of virtual connection numbers (one in each direction), with each member pair of the set defining a path between the root and a base station in the virtual connection tree. When the mobile

user wishes to hand-off to another BS in the same tree, it simply begins to transmit packets with its allocated connection number for the new BS. In the reverse direction, a hand-off is identified by an arriving packet bearing a new connection number. In both cases, the routing table at the root of the tree is updated accordingly. Whenever, the mobile user reaches the boundary of a virtual connection tree, it seeks hand-off to a new tree, and the network call processor is invoked. The network traffic is divided into three classes in decreasing order of priority. Class I comprises of real-time traffic like voice/video and has *hand-off dropping probability* as its QoS metric. Class II connections can be "put on hold" and will suffer packet losses and delay when the system enters an overloaded state. Class III connections utilize the leftover bandwidth of the other two classes of traffic and its QoS metric is *average queuing delay.* Call admission decision is based on the number of ongoing calls of each class, hand-off rate, call duration statistics etc. Acampora and Naghshineh [AN95] also proposed a call admission control algorithm for QoS provisioning for multimedia traffic, based on an adaptive resource sharing policy among two different classes of traffic, namely, real-time and non-real-time, with the former having preemptive priority over the latter. A simple analytical model (based on Poisson call arrivals and departures, and threshold rules for each class of traffic) is proposed to capture the various QoS parameters like call blocking and dropping probabilities and the probability for a minimum bandwidth availability for non-real-time requests which share the available bandwidth equally among themselves.

Some of the above related work, either experimentally evaluates QoS degradation [Sur96b] or model resource scheduling policies in a multi-class user network [AN95, RP96].

5.1.1. Our Contribution

In this Chapter, we propose a framework for modeling QoS degradation strategies for real-time and non-real-time traffic. Two orthogonal QoS parameters are identi-

fied, namely, *total carried traffic* and *bandwidth degradation*, such that improvement of either of them leads to the degradation of the other. A cost function describing the total revenue earned by the system from a *bandwidth degradation policy* is formulated, and methods to compute the optimal policy maximizing the net revenue is discussed. This model is extended to compute combined *call admission* and *degradation* policies, where the incoming calls are grouped into a demand vector. In most systems, the real-time traffic has preemptive priority over the non-real-time traffic, based on which a novel channel sharing scheme for non-real-time users is proposed. This scheme is analyzed using a Markov modulated Poisson process (MMPP) based queuing model, and the *average queue length*, a QoS metric for non-real-time traffic, is also derived from the model. Detailed simulation experiments are conducted to validate our proposed models and policies for real-time and non-real-time traffic.

The rest of this chapter is organized as follows. The framework for QoS degradation in case of real-time traffic is described in Section 5.2, along with some numerical results and detailed simulation results. In Section 5.3, we develop the MMPP model capturing the preemptive priority of real-time calls over non-real-time ones. Simulation results for the average queue length is also presented. A novel approach called *bandwidth compaction* to maximize spectrum utilization for multi-rate traffic requiring contiguous allocation of bandwidth is described in Section 5.4.

5.2. Graceful Degradation of User Traffic: the Real-Time Case

Various types of real-time multimedia services require different bandwidth requirements depending on the application. For example, low motion video telephony requires about 25 Kbps and generic video telephony over 40 Kbps, whereas a bandwidth of 9.6 Kbps is sufficient for voice services. In a practical wireless system, a *channel* is defined as a fixed block of communication medium such as (time slot, carrier frequency) tuple in TDMA systems or simply a fixed block of radio frequency bandwidth as in FDMA systems. A GSM channel has a data rate of 9.6 Kbps and

multiple such channels can be allocated to a single user application. For example, HSCSD allows up to 8 GSM channels to increase the bandwidth to 76.8 Kbps for transmitting high bandwidth video. The analysis presented below is applicable to both FDMA and TDMA based wireless systems unless otherwise mentioned.

5.2.1. A Framework for Bandwidth Degradation

Let us consider that there are K classes of traffic where a call of class i, for $1 \leq i \leq K$, uses i channels under normal operation. We also define a *degraded mode* of operation in which a call in class i $(2 \leq i \leq K)$ uses $i-1$ channels and releases a channel to a common pool. These released channels are used to accommodate new incoming calls and hence to increase the total carried traffic in the system. For example, if a video transmission in an HSCSD system using 76.8 Kbps is allowed to degrade by one channel (9.6 Kbps), then this released channel can accommodate one more voice call. Bandwidth degradation such as this, translates to poor Quality-of-Service for the call which may not be acceptable to the user. Therefore, all class $i > 1$ calls in the system may not be degradable.

In fact, there is a trade-off between the total carried traffic of various classes and the number of degraded calls in an saturated system. A *saturated system* is one in which there are no free channels available. In order to admit more incoming calls, channels have to be released through bandwidth degradation. While increased carried traffic generates positive revenues for the system, bandwidth degradation leads to a negative revenue generation. Therefore, we define two orthogonal QoS parameters – *carried traffic* and *bandwidth degradation* – and formulate a *cost function* based on these parameters to compute the net payoff generated out of bandwidth degradation.

A *bandwidth degradation policy*, $D_{bw} = \{y_2, y_3, \ldots, y_K\}$, specifies the number y_i of the ongoing class i calls to be degraded, where $2 \leq i \leq K$. We assume that the degradation of $(y_2 + y_3 + \ldots + y_K)$ number of calls take place sequentially in any arbitrary order. Thus the number of channels released by degradation of class i calls

is y_i. An optimal policy D^*_{bw} is one that maximizes the revenue cost function.

Let us assume a static distribution of various classes of calls (call mix) in an overloaded system with C channels where the fraction of class i calls is given by α_i, where $1 \leq i \leq K$. Then the number of class i calls in the system without degradation is $X_i = \frac{\alpha_i C}{\sum_{j=1}^{K} j\alpha_j}$. Let the system degrade with a policy $D_{bw} = \{y_2, y_3, \ldots, y_K\}$ where $y_i \leq X_i$, and the released channels are used to admit incoming calls of various classes.

Let $t_{c,i}(X_i, y_2, y_3, \ldots, y_K)$ denote the total number of admitted class i calls in the system and $C_{t,i}$, the revenue generated by each such call. Then the total revenue earned due to carried traffic in the degraded system is

$$\Phi_t = \sum_{i=1}^{K} C_{t,i} t_{c,i}(X_i, y_2, y_3, \ldots, y_K) \tag{5.1}$$

The system will also lose revenue proportionately to the number of degraded calls in the system. Let $t_{d,i}(y_2, y_3, \ldots, y_K)$ denote the number of class i calls in the system degraded in any arbitrary order. Degradation of a class i call (say, k) will have a cost $C_{d,i,k}$ associated with it. In general, the larger the number of degraded calls in the system the costlier it is to degrade a call. Therefore, $C_{d,i,k}$ will be proportional to the number of degraded class i calls in the system when call k is being degraded.

The number of degraded class i calls increases uniformly from 0 to $t_{d,i}(y_2, y_3, \ldots, y_K)$, so the average degradation cost for a class i call is proportional to $\frac{t_{d,i}(y_2, y_3, \ldots, y_K)}{2}$. Then the total cost due to the $t_{d,i}(y_2, y_3, \ldots, y_K)$ degraded class i calls is $\frac{C_{d,i} t_{d,i}(y_2, y_3, \ldots, y_K)}{2} t_{d,i}(y_2, y_3, \ldots, y_K)$, where the proportionality constant $C_{d,i}$ is the degradation cost per unit of class i traffic. Hence the total revenue loss due to the degradation of all classes of calls is

$$\Phi_d = \sum_{i=2}^{K} \frac{C_{d,i}}{2} (t_{d,i}(y_2, y_3, \ldots, y_K))^2 \tag{5.2}$$

Thus the effective revenue earned by the system from the bandwidth degradation policy D_{bw} is given by

$$\Phi(y_2, y_3, \ldots, y_K) = \Phi_t - \Phi_d \tag{5.3}$$

An optimal policy D^*_{bw} maximizes the revenue function Φ.

5.2.2. Derivation of the Revenue Function

In this section, expressions for $t_{c,i}(X_i, y_2, y_3, \ldots, y_K)$ and $t_{d,i}(y_2, y_3, \ldots, y_K)$ are derived, assuming the same static distribution $\{\alpha_1, \alpha_2, \ldots, \alpha_K\}$ of incoming calls using the channels released by bandwidth degradation. To reduce the complexity of the problem, we also assume that the system admits these new calls (excepting class 1 calls) to operate only in degraded mode. This assumption will be removed later.

It can be shown easily that under the static distribution of calls and using $\sum_{j=2}^{K} y_j$ released channels, the number of admitted class i calls is $\frac{\alpha_i \sum_{j=2}^{K} y_j}{\alpha_1 + \sum_{j=2}^{K}(j-1)\alpha_j}$, where $1 \leq i \leq K$. Hence

$$t_{c,i}(X_i, y_2, y_3, \ldots, y_K) = X_i + \frac{\alpha_i \sum_{j=2}^{K} y_j}{\alpha_1 + \sum_{j=2}^{K}(j-1)\alpha_j} \tag{5.4}$$

where $X_i = \frac{\alpha_i C}{\sum_{j=1}^{K} j\alpha_j}$.

Also, by the policy D_{bw}, the y_i calls of class $i > 1$ are already operating in the degraded mode. Hence, for $2 \leq i \leq K$,

$$t_{d,i}(y_2, y_3, \ldots, y_K) = y_i + \frac{\alpha_i \sum_{j=2}^{K} y_j}{\alpha_1 + \sum_{j=2}^{K}(j-1)\alpha_j} \tag{5.5}$$

The complete form of the revenue cost function is

$$\Phi = \sum_{i=1}^{K}\{X_i + \frac{\alpha_i \sum_{j=2}^{K} y_j}{\alpha_1 + \sum_{j=2}^{K}(j-1)\alpha_j}\}C_{t,i} - \sum_{i=2}^{K}\{y_i + \frac{\alpha_i \sum_{j=2}^{K} y_j}{\alpha_1 + \sum_{j=2}^{K}(j-1)\alpha_j}\}^2 \frac{C_{d,i}}{2} \tag{5.6}$$

where $y_i \leq X_i$ for $2 \leq i \leq K$.

Our objective is to find an optimal policy $D_{bw}^* = \{y_2, y_3, \ldots, y_K\}$ which maximizes $\Phi(y_2, y_3, \ldots, y_K)$ under the constraints $y_i \leq X_i$ for $2 \leq i \leq K$. This is equivalent to a $(K-1)$-dimensional constrained optimization problem which can be solved by Lagrange's multiplier method [GW73]. The procedure for finding the extrema for such a function is as follows. First we ignore the inequality constraints and proceed to optimize the given function. If the solution obtained satisfies the inequality constraints,

then the extremum lies within the boundaries of the constrained region. Otherwise we change the inequality constraints to equality constraints one by one and apply Lagrangian Multiplier's technique until the remaining inequality constraints are also satisfied.

It is appropriate to mention here that the estimation of the parameters, $C_{t,i}$ and $C_{d,i}$, will be driven mainly by market research and the third generation system deployment issues. For this, the various application services have to be defined and their market penetration, calculated. This is a separate research issue by itself and beyond the scope of this work.

Illustrative Example

Equation (5.6) is considered for $K = 3$ leading to the following 2-dimensional constrained optimization problem for three classes of traffic.

Maximize $\Phi(y_2, y_3) = \{X_1 + \frac{\alpha_1(y_2+y_3)}{A}\}C_{t,1} + \{X_2 + \frac{\alpha_2(y_2+y_3)}{A}\}C_{t,2} + \{X_3 + \frac{\alpha_3(y_2+y_3)}{A}\}C_{t,3} - \frac{C_{d,2}}{2}\{y_2 + \frac{\alpha_2(y_2+y_3)}{A}\}^2 - \frac{C_{d,3}}{2}\{y_3 + \frac{\alpha_3(y_2+y_3)}{A}\}^2$

where $y_2 \leq X_2$ and $y_3 \leq X_3$, and $A = \alpha_1 + \alpha_2 + 2\alpha_3$.

The unconstrained optimization problem is solved by setting $\frac{\partial\Phi(y_2,y_3)}{\partial y_2} = 0 = \frac{\partial\Phi(y_2,y_3)}{\partial y_3}$. The function $\Phi(y_2, y_3)$ is maximized at the surface point (y_2, y_3) given by

$$y_3 = \frac{(\alpha_1 C_{t,1} + \alpha_2 C_{t,2} + \alpha_3 C_{t,3})/A}{\{C_{d,2}(1 + \frac{\alpha_2}{A})^2 + C_{d,3}(\frac{\alpha_3}{A})^2\}\Gamma + \{\frac{\alpha_2 C_{d,2}}{A}(1 + \frac{\alpha_2}{A}) + \frac{\alpha_3 C_{d,3}}{A}(1 + \frac{\alpha_3}{A})\}} \tag{5.7}$$

$$y_2 = \Gamma y_3 \tag{5.8}$$

where $\Gamma = \frac{C_{d,3}(1+\frac{\alpha_3}{A}) - C_{d,2}\frac{\alpha_2}{A}}{C_{d,2}(1+\frac{\alpha_2}{A}) - C_{d,3}\frac{\alpha_3}{A}}$.

The function $\Phi(y_2, y_3)$ is plotted for $\mathcal{C} = 50, \alpha_1 = 0.5, \alpha_2 = 0.3, \alpha_3 = 0.2$ and assuming the cost constants $C_{t,1} = 10$ units, $C_{t,2} = 20$ units, $C_{t,3} = 40$ units, and $C_{d,2} = 1$ unit, $C_{d,3} = 2$ units. Among the three classes, class 1 is assumed to constitute

of voice calls, and classes 2 and 3 constitute of data and low speed video. In spite of the rapid popularity of multimedia applications, the predominant service in the next generation wireless networks will still be voice, hence, 50% of the traffic is assumed to be generated by voice users. The assumptions for cost parameters are based on the fact that both the revenue per call and the degradation cost will be proportionately higher for higher bandwidth traffic. The number of class 2 and class 3 calls in the system at the time of degradation is $X_2 = 9$ and $X_3 = 6$, respectively. Using Equations (5.7) and (5.8), it is seen that $\Phi(y_2, y_3)$ is maximized at the point $(y_2 = 8.2, y_3 = 3.6)$, which satisfy the inequality constraints for the given set of parameters. This implies that a degradation of eight out of the nine class 2 calls and three out of six class 3 calls will generate the maximum effective revenue for the system.

5.2.3. Undegraded Admissions and Degradation Constraints

In the last section, we have followed the strategy that all incoming calls (except class 1) scheduled with the channels released by degrading existing calls, will themselves operate in the degraded mode. The maximum revenue generated can be obtained from Equation (5.6) under this strategy which, however, may not be optimal. Allowing the incoming calls the freedom to operate either in a degraded or undegraded mode, the optimal revenue can be attained. We introduce two additional variables y_{id} and y_{in} $(2 \leq i \leq K)$ denoting respectively the number of released channels from class i calls used by the incoming traffic in degraded and non-degraded modes. This increases the dimensionality of the revenue function by $2(K-1)$.

Since an incoming class 1 call cannot operate in degraded mode, it can only access the y_{in} channels released by each class i. Other classes can access all the released channels. Thus the revenue function can be rewritten as

$$\Phi = (X_1 + \frac{\alpha_1 \sum_{j=2}^{K} y_{jn}}{\sum_{j=1}^{K} j\alpha_j})C_{t,1} + \sum_{i=2}^{K}(X_i + \frac{\alpha_i \sum_{j=2}^{K} y_{jn}}{\sum_{j=1}^{K} j\alpha_j} + \frac{\alpha_i' \sum_{j=2}^{K} y_{jd}}{\sum_{j=2}^{K}(j-1)\alpha_j'})C_{t,i} - \sum_{i=2}^{K} \frac{C_{d,i}}{2}(y_i + \frac{\alpha_i' \sum_{j=2}^{K} y_{jd}}{\sum_{j=2}^{K}(j-1)\alpha_j'})^2 \quad (5.9)$$

under the constraints:

(i) $y_i = y_{in} + y_{id}$ and **(ii)** $y_i \leq X_i$ for $2 \leq i \leq K$.

Therefore, the number of degrees of freedom of the $3(K-1)$-dimensional function Φ is $2(K-1)$. Here the parameters α_i', $2 \leq i \leq K$, describe the new distribution of the incoming calls from all classes except class 1, using the $\sum_{i=2}^{K} y_{id}$ released channels.

As observed earlier, not all higher bandwidth calls in the system are capable of undergoing degradation. Consider, for example, the case where bandwidth degradation leads to discarding data in a compressed video stream. The receiver may not always be able to de-compress the data from the received information. Bandwidth degradation is prohibitive in such a situation. If the distribution of such calls among a class of traffic can be estimated, then the allowable number of call degradations in that class can be restricted accordingly. Let $\delta_i \leq 1$ denote the fraction of ongoing calls allowed to degrade in a certain class i. For this case of *constrained degradation* among each class of traffic, the revenue function Φ is also given by Equation (5.9) with constraint **(ii)** modified as $y_i \leq \delta_i X_i$. The problem is again amenable to solution by Lagrange's multiplier method [GW73] in a similar way as described earlier.

5.2.4. Degradation and Admission Policies for a given User Demand

In the previous sections, we assumed a static mix of various classes of calls in the system and that the incoming calls follow the same mix. In this section, we remove that assumption and consider a batch arrival scenario where all the incoming calls are periodically grouped into a *demand vector* and admitted into the system through an optimization cost function similar to that derived in the previous section. Once a call arrives, the system waits for an amount of time equal to the permissible delay in admitting that call, before computing the *degradation* and *admission policies*. Other calls which arrive during the waiting period are included in the demand vector. If we assume that no waiting time is permissible for any call, the demand vectors will only consist of a single request.

Let us denote the demand vector as $\mathcal{D} = \{d_1, d_2, \cdots, d_K\}$, where d_i is the number

of requests of class i. Allocation to all the requests is possible without any degradation if the following condition is satisfied

$$\mathcal{C} - \sum_{i=1}^{K} iX_i \geq \sum_{i=1}^{K} id_i \tag{5.10}$$

which means that the demand is less than the number of available channels.

If allocation of bandwidth to all the requests is not possible without degradation, then let us denote by $\mathcal{A} = \{a_1, a_2, \cdots, a_K\}$ an allocation vector to be computed, where a_i denotes the number of class i calls admitted. We also compute a vector $\Gamma = \{\gamma_2, \gamma_3, \cdots, \gamma_K\}$, where γ_i denotes the fraction of incoming class i calls to be admitted in degraded mode. Recall that the degradation policy $D_{bw} = \{y_1, y_2, \cdots, y_K\}$ is a vector denoting the number of ongoing calls to be degraded. The objective is to compute a solution vector $\mathcal{S} = \{\mathcal{A}, D_{bw}, \Gamma\}$, which maximizes the following reward function

$$\Phi = \sum_{i=1}^{K} a_i C_{t,i} - \sum_{i=2}^{K} (y_i + \gamma_i a_i)^2 \frac{C_{d,i}}{2} \tag{5.11}$$

while satisfying the constraint that,

$$a_1 + \sum_{i=2}^{K} \{(i-1)\gamma_i a_i + i(1-\gamma_i)a_i\} \leq \mathcal{C} - \sum_{i=1}^{K} iX_i + \sum_{i=2}^{K} y_i \tag{5.12}$$

The condition given by Equation (5.12) ensures that the leftover bandwidth and the bandwidth released by degrading some of the existing calls (specified by the degradation policy) are fully utilized to accommodate new incoming calls, some of which will be admitted in degraded mode (γ_i denotes the fraction of class i calls admitted in degraded mode). The obvious constraints on the variables y_i, a_i and γ_i are $y_i \leq X_i$, $a_i \leq d_i$ and $\gamma_i \leq 1$. If some of the X_i ongoing calls are not allowed to degrade, the constraint will be changed to $y_i \leq X_i'$, where, $X_i' < X_i$. Similarly, if some of the incoming calls of class i are not allowed to degrade, γ_i will have a constraint value less than 1.

This is a $(3K-2)$-dimensional constrained optimization problem. The solution vector $\mathcal{S}$ determines a combined allocation and degradation policy which maximizes

the utilization of the wireless bandwidth from the point of view of the net revenue earned by the system.

5.2.5. Experimental Results

Simulation experiments are carried out for 5 classes of user traffic. In a practical system like GPRS, the maximum number of different classes of bandwidth that can be allocated to users is 8, but the different types of multimedia services will hardly exceed 5. The maximum number of full-duplex radio-frequency (RF) channels supported by any cell is assumed to be 50. In our simulation model, the maximum number of channels required by any service is 5. We assumed that class 1 service will be voice requiring 1 channel, class 2 is low speed data requiring 2 channels on the average, and the classes 3, 4 and 5 are for various types of high speed data and video services. A static call mix for the different types of services is assumed. Again, there is no practical evidence to support the call mix because some of these services are futuristic, but it is predicted that even for the third generation wireless multimedia, voice and data services will predominate. On an average, we assumed around 50% of the traffic will be voice calls, 20% data and the rest equally distributed among the other three service classes.

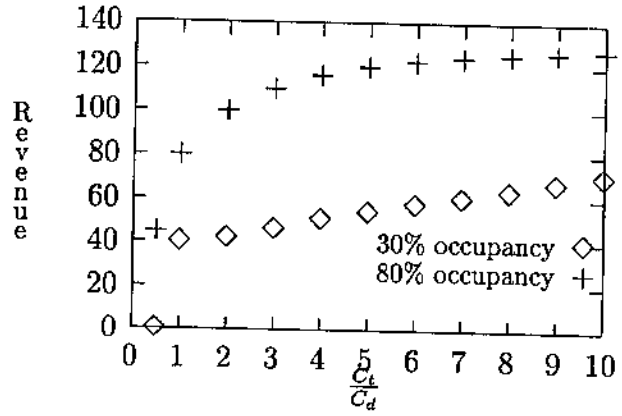


Figure 5.2: Net revenue earned with various ratios of $\frac{C_t}{C_d}$

In the absence of accurate estimates for the quantities $C_{t,i}$'s and $C_{d,i}$'s from a real system, a set of values for $C_{t,i}$'s is assumed and experiments are conducted for various ratios of $C_{t,i}$ to $C_{d,i}$, each ratio assumed to be the same for all classes. We primarily focus on call degradation and admission policies in a single cell.

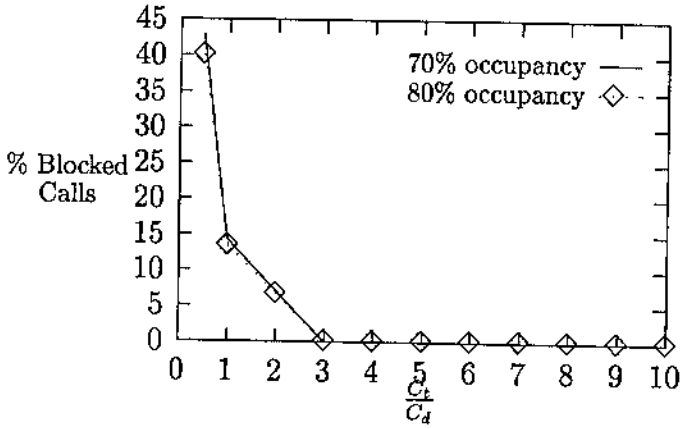


Figure 5.3: Percentage of blocked calls with various ratios of $\frac{C_t}{C_d}$

The call arrival patterns in a cell are generally distinctive in nature. Generally, the cell traffic peaks around mid-day or late afternoon and the average load at that time stays around 70-80% of the cell capacity. In other parts of the day, the typical load varies from 5% to 30% of the capacity. The following experiments were conducted with these types of load values.

Figure 5.2 shows the effective revenue earned by the system from the degradation and call admission policies using our scheme. The revenue earned per unit call is assumed as $C_{t,i} = 5i$ where $1 \leq i \leq 5$. The ratio $\frac{C_{t,i}}{C_{d,i}}$ is assumed the same for all classes and varied from 0.5 to 10. The total number of channels occupied at the time of running our algorithm is denoted by ch_occupancy. The interesting observation is that the net revenue is very sensitive to changes in the $\frac{C_t}{C_d}$ ratio when $C_t \leq C_d$. The revenue progressively increases as the ratio increases. Another observation is that the revenue earned is higher for a highly loaded system with 80% occupancy than

a moderate 30% loaded system. This is because of our assumption of almost the same call mix for incoming calls as the present mix of ongoing calls in the system. This mix is observed to change quite slowly compared to the periodicity of our call degradation/ call admission strategy, which is of the order of hundreds of seconds. Hence, an already overloaded system is expected to receive a high number of calls, and our degradation and admission policies ensure that they are admitted in such a way that maximum revenue is earned by the system.

Figure 5.3 shows the variation of % blocked calls with the ratio $\frac{C_t}{C_d}$ for a highly loaded system with 70% and 80% channel occupancies. The variation is, however, very slight between the two cases. For nearly equal values of C_t and C_d, call blockade is quite high. For relatively low degradation costs, there is provision of degrading more calls to accommodate new ones and hence % blockade sharply decreases. There is almost 0% call blockade for $\frac{C_t}{C_d} \geq 3$. For channel occupancies less than 70%, the call blocking is almost zero. Thus we conclude that for the bandwidth degradation strategy to be fully utilized, the system must ensure that the *ratio of net revenue per call to the cost of degrading the call is at least 3.*

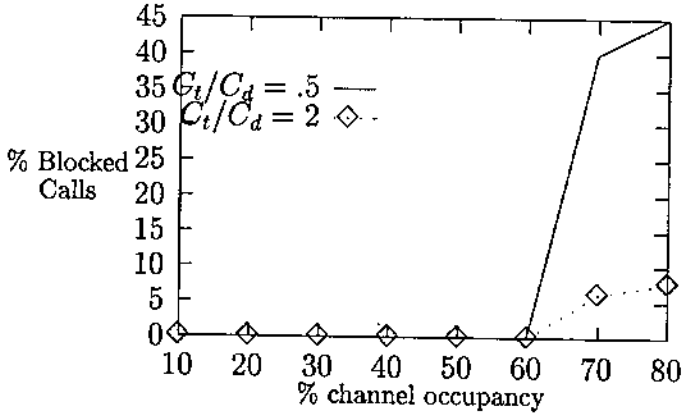


Figure 5.4: Percentage of blocked calls with various channel occupancy

Figure 5.4 shows call blockade percentage with various system loads (channel occupancies). Till about 60% channel occupancy, there is insignificant call blockade.

For higher system load, the call blockade % increases depending on the $\frac{C_t}{C_d}$ ratio. There is about 45% call blockade for $\frac{C_t}{C_d} = 0.5$, but it decreases to 7% when $\frac{C_t}{C_d} = 2$. This shows that the system performance is highly sensitive to the proper choice of the $\frac{C_t}{C_d}$ ratio and the system should be engineered accordingly.

5.3. Graceful Degradation of Non-Real-Time Traffic

Real-time traffic for voice and video is unacceptable to the user perception if the end-to-end delay is greater than 200 msec. Since non-real-time traffic packets can sustain much longer delays, real-time traffic usually has preemptive priority over the former in case of scarcity of available channels in the system. The preempted non-real-time traffic are buffered for future scheduling to the available channels or when free channels become available. Let us describe how this simple idea will be implemented in a TDMA/FDMA based wireless system.

Wireless systems use circuit switched dedicated channels between the mobile user and the base station or MSC in order to send data or voice packets. A contention-less multiple access protocol is required for certain types of services because of their real-time requirements, where a dedicated channel or bandwidth is allocated to the user for the entire duration of its call. For non-real-time packet services, assuming that a large but finite amount of delay is tolerable, the free channels (after allocating the required number of channels to every real-time user) can be equally shared by the receivers [AN95]. Next, we propose and analyze a new channel sharing scheme.

One approach for *channel sharing* is used by the CDPD, in which a non-real-time data user *hops* to a different free channel when there is a voice (real-time) call in the currently occupied channel. The disadvantage of this scheme is that the available resources (channels) are not utilized uniformly and a data call gets blocked when there is no available channel. In our proposed strategy, all the non-real-time calls share the available channels equally, in a synchronized manner. For this purpose, it is assumed that each channel has a buffer associated with it to store incoming packets

of the non-real-time calls. This queue is used only by the arriving non-real-time packets being serviced by that channel. If the channel is servicing a real-time call, its queue remains empty. The incoming packets of the non-real-time calls are equally distributed among these queues. Such a distribution ensures that all the available channels are always fully utilized.

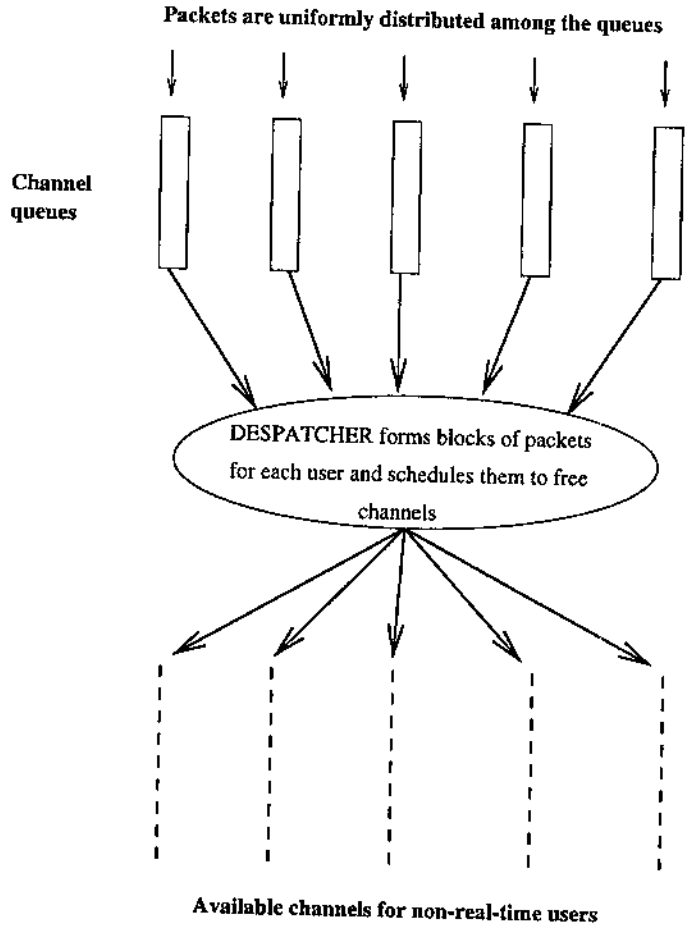


Figure 5.5: A channel sharing scheme for non-real-time users

The problem now is to dispatch the packets for multiple users residing in a queue to the corresponding users using the single channel associated with the queue. For this purpose, there is a *despatcher* function (see Figure 5.5) which divides the packets for each user in the queue into equal sized blocks. The despatcher constructs these

packet blocks for each user from the data arriving in the queue and schedules them, in a synchronous manner, to be transmitted to the user through the channel. Block transmission for all the channels being used by non-real-time users begins and ends at the same time. In case of a transmission error, the entire block is scheduled for retransmission at a later time. For the duration of the block transmission, the user is linked to the base station through a particular channel and may have to re-tune to a different channel to receive the next data block. In that case, the new channel id is sent out to the user in the last message of the current block.

When an user, U, does not have any block of data to receive, he is asked by the system in the last message of the current block, to drop the connection. The released channel can then be used to transmit blocks to other non-real-time users. When sufficient data for the user U has arrived in the queue(s) of one (or more) channel(s) to form block(s), U is paged and connection is re-established with the base station. In spite of the possibility that this scheme may lead to some additional paging, the main advantages are as follows :

- A large number of non-real-time users can be accommodated even with a small number of available channels and finite delay is guaranteed provided a fair despatcher exists.

- The activities of the despatcher can be overlapped in order to ensure maximum possible channel utilization. For example, when it is scheduling the block of user 1, it will construct the block for user 2, to be scheduled next.

In order to reduce the amount of paging, the user U, when asked to drop connection, will enter a *doze* mode where he will not occupy any channel but will update whenever he enters a new cell. In that way, whenever the system acquires some new blocks to be sent out to U, he is paged only in one cell.

With this channel sharing mechanism in mind for the non-real-time users, let us formulate a QoS parameter.

5.3.1. A QoS Parameter for Non-Real-Time Users

A suitable QoS parameter describing the performance of non-real-time traffic in a cellular mobile system is the *average queue length* for a channel queue. A gracefully degrading cellular mobile system will preempt the ongoing non-real-time users in case of arrival of a real-time call and release channels being used by the non-real-time calls. The non-real-time calls will be rescheduled later using the available bandwidth. As the non-real-time communications are preempted and directed towards the queues being used by other non-real-time communications, the average queue length (and hence, the queuing delay) for these packets increases. In the analysis that follows for estimating the average queue length for non-real-time packets, the following system related assumptions are made:

1. The total number of channels in the base station is C and both real-time and non-real-time traffic use a single channel.

2. Real-time packets are never buffered and always allocated a channel if one is available or being used by non-real-time traffic. Hence, the real-time traffic has not only preemptive priority over non-real-time traffic but uses a contentionless multiple access scheme where each user communicates via a single channel for the entire call duration.

3. The non-real-time users, on the other hand, may use different channels at different points of time (based on channel availability) as discussed in the previous subsection.

4. When a non-real-time call is preempted from a channel, the system distributes the incoming packets (from the queue associated with the channel) equally among the queues servicing other ongoing calls of similar type (non-real-time).

5.3.2. Queuing Analysis

Let the call arrivals and departures for real-time traffic be Poisson with mean rates λ_R and μ_R, respectively. For the real-time calls, the system follows an $M/M/C/C$ queuing discipline, leading to the simple birth and death Markov chain as shown in Figure 5.6. Let us also assume that there is always high traffic overload for non-real-time calls and whenever a real-time call arrives, it preempts an ongoing non-real-time communication. Suppose the packet arrival and departure rates for non-real-time traffic are Poisson distributed with mean λ_{PNR} and μ_{PNR}, respectively.

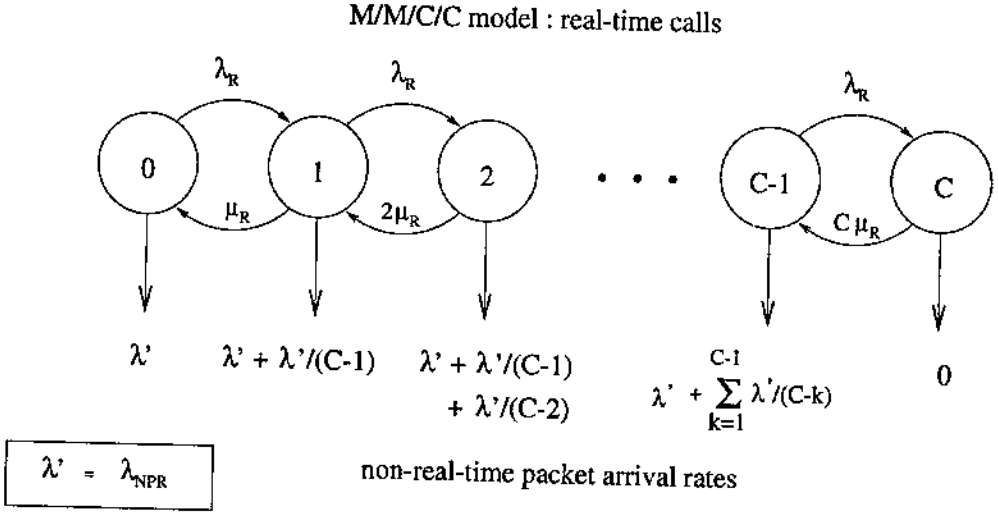


Figure 5.6: A Markov Modulated Poisson Process (MMPP)

As shown in Figure 5.6, with no real-time calls in the system, the packet arrival rate for a queue is λ_{PNR}. With the arrival of a single real-time call, the packet arrival rate in an active queue increases by $\frac{\lambda_{PNR}}{C-1}$. For i real-time calls in the system, the packet arrival rate in an active queue is $\lambda_{PNR} + \sum_{k=1}^{i} \frac{\lambda_{PNR}}{C-k}$. Since, on preemption, a Poisson stream of arriving packets is split among all other active queues in the system with equal probability, the new arrival process for an active queue is also Poisson [NEL96]. The fact that the mean arrival rate for non-real-time packets in

any active queue is dependent on the state of the Markov chain for the real-time call arrival and departure process, leads to a Markov modulated Poisson process with the two dimensional state space $\langle i, j \rangle$, where i denotes the number of non-real-time packets in a queue and j denotes the number of ongoing real-time calls. An absence of packets in a queue is possible with all C channels occupied by real-time traffic, so $0 \leq j \leq C$ for $i = 0$ and $0 \leq j \leq C - 1$ for $i > 0$. Let $p_{i,j}$ be the joint probability that i packets are in the queue with j ongoing real-time calls.

Solving this system using traditional matrix-geometric solution techniques [NEL96] leads to a compound matrix equation of the form

$$\mathbf{p} = \mathbf{pT} \tag{5.13}$$

where (i) $\mathbf{p} = [\mathbf{p_0}, \mathbf{p_1}, \mathbf{p_2}, \cdots, \mathbf{p_i}, \cdots]$ and $\mathbf{p_i}$ is a C-element row vector defined as $\mathbf{p_i} = [p_{i,0}, p_{i,1}, \cdots, p_{i,C-1}]$, and (ii) $\mathbf{T}$ is a compound tri-diagonal matrix of the form

$$\mathbf{T} = \begin{bmatrix} \mathbf{B_0} & \mathbf{A_0} & 0 & 0 & \cdots \\ \mathbf{B_1} & \mathbf{A_1} & \mathbf{A_0} & 0 & \cdots \\ 0 & \mathbf{A_2} & \mathbf{A_1} & \mathbf{A_0} & \cdots \\ 0 & 0 & \mathbf{A_2} & \mathbf{A_1} & \cdots \\ 0 & 0 & 0 & \mathbf{A_2} & \cdots \\ \cdot & \cdot & \cdot & \cdot & \cdots \end{bmatrix} \tag{5.14}$$

The matrices $\mathbf{B_0}, \mathbf{B_1}, \mathbf{A_0}, \mathbf{A_1}$ and $\mathbf{A_2}$ are given below.

$\mathbf{B_0}$ is an $C \times C$ tri-diagonal matrix given by

$$\mathbf{B_0} =$$

$$\begin{bmatrix}
1-\lambda_{PNR}-\lambda_R & \lambda_R & 0 & 0 & . & . & . & 0 \\
\mu_R & 1-\lambda_R-\mu_R-\lambda_{PNR} & \lambda_R & 0 & . & . & . & 0 \\
 & -\frac{\lambda_{PNR}}{\mathcal{C}-1} & & & & & & \\
2\mu_R & 1-\lambda_R-2\mu_R-\lambda_{PNR} & \lambda_R & . & . & . & 0 & \\
 & -\sum_{k=1}^{2}\frac{\lambda_{PNR}}{\mathcal{C}-k} & & & & & & \\
 & \lambda_R & . & . & . & 0 & & \\
. & . & . & . & . & . & . & . \\
0 & 0 & 0 & 0 & 0 & 0 & (\mathcal{C}-1)\mu_R & 1-(\mathcal{C}-1)\mu_R-\lambda_{PNR} \\
 & & & & & & & -\sum_{k=1}^{\mathcal{C}-1}\frac{\lambda_{PNR}}{\mathcal{C}-k}
\end{bmatrix}$$

$\mathbf{B_1}$ and $\mathbf{A_2}$ are $\mathcal{C} \times \mathcal{C}$ diagonal matrices given by

$$diag\{\mu_{PNR}, \mu_{PNR} \cdots \mu_{PNR}\}$$

$\mathbf{A_0}$ is an $\mathcal{C} \times \mathcal{C}$ diagonal matrix given by

$$diag\{\lambda_{PNR}, \lambda_{PNR} + \tfrac{\lambda_{PNR}}{\mathcal{C}-1}, \cdots, \lambda_{PNR} + \textstyle\sum_{k=1}^{\mathcal{C}-1}\frac{\lambda_{PNR}}{\mathcal{C}-k}\}$$

$\mathbf{A_1}$ is an $\mathcal{C} \times \mathcal{C}$ tri-diagonal matrix given by

$$\mathbf{A_1} =$$

$$\begin{bmatrix}
1-\mu_{PNR}-\lambda_{PNR}-\lambda_R & \lambda_R & 0 & 0 & . & . & . & 0 \\
\mu_R & 1-\mu_{PNR}-\lambda_R-\mu_R & \lambda_R & 0 & . & . & . & 0 \\
 & -\lambda_{PNR}-\frac{\lambda_{PNR}}{\mathcal{C}-1} & & & & & & \\
2\mu_R & 1-\mu_{PNR}-\lambda_R-2\mu_R & \lambda_R & . & . & . & 0 & \\
 & -\lambda_{PNR}-\sum_{k=1}^{2}\frac{\lambda_{PNR}}{\mathcal{C}-k} & & & & & & \\
. & . & . & . & . & . & . & . \\
0 & 0 & 0 & 0 & 0 & 0 & (\mathcal{C}-1)\mu_R & 1-\mu_{PNR}-(\mathcal{C}-1)\mu_R \\
 & & & & & & & -\lambda_{PNR}-\sum_{k=1}^{\mathcal{C}-1}\frac{\lambda_{PNR}}{\mathcal{C}-k}
\end{bmatrix}$$

Following [NEL96], the desired solution for the $\mathcal{C}$-element row vector, $\mathbf{p_i}$, is given by the recursive matrix equation

$$\mathbf{p_i} = \mathbf{p_{i-1}}\mathbf{R} \tag{5.15}$$

where $\mathbf{R}$ is the minimal non-negative solution to the matrix equation $\mathbf{R} = \mathbf{A_0} + \mathbf{R}\mathbf{A_1} + \mathbf{R^2}\mathbf{A_2}$. From Equations (5.13) and (5.15), the $\mathcal{C}$-element vector $\mathbf{p_0}$ is obtained as the eigen-vector solution of the equation $\mathbf{p_0} = \mathbf{p_0}(\mathbf{B_0} + \mathbf{R}\mathbf{B_1})$. Knowing $\mathbf{p_0}$ and $\mathbf{R}$, the vectors

p_i can be computed. The marginal probability distribution of the queue lengths is then given as

$$p_i = \begin{cases} \sum_{j=0}^{C} p_{i,j} & : \quad i = 0 \\ \sum_{j=0}^{C-1} p_{i,j} & : \quad i > 0 \end{cases}$$

The average queue length is given by $L_{avg} = \sum_{i=0}^{\infty} i p_i$.

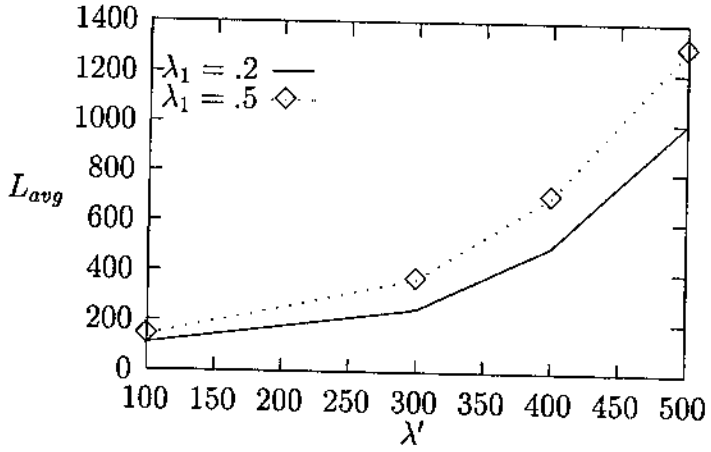


Figure 5.7: L_{avg} versus non-real-time packet arrival rate. $\lambda_1 \equiv \lambda_R$

Figure 5.7 shows the average queue length, L_{avg}, for non-real-time traffic for various values of packet arrival rates, λ_{PNR}. As expected, L_{avg} increases with a progressively faster rate with λ_{PNR}. It is also observed that as the call arrival rate λ_R of real-time traffic increases from 0.2 to 0.5, the average queue length increases. As the real-time calls arrive at a faster rate, more non-real-time calls are preempted increasing the average queue length. This clearly demonstrates the QoS degradation of the non-real-time traffic in the presence of the real-time traffic.

5.3.3. Experimental Results

A real-time and non-real-time user traffic profile is simulated in a particular cell. The maximum channel capacity in the cell is assumed to be $C = 50$. The call arrival process for real-time and non-real-time calls are both Poisson with different rates λ_R and λ_{NR}, respectively. Within the call holding period of the non-real-time calls, the packet arrival process is also assumed to be Poisson with rate λ_{PNR}. It is also assumed that $\lambda_{PNR} \gg \lambda_R, \lambda_{NR}$.

Each channel is assumed to have a bandwidth of 10 Kbps and the packet size is 500 bits. Hence, the channel packet rate is 20 packets/sec. A queue is assumed to be associated with each channel. As discussed previously, the packets are distributed evenly among the queues associated with the available channels for non-real-time calls and, hence, the packet rate is dependent on the number of real-time calls in the system. Since all the queues will have similar packet arrival rate, the average queue length for a particular queue is studied in the simulation experiment.

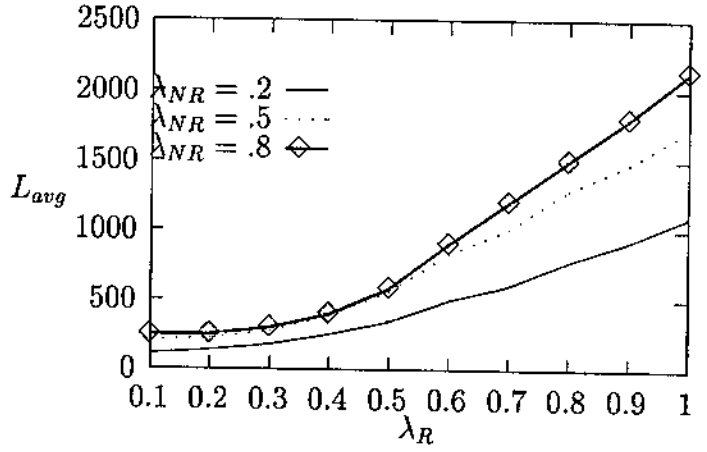


Figure 5.8: Average queue length versus real-time call arrival rate

Figure 5.8 shows the variation of average queue length, L_{avg}, with the arrival rate λ_R of real-time calls which is varied from 0.1 to 1 call per second. The packet arrival rate is assumed to be 500 packets/sec for all non-real-time calls. The average service

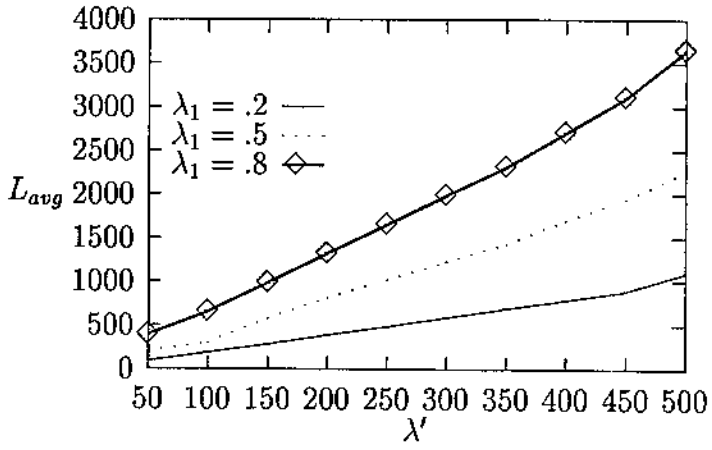


Figure 5.9: Average queue length versus packet arrival rate for nonreal-time calls. $\lambda_1 \equiv \lambda_R$

rate for both real and non-real time calls is assumed to be $\mu_R = \mu_{NR} = 0.05$ in our simulation.

Three curves are shown in Figure 5.8 for three different values of non-real-time call arrival rate λ_{NR}. It is observed that for low values of λ_R, the effect of increase in non-real-time call arrival rate is small on the average queue length. For $\lambda_R > 0.5$, the effect of change in λ_{NR} is more noticeable and the rate of increase of L_{avg} is higher. Increase in the values of λ_R and λ_{NR} leads to the same effect – increase in the packet arrival rate in the queue. Hence, higher the values of both these parameters, the faster is the rate of build-up of the queue, *i.e.*, the greater is the degradation.

Figure 5.9 shows the variation of L_{avg} with the packet arrival rate λ_{PNR} for non-real-time calls for various values of λ_R, when $\lambda_{NR} = 0.3$. It is observed that L_{avg} increases with a greater rate as λ_{PNR} increases. Also, the rate increases as the value of λ_R increases, and the curves spread apart for higher values of λ_{PNR} and λ_R. Comparing the curves for $\lambda_R = .2$ and .5 in Figure 5.9 with those in Figure 5.7 obtained from analytical model, we observe that our analytical model very closely follows the simulation results.

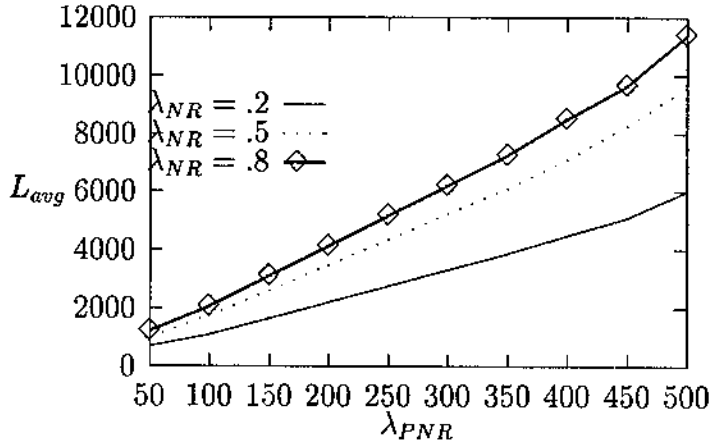


Figure 5.10: Average queue length versus packet arrival rate for nonreal-time calls

Figure 5.10 shows the variation of L_{avg} with the packet arrival rate λ_{PNR} for non-real-time calls for various values of λ_{NR}, and with $\lambda_R = 0.9$. The curves follow the same trend as with λ_R as parameter (see Figure 5.9), but the average queue lengths are observed to be higher in this case because of the higher call arrival rate of real-time traffic.

5.4. Bandwidth Compaction : A Technique for Maximizing Spectrum Utilization for Multi-rate Real-time and Nonreal-time Traffic

As discussed earlier, a *channel* is defined to be a fixed block of communication medium such as ⟨time slot, carrier frequency⟩ tuple in TDMA systems or simply a fixed block of radio frequency as in FDMA systems. Multiple channels can be allocated to a single user to satisfy high bandwidth requirements, e.g. HSCSD allows up to 8 time slots per carrier to transmit high bandwidth video. Let us define a *bandwidth segment* as a block of radio frequency bandwidth constituting a variable number of channels in a real system (e.g. 1-8 time slots/carrier for HSCSD). A fixed number of channels (e.g., one , in the simplest case) defines a *bandwidth page.*

Given the concepts of bandwidth segments and pages, let us now describe a simple

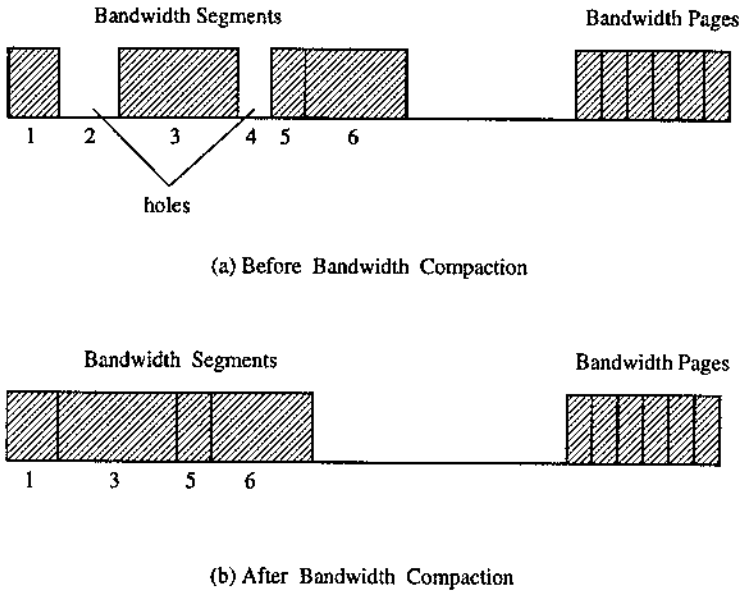


Figure 5.11: Bandwidth allocation pattern showing segments and pages

call admission algorithm (CAA) for real-time and non-real time traffic (please refer to Fig. 5.11). The originator of real-time traffic provides a minimum bandwidth requirement for that service. Based on that, a bandwidth segment is allocated by the system. Bandwidth pages will be allocated to non-real time traffic successively from the rear end of the spectrum allocated to the cell. The system will have the provision of "removing" a bandwidth page depending on the traffic load. This means that the non-real time communication using the bandwidth page is temporarily buffered. Bandwidth allocation in this fashion ensures that the segments and pages form two disjoint sets for efficient utilization of the spectrum. The "page size" can vary dynamically depending on non-real-time traffic load and the maximum tolerable delay of the system.

Since real-time users are assigned bandwidth segments, there can be unutilized "holes" in the frequency spectrum caused by call terminations. For example, in Figure 5.11, call terminations for users 2 and 4 lead to the creation of unutilized bandwidth "holes". In a system requiring contiguous allocation of bandwidth for

multimedia traffic, such a hole can only be filled again by a call request of equal or smaller bandwidth size. We describe below a method called *bandwidth compaction* for efficient utilization of the available spectrum in such a system.

The idea behind bandwidth compaction is to shuffle the ongoing communications to place all free bandwidth together in one contiguous block. For example, the spectrum with holes as in Figure 5.11(a) becomes as in Figure 5.11(b) after compaction.

Moving a bandwidth segment to a different part in the frequency spectrum implies re-tuning the channels (constituting the segment) to a different set of carrier frequencies, also known as *carrier re-assignments*. The bandwidth compaction algorithm will sequentially shuffle the existing bandwidth segments towards the lower frequency side to absorb the holes, requiring the communicating users to tune to new carrier frequencies almost instantaneously. This scheme can sometimes be very expensive and time consuming as a large number of channel re-assignments needs to be done.

A real-time traffic request will generally not tolerate long delays to receive acknowledgment. Hence, a *partial compaction* of bandwidth will be executed during the call admission phase. The partial compaction scheme checks the spectrum allocation pattern to determine if by removing a few (in the range of 2 or 3) bandwidth segments to adjoining holes, sufficient contiguous spectrum can be released in order to satisfy the user request. This scheme only involves very few channel reassignments and sustains lower delays.

5.4.1. Experimental Results

A sequential simulation experiment is conducted to evaluate the effectiveness of our bandwidth compaction algorithm against a bandwidth allocation scheme without any means of spectrum management. The QoS parameter used for the comparison is called *bandwidth utilization*, γ, defined as the ratio of the average amount of bandwidth in active use to the total amount allocated to the system.

Five classes of traffic are used where a class i call uses i contiguous channels,

where $1 \leq i \leq 5$. The call origination and termination processes are both assumed as Poisson. For simplicity, we assumed the same average rates for all classes of traffic.

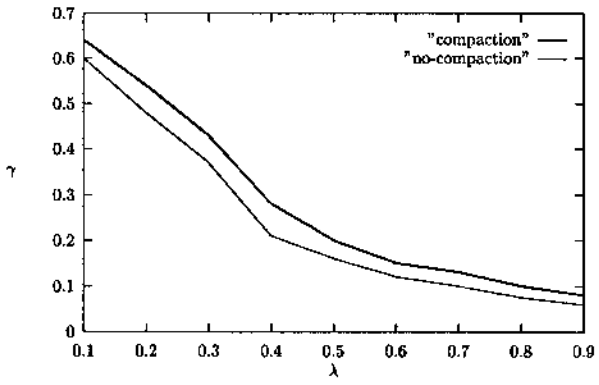


Figure 5.12: Bandwidth utilization versus traffic load

Figure 5.12 shows the improvement in spectrum efficiency achieved if bandwidth compaction is implemented, over a scheme which does not use compaction. Bandwidth utilization is defined as the ratio of the average amount of bandwidth in active use to the total amount allocated to the system. An improvement in bandwidth utilization of about 6% and 24% are observed with call arrival rates $\lambda = 0.1$ and 0.9, respectively. Therefore, we conclude that for mulitrate traffic using contiguous bandwidth allocation, a spectrum management technique like bandwidth compaction is a necessity in the low bandwidth wireless domain.

5.5. **Summary**

Bandwidth degradation and call admission strategies are proposed based on modeling quality-of-service (QoS) degradation of real-time and non-real-time traffic in a multimedia wireless network. The real-time traffic is classified into K classes, where a class i call uses i channels. Two orthogonal QoS parameters are identified, namely, *total carried traffic* and *bandwidth degradation*, such that an improvement of either of them leads to the degradation of the other. A cost function describing the total revenue

earned by the system from *bandwidth degradation* and *call admission* policies is formulated. Methods to compute the optimal policy maximizing the net revenue earned are discussed. Simulation results demonstrate that the system performance is highly sensitive to the proper choice of the cost of degradation and the revenue earned per admitted call, and the system should be properly engineered for maximum benefit.

In most systems, the real-time traffic has preemptive priority over the non-real-time traffic, based on which a novel channel sharing scheme for non-real-time users is proposed. This scheme is analyzed using a Markov modulated Poisson process (MMPP) based queuing model, and the *average queue length*, a QoS metric for non-real-time traffic, is also derived. The sensitivity of the average queue build-up (for the non-real-time traffic packets) to the call arrival rates of both types of traffic and also the packet arrival rate, are observed from simulation experiments. However, the degradation is minimal as long as the system works in the stable region.

For multi-rate traffic requiring contiguous spectrum allocation, a novel scheme called bandwidth compaction (conceptually similar to memory compaction in commercial operating systems) is described. Simulation results show an improvement in spectrum utilization as high as 24% for high traffic load.

CHAPTER 6

A LOCATION UPDATE STRATEGY FOR PCS USERS AND ITS GENETIC ALGORITHM IMPLEMENTATION

Location management in a *personal communication services* (PCS) network environment deals with the problem of tracking down a mobile user. The cellular wireless service area is composed of a collection of *cells*, each serviced by a *base station* (BS) (see Figure 1.2). A number of base stations are wired to a *mobile switching center* (MSC) as clusters. The *backbone* of the PCS network consists of all these BSs, MSCs and the existing wire-line networks (such as PSTN, ISDN etc.) between them. These MSCs act as the gateway for the base stations to the wire-line counterpart of the network.

In order to route an incoming call to a PCS user, his (or her) whereabouts in the cellular network need to be determined within a fixed time delay constraint. Location management as a whole involves two kinds of activities, one on the part of the user and the other on the part of the system providing service. A mobile user can report as soon as he crosses a cell boundary. This reporting, initiated by the user to limit the search space at a later point of time, is called a *location update*. The system, on the other hand, can initiate the search for a user, called *paging*, by simultaneously polling all the cells where the user can possibly be found.

Both paging and updates consume scarce resources like wireless network bandwidth and mobile equipment power, and each has a significant cost associated with it. The mechanism of paging involves the MSC sending a *page* message over a *forward control channel* in all the cells where the user is likely to be present. The mobile user listens to the page message and sends a response message back over a *reverse control channel*. The update mechanism involves the mobile user sending an update message over a reverse control channel, which may initiate a good amount of traffic

and switching in the backbone network.

6.1. Previous Work on Location Management

In the simplest case, the user is paged simultaneously in the whole network. The resulting signaling traffic will be enormous even for moderately large networks [PL94]. An improvement to this scheme is to split the whole area into *paging areas* (PAs), where all the cells within a PA are paged simultaneously. The number of PAs in the system is governed by a total allowable delay to locate an user, termed as the *delay constraint* [RY95]. In case of an incoming call, the PAs are sequentially paged for the user following a *paging strategy*, which lists the order in which the PAs are to be paged. It has been shown by Rose and Yates [RY95] that given the steady-state location probability distribution of a user for each paging area, the optimal paging strategy (which minimizes the average cost of paging) under no delay constraint, pages the PAs in the order of decreasing location probabilities. This is achieved by minimizing the average paging cost over all the paging areas. For the constrained delay problem, three types, namely, maximum, weighted and mean delay constraints, are considered. Problems such as constraining the maximum delay and average weighted delay can be easily solved using a dynamic programming approach. Using variations of Lagrange's multiplier technique, the authors determine optimal paging sequences in the case of constrained mean delay. However, system-wide paging needs to be done in the worst case, which incurs a huge amount of paging cost.

In order to alleviate the disadvantage of pure paging based strategies, another class of schemes rely on *location updates* by the users at certain instants, in order to restrict paging within a certain limited area near the last updated location of the user. Four such schemes have been proposed in the literature: (*i*) time based [ROS95], (*ii*) movement based [BKS95], (*iii*) distance based [AH95, MHS94], and (*iv*) zone based [MBR94, SM96, XTG93]. In a *time based* update scheme, the mobile user updates the location management database about his current location every t_u units of time

(e.g., $t_u = 1$ hour). Similarly in *distance based* schemes, the user updates his current location every D_u units of distance (e.g., $D_u = 20$ miles). In *movement based* schemes, the mobile user counts the number of boundary crossings between cells and updates when this count equals certain threshold m_u. Finally, in *zone based* schemes, the entire cellular network is partitioned into *location areas* (LA) and the user updates whenever a location area boundary is crossed. In case of an incoming call, the current LA of the user (where he has last updated) is paged. To our knowledge, all the existing cellular networks use the zone based approach.

The total cost of *location management* over a certain period of time T, is the sum of the two complementary cost components due to location updates and paging, incurred over that period. The total update cost will be proportional to the number of times the user updates, while the net cost of paging increases with the number of calls received over that period T. The complementary nature of the two components is evident from the fact that, the more frequently the user updates (incurring more update cost), the less is the number of paging attempts required to track him down.

Several strategies have been proposed which attempt to minimize either the total location management cost, or the individual costs of paging and update [BKS95, AH95, MBR94, MHS94, DAV90, OOYM91, PL94, ROS95, RY95, XTG93]. For the zone based update scheme, Xie, Tabbane and Goodman [XTG93] have proposed a method of computing the optimal location area size given the location update and paging costs as functions of the number of cells in an LA, call arrival rate and the user mobility model. They have assumed square cells and location areas (LA), with the latter containing num_{cell} number of cells. A location management cost metric as a function of num_{cell}, user call arrival rate and mobility, is derived and the value of num_{cell} which minimizes the cost is computed. Two variants of the above scheme are proposed. The first, called the *static* scheme, considers average call arrival rate and mobility index for all users. In the *dynamic* variant, each user is given a particular value of num_{cell} based on individual call arrival and mobility patterns. This scheme

makes many simplifying assumptions such as square cell and LA shape, equal number of cells in every LA, and a fluid model of user mobility.

The paging strategies due to Madhavapeddy, Basu and Roberts [MBR94] assume the knowledge of a global user location and call arrival probability distribution in a cell in order to partition the set of cells into optimal paging zones. The scheme assumes a single global model to represent the user mobility patterns, and another such model for the call arrival rates of all the users in the system. The user mobility pattern is represented by location probabilities of any mobile user in a cell given its last cell location. These probabilities are estimated using a data structure called the *location accuracy matrix* (LAM). Using the LAM data, an iterative algorithm is proposed which partitions the set of cells into paging zones which minimizes the average cost of paging for all users. In most practical systems, the user mobility and call arrival patterns exhibit a wide range of variability depending mainly on the commuting habits and needs of individual users. Additionally, they do not consider the update costs at all. Another drawback of this kind of zone based scheme is that the user update traffic is prevalent only in the boundary cells of the fixed LAs. Although the most ideal scenario would be determination of LAs on a per-user basis, it is not easy to keep track of the information for every single user, which will require a significant amount of CPU computation power, storage capacity and database access.

Bar-Noy, Kessler and Sidi [BKS95] have compared the time, distance and movement based location update schemes in terms of the location management cost, assuming two types of user movement models – independent and Markovian random on a ring topology of cells. It is proved that, in case of the independent movement model, the distance-based strategy is the best among the three while the time-based is the worst. In case of the Markovian movement model, the movement-based scheme sometimes performs better than the time-based scheme. However, the authors do not consider incoming calls in their models. Bar-Noy and Kessler [BK93] have introduced the concept of *reporting cells* which is a subset of all the cells in the system. Mobile

users update only on entering a reporting cell. On arrival of an incoming call, the user is paged in the vicinity of the reporting center he last updated from. With the help of an interference graph formulation, the authors have showed that computing the optimal set of reporting centers is an NP hard problem. Optimal and near-optimal solutions are presented for important special cases of the interference graph, e.g., tree, ring and grids. For an arbitrary graph, a simple approximation algorithm is presented.

Madhow, Honig and Steiglitz [MHS94] have developed a distance based update policy such that the expectation of the sum of the update and paging costs of the next call is minimized. The location of an user at time t relative to its last known position is denoted by a random vector $X(t)$ which is incremented by a vector of independent and identically distributed random variables at time $t+1$. The paging cost is assumed to be a function of $X(t)$. Under a memoryless movement model, the user updates at an optimal threshold distance from its last known position. Using a dynamic program formulation, an iterative algorithm, which considers the evolution of the system in between call arrivals, is used to compute the optimal threshold distance.

A similar iterative approach is used by Ho and Akyldiz [AH95] to compute the optimal threshold distance D_u, assuming a two-dimensional random walk model for each user and a hexagonal cell geometry. The random walk is mapped into a discrete-time Markov chain, where the state is defined as the distance between the user's current location and its last known cell. From this model, the probability distribution of terminal location is computed. Based on this, an average location update and terminal paging cost under given D_u and delay constraint, is determined. Given this average cost function, the optimal threshold distance, D_u, is computed using an iterative algorithm. Thus, the residing area of the user contains D_u layers of cells around its last known cell, which is partitioned into paging areas according to the delay constraint. This scheme combines a distance based update mechanism with a paging scheme subject to delay constraints.

6.1.1. Motivation of Our Work

Although the schemes in [BKS95, AH95, MHS94] achieve the goal of optimizing the location management cost on a per-user basis which is an improvement over the schemes proposed in [MBR94, XTG93], they still suffer from the following drawbacks:

- Schemes in [BKS95, AH95] assume particular topologies of the cells (e.g. ring, linear array or hexagonal) in order to simplify the analysis, whereas the cell topology in a practical cellular network has a more random structure.

- Existing mobile networks use the conventional zone based approaches similar to the models described in [MBR94, SM96, XTG93], which however ignore the per-user mobility model and call arrival pattern. Deployment of the distance based schemes [MHS94], which consider user mobility and call arrival patterns, may require an unacceptable amount of changes in the existing infrastructure. Deployment of different LAs (or LA sizes [XTG93]) for different users also calls for changing the network infrastructure. Some means have to be provided by the system to each user for detecting a cross-over between two LAs assigned to him.

- Mobility tracking over a wide area can impose a significant burden on the wire-line network [MHS94] in terms of exchange of messages between the MSCs (inter-VLR updates). This problem is ignored in almost all proposed schemes in the literature.

We insist that a truly optimal and easily implementable location management scheme must consider per-user mobility pattern on top of the conventional zone based approach. Moreover, existing systems in which all users are made to update whenever they cross a location area boundary, will lead to a significant number of redundant updates because LAs are formed assuming a common mobility model for all the users in the system. Consider, for example, a daily commuter who crosses a number of LAs

on his way from home to office. Under the existing scheme, he updates on entering each LA, although he stays there for a very short interval of time and has an extremely low probability of receiving a call. We argue that, in this particular case, most of his updates are redundant which not only consume scarce wireless bandwidth but might impose a significant burden on the wire-line network as well. Subramanian and Madhavapeddy [SM96] have demonstrated that the inter-MSC networking traffic arising from this type of activity can account for as much as 30% load on the switches. This typically includes inter-MSC hand-off and registrations involving the exchange of subscriber informations over the wire-line network and can overburden the switches, thus reducing their call carrying capacity.

6.1.2. Our Contributions

In this dissertation, we attempt to solve the location management problem under the following models and assumptions:

Flexible network model: The cellular network is modeled as a connected graph whose nodes represent the location areas. We assume no particular geometry for a cell. To our knowledge, all existing schemes assume some typical interconnections between cells. Since, we are not considering cell level granularity, this considerably reduces both the complexity and dimensionality of the problem.

Flexible mobility model: Simple models based on uniform distribution of vehicles [TGM88], fluid flow [SMHW92] and also Markovian models [AH95, MHS94] have been used earlier to characterize user mobility. Uniform and fluid flow models do not characterize the PCS user traffic well. In case of the Markovian models, the transition probability distributions between the states are assumed to be known a priori. Actual estimation of these probabilities may impose a severe burden on the switch. We consider random walk on the graph of LAs to model user mobility, leading to discrete time Markov chains. We shall show

later that consideration of the random walk on an LA level hierarchy provides a very practical and reliable way of estimating the transition probabilities of the Markov chain with minimal effort on the part of the system.

Selective update strategy: Using the Markovian mobility model for the user and memoryless call arrival patterns, we derive a location management cost (LMC) function. The problem of minimizing this cost function is amenable to near-optimal solutions by practical optimization techniques like a genetic algorithm due to large solution space and computational complexity of cost factors. This leads to a near-optimal location update strategy for individual users. Each user updates only in certain pre-selected location areas, called *update areas*, following his own mobility pattern. Optimizing on individual user rather than the majority would result in a lower aggregate cost, especially where users do not show any clear central tendency so far as the mobility is concerned. Our strategy also considers the inter-MSC update traffic [SM96] in computing the optimal location management cost.

The total location management cost is the weighted average of the location management costs in the individual LAs which themselves are functions of the user's update strategy. Expressions for these cost functions are derived. The optimization of the total location management cost is implemented using a genetic algorithm. The experimental results show that for low user location probability, low call arrival rate and/or comparatively high update cost, skipping updates in several LAs leads to minimization of the location management cost.

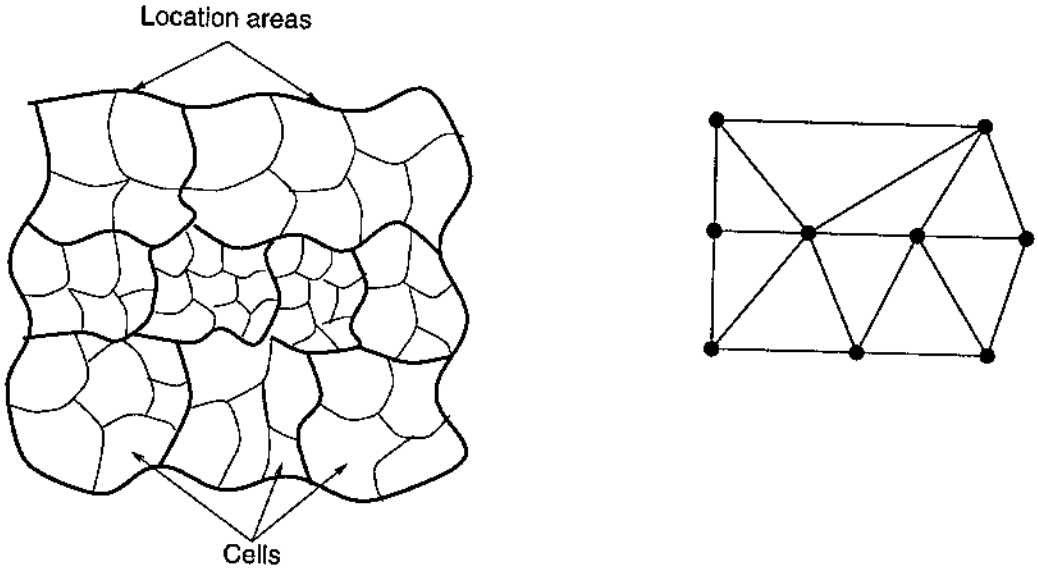
The rest of this chapter is organized as follows. Section 6.2 describes the complete model for the system, including the network and the mobility and call receiving pattern for the user under consideration. The location management cost function assuming a discrete time mobility model for the user is derived in Section 6.3. The functionality of our location management scheme is illustrated with a numerical example in Section 6.4. Section 6.5 describes a genetic algorithm formulation of our

scheme. Detailed simulation results are presented in Section 6.6, where our "selective update" scheme is compared with the existing "all-update" scheme.

6.2. System Model

6.2.1. Network Model

Existing location management schemes use a structured graph to model a cellular network. For example, circular, hexagonal or square areas are used to model the cells, whereas various regular graph topologies such as rings, trees, and one- or two-dimensional grids are used to model the interconnection between the cells. However, this model does not accurately represent the real cellular networks, where the cell shapes can vary depending on the antenna radiation pattern of the base stations and any cell can have an arbitrary (but bounded) number of neighbors.



(a) Cell and location area plan in a cellular System

(b) Network model showing interconnections of the location areas

Figure 6.1: Modeling an actual cellular system

In this work, we make no assumption about either the cellular geometry or the

interconnections between the cells. However, similar to the models described in [MBR94, SM96, XTG93], we assume the existence of a zone (or location area) based cellular mobile system. As a result, our network is represented by a bounded-degree, connected graph $G = (V, E)$ where the node-set V represents the location areas (LAs) and the edge-set E represents the access paths (roads, highways etc.) between pairs of LAs. Let $N = |V|$ be the number of nodes or LAs in the network G. For a node $v \in V$, let $\Gamma_G(v)$ denote the set of neighbors of v in G. The proposed model is illustrated in Figure 6.1.

6.2.2. Selective Update

At each LA, a user has two options to choose from, to update or not to update. Let u_i denote a binary decision variable for the user in LA i such that

$$u_i = \begin{cases} 1 & : \quad \text{update occurs in LA } i \\ 0 & : \quad \text{otherwise.} \end{cases}$$

An *update area* for the user is an LA i such that $u_i = 1$. An *update strategy* $\mathcal{S}_u = \{u_i\}$ for the user consists of a set of binary decision variables (to update or not to update) for all LAs. As a result, the strategy $\mathcal{S}_u$ can simply be represented by a binary vector $\mathbf{u} = (u_1, \ldots, u_N)$.

Assuming that the user updates immediately upon entering one of his update areas, any subsequent call will page the user only in that area. Let the cost of paging LA i be $C_p(i)$ where $1 \leq i \leq N$. This cost will be different for different LAs depending on the number of cells and the cost of paging each cell. Let $C_u(i, j)$ be the update cost for a user transition from LA i to LA j. We assume that the update costs are deterministic for all such transitions. This can be easily estimated for an operational system from the number of messages exchanged, consumption of CPU cycles etc, when a user does a location update after crossing an LA boundary. In such a system, those LA transitions which involve inter-MSC (i.e. inter-VLR) update traffic will be much more expensive from wire-line resource consumption point of view than the

intra-system update traffic [MBR94].

If LA i is an update area of the user, the cost of paging is $C_p(i)$ for a call arrival. For a non-update area, the cost of paging for the first call the user receives in that LA, will be determined by his last known LA and the paging strategy, $\mathcal{S}_p$, employed by the system to locate the user given its last known LA. Once the first call goes through, the user's location area is updated in the location management database and the paging cost for all the remaining calls received in the current LA is that for paging the same LA.

6.2.3. User Mobility Model

A simple model based on uniform distribution of vehicle locations is typically used to characterize user mobility [TGM88], which neglects directional movements of vehicular or pedestrian traffic. Another approach models vehicular traffic as fluid flow, which assumes blocks of vehicles moving with equal speed [SMHW92]. Since our objective is to develop an optimal location management strategy on a per-user basis, we use instead the *random walk model* which very well characterizes the user traffic flow for PCS networks. While random walk models on special graph topologies like one-dimensional ring or two-dimensional hexagonal cell structure have been considered earlier [AH95, MHS94], we propose to use random walk on a connected graph G representing our network model.

A random walk on a graph [MOT96, ROS80] is a stochastic process which occurs in a sequence of discrete steps. As depicted in Figure 6.2, starting at a node i (representing LA i in our case) at each discrete time slot, there is a predefined probability $P_{i,j}$ for a user to reach any neighboring node j in $\Gamma_G(i)$, or staying in node i itself. Let the slot duration be τ time units. The movement of the user at each step is independent of all previous choices. A random walk on the graph G induces a Markov chain M_G as follows [MOT96]. The states of M_G are the nodes of G and for any two nodes i and j, the transition probability between the corresponding states is given by

$P_{i,j}$.

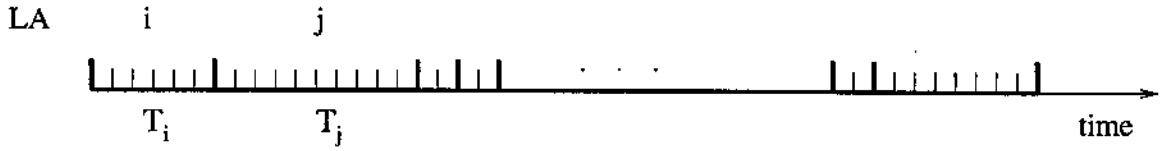


Figure 6.2: A typical movement profile of a user with time

Assuming that $P_{i,i} > 0$ for a user in any state i, this Markov chain can be shown to be irreducible, finite and aperiodic.

Lemma 1. *The discrete Markov chain, M_G, induced by the random walk on graph G is irreducible, finite and aperiodic.*

Proof: (sketch) (i) M_G is irreducible because G is a connected graph. (ii) M_G is finite because N is finite. (iii) The periodicity of M_G is the greatest common divisor of the length of all closed walks in G [MOT96]. Since at any particular step of the stochastic process, the user has a non-zero probability of remaining in the same node as in the previous step, there are closed walks of length 1 and hence M_G is aperiodic. □

Therefore, there exists a unique stationary (or steady-state) probability distribution vector $\mathbf{\Pi} = (\Pi_1, \Pi_2, \ldots, \Pi_N)$ such that $\Pi_i > 0$ for $1 \leq i \leq N$. The steady state probability Π_i of state i estimates the location probability of a user in LA i. Thus, if we know the transition probability distribution between the location areas for the user, his location probabilities at individual LAs can be estimated using the random walk model. For the purpose of our model, we will assume that this transition probability distribution matrix $\mathbf{P} = [P_{i,j}]_{N \times N}$ is known for each user. Then the steady state location probability vector can be obtained simply by solving $\mathbf{\Pi} = \mathbf{\Pi}\,\mathbf{P}$ for $\mathbf{\Pi}$.

The *sojourn time* (in number of slots) of the user in state i is a discrete random variable T_i. It also follows from the Markovian nature of transitions that the sojourn

time T_i within LA i is a geometric random variable with parameter

$$P_i = \sum_{j=1, j \neq i}^{N} P_{i,j} = 1 - P_{i,i} \tag{6.1}$$

where P_i is the probability of a transition out of LA i happening within a slot.

A few words are due as how to estimate the transition probabilities, $P_{i,j}$, in an already deployed system. One of the problems we face for the introduction of a new strategy is that, at the introductory stage, the new must co-exist with the old. Keeping this in mind, we propose gradual transition from the existing LA based location management technique to the proposed "selective update" strategy. To estimate the transition probabilities between LAs for a particular user, his movements throughout the day are observed over a long period of time (of the order of weeks). Since in the current system, the user updates whenever he changes his LA, the information about the frequency of his transitions from a certain LA i to another LA j can be easily obtained from the system database. This information helps in building up the user's movement profile. The various ways to obtain a good estimate of the transition probabilities from the information gathered is an important issue by itself and is beyond the scope of this work. An easy way of estimation is from the first principle that the probability, $P_{i,j}$, is in direct proportion to the frequency of user's transitions from LA i to LA j in the observation period.

6.2.4. Call Arrival and Duration

We assume that the call arrival process for a user is Poisson with mean rate λ. In other words, the inter-arrival time between calls are exponentially distributed with this rate. But not all of these calls are liable to trigger paging. Specifically, if the user is already in the middle of a call, the system (at the MSC level) being aware of this situation, does not page. Such a call arrival is dealt with in different ways which may include sending a call waiting tone, sending back busy tone or forwarding to voice mailbox. If we also assume that the duration of a typical call is exponentially distributed with

rate μ, the user can be modeled by a M/M/1/1 queuing system having only two states, viz. busy (with one call at most) and not busy. The steady-state probabilities for these two states are $\frac{\lambda}{\lambda+\mu}$ and $\frac{\mu}{\lambda+\mu}$, respectively. The call arrivals which find the user not busy are the ones to trigger paging (i.e., cause transitions from not busy to busy state), and constitute a Poisson process with rate $\lambda\left(\frac{\mu}{\lambda+\mu}\right) = \lambda_p$ (say). On the other hand, the call termination process (i.e., transitions from busy to not busy state) is also Poisson with rate $\mu\left(\frac{\lambda}{\lambda+\mu}\right)$, which is the same as λ_p.

The probability of one effective call arrival or termination within a slot of duration τ units (Figure 6.2) is thus $\lambda_p\tau + o(\tau)$ and that of more than one arrival or termination is given by $o(\tau)$, where $o(\tau)$ vanishes as $\tau \to 0$ implying the probability of more than one call arrival or termination per slot to be negligible. Since the call arrival process is Poisson which possesses stationarity and independent increments, the number of calls arriving for the user within a particular time duration depends only on the length of the duration. Hence, the number of calls received by the user within an LA depends only on its length of stay within that LA. In effect, we assume that the call arrival process is independent of the user mobility pattern.

If we now assume that the number of slots t_i the user spends in LA i is large and the probability $r = \lambda_p\tau$ of a call arrival or termination in a slot is small such that $t_i r$ is finite, then the Poisson call arrival process can be replaced by a Bernoulli process [ROS80]. Hence, the probability of arrival of c number of calls in LA i is given by the binomial distribution, $B(t_i, r) = \binom{t_i}{c} r^c (1-r)^{t_i - c}$.

6.3. Computation of Location Management Cost

6.3.1. Average Paging and Update Costs for Update Area i

Recall that, the sojourn time T_i in LA i is a geometric random variable with parameter P_i i.e., $\mathbf{Pr}[T_i = t_i] = (1-P_i)^{t_i - 1} P_i$. The average paging cost per slot for the user in

the update area i is

$$\begin{aligned}
\mathbf{E}[\text{Paging cost per slot}] &= \sum_{t_i=1}^{\infty} \frac{1}{t_i} E[\text{Paging cost over } t_i \text{ slots} \mid T_i = t_i] \cdot \Pr[T_i = t_i] \\
&= \sum_{t_i=1}^{\infty} \frac{1}{t_i} \left\{ \sum_{c=0}^{\infty} c C_p(i) \binom{t_i}{c} r^c (1-r)^{t_i - c} \right\} (1-P_i)^{t_i - 1} P_i \\
&= r\, C_p(i), \qquad \text{where } r = \frac{\lambda\mu\tau}{\lambda+\mu}.
\end{aligned}$$

The average update cost per slot for entering an update area i is given by $\sum_{j=1, j\neq i}^{N} \Pi_j P_{j,i} C_u(j,i)$, where $P_{j,i}$ is the probability of a transition from LA j to LA i, and Π_j is the location probability of the user at LA j. Note that this cost is already computed on a per slot basis as a transition takes place on a slot-by-slot basis in the discrete model. Hence, the per slot average location management cost for all the update areas is given by

$$\text{LMC}^{(1)} = \sum_{i=1}^{N} u_i \left\{ \Pi_i r C_p(i) + \sum_{j=1, j\neq i}^{N} \Pi_j P_{j,i} C_u(j,i) \right\}. \tag{6.2}$$

The superscript on LMC indicates that this cost is contributed by the update areas with $u_i = 1$.

6.3.2. Average Paging Cost in Non-Update Area i

First-time vs. Subsequent Paging

If the user does not update immediately on entering LA i, then the paging associated with the first call received is costlier than the subsequent ones. The *expected paging cost $C_p^0(i)$ for the first call in LA i* is determined by the paging strategy, $\mathcal{S}_p$, used by the system in order to track him down to *current LA i* starting from the *last known LA k*. After the first call has been successfully received in LA i, the user's LA information remains updated in the system during the entire call holding time or the remaining duration of his sojourn. Hence, each subsequent call that triggers paging involves only the cost of paging, $C_p(i)$, within the current LA i.

First we condition on the sojourn time as in the preceding case (Section 6.3.1). Given $T_i = t_i$, we also condition on the slot number t_f in which the first call arrives. We know,

$$\mathbf{Pr}\left[\text{First call in slot } t_f \leq t_i \mid T_i = t_i\right] \;=\; (1-r)^{t_f-1}r.$$

Suppose c calls arrive during the remaining slots of the sojourn. Clearly, the number of such calls can be at least zero and at most $t_i - t_f$, having a binomial distribution $B(t_i - t_f, r)$. Hence the expected paging cost for all calls excluding the first one is

$$\sum_{c=0}^{t_i-t_f} c\, C_p(i) \binom{t_i - t_f}{c} r^c (1-r)^{t_i-t_f-c} \;=\; (t_i - t_f)\, r\, C_p(i).$$

Thus, we have

$$\begin{aligned}
&\mathbf{E}\left[\text{Cumulative cost of paging within LA } i \mid T_i = t_i\right] \\
&= C_p^0(i) + \sum_{t_f=1}^{t_i} \left\{(t_i - t_f)\, r\, C_p(i)\right\} (1-r)^{t_f-1} r \\
&= C_p^0(i) + \left\{rt_i + (1-r)^{t_i} - 1\right\} C_p(i).
\end{aligned}$$

Once again unconditioning over the distribution of geometric random variable T_i, the expected cumulative paging cost for one visit to non-update LA i is obtained as

$$\begin{aligned}
\mathrm{LM}^{(0)}(i) &= C_p^0(i) + \sum_{t_i=1}^{\infty} \{rt_i + (1-r)^{t_i} - 1\} C_p(i) (1-P_i)^{t_i-1} P_i \\
&= C_p^0 + r\, C_p(i) \left\{ \frac{1}{P_i} - \frac{1}{P_i + (1-P_i)r} \right\}. \qquad (6.3)
\end{aligned}$$

The superscript indicates the decision variable u_i, which is zero for a non-update area. If the average number of calls the user receives in LA i is much greater than one, i.e. $\frac{r}{P_i} \gg 1$, a good approximation of Equation (6.3) is

$$\mathrm{LM}^{(0)}(i) \approx C_p^0(i) + \frac{r}{P_i} C_p(i) \qquad (6.4)$$

where the first term corresponds to the average cost of paging for the first call and the second term for all subsequent calls. However, it is not a good idea to choose large values of τ to force $r \gg P_i$, as that would invalidate the binomial approximation of Poisson arrivals and departures.

Cost of First-time Paging

It has been noted earlier that the cost associated with first-time paging in a non-update area is a resultant of the user's last known LA and the paging strategy used by the system. In order to maintain generality in this model, we have not made any assumption on the nature of the paging strategy. We have simply assumed that such a strategy, $\mathcal{S}_p$, exists that gives rise to the appropriate cost function. Hence, the proposed solution may not be truly optimal, but optimal in a restricted sense as certain parameters (e.g. $\mathcal{S}_p$, location area sizes, etc.) are assumed to have already been designed for the system and not under our control. As a result, we can express the expected cost of paging the user for the first call in LA i as

$$C_p^0(i) = \sum_{k=1, k\neq i}^{N} C(\mathcal{S}_p, k, i) \times \mathbf{Pr}\left[\text{Last known LA is } k \cap \text{Current LA is } i \text{ when paged}\right] \tag{6.5}$$

where $C(\mathcal{S}_p, k, i)$ is the cost of tracking the user in LA i given his last known LA k using the paging strategy $\mathcal{S}_p$. The remainder of this subsection derives an expression for the probability that last known LA is k and current LA is i.

Consider the situation where the user is paged in LA i at slot h_i, where the slot counting is started from the beginning of his movement. Suppose he is last known to be at LA k in slot h_k where $1 \leq h_k < h_i$. This implies all of the following three independent events: (i) An update or call termination happening in slot h_k, (ii) no call arrival during slots $h_k + 1$ and $h_i - 1$ followed by an arrival in slot h_i, and (iii) no entry into an update area during the $(h_i - h_k)$-step transition.

Event (i) again implies either of the following two *independent* sub-events in slot h_k: (a) *Update*: h_k is the first slot (in which user updates) of his sojourn in update area k, the probability of which is $\sum_{l\neq k} \Pi_l P_{l,k} u_k$. (b) *Terminate*: User terminates a call in h_k irrespective of whether k is an update area or not, the probability of this being r.

Hence, the probability of Event (i) is

$$\begin{aligned}
&\mathbf{Pr}\,[\,\text{Update or call termination in slot } h_k\,] \\
&= \mathbf{Pr}\,[\,\text{Update}\,] + \mathbf{Pr}\,[\,\text{Terminate}\,] - \mathbf{Pr}\,[\,\text{Update} \cap \text{Terminate}\,] \\
&= r + (1-r)\sum_{l \neq k} \Pi_l P_{l,k} u_k.
\end{aligned}$$

Event (ii) merely signifies no call arrival in $h_i - h_k - 1$ successive slots and one subsequent arrival. The Bernoulli arrivals being independent from slot to slot, the probability of this event is $(1-r)^{h_i - h_k - 1} r$.

For Event (iii), let the set of non-update areas be denoted by $\bar{U}$. Then, the user never updates after the slot h_k till h_i implies that he enters only the non-update LAs in $\bar{U}$ after completion of his sojourn in LA k. Let the user spend till slot h'_k in LA k during which he need not update anyway, the probability of which is $(1-P_k)^{h'_k - h_k}$. Let $\mathbf{Pr}\,[\,(j \xrightarrow{h} i) \in \bar{U}\,]$ denote the probability that the user starting at LA $j \in \bar{U}$ ends up in LA $i \in \bar{U}$ after passing through non-reporting LAs in h transitions. Then the probability that the user does not enter an update area at all after spending $h'_k - h_k$ slots beyond h_k in LA k, is given by

$$\sum_{h'_k = h_k}^{h_i - 1} (1-P_k)^{h'_k - h_k} \left\{ \sum_{j \in \bar{U}} P_{k,j}\, \mathbf{Pr}\,[\,(j \xrightarrow{h_i - h'_k} i) \in \bar{U}\,] \right\}.$$

Multiplying the probabilities of the independent Events (i), (ii) and (iii), and summing over all possible values of h_k, we arrive at the following final expression

$$\begin{aligned}
&\mathbf{Pr}\,[\,\text{Last known LA is } k \cap \text{Current LA is } i\,] \\
&= \sum_{h_k=1}^{h_i - 1} \left\{ r + (1-r)\sum_{l \neq k} \Pi_l P_{l,k} u_k \right\} \times \left\{ (1-r)^{h_i - h_k - 1} r \right\} \times \\
&\qquad \left\{ \sum_{h'_k = h_k}^{h_i - 1} (1-P_k)^{h'_k - h_k} \left(\sum_{j \in \bar{U}} P_{k,j} \mathbf{Pr}\,[\,(j \xrightarrow{h_i - h'_k} i) \in \bar{U}\,] \right) \right\} \qquad (6.6)
\end{aligned}$$

To compute $\mathbf{Pr}\,[\,(j \xrightarrow{h_i - h'_k} i) \in \bar{U}\,]$, we can write the following recurrence relation using Chapman-Kolmogorov equation [ROS80],

$$\mathbf{Pr}\,[\,(j \xrightarrow{h} i) \in \bar{U}\,] = \sum_{l \in \bar{U}} P_{j,l} \cdot \mathbf{Pr}\,[\,(l \xrightarrow{h-1} i) \in \bar{U}\,]$$

which implies that the probability $\mathbf{Pr}[(j \xrightarrow{h} i) \in \bar{U}]$ can be written as the product of the 1-step transition probability to another non-update LA l and the $(h-1)$-step probability that the user remains in $\bar{U}$ starting from LA l reaching LA i. This can be succinctly represented in a matrix notation. Consider the $N \times N$ matrix $\mathbf{P}_{\bar{U}}$ obtained from the transition matrix $\mathbf{P}$ by setting the entries of the jth column to zero for all $j \notin \bar{U}$. In other words,

$$\mathbf{P}_{\bar{U}} = \mathbf{P} \times \mathrm{diag}(\bar{\mathbf{u}}_{\mathbf{s}}),$$

where $\bar{\mathbf{u}}_{\mathbf{s}}$ is the bitwise inverse of the update vector $\mathbf{u}_{\mathbf{s}}$. Then $\mathbf{Pr}[(j \xrightarrow{h_i - h'_k} i) \in \bar{U}]$ is the entry in the jth row and ith column of the matrix $\mathbf{P}_{\bar{U}}^{h_i - h'_k}$, i.e.

$$\mathbf{Pr}[(j \xrightarrow{h_i - h'_k} i) \in \bar{U}] = (\mathbf{P}_{\bar{U}}^{h_i - h'_k})_{j,i}$$

Equation (6.6) can now be written as

$$\begin{aligned}
&\mathbf{Pr}[\text{Last known LA is } k \cap \text{Current LA is } i] \\
&= \left\{ r + (1-r) \sum_{l \neq k} \Pi_l P_{l,k} u_k \right\} \left[\sum_{h_k=1}^{h_i-1} \{(1-r)^{h_i - h_k - 1} r\} \left\{ \sum_{h'_k = h_k}^{h_i - 1} (1-P_k)^{h'_k - h_k} \left(\sum_{j \in \bar{U}} P_{k,j} (\mathbf{P}_{\bar{U}}^{h_i - h'_k})_{j,i} \right) \right\} \right] \\
&= \left\{ r + (1-r) \sum_{l \neq k} \Pi_l P_{l,k} u_k \right\} \left[\sum_{j \in \bar{U}} P_{k,j} \left(\sum_{h_k=1}^{h_i-1} \left\{ (1-r)^{h_i - h_k - 1} r \right\} \left\{ \sum_{h'_k = h_k}^{h_i - 1} (1-P_k)^{h'_k - h_k} \mathbf{P}_{\bar{U}}^{h_i - h'_k} \right\} \right)_{j,i} \right] \\
&= \left\{ r + (1-r) \sum_{j \neq k} \Pi_j P_{j,k} u_k \right\} \left\{ \sum_{j \in \bar{U}} P_{k,j} (\hat{\mathbf{P}}_{\bar{U}})_{j,i} \right\}, \qquad (6.7)
\end{aligned}$$

where

$$\begin{aligned}
\hat{\mathbf{P}}_{\bar{U}} &= \sum_{h_k=1}^{h_i-1} \left\{ (1-r)^{h_i - h_k - 1} r \right\} \left\{ \sum_{h'_k = h_k}^{h_i - 1} (1-P_k)^{h'_k - h_k} \mathbf{P}_{\bar{\mathbf{u}}}^{h_i - h'_k} \right\} \\
&= \sum_{l=0}^{h_i-2} r\{(1-r)(1-P_i)\}^l \left\{ \sum_{m=0}^{l} (1-P_k)^{l-m} \hat{\mathbf{P}}_{\bar{U}}^{m+1} \right\} \qquad (6.8)
\end{aligned}$$

Since we are interested in the long term behavior of the system, we only take into account the limiting value of matrix $\hat{\mathbf{P}}_{\bar{U}}$ as $h_i \to \infty$. Equation (6.5) along with Equations (6.7) and (6.8) gives the final form of the average paging cost for the first call received in a non-update area i.

Average Paging Cost per slot in Non-Update Areas

We have computed $\mathrm{LM}^{(0)}(i)$ as the mean cost per sojourn in LA i. To obtain the average location management cost per slot, let us observe that the average number of slots spent in LA i in each sojourn is given by $\frac{1}{P_i}$. By renewal theoretic argument, it can be shown that the average paging cost per slot is $\mathrm{LM}^{(0)}(i)/(\frac{1}{P_i})$. Thus, the average paging cost for all non-update areas is

$$\mathrm{LMC}^{(0)} = \sum_{i=1}^{N} \bar{u}_i \, \Pi_i \, P_i \, \mathrm{LM}^{(0)}(i) \tag{6.9}$$

which is also the average location management cost for these areas.

6.3.3. User's Average Location Management Cost (LMC)

The final form of the average cost function for a user is the sum of the expressions given in Equations (6.2) and (6.9). Hence

$$\mathrm{LMC} = \sum_{i=1}^{N} \{ u_i \sum_{j=1, j \neq i}^{N} \Pi_j \, P_{j,i} \, C_u(j,i) + u_i \, \Pi_i \, r \, C_p(i) + \bar{u}_i \, \Pi_i \, P_i \, \mathrm{LM}^{(0)}(i) \} \tag{6.10}$$

where $C_u(j,i)$ is the cost of update for a transition from LA j to LA i; $C_p(i)$ is the cost of paging LA i; and $\mathrm{LM}^{(0)}(i)$ is given by Equation (6.3).

6.4. Numerical Studies

To illustrate our "selective update" strategy let us first consider a service area made of eight LAs. The model for the network is a graph with eight nodes and eleven edges as depicted in Figure 6.3. The maximum edge distance between any two nodes is three. The transition probability matrix and the resulting steady-state probability vector have been tabulated in Table 6.1. The rationale behind such a choice of transition probabilities is that a large percentage of mobile users who commute from their homes to offices everyday, typically stay at these two location areas for most part of the day. This implies a high location probability (and hence, a high call receiving

probability) in those two LAs. Let us assume that LA 2 and LA 5 correspond to the user's home and office areas respectively in this example. High values of $P_{2,2}(= .94)$ and $P_{5,5}(= .92)$ have been chosen to induce such a mobility pattern that results in considerably large steady-state probabilities $\Pi_2(= .4046)$ and $\Pi_5(= .3017)$. For these two LAs, $P_{i,j} = \frac{1}{|\Gamma_G(i)|}$ if $j \in \Gamma_G(i)$, and zero otherwise. For all other LAs, $P_{i,i} = P_{i,j} = \frac{1}{|\Gamma_G(i)|+1}$ for $j \in \Gamma_G(i)$, and zero otherwise.

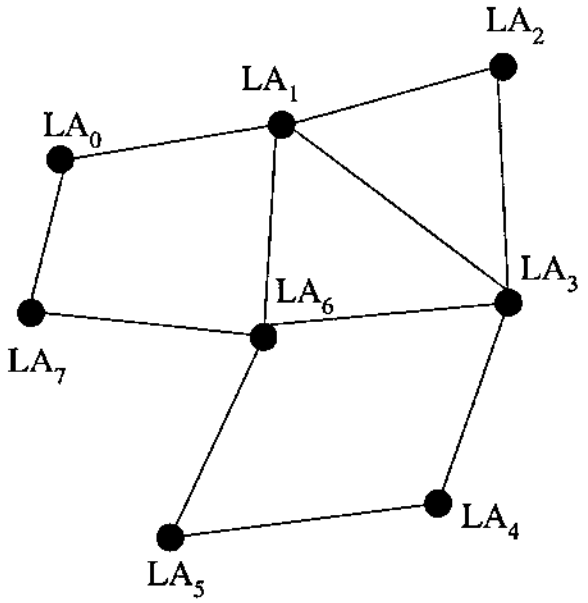


Figure 6.3: The location area graph for the system

The above scenario suggests the possibility that the user might skip updating in LAs except for a few strategic ones. If his mobility pattern is more or less uniform over all location areas, then our "selective update" scheme boils down to the existing "all-update" scheme which, therefore, is a special case of our more general update strategy. We represent such an update strategy by a bit pattern $S_u = \{u_7\, u_6\, u_5\, u_4\, u_3\, u_2\, u_1\, u_0\}$, so that the strategies can be easily indexed by the binary number they represent, for computational purposes. We chose $u_i = 1$ for an update in LA i and zero otherwise. In principle, we need to compute the minimum location management cost LMC* and

Table 6.1: Transition probability matrix and steady-state probabilities

LAs	Transition probability								Steady-state probability
	LA0	LA1	LA2	LA3	LA4	LA5	LA6	LA7	
LA0	0.33	0.33	0.00	0.00	0.00	0.00	0.00	0.33	0.0369
LA1	0.20	0.20	0.20	0.20	0.00	0.00	0.20	0.00	0.0608
LA2	0.00	0.03	0.94	0.03	0.00	0.00	0.00	0.00	0.4046
LA3	0.00	0.20	0.20	0.20	0.20	0.00	0.20	0.00	0.0606
LA4	0.00	0.00	0.00	0.33	0.33	0.33	0.00	0.00	0.0361
LA5	0.00	0.00	0.00	0.00	0.04	0.92	0.04	0.00	0.3017
LA6	0.00	0.20	0.00	0.20	0.00	0.20	0.20	0.20	0.0611
LA7	0.33	0.00	0.00	0.00	0.00	0.00	0.33	0.33	0.0381

pick the update strategy which makes it possible. The factors that affect this cost include the pattern of call arrival and duration, the relative values of update and paging cost within an LA, and the paging strategy used by the system. Let us take a look at all of these three.

A user might typically get 3-4 short calls during a busy hour with average duration of 1-2 minutes. We can choose λ and μ to be 0.001 per second and 0.01 per second respectively. The value of $\frac{\lambda\mu}{(\lambda+\mu)}$ reduces to approximately 0.009 per second. Various values of r may be selected based on the choice of slot length τ, e.g. $\tau = 100$ sec. gives rise to $r = 0.9$. Since r is the success probability of Bernoulli trials, we should observe variations of LMC* with r varying from 0.1 to 0.9 in the increment of 0.1. To keep computations simple, we assign equal paging cost of 10000 units for all LAs. As figures from a real system for the cost of update for a single user and the cost of paging in an LA are not available, we experimented with various ratios of these two costs. To reflect the variation of these relative values, LMC* is computed using Equation (6.10) for various values of effective arrival/termination rate r and the ratio

$\frac{C_u}{C_p}$ of the update cost to the paging cost. The paging strategy $\mathcal{S}_p$ is similar to that proposed in [AH95] and works as follows. The last known LA is paged first. If no response is obtained, the set of neighboring LAs are paged. If still unsuccessful, the set of LAs at edge distance two, and if required, those at distance three are paged until the user is tracked down.

Table 6.2 shows the minimum location management cost LMC* for various values of r and $\frac{C_u}{C_p}$. The optimal update strategies corresponding to each LMC* are shown within parenthesis. We observe from Table 6.2 that, as the cost of an update relative to that of paging increases for constant call arrival/termination rate r, the required number of updates to achieve the optimal cost decreases. For example, with $r = 0.6$, the user needs to do five location updates for $\frac{C_u}{C_p} = 0.1$, while three such updates are required for $\frac{C_u}{C_p} = 1$. As the call arrival/termination rate r increases for constant $\frac{C_u}{C_p}$, the number of updates increases. For low values of $r(= 0.1)$, the user need not update at all. This implies that on the average, systemwide paging will be less costly as compared to the cost of one or more updates and the reduced paging cost that results. For higher values of r, updates in certain strategic location areas is necessary and the maximum number of such updates is five for $r = 0.9$. Hence, we conclude that even for high call arrival rates, there is scope of improving the "all-update" strategy by computing individual update strategies for the users.

Figures 6.4 and 6.5 pictorially represent the variation of the optimal location management cost with r and $\frac{C_u}{C_p}$. It is observed that, although the value of LMC* increases with increase in both the parameter values, it is more sensitive to the variations of rate r than the $\frac{C_u}{C_p}$ ratio. Since r depends on the slot lenth τ, the choice of τ is very critical in choosing the best update strategy based on this discrete-time model.

The component found to have the maximum impact on the location management cost is the average cost of the first time paging $C_p^0(i)$ in a non-update LA i. These values capture the penalty the system has to pay for not making the user update in all location areas. The genetic algorithm implementation of the above problem

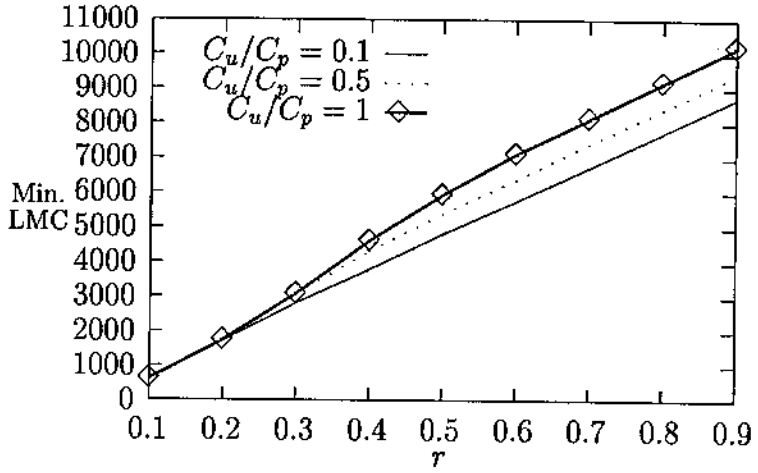


Figure 6.4: Optimal location management cost vs. r

(described later) is required to compute these values for different update strategies. Figure 6.6 shows the value of $C_p^0(i)$ for LA i in the worst case when the user does not update at all. Under this "no-update" strategy, the average cost of paging the user for the first call he receives in any LA is expected to be maximum because, the only time the system can know about his whereabouts is when he receives a call. In Figure 6.7 we also display another set of values of C_p^0 for the update strategy {10101010} with equal number of updates and no-updates to show the impact of these updates. Note that, the value of C_p^0 is zero in update areas as is obvious from the definition.

6.5. Genetic Algorithm Formulation

For small number of LAs, the optimal strategy for location update can be obtained by enumerating all possible solutions (represented as bit-strings) of decision vectors in the solution space, as in the previous example. But as the number of LAs increases, the solution space grows exponentially. Also, the nature of the optimization problem calls for an iterative method of computing the optimal solution. These considerations lead us to use genetic algorithm to compute the optimal (or near-optimal) solution

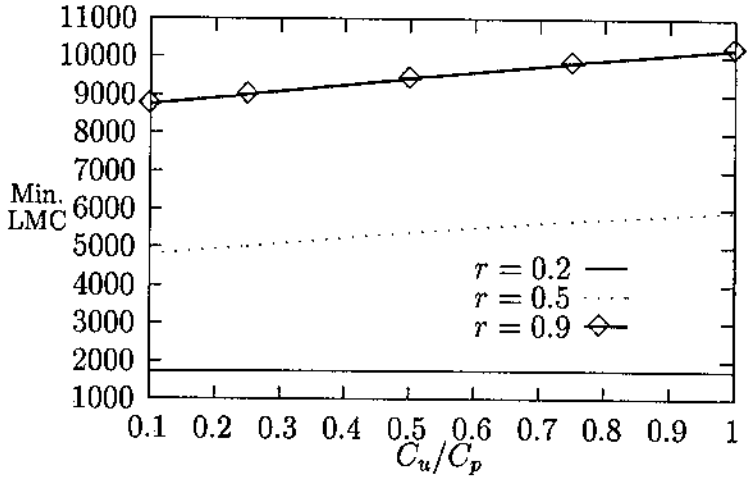


Figure 6.5: Optimal location management cost vs. $\frac{C_u}{C_p}$

to this combinatorial optimization problem.

A genetic algorithm for a given problem has the following features [MIC95]: (i) the potential solutions are represented by bit-strings, each called a *chromosome*; (ii) a function evaluates the *fitness* of the chromosome and hence the entire population; and (iii) genetic operators like *cross-over* and *mutation* alter the composition of the children with certain probabilities.

In our case, an update strategy for a user is represented by the update vector, which is a bit-string corresponding to update ($u_i = 1$) and non-update ($u_i = 0$) for each LA. Thus the length of the bit-string is equal to the number of LAs. A group of strategies is chosen at random as the initial population. However, if the initial population is chosen properly, the iterations might converge faster to the optimal solution.

We evaluate the fitness of a chromosome by the function $\frac{1}{\text{LMC}}$, where LMC is given by Equation (6.10). This implies that the smaller the location management cost, the greater the fitness of the chromosome. The selection function is implemented based on a roulette-wheel spinning mechanism [MIC95]. There are two associated

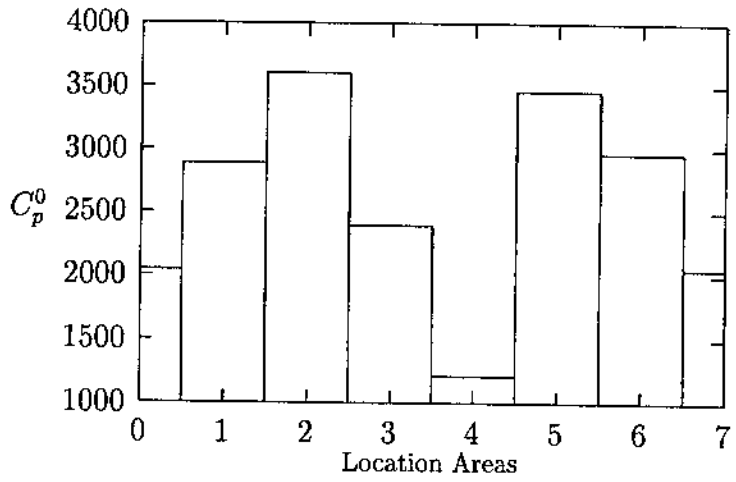


Figure 6.6: Average first time paging cost for the user at various LAs with "no-update" strategy

parameters, namely the probabilities of crossover and mutation, which are assumed to be 0.8 and 0.01, respectively.

At each iteration of the genetic algorithm, we keep track of the best chromosome from the initialization phase till that iteration cycle, which gives the optimal (or near-optimal) solution at the termination of the algorithm. The population size for each generation is kept constant at 50 and the number of bits (the number of LAs) in the chromosome is chosen as 8 (the same network as shown in Figure 6.3 is considered). The cost function LMC is computed using the transition probabilities shown in Table 6.1 and identical values of C_u and C_p as in the numerical example. The values for $C_p^0(i)$s are computed using a mathematical software package and provided as input to the genetic algorithm at every iteration. It was found that the genetic algorithm converged very fast to the optimal solution in all cases and identical results as shown in Table 6.2 were obtained.

In this genetic algorithm implementation, the best and average values of fitness of the chromosome as well as the standard deviations are computed for each generation.

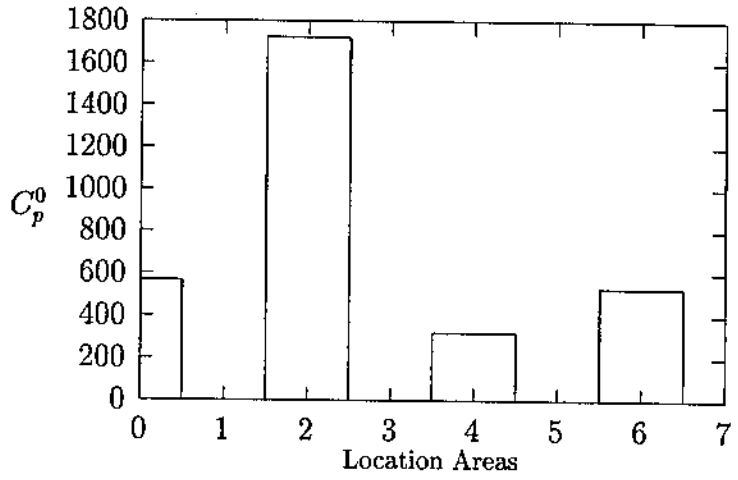


Figure 6.7: Average first time paging cost for the user at various LAs with a selective update strategy of (10101010)

We present in Table 6.3 a sample run from the experiment with $r = 0.5$ and $\frac{C_u}{C_p} = 0.33$.

6.6. **Simulation Experiments**

In Sections 6.3, a location management cost function is derived based on a particular mobility model for the user. In Section 6.5, a genetic algorithm is presented to optimize the cost function and compute an update strategy for the user such that the system resource consumption is minimized on the average. In this section, we present the results of our simulation experiments which attempt to capture the real life movement profile of a user in a PCS network, which is a mixture of deterministic and randomized movements. We also compare the location management cost of the update strategy predicted by our analytical model, with a system where an update always takes place (worst case scenario) and the minimum cost from the simulation experiments. The cost function computed by our model gives the minimum *average* cost and leads to an optimal update strategy for the user assuming static movement probabilities.

For the simulation model, two types of movement profiles for a user is simulated: (*i*) *random* in which the user has equal probability of being located in any LA at any instant, and (*ii*) the user has a certain directional component in his movement most of the times having much higher location probabilities in certain LAs than others, and random in other areas. Note that the latter movement profile has closer resemblance to the real life scenario than the former. Indeed, experiments show that for the first type of movement, our update strategy degenerated to the all-update scheme which is also optimal in this case. Therefore, for the rest of this work we shall be concerned with the second type of movement profile only.

The call arrival process is assumed to be Poisson. In our analytical model, we assumed that call arrival is independent of the user mobility model. In real life scenario, this is hardly the case since a user is expected to receive more calls in certain areas (e.g., his home or office) than in other areas. In the simulation, the call arrival process is dependent on whether the user updates or not in a location area and the rate is increased by a small fraction if he does. The network model for the simulation is the same graph depicted in Figure 6.3.

6.6.1. Experimental Results

In the following set of experiments, the costs of our selective update strategy and that of the all update scheme are compared with the minimum cost obtained for each from simulation, by plotting the corresponding cost ratios with traffic load λ in Erlang (Figures 6.8-6.11). Note that, our analytical model computes the minimum average location management cost. Hence, the necessity of comparing this minimum value of the average cost with the true optimal cost obtained from the simulation run.

For simplicity, the cost of an update, C_u, is assumed to be constant and the cost of paging, C_p, is same for all location areas. Simulation experiments are performed for particular $\frac{C_u}{C_p}$ ratios which are greater than one. Figures 6.8-6.11 display the results for $\frac{C_u}{C_p} = 4$, $\frac{C_u}{C_p} = 2$, $\frac{C_u}{C_p} = 1.33$ and $\frac{C_u}{C_p} = 1$, respectively. The traffic load generated by

the user is assumed to be quite high (in busy hour) and varies from 0.1 to 0.9 Erlang. We assumed a two minute average call holding time for the user.

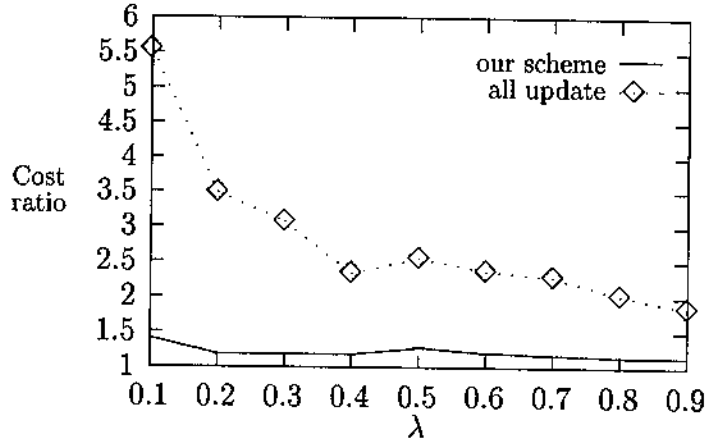


Figure 6.8: Ratio of location management costs with the minimum cost from simulation for $\frac{C_u}{C_p} = 4$

In all four cases we note that the cost due to our selective update strategy is less than a factor of 1.5 the minimum cost from actual experiments. The improvement factor of our scheme over the existing all-update scheme is significant (as high as 3.9 times) for low traffic loads. As the load increases, more updates are necessary to decrease the excessive paging costs. Hence, the difference between their performances diminishes. For $\frac{C_u}{C_p} = 1$ (in Figure 6.11), both the schemes merge into the optimal strategy for $\lambda \geq 0.6$ Erlang, implying that when paging and update costs are comparable and tele-traffic load is high, updating in all the location areas produces the best result.

Figures 6.8-6.11 also display the low variability of our scheme in terms of location management cost compared to the all-update scheme for a wide range of traffic load. For example, for $\frac{C_u}{C_p} = 1.33$, the ratio of the cost for the all update scheme to the minimum cost varies between 2.6 and 1.15, while for our scheme this varies only between 1.2 and 1.12, for a load varying from 0.1 to 0.9 Erlang. This demonstrates

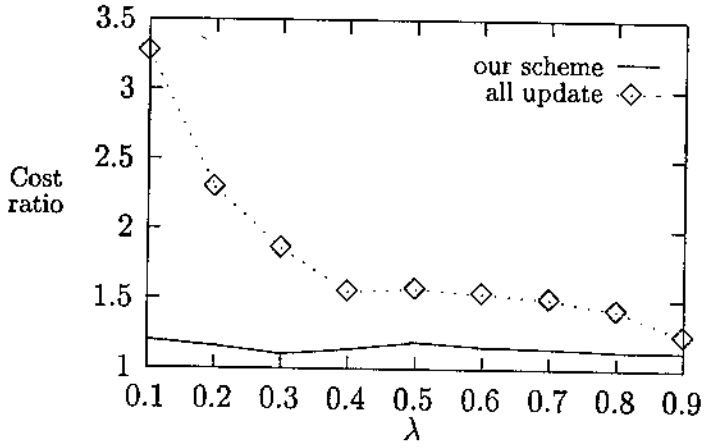


Figure 6.9: Ratio of location management costs with the minimum cost from simulation for $\frac{C_u}{C_p} = 2$

the stability of our proposed update strategy which makes it not-so-sensitive to the choice of the paging strategy, $\mathcal{S}_p$, used in the system.

Finally, Figures 6.12 and 6.13 compare the actual paging and update costs of the selective update and the all-update schemes to the corresponding quantities obtained from the minimum cost simulation experiment. For the "all-update" scheme (see Figure 6.13), the update costs are many-fold higher (about 8 times for light load) than the minimum cost simulation. Note that, the minimum cost simulation might produce completely different update strategies under various loads, hence the variation of the ratios of the update costs with λ. For our "selective update" scheme (see Figure 6.12), we note a similar kind of decreasing variability of the update costs with load, compared to the minimum cost simulation, but our update costs are almost always lower (except for very light loads, when they are equal) than that produced by the latter. Our paging costs are about 1.1 to 1.4 times higher than those for the minimum cost simulation. The paging costs for the all-update scheme are smaller than those for the minimum cost case. But the many-fold increase in the update costs leads to a significant increase in the total location management cost. The complementary nature of the paging and

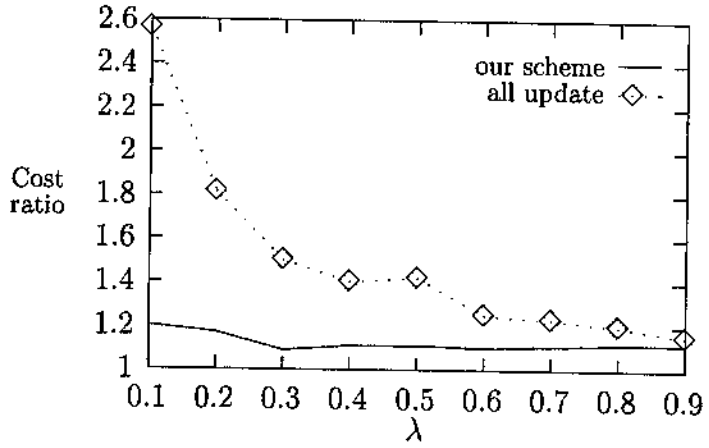


Figure 6.10: Ratio of location management costs with the minimum cost from simulation for $\frac{C_u}{C_p} = 1.33$

update costs are also apparent from the graphs in Figures 6.12 and 6.13. Again we note the stability of our scheme in terms of low variation paging and update costs for a wide range of traffic load.

6.7. Summary

We have proposed a novel location update strategy and its implementation using a genetic algorithm. Assuming the existence of location areas in the system to start with, our scheme develops an update strategy specifying a set of update areas for each user. A user updates only in his update areas. The update strategy minimizes the average location management cost derived from a user-specific model and call generation pattern. Experimental results demonstrate that for low residing probability in certain LAs, low call arrival probability and high update cost for a user, a location update in some of those LAs leads to increase in the average location management cost than no update.

Let us qualitatively evaluate the practical merit of our approach. We have shown that the currently implemented zone based "all-update scheme" is a special case of

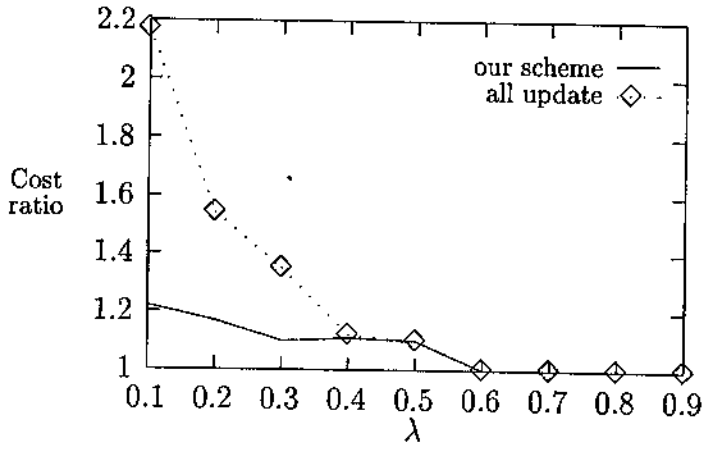


Figure 6.11: Ratio of location management costs with the minimum cost from simulation for $\frac{C_u}{C_p} = 1$

our "selective update" strategy. This is also confirmed by the simulation experiments. Our scheme improves on the existing zone based schemes by minimizing the total location management cost for individual users by devising an update strategy for each. In this way, the mobility and call arrival patterns are taken into account for each user. The only required parameters to implement our strategy on top of the traditional zone based approach involve an LA-LA transition probability distribution for each user and its average call arrival rate. Both of these parameters can be easily estimated through a period of observance. For example, in the traditional approach whenever the user changes his LA, he updates. By keeping track of the number of such updates from LA i to LA j over a certain period of time, his transition probability from i to j can be estimated. Similarly, the call arriving rate for the user can be estimated from his call receiving records over a time interval. Once these parameters are known, the optimal update strategy for the user can be computed using the genetic algorithm solution.

It has been shown in [BKS95] that the distance based update scheme generally performs better than time and movement based schemes. But deployment of a distance

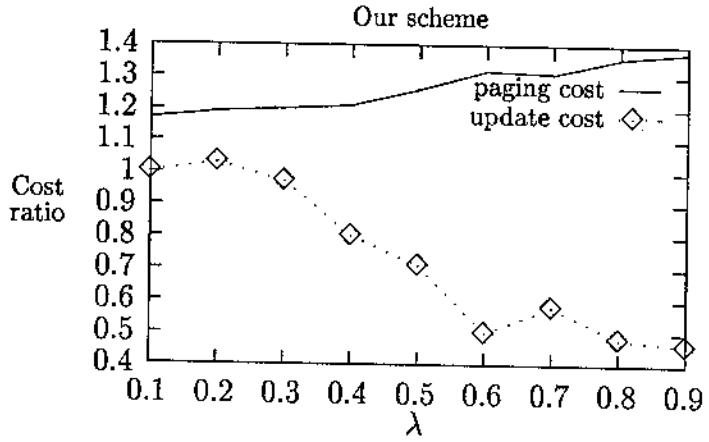


Figure 6.12: Ratio of paging and update costs with those from the minimum cost simulation

based scheme might require an unacceptable amount of changes in the existing infrastructure. Also, in all distance based schemes, the mobile user has to continuously monitor its distance from the point of last update and compare it with the optimal threshold distance to decide whether to update or not. This requires knowledge of the cellular network topology in the part of the mobile user. Hence, implementation of a distance based scheme is not only difficult but also power consuming from the viewpoint of the mobile user. Since our location update scheme does not have any of these drawbacks, it is easy to implement, offers practical solution to the location management problem while performing better.

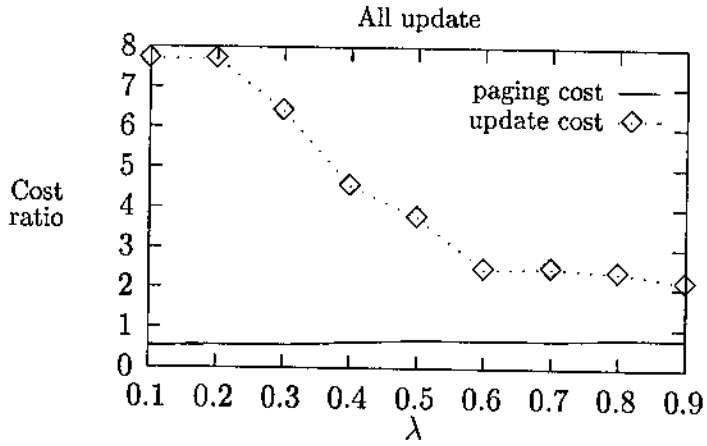


Figure 6.13: Ratio of paging and update costs with those from the minimum cost simulation

Table 6.2: Minimum location management costs and corresponding update strategies for various r and $\frac{C_u}{C_p}$

r	$\frac{C_u}{C_p}$				
	0.1	0.25	0.5	0.75	1.0
0.1	610.5 (00000000)	610.5 (00000000)	610.5 (00000000)	610.5 (00000000)	610.5 (00000000)
0.2	1706.7 (01000000)	1728.3 (00000000)	1728.3 (00000000)	1728.3 (00000000)	1728.3 (00000000)
0.3	2801.3 (01001001)	2943.4 (01000001)	3079.0 (00000000)	3079.0 (00000000)	3079.0 (00000000)
0.4	3795.0 (10101010)	4013.5 (10101010)	4321.1 (01000001)	4500.6 (01000000)	4586.7 (00000000)
0.5	4796.3 (01101101)	5014.9 (10101010)	5379.0 (10101010)	5700.2 (01000101)	5934.1 (01000001)
0.6	5708.0 (01101101)	6023.8 (01101101)	6391.5 (10101010)	6755.7 (10101010)	7112.5 (01000101)
0.7	6749.9 (01101101)	7005.7 (01101101)	7413.3 (10101010)	7777.5 (10101010)	8141.7 (10101010)
0.8	7740.0 (01101101)	7996.4 (01101101)	8422.7 (01101101)	8807.0 (10101010)	9171.1 (10101010)
0.9	8738.9 (01101101)	8904.6 (01101101)	9421.0 (01101101)	9842.8 (10101010)	10207.0 (10101010)

Table 6.3: A sample run of the genetic algorithm ($r = 0.5, \frac{C_w}{C_p} = 0.33$)

Generation no.	Best value	Average	Std. deviation
1	0.954	0.883	0.043
10	0.963	0.900	0.027
20	0.973	0.909	0.030
50	0.973	0.938	0.021
100	0.973	0.964	0.010
200	0.973	0.972	0.001
300	0.973	0.972	0.001

CHAPTER 7

CONCLUSIONS

In this dissertation, efficient resource utilization techniques are proposed in two important areas of wireless mobile computing, namely, *bandwidth management* and *location management.* Under bandwidth management, we investigated resource utilization problems in two important areas, namely, *channel assignment problem* and *wireless multimedia.* Under location management, a resource-optimal update strategy for mobile PCS users is proposed. We summarize our results and findings briefly in the next few sections.

7.1. Load Balancing Heuristics for the Channel Assignment Problem

We have developed dynamic load balancing strategies for the hot cell problem in cellular mobile environment, which can be implemented in a centralized or distributed manner. These strategies constitute two main parts. In the first part, a channel borrowing strategy is proposed where channels are borrowed by a hot cell from suitable cold cells. The suitability of a cold cell as a lender is determined by an optimization function constituting three cell parameters – coldness, nearness and hot cell channel blockade. In the second part, a channel assignment strategy is proposed where the assignment is done on the basis of different priority classes in which the user demands are classified. The relative merits and demerits of the centralized and distributed schemes are discussed both quantitatively and qualitatively. A Markov model for an individual cell is also proposed. One important parameter of our model – the threshold h, below which a cell is classified as hot – is estimated from this model. Exhaustive simulation is also carried out to compare the centralized and distributed schemes. From these results, we conclude that in a region with a large number of

hot cells, the distributed scheme performs better. Simulation experiments showed a significant improvement of our scheme in system performance as compared to many existing schemes like, fixed channel assignment, simple borrowing, directed retry and CBWL.

We have also developed a load balancing scheme to cope with the problem of hot spot, which is a region of adjacent hot cells. A hot spot is conceived as a stack of hexagonal 'Rings', each containing at least one hot cell. In our load balancing approach, a hot cell in 'Ring i' borrows channels from its adjacent cells in 'Ring i+1' with the help of a channel *demand graph*, to ease out its high channel demand. This structured lending mechanism decreases excessive co-channel interference and borrowing conflicts, which are prevented through channel locking in other schemes. Detailed analytical modeling of the system with our load balancing scheme captured certain useful performance metrics like call blocking probability, the probability of a cell being hot and the evolution of the hot spot size. Simulation experiments are carried out proving that our load balancing algorithm is robust under severe load conditions. Also, comparison of our scheme with the CBWL strategy demonstrates that under moderate and even very high load conditions, a performance improvement of as high as 12% in terms of call blockade is achievable with our load balancing scheme.

7.2. Resource Management in Wireless Multimedia

While load balancing strategies attempt to eliminate tele-traffic imbalances in the system by resource (channel) migration, we have also investigated possible ways of improving system capacity by scheduling bandwidth allocation in case of wireless multimedia communications where certain classes of real-time traffic use multiple channels for a single application. The real-time traffic is classified into K classes, where a class i call uses i channels. Two orthogonal QoS parameters are identified, namely, *total carried traffic* and *bandwidth degradation*, such that an improvement of

either of them leads to the degradation of the other. A cost function describing the total revenue earned by the system from *bandwidth degradation* and *call admission* policies is formulated and the optimal policy, maximizing the net revenue earned, is computed.

In most systems, the real-time traffic has preemptive priority over the non-real-time traffic, based on which a novel channel sharing scheme for non-real-time users is proposed. This scheme is analyzed using a Markov modulated Poisson process (MMPP) based queuing model, and the *average queue length*, a QoS metric for non-real-time traffic, is also derived. Simulation experiments demonstrate the sensitivity of the effective revenue earned to the proper choice of degradation cost and revenue in the first case, and also that of the average queue build-up (for the non-real-time traffic packets) to the call arrival rates of both types of traffic and the non-real-time packet arrival rate, in the second case.

For multi-rate traffic requiring contiguous spectrum allocation, a novel scheme called bandwidth compaction (conceptually similar to memory compaction in commercial operating systems) is described. Simulation results show an improvement in spectrum utilization as high as 24% for higher traffic load.

7.3. A Location Update Strategy

An optimal and easily implementable location update scheme is described which considers per-user mobility pattern on top of the conventional zone (LA) based approach. Most of the practical cellular mobile systems partition a geographical region into location areas (LAs) and users are made to update on entering a new LA. The main drawback of this scheme is that it does not consider the individual user mobility and call arrival patterns. Assuming the existence of location areas in the system to start with, our scheme develops an update strategy specifying a set of update areas for each user. A user updates only in his update areas. The update strategy minimizes the average location management cost derived from a user-specific model and call

generation pattern. For a large number of location areas, a genetic algorithm implementation of the above optimization problem is proposed. Experimental results demonstrate that for low residing probability in certain LAs, low call arrival probability and high update cost for a user, a location update in some of those LAs might lead to increase in the average location management cost than no update. By a qualitative comparison with other types of existing location update schemes, it is shown that our "selective update" scheme is easy to implement, offers practical solution to the location management problem while performing better.

7.4. Future Work

There are possibilities of extending the work done in this dissertation in several directions. The load balancing problem is solved assuming static hot spots. The next step will be to investigate the consequences of the hot spot changing its shape and size during the running time of the load balancing algorithm.

In our proposed bandwidth degradation framework, two cost (or revenue) parameters – C_p and C_u – have been used. We would like to conduct some test measurements in an already deployed switch to collect informations like CPU occupancy, message complexity etc, to estimate the actual values of these parameters. We are also working on applying this QoS framework in an existing standard for wireless data, e.g. IS-99 CDMA data standard.

REFERENCES

[CR73] D. C. Cox and D. O. Reudink, "Increasing channel occupancy in large scale mobile radio systems: dynamic channel reassignment", *IEEE Trans. Veh. Technol.*, vol. VT-22, pp 218-222, Nov. 1973.

[DKR93] M. Duque-Anton, D. Kunz and B. Ruber, "Channel Assignment for cellular radio using simulated annealing," *IEEE Trans. Veh. Technol.*, vol. VT-42, Feb. 1993.

[DSJ96] S.K. Das, S.K. Sen, R. Jayaram, "A Dynamic Load Balancing Strategy for Channel Assignment Using Selective Borrowing in Cellular Mobile Environment," *Proceedings of IEEE/ACM Conference on Mobile Computing and Networking* (Mobicom '96), pp. 73-84, Nov. 1996.

[DSJA97] S.K. Das, S.K. Sen, R. Jayaram, P. Agrawal, "A Distributed Load Balancing Algorithm for the Hot Cell Problem in Cellular Mobile Networks", Proceedings of *Sixth IEEE International Symposium on High Performance Distributed Computing (HPDC)*, Portland, Oregeon, August, 1997.

[DSJ97] S.K. Das, S.K. Sen, R. Jayaram, "A Structured Channel Borrowing Scheme for Dynamic Load Balancing in Cellular Networks", Proceedings of *IEEE International Conference on Distributed Computing Systems (ICDCS)*, Baltimore, Maryland, May 1997.

[Eklun86] B. Eklundh, "Channel utilization and blocking probability in a cellular mobile telephone system with directed retry," *IEEE Trans. Communications*, vol. COM-34, No. 4, April 1986.

[ESG82] S. M. Elnoubi, R. Singh, S. C. Gupta, "A new frequency channel assign-

ment in high capacity mobile communication systems", *IEEE Trans. Veh. Technol.*, vol. VT-31, no. 3, Aug. 1982.

[HR86] D. Hong and S. S. Rappaport, "Traffic model and performance analysis for cellular mobile radio telephone systems with prioritized and non-prioritized hand-off procedures," *IEEE Trans. Veh. Technol.*, vol. VT-35, Aug. 1986.

[JR94] H. Jiang and S.S. Rappaport, "CBWL: A new channel assignment and sharing method for cellular communication systems," *IEEE Trans. Veh. Technol.*, vol. 43, No. 2, May 1994.

[KE89] J. Karlsson and B. Eklundh, "A cellular mobile telephone system with load sharing – an enhancement of directed retry," *IEEE Trans. Communications*, vol. 37, No. 5, May 1989.

[Lee96] W.C.Y. Lee, *Mobile Cellular Telecommunication Systems, Analog and Digital Systems*, Second Edition, McGraw-Hill, 1996.

[Mac79] V. H. Macdonald, "Advanced mobile phone service: The cellular concept," *Bell Syst. Tech. J.*, vol. 58, pp. 15-41, Jan 1979.

[NEL96] R. Nelson, *Probability, Stochastic Processes and Queueing Theory*, Springer-Verlag, 1996.

[TI88] J. Tajima and K. Imamura, "A strategy for flexible channel assignment in mobile communication systems," *IEEE Trans. Veh. Technol.*, vol. VT-37, May 1988.

[TJ91] S. Tekinay and B. Jabbari, "Handover and channel assignment in mobile cellular network", *IEEE Communication Magazine*, Nov. 1991.

[WLR93] M.H. Willebeek-LeMair and A.P. Reeves, "Strategies for dynamic load balancing on highly parallel computers," *IEEE Trans. on parallel and distributed systems*, vol. 4, No. 9, Sept. 1993.

[ZY89] M. Zhang and T. S. Yum, "Comparisons of channel assignment strategies in cellular mobile telephone systems", *IEEE Trans. Veh. Technol.*, Vol. 38, Nov 1989.

[Kur93] J. Kurose. "Open Issues and Challenges in Providing Quality of Service Guarantees in High-Speed Networks", *Computer Communication Review*, 23 (1), January 1993.

[OKS96] C. Oliveira, J.B. Kim, T. Suda. "Quality-of-Service Guarantee in High-Speed Multimedia Wireless Networks", *IEEE International Communications Conference 1996*, Dallas Texas, pages 728-734.

[AN94] A.S. Acampora, M. Naghshineh. "Control and Quality-of-Service Provisioning in High-Speed Microcellular Networks", *IEEE Personal Communications Magazine*, Vol. 1, No. 2, Second Quarter 1994.

[AN95] A.S. Acampora, M. Naghshineh, "QOS Provisioning in Micro-Cellular Networks Supporting Multimedia Traffic", *IEEE INFOCOM*, 1995.

[NSA96] M. Naghshineh, M. Schwartz, A.S. Acampora. "Issues in Wireless Access Broadband Networks", *Wireless Information Networks*, edited by J.M. Holtzman, Kluwer Academic Publishers, 1996.

[NT95] D. Newman, K. Tolly. "Wireless LANs: How Far? How Fast?", *Data Communications on the Web*,

`http://www.data.com/Lab\_Tests/Wireless\_LANs.html`

[RW94] D. Raychaudhuri, N.D. Wilson. "ATM-Based Transport Architecture for Multiservices Wireless Personal Communications Networks", *IEEE Journal on Selected Areas in Communications*, Vol. 12(8), October 1994.

[Sur96a] S. Singh, "Quality of Service Guarantees in Mobile Computing", to appear in *IEEE Journal of Computer Communications.*

[PTP94] S. Papavassiliou, L. Tassiulas, P. Tandon, "Meeting QOS requirements in a cellular network with reuse partitioning", *IEEE Infocom*, 1994.

[Sur96b] K. Seal, S. Singh, "Loss Profiles : A Quality of Service measure in mobile computing", *Journal of Wireless Computing*, Vol. 2, pp 45-61, 1996.

[PTP94] S. Papavassiliou, L. Tassiulas, P. Tandon, "Meeting QOS requirements in a cellular network with reuse partitioning", *Proceedings of INFOCOM '94*, pp 4-12.

[RP96] S. Rappaport, C. Purzynski, "Prioritized resource assignment for mobile cellular communication systems with mixed services and platform types", *IEEE Transactions on Vehicular Technology*, vol. 45, no. 3, pp 443-457, Aug. 1996.

[LEE96] *Mobile Cellular Telecommunication Systems, Analog and Digital Systems*, W.C.Y. Lee, Second edition, McGraw-Hill, 1996.

[RS90] *Multiple Access Protocols, Performance and Analysis*, R. Rom and M. Sidi, Springer-Verlag, 1990.

[GW73] *Introduction to Optimization Theory*, B.S. Gottfried and J. Weisman, Prentice-Hall, 1973.

[PL94] D. Plassmann, "Location management strategies for mobile cellular networks of 3rd generation", *Proc. of IEEE Vehicular Technology Conference*, June 1994.

[RY95] C. Rose and R. Yates, "Minimizing the average cost of paging under delay constraints", *ACM Journal of Wireless Networks*, 1995, pp. 211-219.

[ROS95] C. Rose, "Minimization of Paging and Registration Costs Through Registration Deadlines", *Proc. of IEEE Vehicular Technology Conference*, pp. 735-739, 1995.

[BKS95] A. Bar-Noy, I. Kessler, M. Sidi, "Mobile Users: To update or not to update?", *ACM Journal of Wireless Networks*, 1995, pp. 175-185.

[BK93] A. Bar-Noy and I. Kessler, "Tracking Mobile Users in Wireless Communication Networks", *Proc. INFOCOM*, 1993, pp. 1232-1239.

[MHS94] U. Madhow, M.L. Honig, K. Steiglitz, "Optimization of Wireless Resources for Personal Communications Mobility Tracking", *Proc. of IEEE INFOCOM*, pp. 577-584, 1994.

[AH95] J.S.M Ho, I.F. Akyildiz, "Mobile user location update and paging under delay constraints", Wireless Networks, 1995, pp. 413-425.

[TGM88] R. Thomas, H. Gilbert, G. Maziotto, "Influence of the moving of the mobile stations on the performance of a radio mobile cellular network", *Proc. of the Third Nordic Seminar on Digital Land Mobile Radio Communications*, Sept. 1988.

[XTG93] H. Xie, S. Tabbane, D. Goodman, "Dynamic location area management and performance analysis", *Proc. of IEEE Vehicular Technology Conference*, May 1993.

[DS96] S.K. Das and S.K. Sen, "Adaptive Location Prediction Strategies Based on a Hierarchical Network Model in Cellular Mobile Environment", Second International Mobile Computing Conference (IMCC), March 1996, pp. 131-140.

[OOYM91] S. Okasaka, S. Onoe, S. Yasuda, A. Maebara, "A new location updating method for digital cellular systems", *41st IEEE Vehicular Technology Conference*, May 1991.

[DAV90] David Munoz-Rodriguez, "Cluster paging for travelling subscribers", *Proc. of IEEE Vehicular Technology Conference*, May 1990.

[SMHW92] I. Seskar, S. Marie, J. Holtzman, J. Wasserman, "Rate of location area

updates in cellular systems", *Proc. of the 42nd Vehicular Technology Conference*, May 1992.

[BIV92] B. R. Badrinath, T. Imielinski, A. Virmani, "Locating strategies for personal communication networks", Workshop on Networking of Personal Communications Appliances, Dec, 1992.

[MBR94] S. Madhavapeddy, K. Basu, A. Roberts, "Adaptive paging algorithms for cellular systems", *Nortel Technical Report*, 1995.

[SM96] S. Subramanian, S. Madhavapeddy, "System Partitioning in a Cellular Network", *Proc. of IEEE Vehicular Technology Conference*, 1996.

[GKS96] D. Goodman, P. Krishnan, B. Sugla, "Minimizing queueing delays and number of messages in mobile phone location", *ACM Mobile Networks and Applications (MONET) journal*, vol. 1, pp 39-48, 1996.

[BKN96] A. Bar-Noy, I. Kessler, M. Naghshineh, "Topology-based tracking strategies for personal communication networks", *ACM Mobile Networks and Applications (MONET) journal*, vol 1, pp 49-56, 1996.

[YRRB96] R. Yates, C. Rose, S. Rajagopalan, B. Badrinath, "Analysis of a mobile-assisted adaptive location management strategy", *ACM Mobile Networks and Applications (MONET) journal*, vol 1, pp 105-112, 1996.

[ROS80] S. Ross, *Introduction to Probability Models*, Academic Press, 1980.

[MOT96] R. Motwani and P. Raghavan, *Randomized Algorithms*, Cambridge University Press, 1995.

[GIF78] W.C. Giffin, *Queueing, Basic Theory and Applications*, Grid Inc, Ohio, Columbus, 1978.

[MIC95] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, Springer, 1995.