RESOURCE ALLOCATION IN DIGITAL MOBILE SYSTEMS

By

Wan Wai Leung

A Thesis

submitted in partial fulfillment of the requirements

for the Degree of Master of Philosophy

Division of Information Engineering

The Chinese University of Hong Kong

June 1998

Acknowledgement

I would like to express my sincere gratitude to my supervisor, Prof. Wing Shing Wong, for his guidance in this research work. His valuable advice helps me a lot in handling technical problems. Besides, I would also like to thank Mr. Ho Leung Chan for his patience in dealing with mathematical problems with me.

Abstract

With the advancement in radio transmission technology and compression techniques such as the MPEG (The Moving Picture Experts Group) standard, it is now possible to transmit bandwidth demanding multimedia data, such as high quality video streams, over the air. Besides traditional voice data, multimedia data can also be transmitted to and from mobile users via a radio system.

In a hybrid TDMA/FDMA digital communication system such as the GSM system, *Global System for Mobile Communication*, a channel is defined by a tuple consisting of the *frequency band number* and the *slot number*. For a normal voice connection, an idle time slot will be assigned upon a channel request in uplink direction and correspondingly one in downlink direction. However, this would not be the case for the transmission of multimedia data that demand multi-rate connections. To accommodate these multi-rate data, multiple slots have to be assigned to achieve the required rate for transmission. The problem of how to assign the slots to make efficient frequency usage arises if the slots can be chosen from different frequency bands. With the capability of transmitting and receiving data through multiple frequency bands simultaneously in a mobile unit, it is desirable to achieve compact and efficient slot assignments to keep the number of frequency bands used small. The smaller the number of frequencies is being used in a cell, the lower is the amount of interference generated. As a consequence, more efficient frequency reuse can be attained.

Two optimal slot assignment schemes, with optimality defined either in the sense of frequency usage or processing time, are described and analysed in this thesis. These schemes demonstrate the need of trading processing time for efficiency in frequency usage. In view of this, a heuristic slot assignment algorithm called the Buddy algorithm which performs well in both aspects will be presented. This algorithm, with an inference property, takes advantage of the binary clustering of idle time slots to speed up the slot assignment process. The performance of the Buddy algorithm and the two optimal assignment schemes are compared by simulations under both fixed and dynamic channel assignment environments. It is found that the Buddy algorithm is time efficient while keeping the number of frequencies used small.

摘要

隨著資料壓縮技術之發展，例如 MPEG 標準之出現，在空中傳送對頻帶寬需求甚高的多媒體資料，如高質素視像流，已並非是憑空想像的了。除了傳統的話音傳送外，多媒體資料亦能透過無線電系統來往傳送至移動用戶處。

在一個混合了 TDMA 及 FDMA 概念之數字化通訊系統，如 GSM ，當中，一個頻道是由一個頻率號碼及一個時隙號碼所組成。在普通的話音傳輸中，個別的閒置時隙會被分配作使用。但是這種分配方式並不適用於需要多種傳送速度的多媒體資料傳輸。爲配合這種傳輸之需求，多時隙之分配便是無可避免的了。若每部移動手機均能同時以多個頻率作資料的傳輸和接收，在多時隙分配中，有限的頻譜便能得到更緊密及更有效的使用。在每一個蜂窩小區中，愈少頻率被使用，便可產生愈少的干擾。換言之，更有效的頻率覆用便能加以採用。

這份論文首先會討論及分析兩種分別有利於頻率應用及處理時間的時隙分配方案。這兩種方案顯示出如要提高頻率效率，在處理時間中所作出的讓步是無可避免的。有見及此，這份論文提出了一個啓發性的時隙分配方法。這方法採納了二進制閒置時隙組合法，能於合理時間內作出高效能的分配。在論文末段，一個模擬的數字式蜂窩小區系統會被用作這啓發性的分配方法在性能上之評估。結果發現，這種啓發性的時隙分配方法能於短時間內作出有效率的資源分配，盡量減低頻率之使用，以容納更多用戶。

Contents

Chapter 1

Introduction

1.1 Wireless Multimedia System

Advancements in wireless transmission technology and data compression techniques, such as MPEG [1], MPEG2 [2] [3] and H.263 [4] coding shemes, make the transmission of multimedia data over the air now possible. Many challenges are to be addressed before a robust and efficient wireless multimedia system can emerge. Quality of service (QOS) of the transmission is one of the challenges as the error rate in a wireless network is much higher than that in a fixed network. Moreover, modifications in multiple access scheme are required to cope with the high data rate requirement of multimedia traffic. Various architectures such as wireless ATM (Asynchronous Transfer Mode) are being developed to facilitate the transmission of multi-rate data. However, there are many disputes between the pros and cons of a wireless ATM versus a wireless LAN architeture (see [5]).

1.2 Motivation of this thesis

The concept of a wireless multimedia system is now a much investigated topic. While the optimal design for a wireless multimedia system is still in debate, it is possible to obtain a simple implementation of it by means of a hybrid TDMA/FDMA system similar to the GSM standards. (For simplicity, the issue of frequency hopping is ignored.) To achieve the multi-rate channel effect by means of a hybrid TDMA/FDMA system, it is necessary to assign multiple slots in multiple frequency bands in both uplink and downlink directions for each connection. It is reasonable to expect that the maximum number of frequencies used simultaneously in each connection is constrained by the physical design of the transceiver in a mobile unit. The problem considered here is how to assign slots in order to keep the intercell interference and hence the blocking probability low for such a system. The smaller number of frequencies are being used in a cell, the lower is amount of intercell interference generated. As a consequence, more efficient frequency reuse can be attained based on various channel assignment schemes [6], [7].

An exhaustive search over all frequency band combinations is a promising approach to come up with a compact slot combination. However, the search can be accomplished in reasonable time only if the system is small. It can be argued that search time and the compactness of a solution are two contradicting factors. We have to trade off one for the other. Figure 1.1 shows the blocking probability of two slot allocation schemes optimising in searching time and frequency usage respectively. One natural question arises – is it possible to find an allocation strategy that requires short processing time yet gives rise to a low blocking probability depicted by the dotted line in figure 1.1?

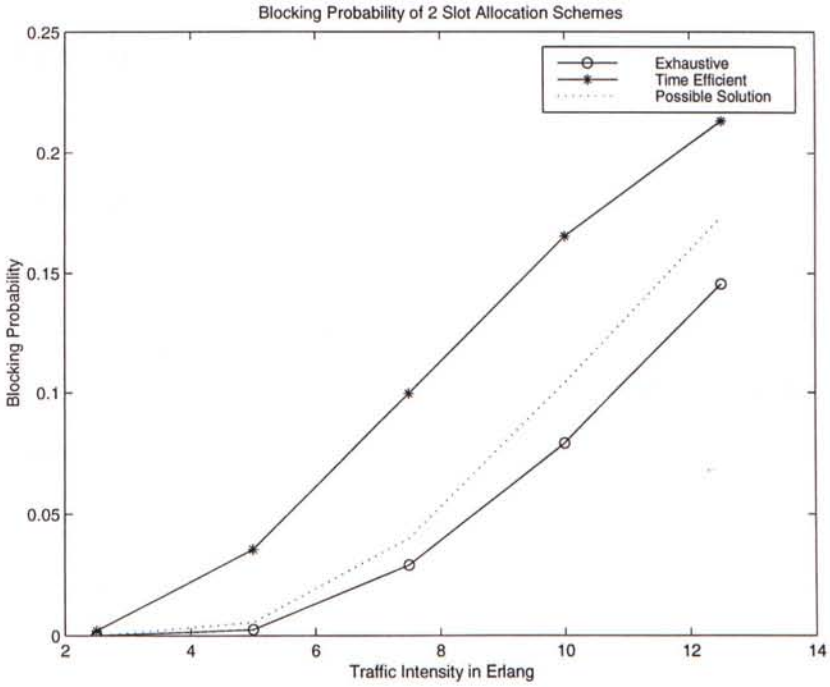


Figure 1.1: Blocking Probability vs Traffic Intensity

In this thesis, three slots assignment schemes are presented. Given a connection requiring multiple time slots, a locally optimal solution with the smallest number of frequencies used can be obtained by searching over all the frequency bands. This approach, however, is very time consuming when the total number of frequency bands available is large. In view of this shortcoming, a heuristic algorithm called the Buddy algorithm is introduced. The goal is to speed up each allocation process and keep the number of frequencies used small. The basic idea of the Buddy algorithm is to reduce the number of idle slot combinations that need to be considered in different frequency bands by managing the idle time slots in a binary fashion.

1.3 The theme of this thesis

1.3.1 System Model and Assumptions

The environment considered in this thesis is a hybrid TDMA/FDMA digital cellular system. The area of coverage is covered by a cluster of cells. The decision on which frequency bands can be used in a particular cell in order to satisfy the co-channel interference constraint is assumed to be handled by some dynamic channel assignment schemes such as the *Channel Segregation* scheme [8]. Each frequency band is divided into slots in the time domain to provide multiple access. Instead of occupying a single slot as that for voice data, multi-rate data connection is accommodated by means of multiple slots within a frame in one or more frequency bands. If a single slot supports a bit rate of m *bps*, the user having the allocation shown in figure 1.2 has a bit rate of $6m$ *bps* through the use of 2 frequencies. However, the allocated slots may not be evenly distributed within a frame. Both the transmitter and receiver have to buffer the data in one frame to maintain a constant bit rate stream.

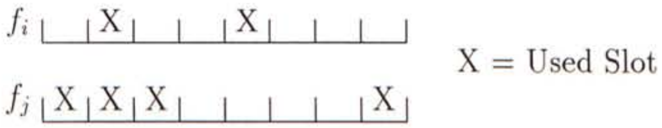


Figure 1.2: Within a Frame

In this thesis, we consider a cellular system with the following parameters and assumptions:

- The cells are in hexagonal shape with reuse factor 1/7.
- The system is a loss system that blocked connection requests are dropped.

- In the DCA environment, the grant of use of channels to a particular cell is on a per frequency band basis.
- The total number of frequency bands available in a cell is denoted by C.
- The total number of slots per frequency band in a frame is denoted by N.
- The maximum number of frequency bands allowed to be used simultaneously in a mobile unit is denoted by M.
- The requested bit rates are multiples of the bit rate achieved by a single slot.
- The number of slots requested in a connection request is denoted by D and is uniformly distributed between 1 to $2N$.
- The call arrivals for every mobile unit follow Poisson distribution with an average rate λ per second.
- The holding time of a connection is exponentially distributed with mean $1/\mu$ second.

1.3.2 Outline of the thesis

This thesis will focus on the descriptions, analyses and comparisons of three different slot allocation strategies: the No-Split algorithm, the Best Fit algorithm and the Buddy algorithm. They are compared by simulations under both the FCA and DCA environments. The comparison is based on blocking probability, processing time and frequency usage.

In the next chapter, the fixed and dynamic channel assignment schemes and two different multiple access techniques (FDMA, TDMA) are overviewed. One of the most popular TDMA/FDMA systems, GSM, is briefly described.

After that, two optimal slot allocation strategies either in the sense of processing time or efficient use of frequency bands are described and analysed. In chapter 5, a heuristic algorithm called the Buddy algorithm together with its inference property is introduced. Simulations are used to compare the three different slot allocation strategies and the Buddy algorithm is shown to be not only efficient in processing time but also in frequency usage with minimal tradeoff. Finally, in chapter 7, a case study is investigated. The proposed Buddy algorithm applies to the transmission of video streams coded in H.263 using GSM data channels.

Chapter 2

Overview of TDMA/FDMA Digital Cellular Systems

2.1 The Cellular Concept

The cellular concept was a major breakthrough in tackling the problem of user capacity in wireless telecommunication system. Without the cellular concept, a large coverage area is covered by a single, high power transimitter located on a tall tower. This greatly limits the number of users simultaneously supported by the system. In a cellular system, the whole coverage area is divided into small geographic areas called *cells* as shown in figure 2.1. A group of radio channels is used for data transmission within a cell by a low power transmitter. Owing to the attenuation of radio signals over distance, this group of radio channels can be reused in cells situated far enough from this cell. By using this concept, we can serve a large number of users simultaneously with a small group of radio channels. The hexagonal cell shape shown in figure 2.1 is conceptual and is a simplistic model of the radio coverage for each base station. Cells with the same letter use the same set of frequencies. The radio channels are now collectively shared by a cluster of 7 cells with different letters and the cell cluster is replicated over the whole coverage area. The *frequency reuse factor* here is 1/7 since each cell now contains one-seventh of the total number of available channels.

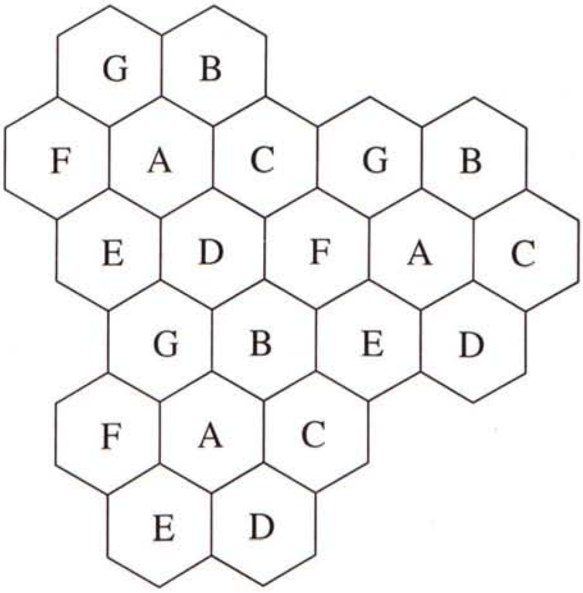


Figure 2.1: Clusters of Cells with reuse factor 1/7

The cellular concept is now being adopted by most wireless telecommunication systems such as the GSM system, *Global System of Mobile communications* and the PHS, *Personal Handyphone System*. Different channel assignment schemes [6] are employed by different systems to effectively provide quality telecommunication services to a large number of users.

2.2 Channel Assignment Strategies

2.2.1 Fixed Channel Assignment

In a *fixed channel assignment (FCA)* strategy, each cell is allocated a predetermined set of radio channels. The determination of how many and which radio channels are used in a cell is normally based on the traffic density of the cell. A busy cell may be assigned a larger set while the others may be assigned a smaller set of radio channels. This strategy is easy to implement but not very effective in the use of radio channels. Any call attempt can only be served by the unused channels in the cell it resides in. Once all the channels are occupied, the call is *blocked* even there are unused channels in neighbouring cells. The fixed channel assignment strategy performs well if the traffic in the system is relatively static. However, it can not cope with the short-term temporal and spatial variations of traffic in the system to attain high channel efficiency. Several variations of the fixed assignment strategy such as the *borrowing strategy* [9] and the *reuse partitioning strategy* [10] exist in order to make the channel usage more efficient.

2.2.2 Dynamic Channel Assignment

To overcome the low trunking capacity problem of FCA , different *dynamic channel assignment (DCA)* strategies have been studied over the past 20 years. In contrast to FCA, there is no fixed relationship between channels and cells in DCA. Channels are dynamically allocated for use in a cell upon a call request. Complicated algorithm [11] has to be applied to take future blocking probability, the reuse distance of the channel, etc., into account in deciding which channel to allocate. After a call is completed, its channel is returned to the central pool.

This assignment strategy can increase the trunking capacity of the system since all the available channels are now accessible to all of the cells.

Dynamic channel assignment strategies can basically be categorized into either *centralized* or *distributed* [12] approach according to who makes the decision on channel assignment.

In centralized DCA schemes, a channel from the central pool is assigned to a call for temporary use by a centralized controller. A variety of cost functions [13] are developed for the selection of candidate channels. For example, in the RING strategy [14], the channel with the highest usage in the co-channel set is selected for use. This strategy obviously aims at maximizing the frequency reuse efficiency which is a shortcoming of DCA when compared with FCA. However, the high centralization overhead [15] [16] of centralized DCA makes it not attractive for implementation in microcellular systems. The distributed DCA schemes then become proper choices.

In distributed DCA schemes, each base station gathers information on either the channel usage pattern in neighbouring cells or signal strength of channels [17] for the channel selection process instead of having a central controller. These schemes are appealing in the sense that only local information is needed for decision making resulting in a self-organized system.

2.3 Multiple Access Techniques

2.3.1 Introduction to Multiple Access

Multiple access schemes are used to allow many users to share the limited radio spectrum simultaneously. *Frequency division multiple access* (FDMA) and *Time* *division multiple access* (TDMA) are two major access techniques for sharing the available bandwidth in a wireless communication system.

2.3.2 Frequency Division Multiple Access - FDMA

In *frequency division multiple access* (FDMA), each user is allocated a unique frequency band. During the period of the call, no other user can share the same frequency band. Figure 2.2 shows the concept of the FDMA scheme. N simultaneous users can get served if there are N unique frequency bands for use. Once a channel is acquired, data can be transmitted continuously as the channel is now dedicated to this transmission. Therefore, applications demanding continuous transmission can get benefit from this multiple access scheme. On the other hand, an FDMA channel sits idle when the channel is not in use in the case of bursty transmission. This situation results in a waste of the scarse radio spectrum.

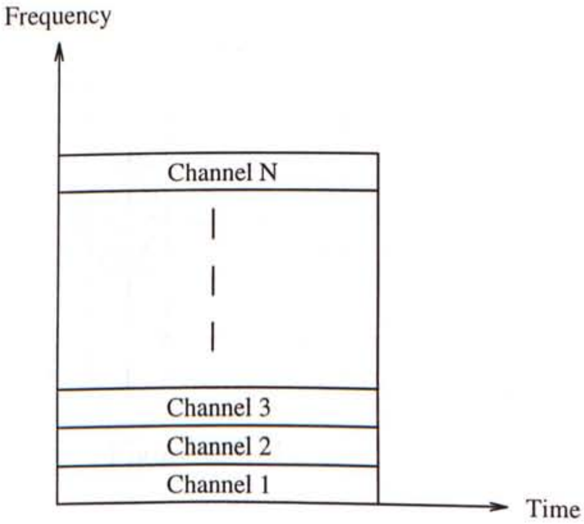


Figure 2.2: FDMA Scheme

2.3.3 Time Division Multiple Access - TDMA

Time division multiple access (TDMA) [18] systems divide the radio spectrum into time slots. In each time slot, only one user is allowed to transmit or receive. A channel here is actually a time slot that repeats cyclically as shown in figure 2.3. The transmissions from different users are interlaced and constitute a repeating frame structure. A single carrier frequency is then shared by several users with nonoverlapping time slots. In contrast to FDMA, data transmission in a TDMA system is not continuous, but occurs in bursts. In most of the time, a mobile unit can be turned off leading to a save in battery consumption when not in use. In addition, *mobile assisted handoff* (MAHO) can be carried out by mobile units as they can now listen on idle slots when they are not transmitting.

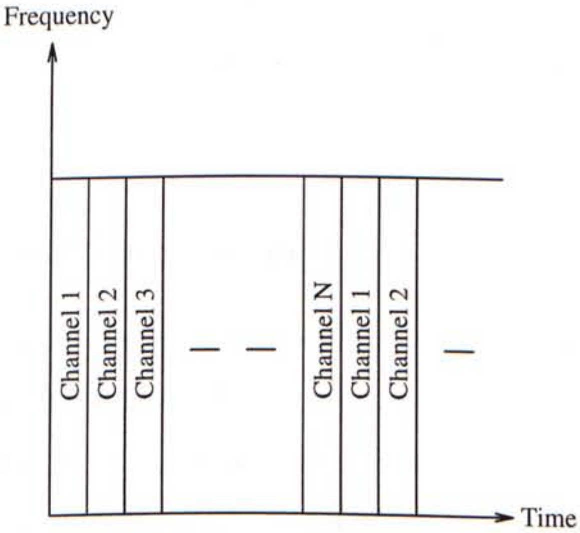


Figure 2.3: TDMA Scheme

TDMA systems are not without disadvantages. They are more complicated than FDMA systems as synchronization is required in TDMA systems because of bursty transmissions.

2.4 A TDMA/FDMA System – GSM

2.4.1 Global System for Mobile

Global System for Mobile (GSM) [19] is a second generation cellular system introduced in 1991. It was developed to solve the fragmentation problems of the first cellular systems in Europe. Before GSM, different cellular standards were used in European countries. This prohibited the use of a single mobile unit throughout Europe. Now, many nonEuropean countries in South America, Asia and Australia have adopted GSM to provide telecommunication services.

GSM services are classified into *teleservices* or *data services*. Teleservices include standard mobile telephony and mobile-originated or base-orginated traffic while data services include computer-to-computer communication and packet-switched traffic.

2.4.2 GSM radio subsystem

Two bands of 25 MHz have been set aside for GSM system use in all member countries. The 890-915 MHz band is for mobile-to-base transmissions and the 935-960 MHz band is for base-to-mobile transmissions. Each 25 MHz band is divided into 200 kHz wide channels called ARFCNs (Absolute Radio Frequency Channel Numbers). A pair of forward and reverse channel separated in frequency by 45 MHz is denoted by a ARFCN. With a guard band of 100 kHz at both the upper and lower ends of the GSM spectrum, 124 radio channels are then implemented. Using the TDMA scheme, each of these 124 radio channels consists of 8 time slots (TS) shared by eight users at the same time as shown in figure 2.4. There are thus a total of 992 traffic channels within GSM. Table 2.1 extracted from [20] summarizes the GSM air interface.

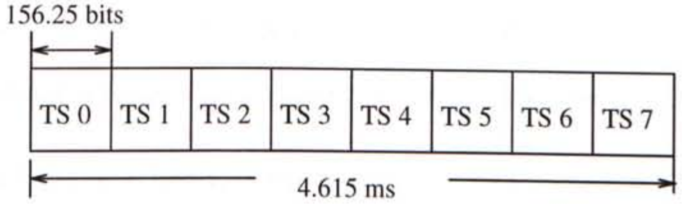


Figure 2.4: The Speech DCCH Frame

Parameter	**Specifications**
Reverse Channel Frequency	890-915 MHz
Forward Channel Frequency	935-960 MHz
AFRCN Number	0 to 124 and 975 to 1023
Tx/Rx Frequency Spacing Tx/Rx Time Slot Spacing	45 MHz 3 Time slots
Modulation Data Rate	270.833333 kbps
Frame Period	4.615 ms
Users per Frame (Full Rate)	8
Time slot Period	576.9 μs
Bit Period	3.692 μs
Modulation	0.3 GMSK
ARFCN Channel Spacing	200 kHz
Interleaving (max. delay)	40 ms
Voice Coder Bit Rate	13.4 kbps

Table 2.1: GSM Air Interface Specifications Summary

There are two types of GSM logical channels, called *traffic channels* (TCH) and *control channels* (CCH) carrying user speech or data and control signals respectively. The GSM traffic channels may be either full-rate of half-rate. User data is contained within one TS per frame in full-rate TCH. When transmitted as half-rate, two half-rate channel users would share the same time slot but transmit alternately during every other frame. The following full rate and half rate traffic channels are supported in GSM.

- **Full-Rate Speech Channel (TCH/FS)**

  User speech is digitized at a raw data rate of 13 kbps. Along with the GSM channel coding, the full-rate speech channel carries 22.8 kbps.

- **Full-Rate Data Channel for 9600 bps (TCH/F9.6)**

  Raw user data is sent at 9600 bps. With additional forward error correction coding, the 9600 bps data is sent at 22.8 kbps.

- **Full-Rate Data Channel for 4800 bps (TCH/F4.8)**

  Raw user data is sent at 4800 bps. With additional forward error correction coding, the 4800 bps data is sent at 22.8 kbps.

- **Full-Rate Data Channel for 2400 bps (TCH/F2.4)**

  Raw user data is sent at 2400 bps. With additional forward error correction coding, the 2400 bps data is sent at 22.8 kbps.

- **Half-Rate Speech Channel (TCH/HS)**

  User speech is digitized at a raw data rate of 6.5 kbps which is sampled at a rate half that of the full-rate channel. Along with the GSM channel coding, the half-rate speech channel carries 11.4 kbps.

- **Half-Rate Data Channel for 4800 bps (TCH/H4.8)**

  Raw user data is sent at 4800 bps. With additional forward error correction coding, the 4800 bps data is sent at 11.4 kbps.

- **Half-Rate Data Channel for 2400 bps (TCH/H2.4)**
  Raw user data is sent at 2400 bps. With additional forward error correction coding, the 2400 bps data is sent at 11.4 kbps.

There are three main types of control channel in the GSM system. These are the *broadcast channel* (BCH), the *common control channel* (CCCH) and the *dedicated control channel* (DCCH). Unlike traffic channels which are duplex, BCHs only use the forward link. The BCH provides synchronization for all mobiles within the cell. Mobile units in neighbouring cells occasionally monitor the BCH to assist in MAHO (*Mobile Assisted Handoff*) decisions. CCCHs are the most commonly used control channels and are used to page, assign signaling channels to and receive connection requests from users. Both the BCH and CCCH occupy TS 0 of every GSM frame. Whenever the frame is not used by the BCH or is idle, CCCH comes into use. Like TCHs, DCCHs are bidirectional and may exist in any time slot on any ARFCN except TS 0. The DCCH carries supervisory data between the mobile unit and the base station during a call and provides signal services required by users in call setup.

The three types of control channels, together with the traffic channels constitute the framework of the GSM system. Both telephony and data services are readily provided by the GSM system.

Chapter 3

Multi-rate Data in TDMA/FDMA Digital Cellular Systems

3.1 Incorporation of Multimedia Data

Most well-developed wireless telecommunication systems such as the GSM and the PHS are mainly designed to provide mobile telephony services and data transmission at low rate. In hybrid TDMA/FDMA systems, the transmission rate is constrained by the slot duration within a frame. The more slots defined in a frame, the lower is the data rate achieved by a single slot connection. Take an example from the GSM system, to support speech, a pair of slots in speech channel providing transmission in forward and reverse direction satisfies the need of quality telephony service at a rate of 13 kbps. Apart from speech channels, a single slot can only support the transmission of data at a maximum rate of

9600 bps in a data channel. To cope with the need of high rate transmission, it becomes necessary to provide multi-slot connections.

It would be an easy task to have consecutive slots in a single frequency band assigned to a connection requiring high transmission rate if users depart in a "first in last out" pattern as shown in figure 3.1. In real situation, the arrivals and departures of users are random in nature. This random nature makes it difficult to have a consecutive slot assignment in a single frequency band without sacrificing efficiency in frequency usage. Departures of users may leave a frequency usage pattern shown in figure 3.2. A request of 6 consecutive slots for transmission would only be satisfied by the use of a new frequency band. Thus, many idle slots will be left unused.

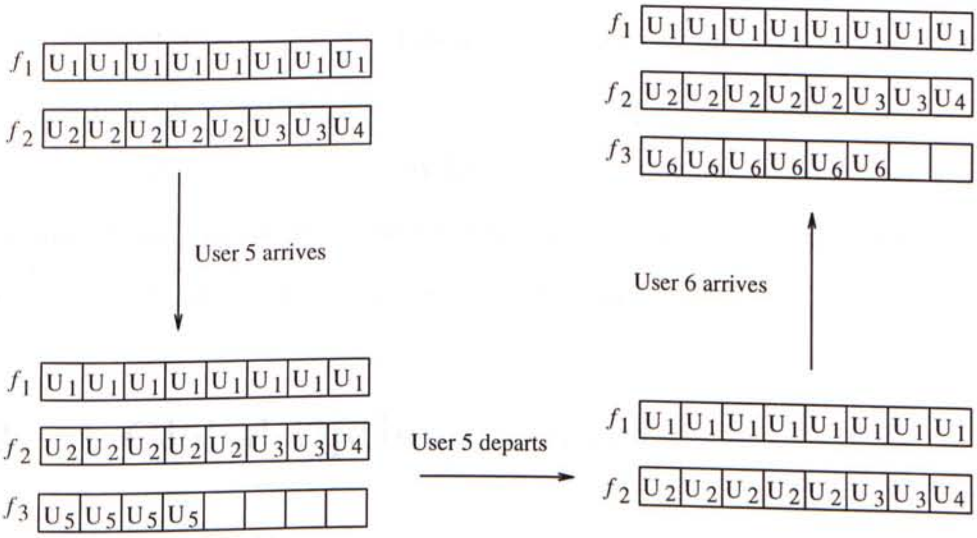


Figure 3.1: Frequency Usage with First In Last Out Departure

With the advancement in technology, it is possible to implement transceivers which can transmit or receive in multiple frequency bands simultaneously. This type of transceivers provides more flexibility in the choice of slot combinations

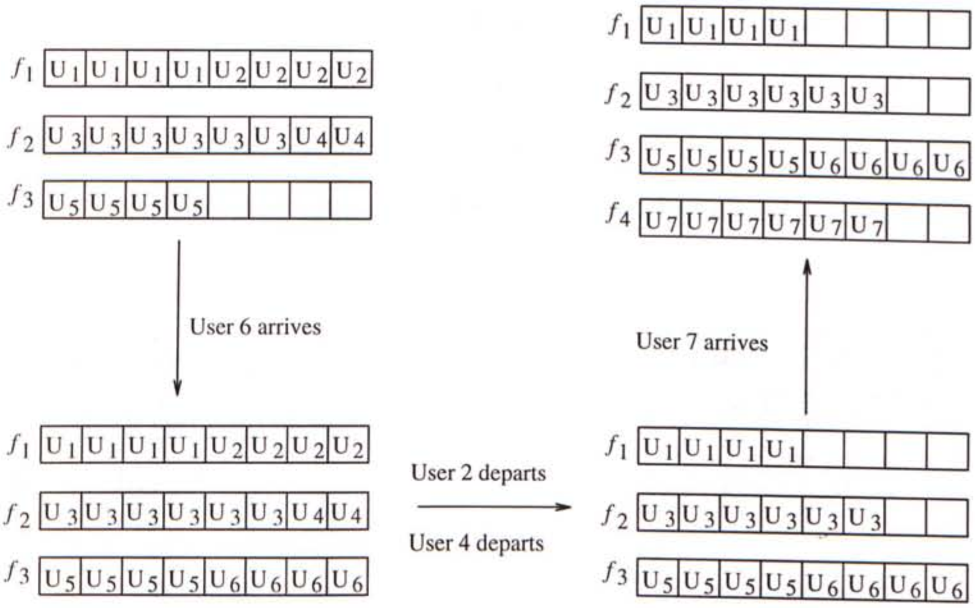


Figure 3.2: Frequency Usage with Random Departure

to satisfy connection requests. More unused slots can be selected before a new frequency band is acquired.

The assignment of idle slots may be an easy task or may need a large amount of computation. This depends on the current frequency usage pattern and how many frequency bands are supported by the mobile unit.

3.2 A Global Optimal Strategy

3.2.1 Channel Rearrangement

One way to save computation time in slot allocation is to rearrange the radio channels for every departure. Those users using radio channels after the current departing user in the usage map are shifted to fill the idle slots as shown in figure 3.3. With this rearrangement, channel utilization is the highest as there is no longer any unnecessary idle slot. Frequency usage can then be restructured into the one with first in last out departures shown in figure 3.1. Whether a new frequency is needed or not can be answered quickly because only the last frequency band has to be examined. Thus, the slot allocation time is minimized.

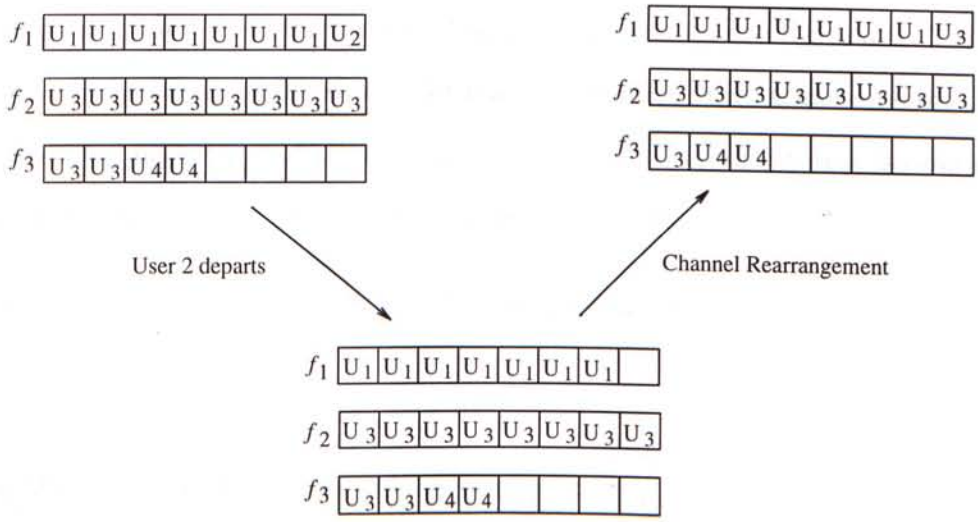


Figure 3.3: Channel Rearrangement

However, it is not always possible to obtain the highest frequency usage patten simply by shifting users to the front. If a mobile unit can only transmit and receive in two frequency bands at the same time, the rearrangement performed in figure 3.3 is not a valid one. User 3 is forced to use three different frequency bands for transmission which is out of its capability. Thus, a special algorithm should be carried out in the rearrangement of radio channels.

3.2.2 Analytical Performance Analysis of a Special Case

In general, a Markov chain study can be carried out for the channel rearrangement strategy. However, states in this Markov chain would depend on the number of users and the slot requirement of each user within the system. This Markov chain is too complex to analyse. To illustrate the approach, the following is a simplified special example. Users in the system are limited to 2 types. Type one users request for single slot transmission for voice transmission. Type two users request 3 slots for data transmission. The cell of interest is operating in a FCA environment with the following parameters:

- The cell is assigned with $C = 2$ frequency bands.
- Number of slots per frequency band is $N = 3$.
- Poisson arrivals with rate λ per second.
- Holding time is exponentially distributed with means $1/\mu_1$ s and $1/\mu_2$ s for type 1 and type 2 users respectively.
- An arrival is a type 1 or type 2 user with equal probability.
- Each mobile unit can handle simultaneous transmission using 2 frequency bands.

With these parameters, the scenerio shown in figure 3.3 is not going to happen. All users are transmitting using no more than 2 frequency bands. This further simplifies the following analysis. We can assume the on-going users are always packed in the most compact form without idle slots in it.

A Markovian model is used to analyse the system. Each state in this model is denoted by an ordered pair $(t_1,\ t_2)$. t_1 and t_2 denote the number of type 1 users and type 2 users in the system respectively. The system can now be at any one of the 12 states at any time and the transition diagram is shown in figure 3.4.

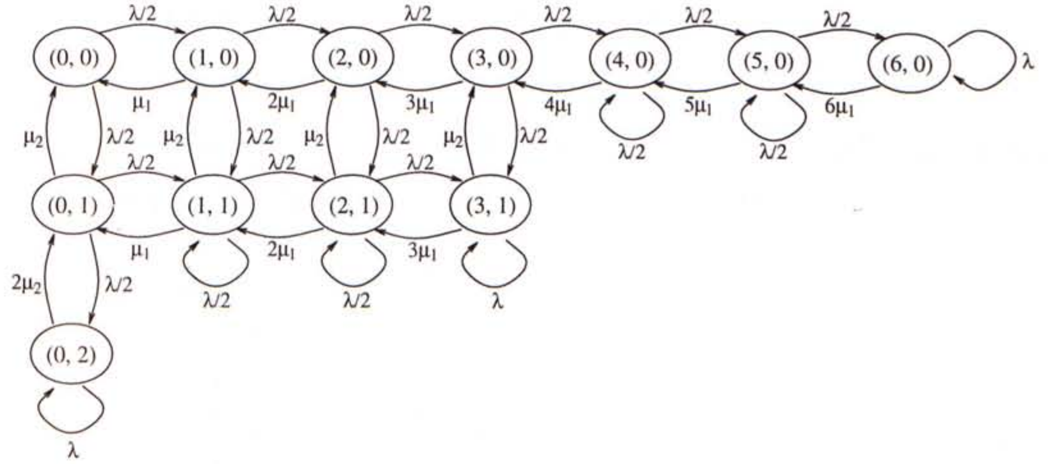


Figure 3.4: Transition Diagram for Channel Rearrangement Scheme

By denoting the limiting probability of state $(i,\ j)$ as $P_{(i,j)}$, we can set up the following balance equations for all the 12 states.

$$
\begin{array}{llcll}
\underline{\text{States}} & \underline{\text{Rate of Leaving}} & = & \underline{\text{Rate of Entering}} & \\
(0,0) & \lambda P_{(0,0)} & = & \mu_1 P_{(1,0)} + \mu_2 P_{(0,1)} & \\
(i,0) & (\lambda + i\mu_1) P_{(i,0)} & = & (i+1)\mu_1 P_{(i+1,0)} + \mu_2 P_{(i,1)} + \frac{\lambda}{2} P_{(i-1,0)}, & 1 \le i \le 3 \\
(i,0) & (\frac{\lambda}{2} + i\mu_1) P_{(i,0)} & = & (i+1)\mu_1 P_{(i+1,0)} + \frac{\lambda}{2} P_{(i-1,0)}, & 4 \le i \le 5 \\
(6,0) & 6\mu_1 P_{(6,0)} & = & \frac{\lambda}{2} P_{(5,0)} & \\
(0,1) & (\lambda + \mu_2) P_{(0,1)} & = & \mu_1 P_{(1,1)} + 2\mu_2 P_{(0,2)} + \frac{\lambda}{2} P_{(0,0)} & \\
(0,2) & 2\mu_2 P_{(0,2)} & = & \frac{\lambda}{2} P_{(0,1)} & \\
(i,1) & (\frac{\lambda}{2} + i\mu_1 + \mu_2) P_{(i,1)} & = & (i+1)\mu_1 P_{(i+1,1)} + \frac{\lambda}{2} P_{(i,0)} + \frac{\lambda}{2} P_{(i-1,1)}, & 1 \le i \le 2 \\
(3,1) & (3\mu_1 + \mu_2) P_{(3,1)} & = & \frac{\lambda}{2} P_{(3,0)} + \frac{\lambda}{2} P_{(2,1)} &
\end{array}
$$

The solution set is

$$
\left\{
\begin{array}{lcll}
P_{(i,0)} & = & \frac{1}{i}\left(\frac{\lambda}{2\mu_1}\right) P_{(i-1,0)}, & 1 \le i \le 6 \\
P_{(0,j)} & = & \frac{1}{j}\left(\frac{\lambda}{2\mu_2}\right) P_{(0,j-1)}, & 1 \le j \le 2 \\
P_{(i,1)} & = & \frac{1}{i}\left(\frac{\lambda}{2\mu_1}\right)\left(\frac{\lambda}{2\mu_2}\right) P_{(i-1,0)}, & 1 \le i \le 3
\end{array}
\right.
\tag{3.1}
$$

We can solve for the 12 limiting probabilities with the above solution set and the following equation.

$$
\sum_i \sum_j P_{(i,j)} = 1 \tag{3.2}
$$

A type 1 request will be blocked when the system is in states (6,0), (0,2) and (3,1) in which all slots are being used. A type 2 request will be blocked when the system is in states (4,0), (5,0), (6,0), (0,2), (1,1), (2,1) and (3,1). In these states, there are less than 3 idle slots. Thus, the blocking probability for the above system is given by:

$$
P_{(6,0)} + P_{(0,2)} + P_{(3,1)} + \frac{1}{2}\left(P_{(4,0)} + P_{(5,0)} + P_{(1,1)} + P_{(2,1)}\right) \tag{3.3}
$$

3.2.3 Numerical Results

A general data + voice system using channel rearrangement would have C frequency bands with N slots per frequency band. Type 1 users request for a single slot for voice transmission. Type 2 users request for t slots for data transmission. The approach applied to the previous special case can be used to analyse this general data + voice system. There are a lot more states and the analysis is shown in appendix A.

Figure 3.5 shows the blocking probabilities for four environments under which the channel rearrangement scheme carries out. These blocking probabilities serve as the lower bounds for all other slot allocation schemes. This is because the channel rearrangement scheme can always give rise to the most efficient frequency usage and hence the lowest blocking rate.

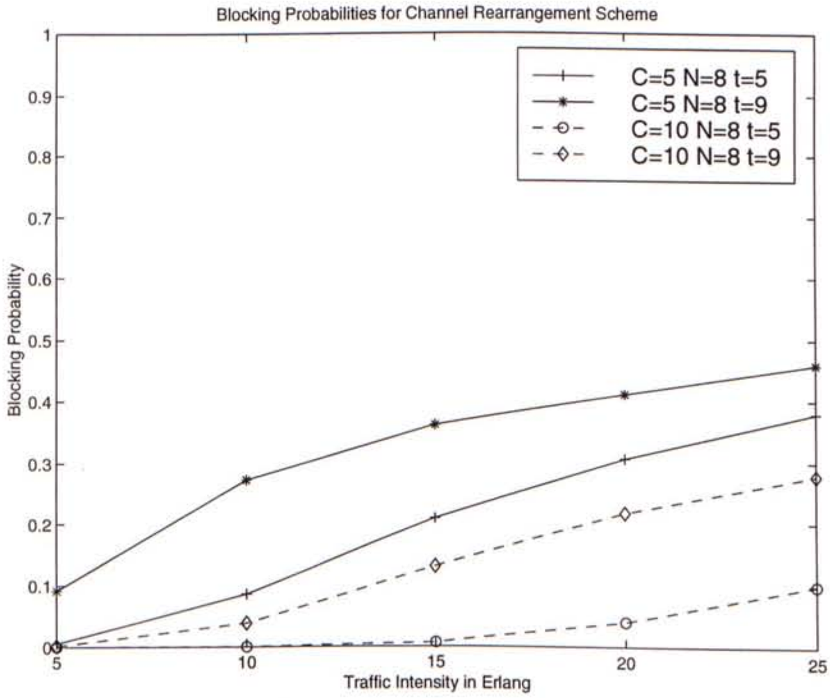


Figure 3.5: Blocking Probabilities for Channel Rearrangement Scheme

3.2.4 Issues in Channel Rearrangement

Despite the low blocking probabilities and short processing time, the channel rearrangment scheme is not practical to be exercised. There are several issues in rearranging channels.

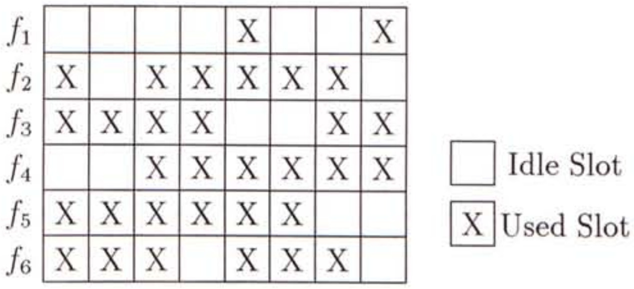
After each user departs, on average, half of the existing users have to shift their existing channels. This inevitably generates a large amount of control traffic for instructing mobile units to have a rearrangement. Moreover, special time consuming algorithm is needed to guarantee the highest frequency usage if the problem of violating mobile unit capability would happen. In conclusion, the channel rearrangement scheme can serve as a theorectical optimal solution but not a practical approach in solving the slot allocation problem.

Chapter 4

Multiple Slots Allocations

4.1 Introduction

In the previous chapter, we can see that for the system under consideration, it is inevitable to have a fragmented channel usage at most of the time. Under a dynamic channel assignment (DCA) scheme, radio bandwidth is assigned dynamically. It is favourable to use as few frequency bands as possible to serve users in a cell. The feasibility question and the actual slot assignment for a new connection request under a given frequency usage pattern such as the one shown in figure 4.1, require careful analysis. Different slot allocation strategies yield different solutions with different processing times. Two different strategies with different targets are described in this chapter. The No-Split algorithm puts focus on the allocation time while the Best Fit algorithm aims at an compact frequency usage.



f_1					X			X
f_2	X		X	X	X	X	X	
f_3	X	X	X	X			X	X
f_4			X	X	X	X	X	X
f_5	X	X	X	X	X	X		
f_6	X	X	X		X	X	X	

☐ Idle Slot

☒ Used Slot

Figure 4.1: A Fragmented Channel Usage

4.2 No-Split Algorithm

4.2.1 No-Split Algorithm

The No-Split Algorithm satisfies a allocation request with the minimal number of frequencies. There is no unnecessary splitting of user demands. For example, with 8 slots per frequency band, a demand of 14 slots will be assigned exactly with 2 frequencies. This algorithm is not going to deal with combinations of fragments of idle slots in different frequencies. Users in the No-Split Algorithm are classified into 2 categories. Category 1 consists of those requesting less than N slots. On the other hand, category 2 users request N or more slots to acheive the required transmission rate.

The allocation for category 1 users is one-pass. All the C currently used frequency bands are scanned once to look for a single one containing enough idle slots for the request. A new frequency band will be needed if the scanning fails. The allocation is demonstrated in figure 4.2.

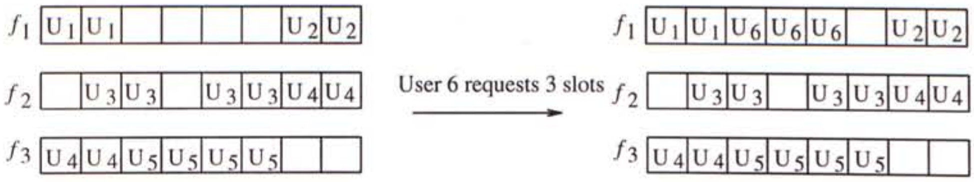


Figure 4.2: No-Split Allocation for Category 1 Users

One modification is made to accommodate category 2 users. Since each user requests for N or more idle slots, new frequency bands are acquired to fit part of the request until the remaining request is less than N. The same scanning procedure as that for category 1 users then applies. Figure 4.3 shows the allocation.

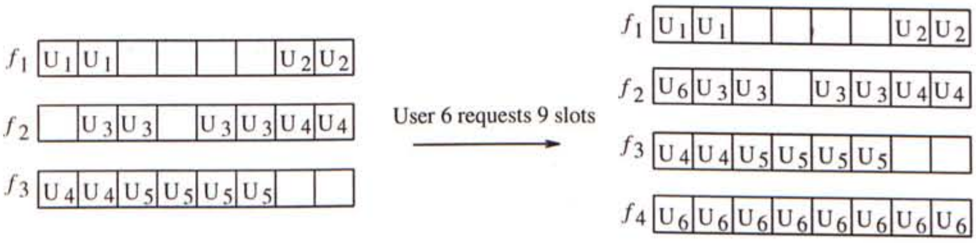


Figure 4.3: No-Split Allocation for Category 2 Users

We can see from the 2 figures that the splitting of a request is a minimal and each of the users is served with a minimial number of frequency bands. This feature accounts for the algorithm's name.

4.2.2 Pros and Cons

The No-Split algorithm is a very time efficient algorihtm as it requires only a single scan of the C currently used frequency bands. Each allocation done is guaranteed to suit the frequency usage capability of the mobile unit. The key reason is the minimization of frequency bands used for each allocation.

However, the channel utilization is not high. New frequency bands are acquired even there are enough idle slots for a request in different frequency bands. Efficiency in frequency usage is traded for the reduction in processing time.

4.3 Best Fit Algorithm

4.3.1 Best Fit Algorithm

The Best Fit algorithm is a locally optimal strategy in terms of frequency usage. Different frequency band combinations are examined to see whether it is possible to accommodate the current request. A hash table, *Hash1*, indexed by the number of idle slots in a frequency band is employed to facilitate the search. The sequence of search follows the following 3 criteria with decreasing importance.

1. The search proceeds with increasing demand.
2. The search proceeds with increasing number of frequencies needed.
3. The search proceeds from "inner" partitions to "outer" partitions.
   (i.e. the combination {4,4} before {3,5} to fit an 8-slot request)

The intention of the first criterion is to try to find an idle slot combination from different frequencies to fit as close as possible the requested demand. Thus, the idle slot combination that is a best fit to the requested demand is allocated under the current channel usage. The second criterion is to minimize the number of frequencies to fit a demand. For example, to provide 10 idle slots, a {5,5} combination that needs 2 frequencies is preferable than a {2,4,4} combination that needs 3 frequencies. The third criterion is to leave larger holes for future allocations. For example, a {5,5} combination is preferable than a {4,6} combination.

A new frequency band is acquired when no suitable allocation is found under current frequency usage. The search restarts again with this newly added frequency band. A suitable allocation here denotes one with enough idle slots for the requested bit rate transmission and does not violate the capability of frequency usage in a mobile unit. A connection request is blocked if and only if no suitable allocation is found and no extra bandwidth can be assigned to this cell without violating the co-channel interference constraint.

For instance, with the current channel usage shown as in figure 4.4, the best choice for a request of 8 time slots can be obtained by using 5 slots in f_4 and 3 slots out of the 4 in f_2. If the frequency band constraint M is equal to 3 (i.e. each mobile unit can use up to 3 frequencies for simultaneous transmissions, the algorithm will come up with the solution based on *Hash1* after 11 iterations when all the following slot combinations are examined.

D = 8:

{8}, {4,4}, {3,5}, {2,6}, {1,7}, {2,3,3}, {2,2,4}, {1,3,4}, {1,2,5}, {1,1,6}

D = 9:

{4,5}

Each element of a set signifies the need of a frequency band with exactly the corresponding number of idle slots. Thus, in the second iteration, one tries to find 2 frequency bands with exactly 4 idle slots in each of them. (Clearly such a solution does not exist.)

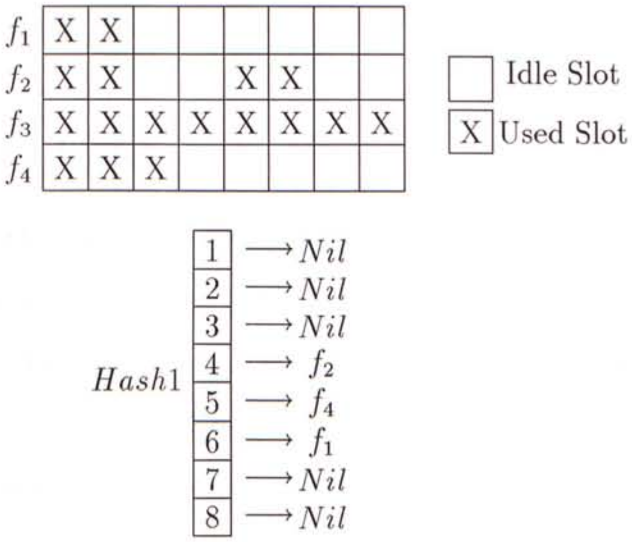


Figure 4.4: Data Structure for Best Fit Algorithm

4.3.2 Optimization

For each demand D in an allocation, there is no need to start with a single frequency (sets with 1 elements), then 2-frequency combinations (sets with 2 elements), etc. for accomodation. The minimum number of frequencies needed for a demand can be easily calculated from *Hash1*. We can accumulate idle slots under current channel usage and proceed with decreasing number of idle slots. From *Hash1* in figure 4.4, the first accumulation involves 6 idle slots. The second accummulation will accumulate 5 idle slots to give rise a total of 11 slots which is larger than the demand of 8 slots. We can then claim that at least 2 frequencies are needed as even the largest available hole with 6 idle slots in f_1 cannot fit the request. Thus, all attempts in using single frequency to fit the request during the search can be saved. For example, the first set $\{8\}$ in the previous sequence of search can be saved. The larger the number of minimum number of frequency needed is, the more iterations can be saved during a Best

Fit search.

4.3.3 Pros and Cons

The Best Fit algorithm is locally optimal as it maximizes the channel usage in each connection request. In a Best Fit search, an exact (best fit) match is first tried for allocation. This act can fill up holes of idle slots in differenct frequencies in order to combat the problem of fragmentation upon departures. Every acquisiton of new frequency bands is unavoidable in order to provide service. However, this searching technique is not time efficient. It has to examine different frequency band combinations until one that contains enough idle time slots for the current request can be found. An increase in the number of available frequency bands (C) in a cell as well as an increase in the capability of a mobile unit in using multiple frequency bands for transmission (M) will dramatically increase the allocation time. Indeed, this exhaustive searching technique can be shown as NP-complete (see appendix B) and is not practical in large communication systems that demand quick allocations.

4.4 Comparison of the two algorithms

In this section, the No-Split algorithm and the Best Fit algorithm are compared by simulation. The simulation is performed under the following environment.

- A 7-cell cluster with totally 49 frequency bands for sharing dynamically.
- Slots per frequency band $N = 8$.
- Number of usable frequency bands in each connection $M = 3$.

- Poisson arrivals with rates from 3 to 10 per second apply to each cell.
- Connection holding time is exponentially distributed with mean 0.5s.

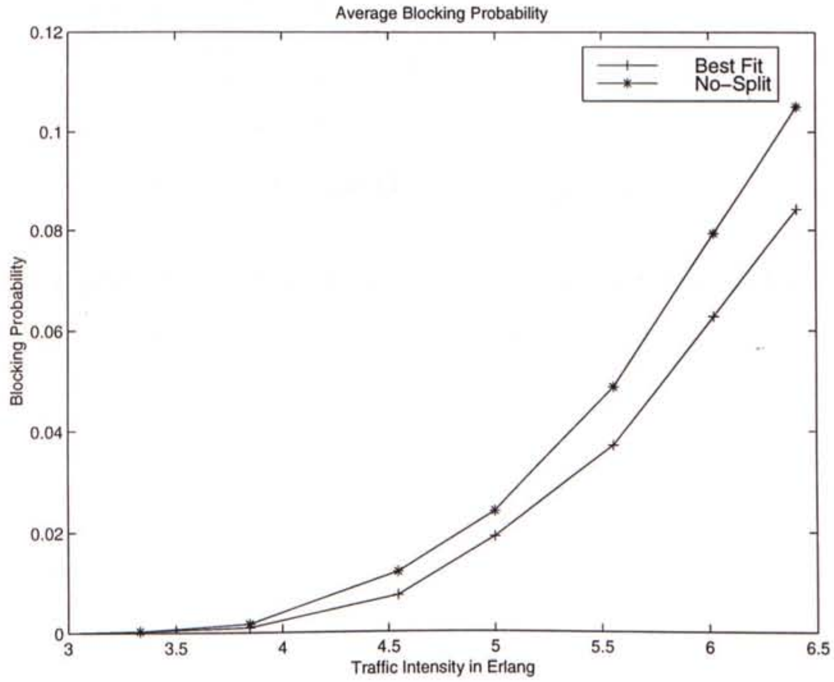


Figure 4.5: Blocking Probabilities for Best Fit and No-Split algorithms

Figure 4.5 shows the blocking probabilities for both the Best Fit and No-Split algorithms. A connection request is blocked when either there are not enough idle slots in the system and no more frequency band can be assigned or the usable frequency bands number constraint (M) is violated. The figure shows that the Best Fit algorithm can always have a lower blocking probability when compared with the No-Split algorithm. Consequently, more users can be served by the Best Fit algorithm with the same amount of resources.

The slot occupancies for both algorithms are shown in figure 4.7. The slot occupancy of a system at any instance is defined by the ratio of non-idle slots to the total number of slots assigned to a cell. For example, the slot occupancy for the channel usage shown in figure 4.6 is 0.75.

f_1	X	X	X	X			X	X
f_2			X	X	X	X	X	X
f_3	X	X	X	X	X	X		
f_4	X	X	X		X	X	X	

☐ Idle Slot

☒ Used Slot

$$\text{Occupancy} = \frac{24}{32} = 0.75$$

Figure 4.6: Slot Occupancy Calculation

Figure 4.7 plots the slot occupancies of both algorithms in different traffic conditions. Natural Log-Scale applies in traffic intensities to view the plots' tendencies.

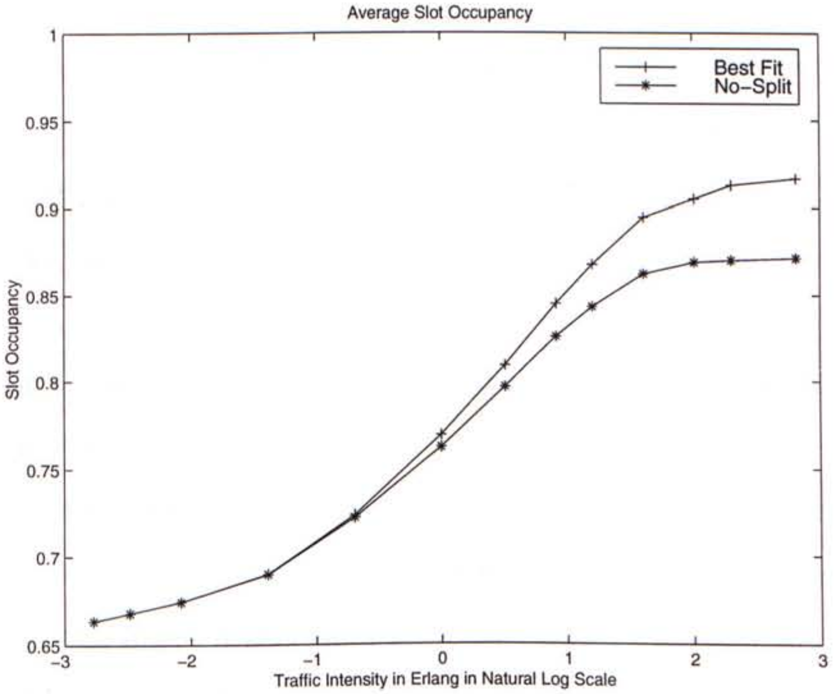


Figure 4.7: Slot Occupancies for Best Fit and No-Split algorithms

When the traffic rate is low (less than 1 Erlang, i.e. 0 in natural log scale), both algorithms perform about the same. Low occupancies are attributed to low traffic rates. At most of the time, the system is idle and it makes no difference in applying both algorithms for slot allocations. When the traffic increases (from 1 Erlang to 5 Erlang), the slot occupancies for both algorithms rise up. These show that the algorithms are now trying to use up idle slots for accommodations of new arrivals. When the traffic is heavy (more than 5 Erlang, i.e. 1.6 in natural log scale), the channels become saturated and it is difficult to get more penetrations in channel usage for both of them.

From the two plots, we can see that when the traffic rate increases, the Best Fit algorithm can alway use the channel more efficiently. The saturation points for slot occupancies are 0.92 and 0.87 for the Best Fit algorithm and the No-Split algorithm respectively. About 13% of the total acquired slots are left idle by the No-Split algorithm. Within them, more than 38% can be effectively utilized by the Best Fit algorithm. The high slot occupancies for the Best Fit algorithm accounts for its low blocking probabilities. These show that it is not a good idea to employ No-Split algorithm for slot allocations as the radio spectrum is scarse.

It is not without expense in return for the high channel usage. Figure 4.8 shows the average processing time needed for both algorithms. Processing time is the time needed when a connection request is dequeued from the arrival queue until a successful slot allocation for it is done. About 0.32 ms is needed for an allocation by the Best Fit algorithm while only 0.04 ms is required by the No-Split algorithm. This makes it not appealing to adopt the Best Fit algorithm in slot allocations. A long processing time generates excessive delays in connection setup and is not tolerable by delay sensitive applications.

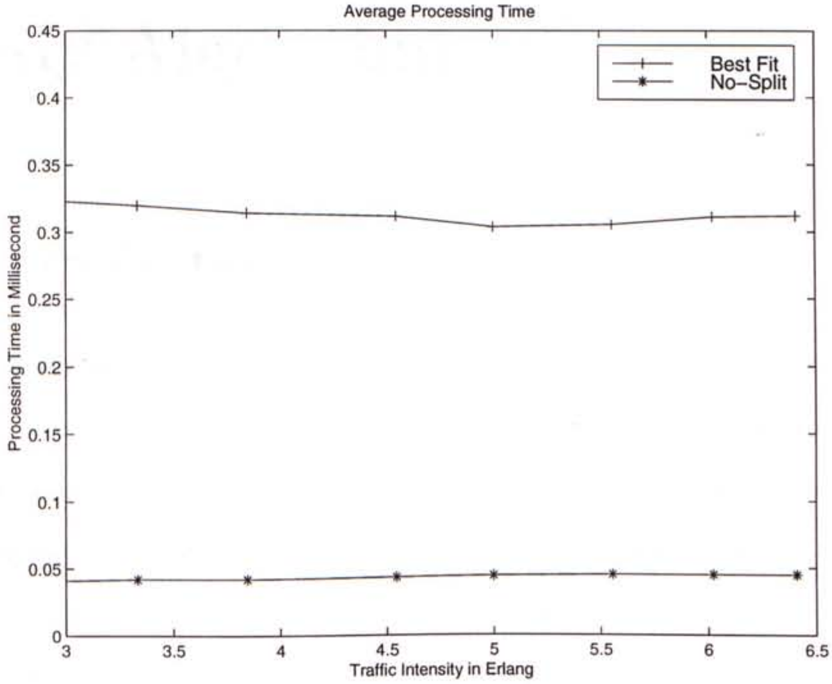


Figure 4.8: Processing Time for Best Fit and No-Split algorithms

Chapter 5

Buddy Algorithm

5.1 Introduction

Two slot allocation algorithms are described and compared in the previous chapter. They are not practical in use as they are either too slow or too wasteful in using resources. The problem in the Best Fit algorithm is that there are too many frequency band combinations to consider. In contrast, none of these combinations is taken into account before the No-Split algorithm decides to get new frequency bands. Is it possible to have an all-round approach to take care of both aspects? The following proposed Buddy algorithm is a heuristic algorithm based on binary clustering of idle slots to speed up the slot allocation process yet still provides an efficient frequency usage.

5.2 Buddy System in Memory Management

Management in memory allocation and deallocation is an important issue in operation system (OS) design. Its effectiveness directly affects the performance of the OS. Like wireless communication systems, the random nature of process arrivals and departures creates lots of fragmentations in memory. One way to manage these fragmentations is to keep linked lists of allocated and free memory segments as shown in figure 5.1. The hole list is kept sorted on size to make allocation faster. However, this arrangement adversely makes deallocation expensive. When a process terminates, it is neccessary to find its neighbors to see if a merge is possible in the hole list which is sorted on sizes of holes, not addresses.

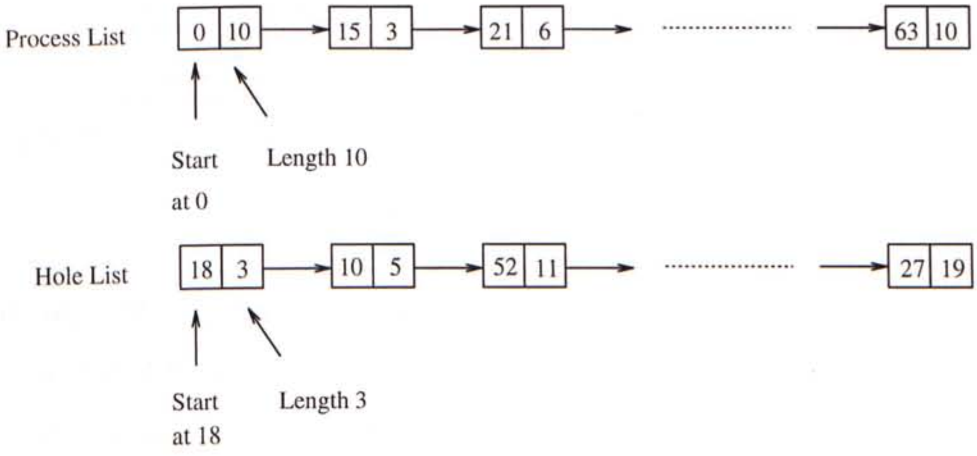


Figure 5.1: Linked Lists for Memory Management in OS

The Buddy System (Knuth 1973; Knowlton 1965) [21] is a memory management algorithm that addresses itself to the speeding up of merging adjacent holes when a process terminates. Hole sizes are restricted to power of 2. Lists of free blocks of sizes 1, 2, 4, 8, etc., bytes, up to the size of memory are maintained. A process request is rounded up into power of 2 and a check for a free block of that size is performed. If found, the free block is allocated. Otherwise, a larger block is split into small blocks called *buddies* for use. From figure 5.2, a 70K memory request will cause the 1M free block split into two 512K blocks. The splitting continues until a 128K free block is formed and the request is then satisfied. In deallocation, the return of user B (holding a 64K block) in figure 5.2 will cause a merge to form back a 256K block.

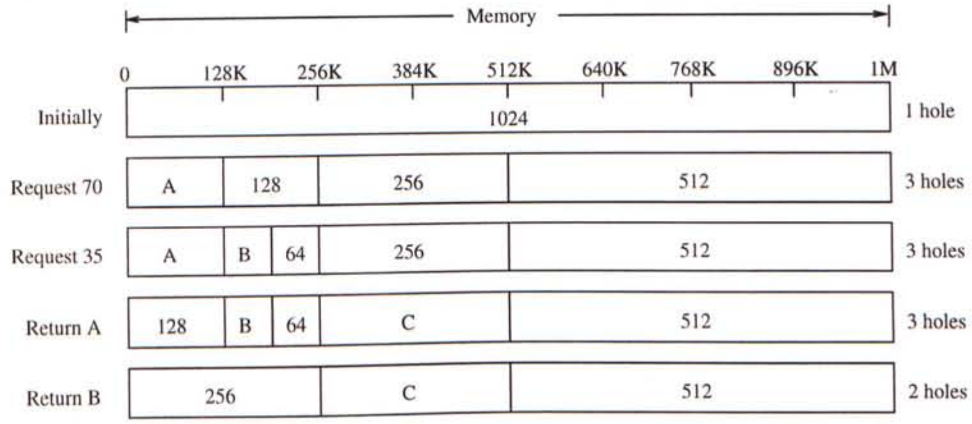


Figure 5.2: The Buddy System

Buddy systems have an advantage that when a block of size 2^k bytes is freed, only the list of 2^k holes has to be searched to see if a merge is possible. This solves the problem of slow deallocation in other systems that sort holes by size. Unfortunately, it is extremely inefficient in terms of memory utilization. The problem of *internal fragmentation* exists as all requests must be rounded up to a power of 2. In worst case, nearly half of the memory space is wasted.

5.3 Buddy Algorithm

5.3.1 Adaptation in slot allocation

The idea in the Buddy system is modified to solve the slot allocation problems described in previous chapters. Idle slots in each frequency band are grouped into holes with sizes of power of two. A particular frequency band may now contain more than one hole. For instance, a frequency band with 7 idle slots contains 3 holes with sizes 2^2, 2^1 and 2^0 respectively. By this restriction in hole size, both the allocation and deallocation process can be speeded up.

5.3.2 Data structure

An additional hash table, *Hash2*, is employed along with *Hash1* as used in the best fit algorithm. The function of *Hash2* is similar to *Hash1* but it contains a smaller number of entries. Each entry is indexed by the hole size. Figure 5.3 indicates the state of *Hash2* with the channel usage status shown. One can see that the number of entries in *Hash2* is reduced to $\lfloor \log_2 N \rfloor + 1$, instead of N.

5.3.3 Slot allocation

The slot allocation in the buddy algorithm [22] is divided into 2 phases. In the following, we denote D as the number of slots required in a particular request.

Phase I

Those entries not less than D in $Hash1$ are examined to find a single frequency band that contains enough idle slots for the request. This phase is highly efficient in a light load environment. If the traffic is heavy, the system tends to be heavily

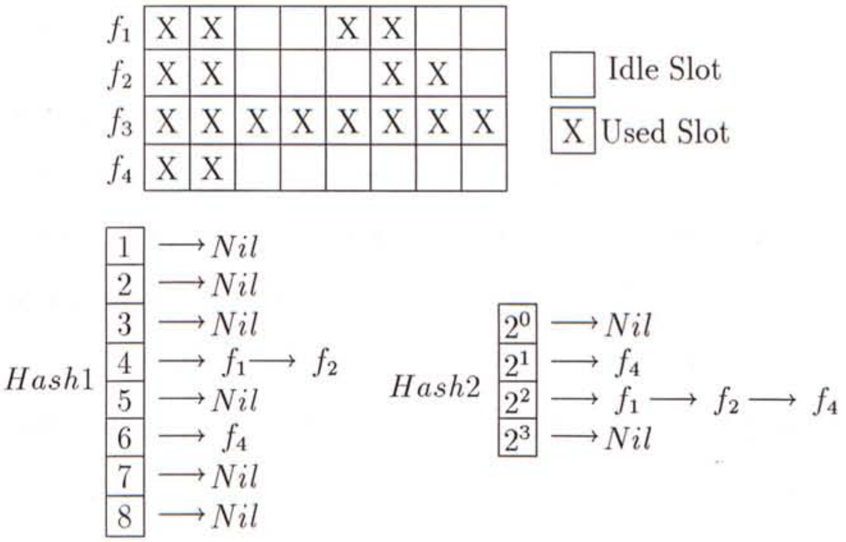


Figure 5.3: Hash2 for Buddy Algorithm

loaded and phase II has to be adopted when phase I fails.

Phase II

Each request D is transformed into a linear combination of the base $\{2^n, 2^{n-1}, ..., 2^0\}$ with non-negative integer coefficients where $2^n \leq D < 2^{n+1}$. Notice that the representation is not necessarily unique and coefficient vectors in the form $(a_n, a_{n-1}, ..., a_0)$ are obtained. Such a representation of the demand D signifies that a_n holes of size 2^n, a_{n-1} holes of size 2^{n-1}, etc. are needed to satisfy the request. Only those coefficient vectors that require no more than M frequency bands need examinations for available slots in the system. Since a frequency may contain more than 1 hole, $\sum_{i=0}^{n} a_i$ is greater than or equal to the total number of frequencies needed for a coefficient vector. The following 2 conditions can then be regarded as sufficient conditions for those coefficient vectors that needed examinations.

1. $\sum_{i=0}^{n} a_i \leq M$ (Frequency band constraint)
2. $\sum_{i=0}^{n} a_i 2^i \geq D$ (User demand)

For each coefficient vector $\vec{a}$, representing D, the algorithm searches for a set of frequency bands to satisfy it. If no valid set of frequency bands is found, one proceeds with $D+1$. This process continues until D is advanced to 2^{n+1} and the corresponding search is found failed where $2^n \leq D < 2^{n+1}$.

Since $\sum_{i=0}^{n} a_i$ is greater than or equal to the total number of frequencies needed, we start with the binary representation $\vec{p}\cdot\{2^n, 2^{n-1},..., 2^0\}$ of D where $p_i \in \{0,1\}$ $\forall$ i. This binary form of D requires the minimum number of frequencies. Other non-negative integer linear combinations of D can then be generated from this binary form. For example, if we have a demand $D = 10$, we will start with the coefficient vector $\{a_3, a_2, a_1, a_0\} = \{1,0,1,0\}$ to see if there exist two frequencies with a hole of size 2^3 and 2^1 respectively. If we can find this set of frequencies, then we can allocate these 2 frequencies to satisfy the demand and the process stops. Otherwise, we have to proceed with $\vec{a} = \{0,2,1,0\}$ which is generated from the binary form having the '1' in a_3 split into a '2' in a_2. In this case, we will need two frequencies each with a hole of size $= 2^2$ and one frequency with a hole of size $= 2^1$.

We continue to generate these coefficient vectors until the frequency band constraint stated above is violated. Then, we advance D to $D+1$ and start the search again. The allocation stops when D is advanced to 2^{n+1} and the corresponding search is found failed. A new frequency band is then acquired and the search restarts again. The allocation process fails and the connection request is blocked when no more free frequency band can be assigned dynamically without violating the co-channel interference constraint.

A simple example below illustrates the procedures. Consider a simple cellular system with the following parameters.

1. C = total number of frequency bands = 4;
2. N = number of slots per frequency band = 8;
3. M = number of usable frequency bands per mobile unit = 3.

Suppose the current channel usage is shown in figure 5.3 and a request of 9 slots arrives. The buddy algorithm should have the following sequence of search.
$\vec{a} = \{1,0,0,1\}, \{0,2,0,1\}, \{1,0,1,0\}, \{0,2,1,0\}$

From figure 5.3, we see from $Hash2$ that only the coefficient vector $\vec{a} = \{0,2,1,0\}$ in iteration 4 can be satisfied by the system. Thus, we choose a hole of size 2^1 in f_4 and two holes of sizes 2^2 in f_1 and f_4 respectively since the only non-zero coefficients in $\vec{a}$ are $a_2 = 2$ and $a_1 = 1$. The channel usage after the allocation is shown in figure 5.4. Different from the Buddy system in memory management, there is no waste in bandwidth during allocation. Although the 10 idle slots from f_1 and f_4 are selected, one of them remains idle and is re-hashed into $Hash2$ as only 9 slots are needed by the request.

This allocation procedure reduces the number of slot combinations. For example, to check for a demand $D = 8$ with N= 8 and $M = 3$, only the 3 coefficient vectors, namely {1,0,0,0}, {0,2,0,0} and {0,1,2,0}, are checked by the Buddy algorithm. On the other hand, there would be 10 different combination sets, namely {8}, {4,4}, {3,5}, {2,6}, {1,7}, {2,3,3}, {2,2,4}, {1,3,4}, {1,2,5}, {1,1,6}, in the Best Fit algorithm.

f_1	X	X	X	X	X	X	X	X
f_2	X	X				X	X	
f_3	X	X	X	X	X	X	X	X
f_4	X	X	X	X	X	X	X	

☐ Idle Slot
☒ Used Slot

$Hash1$
$1 \longrightarrow f_4$
$2 \longrightarrow Nil$
$3 \longrightarrow Nil$
$4 \longrightarrow f_2$
$5 \longrightarrow Nil$
$6 \longrightarrow Nil$
$7 \longrightarrow Nil$
$8 \longrightarrow Nil$

$Hash2$
$2^0 \longrightarrow f_4$
$2^1 \longrightarrow Nil$
$2^2 \longrightarrow f_2$
$2^3 \longrightarrow Nil$

Figure 5.4: After Buddy Allocation

5.3.4 Slot deallocation

Upon the termination of a connection, the assigned slots are returned to the system. Slots for each frequency band originally assigned are grouped into holes in the binary fashion as mentioned above. A hole of size 2^k will need only a check in the 2^k entry of $Hash2$ to see if a merge is possible. This simplification in merging is resulted from the binary management of idle slots. For example, a return of 4 slots in f_2 for channel usage in figure 5.4 will lead to a merge in f_2. $Hash2$ is updated and shown in figure 5.5.

$Hash2$
$2^0 \longrightarrow f_4$
$2^1 \longrightarrow Nil$
$2^2 \longrightarrow Nil$
$2^3 \longrightarrow f_2$

Figure 5.5: Buddy Deallocation after a Return of 4 slots in Frequency band 2

5.4 Inference Property

An improvement in the search can be done to reduce the total number of non-negative integer linear combinations. From the following simple example, one sees why this can be done.

Suppose D is 10 and the number of usable frequency bands M in a mobile unit is 4. We can have the following searching sequence instead of incrementing D by 1 in each iteration.

$D = 10 = 2^3 + 2^1$:

$\vec{a} = \{1,0,1,0\}, \{0,2,1,0\}, \{1,0,0,2\}, \{0,1,3,0\}, \{0,2,0,2\}$

$D = 12 = 2^3 + 2^2$:

$\vec{a} = \{1,1,0,0\}, \{0,3,0,0\}, \{1,0,2,0\}, \{0,2,2,0\}, \{1,0,1,2\}$

$D = 16 = 2^4$:

$\vec{a} = \{1,0,0,0,0\}, \{0,2,0,0,0\}, \{0,1,2,0,0\}, \{0,0,4,0,0\}, \{0,1,1,2,0\}$ (stop)

The rationale for skipping the search at $D = 11$ is due to the fact that the failure of $D = 10$ implies the failure of $D = 11$. The coefficient vectors of $D = 11$ that need to be examined are $\{1,0,1,1\}$, $\{0,2,1,1\}$, $\{1,0,0,3\}$. However, the failure in trying $\{1,0,1,0\}$ indicates that there does not exist a set of 2 frequencies each having a hole of size $= 2^3$ and 2^1 respectively. This directly implies that there does not exist a set of 3 frequencies having holes of sizes $= 2^3$, 2^1 and 2^0 respectively which is exactly what the coefficient vector $\{1,0,1,1\}$ needs for realization. Similarly, the failure of the coefficient vectors $\{0,2,1,1\}$ and $\{1,0,0,3\}$ are implied by the failure of $\{0,2,1,0\}$ and $\{1,0,0,2\}$ respectively. The direct jump from $D = 12$ to 16 is also based on this rationale. This property is referred as the *inference property* of the Buddy algorithm.

Proposition 5.1 *If a demand* $D = \sum_{i=l}^{n} p_i 2^i$ *is examined, we can skip the examination of a demand* $D' = \sum_{i=l}^{n} p_i 2^i + m$ *where* $0 \leq l \leq n$, $0 < m < 2^l$ *and* $p_i \in \{0,1\}\ \forall i$.

The stop of search after 2^{n+1} in the allocation is also due to this inference property and the fact that no hole of size $\geq D$ is found by phase I. The inference property greatly reduces the number of iterations needed in each channel allocation process.

The following shows the procedures of the improved algorithm:

```
Repeat until $D > 2^{n+1}$ where $2^n \leq D < 2^{n+1}$
    $\vec{p}$ is the coefficient vector of the binary form of D
    Check for all coefficient vectors $\vec{a}$ of D s.t. $\sum_{i=0}^{n} a_i \leq M$
    Advance(D, $\vec{p}$)
```

All the possible coefficient vectors can be generated by the function *Span()* below and the *Advance()* function will advance D to the next value that has not been infered by those previous checked.

```
Span($\vec{a}$)
    if $a_i \neq 0$
        decrement $a_i$ by 1
        increment $a_{i-1}$ by 2
```

```
Advance(D, $\vec{p}$)
    set j to 0
    Repeat until $p_j = 1$
```

```
    increment j by 1
Repeat until $p_j = 0$
    increment j by 1
set $p_j$ to 1
set $D = \sum_{i=j}^{n} p_i 2^i$
```

5.4.1 Proof of the Inference Property

In this section, we are going to prove that there always exists a non-negative integer coefficient vector $\vec{b}$ of D that can imply the failure of a non-negative integer coefficient vector $\vec{a}$ of D' where $D = \sum_{i=l}^{n} p_i 2^i$ and $D' = \sum_{i=l}^{n} p_i 2^i + m$ with $0 \leq l \leq n$, $0 < m < 2^l$ and $p_i \in \{0,1\}$ $\forall i$.

An algorithm to systematically find out a coefficient vector $\vec{b}$ of D given a coefficient vector $\vec{a}$ of D' is introduced.

Algorithm:

Starting from a_n down to a_0:

1. If $\sum_{i=k+1}^{n} b_i 2^i + a_k 2^k \leq \sum_{i=l}^{n} p_i 2^i$, set $b_k = a_k$
2. If $\sum_{i=k+1}^{n} b_i 2^i + a_k 2^k > \sum_{i=l}^{n} p_i 2^i$, set $b_k = c$
    where $0 < c < a_k$ such that $\sum_{i=k}^{n} b_i 2^i = \sum_{i=l}^{n} p_i 2^i$

If the condition in 2 is reached, one can stop the algorithm by setting $b_i = 0$ for $0 \leq i < k$. By this algorithm, one can guarantee that $0 \leq b_i \leq a_i$ $\forall i$.

To prove the correctness of this algorithm, assume we cannot find a coefficient vector $\vec{b}$ satisfying the above property by the above algorithm. There are 2 possibilities:

1. We cannot find a $b_k = c < a_k$ such that

$\sum\limits_{i=k}^{n} b_i 2^i = \sum\limits_{i=l}^{n} p_i 2^i$ when $\sum\limits_{i=k+1}^{n} b_i 2^i + a_k 2^k > \sum\limits_{i=l}^{n} p_i 2^i$.

2. $\sum\limits_{i=0}^{n} b_i 2^i < \sum\limits_{i=l}^{n} p_i 2^i$.

Contradictions are shown on both possibilities.

1. Case I: $l \geq k+1$

$$\begin{aligned}\sum_{i=l}^{n} p_i 2^i - \sum_{i=k+1}^{n} b_i 2^i &= 2^k \Big[\sum_{i=l}^{n} p_i 2^{i-k} - \sum_{i=k+1}^{n} b_i 2^{i-k}\Big] \\ &= c 2^k\end{aligned}$$

By the algorithm, we have $\sum_{i=k+1}^{n} b_i 2^i < \sum_{i=l}^{n} p_i 2^i$. Thus, $c > 0$.

Moreover, the situation $\sum_{i=k+1}^{n} b_i 2^i + a_k 2^k > \sum_{i=1}^{n} p_i 2^i$ implies that $c < a_k$.

It contradicts the assumption that we cannot find a $b_k = c < a_k$ such that $\sum_{i=k}^{n} b_i 2^i = \sum_{i=l}^{n} p_i 2^i$.

Case II: $l < k+1$

By the algorithm, $\sum\limits_{i=k+1}^{n} b_i 2^i = \sum\limits_{i=k+1}^{n} a_i 2^i$.

Thus, $\sum\limits_{i=k+1}^{n} b_i 2^i + a_k 2^k > \sum\limits_{i=l}^{n} p_i 2^i$

$$\begin{aligned}&\Rightarrow \sum_{i=k}^{n} a_i 2^i > \sum_{i=l}^{n} p_i 2^i \\ &\Rightarrow \sum_{i=l}^{n} a_i 2^i \geq \sum_{i=k}^{n} a_i 2^i > \sum_{i=l}^{n} p_i 2^i \qquad (l < k+1,\ a_i \geq 0\ \forall i) \\ &\Rightarrow \sum_{i=l}^{n} a_i 2^i - \sum_{i=l}^{n} p_i 2^i = q 2^l > 0 \qquad \text{where } q > 0\end{aligned}$$

Denote $m = \sum\limits_{i=0}^{l-1} p_i 2^i$ where $p_i \in \{0,1\}$.

$$\begin{aligned}&\sum_{i=0}^{n} a_i 2^i - \sum_{i=0}^{n} p_i 2^i \\ &= \sum_{i=l}^{n} a_i 2^i - \sum_{i=l}^{n} p_i 2^i - \sum_{i=0}^{l-1} p_i 2^i + \sum_{i=0}^{l-1} a_i 2^i \\ &= q 2^l - \sum_{i=0}^{l-1} p_i 2^i + \sum_{i=0}^{l-1} a_i 2^i \\ &> 0 \qquad \Big(q 2^l > \sum_{i=0}^{l-1} p_i 2^i \text{ as } p_i \in \{0,1\}\Big)\end{aligned}$$

This contradicts the fact that $\sum_{i=0}^{n} a_i 2^i = \sum_{i=0}^{n} p_i 2^i$ as $\vec{a}$ is the coefficient vector of the number $\sum_{i=0}^{n} p_i 2^i$.

2. $\sum_{i=0}^{n} b_i 2^i < \sum_{i=l}^{n} p_i 2^i$

$$\begin{aligned} &\Rightarrow \sum_{i=0}^{n} a_i 2^i < \sum_{i=l}^{n} p_i 2^i && (b_i = a_i,\ 0 \le i \le n) \\ &\Rightarrow \sum_{i=l}^{n} p_i 2^i + m < \sum_{i=l}^{n} p_i 2^i \\ &\Rightarrow m < 0 \end{aligned}$$

This contradicts the assumption that $0 < m < 2^l$.

The above contradictions assert that we can always find a coefficient vector $\vec{b}$ of the number $D = \sum_{i=l}^{n} p_i 2^i$ corresponding to the coefficient vector $\vec{a}$ of the number $D' = \sum_{i=l}^{n} p_i 2^i + m$ where $0 < m < 2^l$ such that $0 \le b_i \le a_i$ to provide inference. Since $0 \le b_i \le a_i\ \forall i$, the failure of finding a suitable set of frequencies to satisfy the coefficient vector $\vec{b}$ implies the failure of that for $\vec{a}$.

5.5 Pros and Cons

The main advantage of the Buddy algorithm is the reduction in slot combinations. If the number of frequency bands usable simultaneously (M) in a mobile unit = 3, we can show that there is an upper bound on the maximum number of slot combinations from the following proposition.

Proposition 5.2 *Total number of different non-negative linear combinations generated for a request of D slots* $\le 3(\lfloor \log_2 D \rfloor + 1)$ *for* $M = 3$.

The proof of the proposition is shown in appendix C. This reduction speeds up the allocation process. It can be shown from next chapter that it takes only about one-third of the time of the Best Fit algorithm to come up with a solution.

Moreover, the slot occupancy can be kept high and hence the blocking rate of connection requests is improved.

However, the reduction in the number of slot combinations does bring disadvantages as well as the improvement in processing time. One of them is the maintenance of an extra hash table ($Hash2$). But this burden, in terms of time, is compensated by the improvement in processing time. Another tradeoff is the increase in the average number of frequency bands allocated per connection on average. This results from the grouping of idle slots into distinct holes in each frequency band. However, simulation results in next chapter show that this increase is relatively small and is worth for the speeding up in processing.

Chapter 6

Performance Study

6.1 Introduction

In this section, the performance of the proposed Buddy algorithm is compared with the Best Fit and No-Split algorithms. Blocking probability and average processing time are used to evaluate their performance in this study. A connection request is blocked either when there are not enough idle slots or the usable frequency bands number constraint of a mobile unit is violated. Processing time is the time needed when a request is dequeued from the arrival queue until a successful allocation is done.

Blocking probability is a measure to the *quality of service* (QOS) of a system. A connection request is blocked when the system fails to provide the required resources. The lower this probability is, the better the system is to provide services. The processing time needed in allocating slots for user request is another important measure. If the time needed is short, more connection requests can be served by a single server. Also, it reduces the computation time for *Handoff* [23] which is one of the issues in handling mobility in cellular systems. Processing time depends a lot on the algorithm for slot allocation. Thus, it can be used to illustrate which algorithm is more time efficient to be adopted.

Simulations are done on these three algorithms in both the *Fixed Channel Assignment* (FCA) and *Dynamic Channel Assignment* (DCA) environments. In FCA, each cell is assigned a fixed number of frequencies to provide services while in DCA, a central pool of frequencies is shared by all the cells within a cluster. The sharing adopts an on-demand approach. Each cell will try to get more frequency bands only when it is necessary in satisfying a connection request.

6.2 Fixed Channel Assignment

6.2.1 System Parameters

A particular cell in a cluster is focused in the following simulations. FCA is employed to assign a predetermined set of frequencies to the cell with the following parameters:

- Number of frequency bands in the cell $(C) = 8$.
- Number of slots per frequency band $(N) = 8$.
- Number of usable frequency bands in each connection $(M) = 3$.
- Poisson arrivals with rates $\lambda = 1, 1.25, 2, 2.5, 4, 5, 6, 7.5, 8$ per second.
- Exponential service time with mean $\mu = 0.5$s.
- User demand D is uniformly distributed in $[1, 2N]$.

6.2.2 Simulation Results

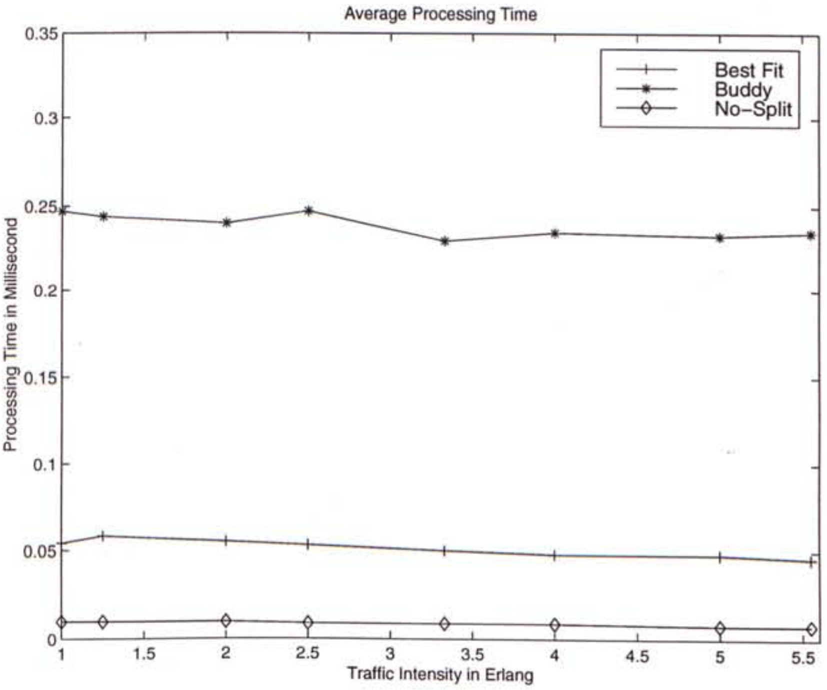


Figure 6.1: Processing Time vs Traffic Intensity under FCA

Figure 6.1 records the average processing time for successful slot allocations. On average, it takes about 0.24 ms for the Best Fit algorithm to come up with a successful allocation. The Buddy algorithm and the No-Split algorithm only require one-fourth and one-tenth of the time. The long proccessing time hinders the use of the Best Fit algorithm in practical systems.

Figure 6.2 shows the blocking probabilities for all the three algorithms. A call request is blocked either when there are not enough idle slots in the system or no suitable set of frequencies is found by the corresponding allocation algorithm. The figure shows that the Buddy algorithm can achieve similar to the exhaustive Best Fit algorithm in blocking. The difference is less than 0.5%. On the other hand, under high traffic, the blocking probability is 4% higher for the No-Split

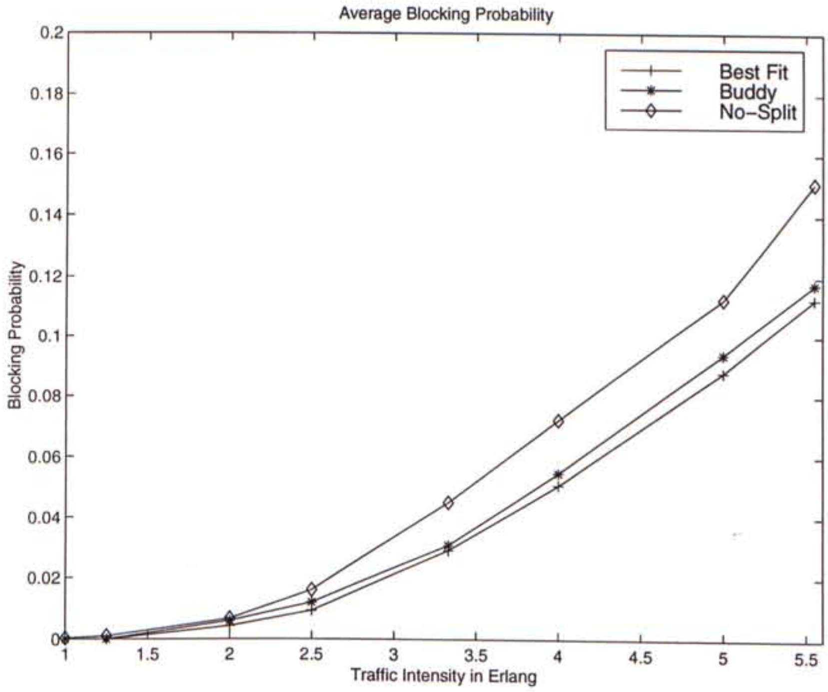


Figure 6.2: Blocking Probability vs Traffic Intensity under FCA

algorithm when compared with the Best Fit algorithm.

One of the tradeoffs for the reduction in processing time for the Buddy algorithm is the increase in average number of frequency bands needed per connection. However, figure 6.3 shows that this increase is relatively small when compared with the Best Fit algorithm.

From this, we can conclude that the proposed Buddy algorithm requires short processing time yet gives rise to a low blocking rate in FCA environment with minimal tradeoff.

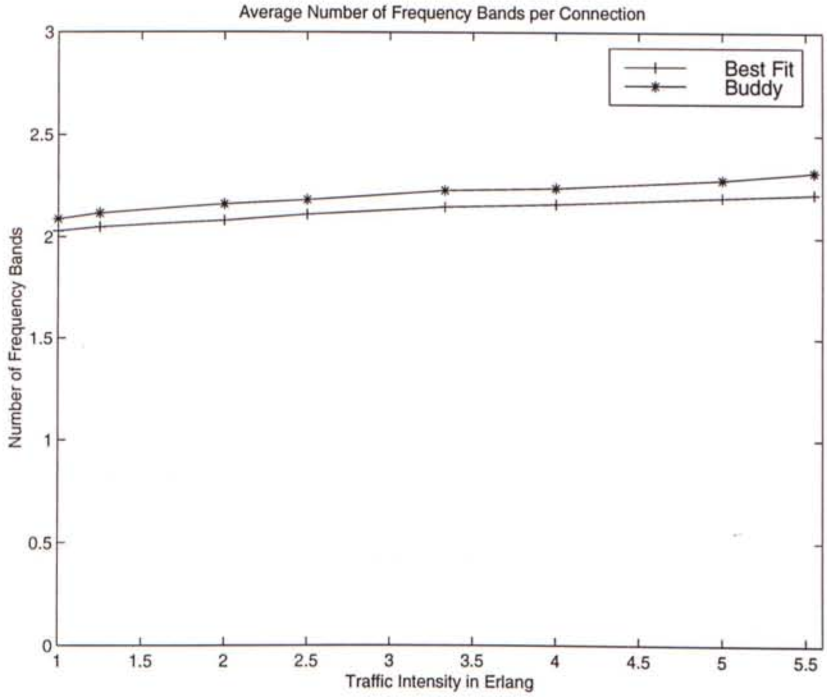


Figure 6.3: Average Fragmentation vs Traffic Intensity under FCA

6.3 Dynmaic Channel Assignment

6.3.1 System Parameters

In this section, the traffic rate of a 7-cell cluster is varied to investigate the performance of the Buddy algorithm compared with the Best Fit and No-Split algorithms. DCA is employed to share a central pool of frequencies among the seven cells. Statistics from all the seven cells are averaged to have a complete picture of the system. Simulations are done under the following parameters:

- The cluster consists of 7 cells.
- Total number of frequency bands in the central pool for sharing = 49
- Number of slots per frequency band (N) = 8.

- Number of usable frequency bands in each connection $(M) = 3$.
- Poisson arrivals with rates $\lambda = 3, 3.33, 5, 6, 7.5, 8, 9, 10$ per second apply to each cell.
- Exponential service time with mean $\mu = 0.5$s.
- User demand D is uniformly distributed in $[1, 2N]$.

6.3.2 Simulation Results

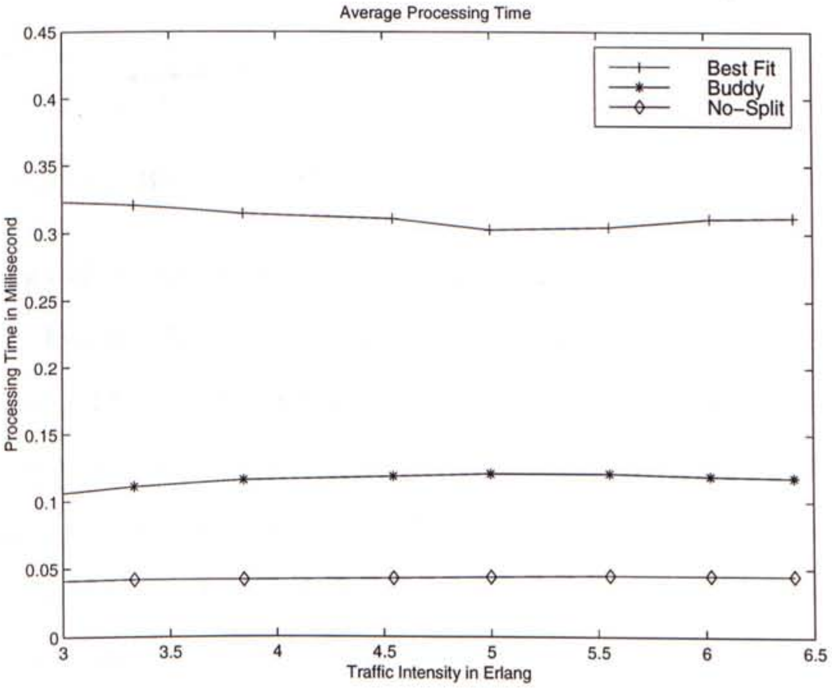


Figure 6.4: Processing Time vs Traffic Intensity under DCA

From figure 6.4, we can see that under the DCA environment, the Best Fit algorithm requires the longest processing time. The Buddy algorithm needs only about one-third of the time to work on an allocation request.

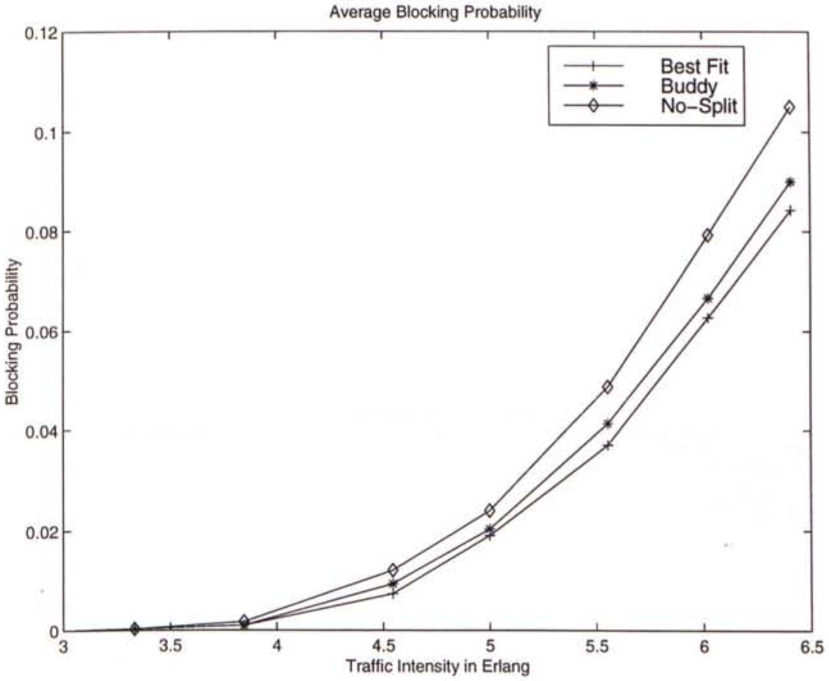


Figure 6.5: Blocking Probability vs Traffic Intensity under DCA

The plots of blocking probability in figure 6.5 agree with the findings in section 6.2. The Buddy algorithm can achieve a low blocking rate that is close to the one obtained by the Best Fit algorithm. This phenomenon can be explained by the slot occupancies shown in figure 6.6. Higher channel usage can be achieved by both the Best Fit algorithm and the Buddy algorithm. More connections can then be supported by the same amount of resources when compared with the No-Split algorithm. When the system is saturated with high traffics, we can see from figure 6.6 that the slot occupancies for the Buddy algorithm and the Best Fit algorithm differ by only about 1% and both can out-perform the No-Split algorithm.

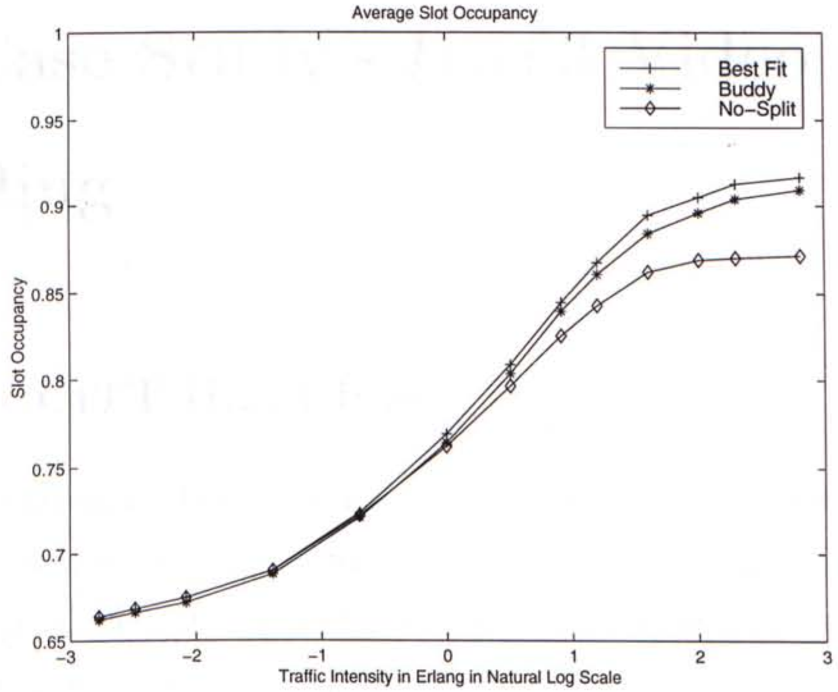


Figure 6.6: Slot Occupancies vs Traffic Intensity under DCA

Chapter 7

A Case Study - H.263 Video Coding

7.1 CCITT H.263 Image Compression

H.263 is a provisional ITU-T standard [24] published in 1995. It was designed as an image compression algorithm for low bit rate transmissions [25]. Early drafts specified data rates at less than 64 kbps. However, this limitation has now been removed and the standard is expected to be used for a wide range of bit rates. One of the applications of H.263 is in video conferencing. Videos with moderate quality can be provided even through low bit rate transmissions.

Half pixel precision is used for motion compensation. There are four optional negotiable options in H.263 for performance improvement. They are Unrestricted Motion Vectors, Syntax-based arithmetic coding, Advance prediction, and forward and backward frame prediction similar to MPEG called P-B frames. Five resolutions are supported in H.263 for different bit rate requirements. They

Picture formats	**Luminance pixels**	**Luminance lines**	**Uncompressed bit rates**			
			10 frames/s		**30 frames/s**	
			Grey	**Colour**	**Grey**	**Colour**
SQCIF	128	96	1.0	1.5	3.0	4.4
QCIF	176	144	2.0	3.0	6.1	9.1
CIF	352	288	8.1	12.2	24.3	36.5
4CIF	704	576	32.4	48.7	97.3	146.0
16CIF	1408	1152	129.8	194.6	389.3	583.9

Table 7.1: H.263 Picture Formats

Description	**Average PSNR(dB)**	**Bitrate (kbit/s)**	**Compression Ratio**
cline2-4 Original, 30 fps	N/A	9124	1:1
10 fps, 20 kbps	29.79	21.83	139:1
10 fps, 50 kbps	32.82	52.76	58:1
10 fps, 100 kbps	36.0	105.47	29:1
10 fps, 500 kbps	44.5	522.4	6:1

Table 7.2: Various Coding Rates by H.263

are summerized in table 7.1.

The 4CIF and 16CIF are 4 and 16 times the resolution of CIF respectively. In SQCIF, video sequences can be coded at 10 frames per second and modulated by 4, 16, 64-QAM to give rise to bit rates of 15, 30, 45 kbps.

7.2 On a GSM Network

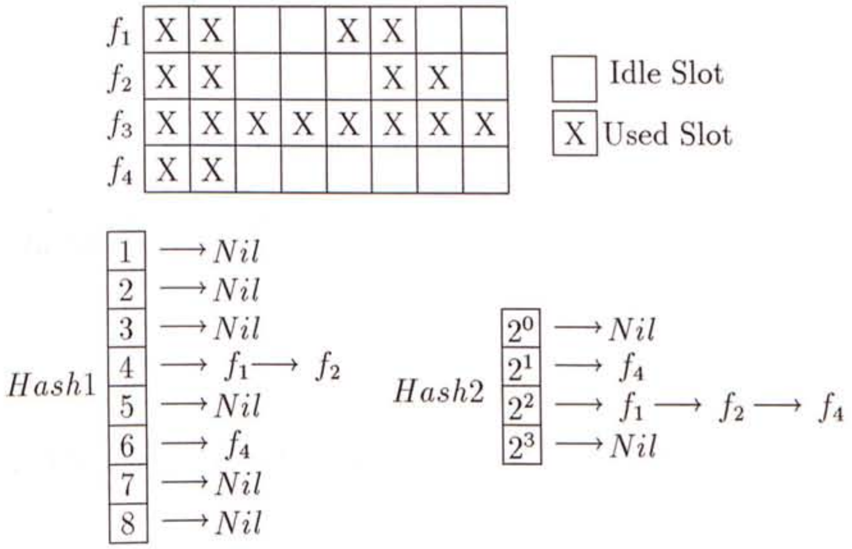
In this section, we apply the H.263 standard in coding video images. An original 30 frames/s MPEG video stream is coded by H.263 to provide low bit rate transmissions. Table 7.2 shows different bit rate requirements resulting from different coding rates.

The data channels in GSM are used for the transmission of a 100 kbps H.263

coded video stream. Since each data channel supports a bit rate of 9.6 kbps, a total of 11 data channels (11 time slots) are needed. Given a channel usage shown in figure 7.1, by the Buddy algorithm we can arrive at a solution after 3 iterations (Assume at most 3 frequencies can be used by a mobile unit).

$\vec{a} = \{1,0,1,1\} \longrightarrow \{1,1,0,0\} \longrightarrow \{0,3,0,0\}$

From $Hash2$, only the coefficient vector {0,3,0,0} can be satisfied by the system. Thus, a total of 12 idle slots (2^2 from each frequency band) from f_1, f_2 and f_4 are selected. Since only 11 slots are needed, one of the 12 slots is rehashed into $Hash2$ as shown in figure 7.2.



f_1	X	X			X	X		
f_2	X	X				X	X	
f_3	X	X	X	X	X	X	X	X
f_4	X	X						

☐ Idle Slot
X Used Slot

Figure 7.1: Channel Usage

With this allocation, the H.263 coded video stream can then be transmitted in its required bit rate over a GSM network.

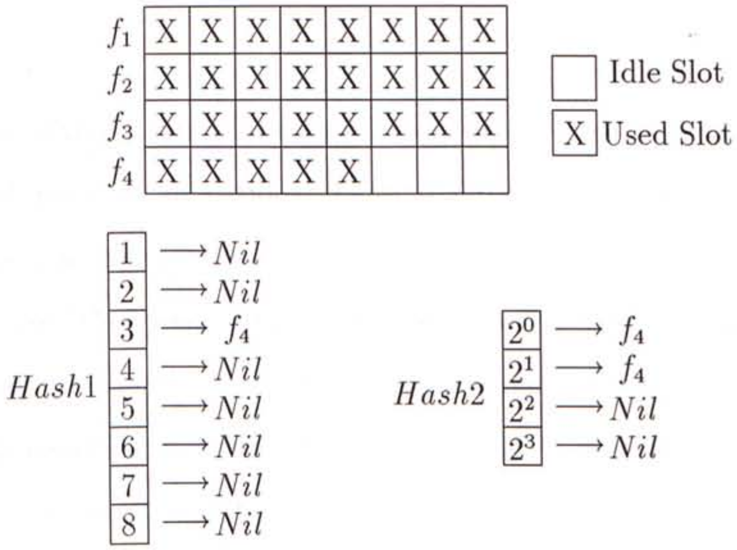


Figure 7.2: Channel Usage after the Allocation for a H.263 Video Stream

Chapter 8

Conclusion

The adoption of multi-rate data transmissions in a hybrid TDMA/FMDA system requires multiple slots allocations. Three different algorithms in slot allocation are described and compared in this thesis. The basic idea of the No-Split algorithm is to avoid the splitting of connection requests. Combinations of idle slots from different frequency bands are not considered. New frequency bands are acquired whenever no single frequency band can fit a request. This algorithm is fast and easy to implement but wastes bandwidth. A high blocking rate is resulted. In contrast, the Best Fit algorithm checks for all different combinations of idle slots in different frequency bands. This ensures a locally optimal use of resources and hence a low blocking rate. However, a long processing time is incurred. The high blocking rate in No-Split algorithm and the long processing time in Best Fit algorithm make them not appealing to be a practical solution for mulitple slots allocations.

In view of their disadvantages, a heuristic algorithm called the Buddy algorithm is proposed in this thesis. The Buddy algorithm groups idle slots in each frequency band into holes with sizes of power of two. This greatly reduces the number of slot combinations for consideration. Moreover, an inference property of the algorithm can further speed up the process. Simulations are done to evaluate this algorithm by means of blocking probability and processing time. The Buddy algorithm is shown to have both a short processing time and a low blocking probability when compared with the Best Fit algorithm and the No-Split algorithm.

Appendix A

A General Data + Voice System with Channel Rearrangement

A.1 System Model

Markovian analysis can be applied to the channel rearrangement scheme in a simplified data + voice system. Users are limited to two types. FCA is assumed to provide channel assignment. Thus, a fixed number of frequency bands are now assigned to the cell of interest. The following parameters and assumptions are employed for the analysis.

- The cell is assigned with C frequency bands.
- Number of slots per frequency band is N.
- Poisson arrivals with rate λ per second.
- Holding time is exponentially distributed with means $1/\mu_1$ s and $1/\mu_2$ s for type 1 and type 2 users respectively.

- Only two types of user exist.
- An arrival is a type 1 or type 2 user with equal probability.
- Type 1 users request for 1 slot for voice transmission. Type 2 users request for t slots for data transmission.
- CN is divisible by t.

States in this model are defined as the number of type 1 and type 2 users in the system. An ordered pair (t_1, t_2) denotes a state with t_1 type 1 users and t_2 type 2 users. If we denote $K = CN$, there are totally $t + \sum_{i=t}^{K} \lfloor \frac{i}{t} \rfloor + 1 = \frac{1}{2t}(K+2)(K+t)$ states.

A.2 Markovian Analysis

There are 10 different types of states in the system and their balance equations are summarized as the following:

States	Rate of Leaving	=	Rate of Entering
$(0,0)$	$\lambda P_{(0,0)}$	$=$	$\mu_1 P_{(1,0)} + \mu_2 P_{(0,1)}$
$(i,0)$	$(\lambda + i\mu_1)P_{(i,0)}$	$=$	$(i+1)\mu_1 P_{(i+1,0)} + \mu_2 P_{(i,1)} + \frac{\lambda}{2} P_{(i-1,0)},$ $1 \le i \le K - t$
$(i,0)$	$(\frac{\lambda}{2} + i\mu_1)P_{(i,0)}$	$=$	$(i+1)\mu_1 P_{(i+1,0)} + \frac{\lambda}{2} P_{(i-1,0)},$ $K - t < i < K$
$(K,0)$	$K\mu_1 P_{(K,0)}$	$=$	$\frac{\lambda}{2} P_{(K-1,0)}$
$(0,j)$	$(\lambda + j\mu_2)P_{(0,j)}$	$=$	$\mu_1 P_{(1,j)} + (j+1)\mu_2 P_{(0,j+1)} + \frac{\lambda}{2} P_{(0,j-1)},$ $t \cdot j \le K - t$
$(0,j)$	$(\frac{\lambda}{2} + j\mu_2)P_{(0,j)}$	$=$	$\mu_1 P_{(1,j)} + \frac{\lambda}{2} P_{(0,j-1)},$ $K - t < j < K$
$(0,\frac{K}{t})$	$\frac{K}{t}\mu_2 P_{(0,\frac{K}{t})}$	$=$	$\frac{\lambda}{2} P_{(0,\frac{K}{t}-1)}$
(i,j)	$(\lambda + i\mu_1 + j\mu_2)P_{(i,j)}$	$=$	$(i+1)\mu_1 P_{(i+1,j)} + (j+1)\mu_2 P_{(i,j+1)} + \frac{\lambda}{2} P_{(i,j-1)} + \frac{\lambda}{2} P_{(i-1,j)},$ $i > 0, j > 0, i + t \cdot j \le K - t$
(i,j)	$(\frac{\lambda}{2} + i\mu_1 + j\mu_2)P_{(i,j)}$	$=$	$(i+1)\mu_1 P_{(i+1,j)} + \frac{\lambda}{2} P_{(i,j-1)} + \frac{\lambda}{2} P_{(i-1,j)},$ $i > 0, j > 0, K - t < i + t \cdot j < K$
(i,j)	$(i\mu_1 + j\mu_2)P_{(i,j)}$	$=$	$\frac{\lambda}{2} P_{(i,j-1)} + \frac{\lambda}{2} P_{(i-1,j)},$ $i > 0, j > 0, i + t \cdot j = K$

The solution set is found to be in product form as:

$$\begin{cases} P_{(i,0)} = \frac{1}{i}\left(\frac{\lambda}{2\mu_1}\right) P_{(i-1,0)}, & 1 \le i \le K \\ P_{(0,j)} = \frac{1}{j}\left(\frac{\lambda}{2\mu_2}\right) P_{(0,j-1)}, & 1 \le j \le \frac{K}{t} \\ P_{(i,j)} = \frac{1}{i}\left(\frac{1}{j}\right)\left(\frac{\lambda}{2\mu_1}\right)\left(\frac{\lambda}{2\mu_2}\right) P_{(i-1,j-1)}, & i > 0, j > 0, i + t \cdot j \le K \end{cases} \tag{A.1}$$

This solution set satisfies all the 10 equations. Together with $\sum_i \sum_j P_{(i,j)} = 1$, we are able to work out the limiting probabilities for all states.

By the rearrangement of channels, a user will be blocked only when there are not enough idle slots. Thus, a type 1 user will be blocked when the system is in state (i,j) where $i + t \cdot j = K$. A type 2 user will be blocked when the system is in state (i,j) where $i + t \cdot j > K - t$. The blocking probability is then given by the following equation:

$$Pr(block) = \sum_{i+t \cdot j = K} P_{(i,j)} + \frac{1}{2} \sum_{i+t \cdot j = K-t+1}^{K-1} P_{(i,j)} \tag{A.2}$$

Appendix B

NP-Completeness Proof of the Best Fit Algorithm

B.1 CONSTRAINT SUBSET-SUM Problem

A variation of the well known NP-complete SUBSET-SUM problem called CONSTRAINT SUBSET-SUM problem is defined and is proved to be NP-complete.

Problem Definition

Given a finite set $H \subset \mathrm{N}$, a target $k \in \mathrm{N}$ and a constraint $M \in \mathrm{N}$, define a CONSTRAINT SUBSET-SUM problem:

$$\text{CONSTRAINT SUBSET-SUM} = \{< H, k, M >\text{: there exists a subset } H' \subseteq H \text{ such that } \sum_{h \in H'} h = k \text{ and } |H'| \leq M\}$$

Verification

For an instance of $< H, k, M >$, let the subset H' be the certificate, checking whether $\sum_{h \in H'} h = k$ and $|H'| \leq M$ can be accomplished by a verification algorithm in polynomial time.

$\Rightarrow$ CONSTRAINT SUBSET-SUM is in NP.

Reduction Algorithm

To show SUBSET-SUM $\leq_p$ CONSTRAINT SUBSET-SUM, the reduction algorithm is as follows:

For any instance $< S, t >$ of the SUBSET-SUM problem, we can map $< S, t >$ into an instance $< H, k, M >$ of the CONSTRAINT SUBSET-SUM problem with

1. $H = S$
2. $k = t$
3. $M = |S|$

Necessary Condition

To show $< H, k, M >$ satisfies CONSTRAINT SUBSET-SUM if $< S, t >$ satisfies SUBSET-SUM.

$< S, t > \in$ SUBSET-SUM $\Rightarrow$ there exists a subset S' such that

a. $S' \subseteq S$

b. $\sum_{s \in S'} s = t$

By the reduction algorithm, we can have a $H' = S'$ such that

1. $H' \subseteq H$

   $H' = S'$ & $S' \subseteq S \Rightarrow H' \subseteq S$

But $H = S \Rightarrow H' \subseteq H$

2. $\sum_{h \in H'} h = k$

   $H' = S' \Rightarrow \sum_{h \in H'} h = \sum_{s \in S'} s = t$

   But $k = t \Rightarrow \sum_{h \in H'} h = k$

3. $|H'| \leq M$

   $H = S \,\&\, M = |S| \Rightarrow M = |H|$

   But $H' \subseteq H \Rightarrow |H'| \leq |H|$

   $\Rightarrow |H'| \leq M$

Sufficient Condition

To show $< S, t >$ satisfies SUBSET-SUM if $< H, k, M >$ satisfies CONSTRAINT SUBSET-SUM.

$< H, k, M > \in$ CONSTRAINT SUBSET-SUM $\Rightarrow$ there exists a subset H' such that

a. $H' \subseteq H$

b. $\sum_{h \in H'} h = k$

c. $|H'| \leq M$

By the reduction algorithm, there exists a $S' = H'$ such that

1. $S' \subseteq S$

   $S' = H' \,\&\, H' \subseteq H \Rightarrow S' \subseteq H$

   But $H = S \Rightarrow S' \subseteq S$

2. $\sum_{s \in S'} s = t$

   $S' = H' \Rightarrow \sum_{s \in S'} s = \sum_{h \in H'} h = k$

   But $k = t \Rightarrow \sum_{s \in S'} s = t$

From the above, one concludes that SUBSET-SUM $\leq_p$ CONSTRAINT SUBSET-SUM.

Conclusion

1. CONSTRAINT SUBSET-SUM is in NP.
2. SUBSET-SUM $\leq_p$ CONSTRAINT SUBSET-SUM.
3. SUBSET-SUM is NP-complete.

$\Rightarrow$ CONSTRAINT SUBSET SUM is NP-complete.

B.2 BEST-FIT Problem

In this section, we are going to show that the best fit algorithm is NP-complete. We can define the best-fit search problem as:

BEST-FIT $= \{< \vec{n}, M, N, D >$: there exists a non-negative integer N -dimensional vector $\vec{x}$ such that $\sum_{i=1}^{N} x_i i = D$, $\sum_{i=1}^{N} x_i \leq M$ and $x_i \leq n_i$ for $i = 1$ to $N\}$

where N = number of slots per carrier.
M = number of usable frequency bands in a mobile unit.
D = current call request's demand.
n_i = number of frequencies with i idle time slots.
x_i = number of allocated frequencies with i idle time slots.

With this definition, the CONSTRAINT SUBSET-SUM problem $< H, k, M >$ defined above can be reduced to the BEST-FIT problem. Denote H_i be a subset of H with positive integer elements and are all equal to i. Then, the followings transform the CONSTRAINT SUBSET-SUM problem to the BEST-FIT problem.

1. $n_i = |H_i|$ where $\bigcup_i H_i = H$
2. $N = max_{h \in H} h$
3. $M = M$
4. $D = k$

Now, an instance (a N-dimensional vector $\vec{x}$) of the BEST-FIT problem is actually an instance (a subset $H' \in H$) of the CONSTRAINT SUBSET-SUM problem an instance (a N-dimensional vector $\vec{x}$) of the BEST-FIT problem with $|H'_i| = x_i$.

Similar to the approach used to prove the NP-completeness of the CONSTRAINT SUBSET-SUM problem, we can prove that the BEST-FIT problem is also NP-complete.

Appendix C

Proof of Proposition 5.2

C.1 Upper Bound on Demand Advancement

In the Buddy algorithm, all those linear combinations that satisfy the frequency band constraint by splitting of coefficients for a demand D will be tried. If all these linear bombinations cannot be satisfied, the demand D will be advanced to the next value that is not implied as failure by the $inferenceproperty$ and the search restarts again. As a result, the total number of coefficient vectors that need examinations equals to the sum of the linear combinations for each demand value. In this section the upper bound on the number of advancement of a demand request is found based on the *Inference Property* stated in section 5.4.

By the procedure *Advance()* defined in section 5.4, we can see that in worst case, we have to stop advancement in every bit of the binary representation of the demand D. The procedure halts until D is advanced 2^{n+1} where $2^n \leq D < 2^{n+1}$. For example, if $D = 17$, the following advancements are made for examinations. $\{1,0,0,0,1\} \longrightarrow \{1,0,0,1,0\} \longrightarrow \{1,0,1,0,0\} \longrightarrow \{1,1,0,0,0\} \longrightarrow \{1,0,0,0,0,0\}$

There are totally 4 advancements and D is advanced to 18, 20, 24, 32. In general, there are $\lfloor log_2 D \rfloor$ advancements in worst case as there are $\lfloor log_2 D \rfloor$ coefficients in the binary representation of D. Hence, at most $\lfloor log_2 D \rfloor + 1$ demands are examined by the Buddy algorithm for a connection request of demand D.

C.2 Proof of Proposition 5.2

In the following proof, the constraint in a mobile unit in using multiple frequency bands (M) is assumed to be 3 to simplify the calculation. Then, the demands Ds for search can be classified into 3 types according to the number of '1's in their binary representation $\vec{p}$:

1. Number of '1's in $\vec{p} = 1$
2. Number of '1's in $\vec{p} = 2$
3. Number of '1's in $\vec{p} = 3$

All other demands are ignored in the allocation as they violate the frequency band constraint with $\sum_{i=0}^{n} p_i > M$. By the *Split()* procedure, for each demand, other linear coefficient vectors $\vec{a}$ are generated by the splitting of non zero coefficients. For each type of demand classified above, we can have at most 2 splittings as shown below without violating the frequency band constraint ($M = 3$):

1. $\{1,0,0,0\} \longrightarrow \{0,2,0,0\} \longrightarrow \{0,1,2,0\}$
2. $\{1,0,1,0\} \longrightarrow \{0,2,1,0\} \longrightarrow \{1,0,0,2\}$
3. $\{1,1,1,0\}$

In conclusion, for each demand D, there are at most 3 different linear combinations for the Buddy algorithm to search. Accompanied with the result obtained from section C.1, we can claim that the total number of different non-negative linear combinations generated for a request of D slots $\leq 3(\lfloor \log_2 D \rfloor + 1)$.

Bibliography

[1] Didier J. Le Gall, "The MPEG Video Compression Standard," *Compcon Spring '91, Digest of Papers.*

[2] Pedro A. A. Assuncao, Mohammed Ghanbari, "Post-Processing of MPEG2 Coded Video for Transmission at Lower Bit Rates," *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol.4, 1996, pp.1998-2001.

[3] Gringeri S., Khasnabish B., Lewis A., Shuaib K. Egorov R., Basch B., "Transmission of MPEG-2 Video Streams over ATM," *IEEE Multimedia*, vol.5, 1998, pp.58-71.

[4] Karel Rijkse, KPN Research, "H.263: Video Coding for Low-Bit-Rate Communication," *IEEE Communications Magazine*, vol.34, issue 12, Dec. 1996, pp.42-45.

[5] Jian Zhang, Michael R. Frater, John F. Arnold and Terence M. Percival, "MPEG2 Video Services for Wireless ATM Networks," *IEEE Journal on Selected Areas in Communications,* vol. 15, No. 1, 1997.

[6] I. Katzela and M. Naghshineh, "Channel Assignment Schemes for Cellular Mobile Telecommunication Systems: A Comprehensive Survey," *IEEE Personal Communication, June, 1996*.

[7] Leonard J. Cimini, Jr., Gerard J. Foschini and Larry A. Shepp, "Single-Channel User-Capacity Calculations for Self-Organizing Cellular Systems," *IEEE Journal in Communications*, vol. 42, No. 12, Dec., 1994.

[8] Y. Akaiwa and H. Andoh, "Channel Segregation-A self Organized Dynamic Allocation Method: Appliciation to TDMA/FDMA Microcellular System," *JSAC*, vol. 11, 1993, pp.949-54.

[9] Hua Jiang and Stephen. S. Rappaport, "Prioritized Channel Borrowing Without Locking: A channel sharing strategy for cellular communication systems," *Proc. IEEE Global Communication Conf., Huston, TX*, Nov. 29-Dec. 2, 1993, pp.276-280.

[10] Tai-Po Chu, Stephen S. Rappaport, "Overlapping Coverage with Reuse Partitioning in Microcellular Communication Systems," *Proc. 6th International Symposium on Personal, Indoor and Mobile Radio Communications*, vol.2, 1995, pp.557-561.

[11] Peter T. H. Chan, Marimuthu Palaniswami, and David Everitt, "Neural Network-Based Dynamic Channel Assignment for Cellular Mobile Communication Systems," *IEEE Transaction on Vehicular Technology*, vol. 43, no. 2, May 1994.

[12] L. J. Cimini, Jr., G. J. Foschini, "Distributed Algorithms for Dynamic Channel Allocation in Microcellular Systems," *Proc. IEEE 42nd Vehicular Technology Conference*, vol.2, 1992, pp.641-644.

[13] Keisuke Nakano, Masakaze Sengoku, Shoji Shinoda, and Takeo Abe, "Clique Packing in Cellular Mobile Communication Systems," *Proc. 4th IEEE International Conference on Universal Personal Communications*, 1995, pp.71-75.

[14] D. C. Cox and D. O. Reudink, "Dynamic Channel Assignment in Two Dimension Large-Scale Mobile Radio Systems," *Bell Sys. Tech. J.*, vol. 51, 1972, pp. 1611-28.

[15] D. Everitt and N. W. MacFadyen, "Analysis of Multicellular Mobile Radio Telephone Systems and Loss," *BT Tech. J.*, vol. 2 1983, pp.37-45.

[16] A. Gamst, "Some Lower Bounds for a Class of Frequency Assignment Problems," *IEEE Transaction on Vehicular Tech.*, vol. 35, 1986.

[17] M. Serizawa and D. Goodman, "Instability and Deadlock of Distributed Dyanmic Channel Allocation," *Proc. 43rd IEEE Vehicular Technology Conference*, 1993, pp.528-531.

[18] Dag Akerberg, "Properties of a TDMA Pico Cellular Office Communication System," *Proc. IEEE 39th Vehicular Technology Conference*, vol.1, 1989, pp.186-191.

[19] Moe Rahnema, "Overview of the GSM System and Protocol Architecture," *IEEE Communications Magazine*, vol.31 issue 4, 1993, pp.92-100.

[20] Theodore S. Rappaport, "Wireless Communications Principles and Practice," *Prentice Hall PTR*, 1996.

[21] Andrew S. Tanenbaum, "Operating Systems Design and Implementation," *Prentice-Hall, Inc.*, 1987.

[22] Wai-Leung Wan, Wing-Shing Wong, "A Heuristic Algorithm for Channel Allocation of Multi-rate Data in Hybrid TDMA/FDMA Digital Cellular Systems," *9th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications* 1998.

[23] Per-Ola Gaasvik, Maria Cornefjord, Viktoria Svensson, "Different Methods of Giving Priority to Handoff Traffic in a Mobile Telephone System with Directed Retry," *Proc. 41th IEEE Vehicular Technology Conference*, 1991, pp.549-553.

[24] ITU-T, SG15, *ITU Recommendation H.263*, "Video Coding for Narrow Telecommunication Channels at ¡ 64 kbit/s," April 1995.

[25] Paramvir Bahl, Imrich Chlamtac, "H.263 Based video Codec for Real-Time Visual Communications Over Wireless Radio Networks," *Proc. 6th International Conference on Universal Personal Communications*, vol.2, 1997, pp.773-779.

[26] Thomson H. Cormen, Charles E. Leiserson and Ronald L. Rivest, "Introduction to Algorithms," *The MIT Press, McGraw-Hill Book Company*, 1990, pp.951-953.