EFFICIENT UTLIZATION OF OVSF CODES FOR CDMA NETWORKS

BY

VIPIN

**A THESIS SUBMITTED IN FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
IN**

ELECTRONICS AND COMMUNICATION ENGINEERING

UNDER THE GUIDANCE OF

Dr. DAVINDER SINGH SAINI

**JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY
WAKNAGHAT, H.P, INDIA (12)**

ROLL NO. 086009 **AUGUST 2013**

SUPERVISOR'S CERTIFICATE

This is to certify that the work reported in the Ph.D. thesis entitled "**EFFICIENT UTLIZATION OF OVSF CODES FOR CDMA NETWORKS**", submitted by **VIPIN** at **Jaypee University of Information Technology University, Solan, India,** is a bonafide record of his original work carried out under my supervision. This work has not been submitted elsewhere for any other degree or diploma.

Davinder S. Saini

Jaypee University of Information Technology University, Solan, India

Date : 23/08/2013

ACKNOWLEDGMENT

The research undertaken during the past 4 and half years would not have been possible without the guidance and assistance I have been able to enjoy. First of all, I would like to sincerely thank my research guide Dr. D. S. Saini who has continued to be an inspiration since the start of this work. His enthusiasm and engagement in giving guidance and sharing knowledge cannot be valued, as well as his editing during the preparation of this document, were invaluable. I would like to thanks my committee members who have each made no small contribution in form of their suggestions Dr. T. S. Lamba, Dr. S. V. Bhoosan, Dr. G. S. Singh and Dr. R. S. Durai. Next I would like to thank my colleague Dr. S. V. Singh, Alok Joshi and my wife Gunjan for their invaluable suggestions to improve thesis. I would like to thank Dr. Y. Medury and Dr. S. V. Bhoosan for providing excellent research environment in JUIT, Waknaghat. I would Finally, I would like to thank Brig. (Retd.) Balbir Singh for his help and advice.

VIPIN

TABLE OF CONTENTS

CHAPTER 7

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

Spread spectrum concept in a wireless communication system [1] allows multiple users to occupy the same transmission band for simultaneous transmission of signals without considerable interference. It is a communication technique which transforms user signal into another form that occupies a larger bandwidth than the original signal would normally need. The process of transformation is known as spreading. It was initially developed for military and intelligence applications. Since last decade, spread spectrum systems have evolved from strictly military applications to commercial navigation and communication systems. The two prominent examples of this are: the Global Positioning System (GPS) navigation system, and the IS-95 (Interim standard) Code Division Multiple Access (CDMA) cellular telephone system. Spreading on a wide bandwidth makes jamming and interception more difficult. The bandwidth spread is accomplished by means of a code which is independent of the data and synchronized reception with the code at the receiver is used for *despreading* and subsequent data recovery. Spread spectrum signals are highly resistant to narrowband interference. This technique decreases the potential interference to other receivers while achieving privacy. To recover the transmitted data sequence at the destination mobile phone side, the same spreading code is used at the receiver. The spreading of signals depends upon *spreading factor* (*SF*). The incoming rate multiplied by spreading factor is called *chip rate*. The chip rate is always larger than the original data rate (symbol rate) i.e., $SF = Chip\ rate / Symbol\ rate$.

1.1 PROPERTIES OF SPREAD SPECTRUM

The purpose of transmitting a signal with a bandwidth wider than required is to improve the communications system performance. Spread spectrum systems have the following properties, which are utilized to improve performance relative to narrow band communication systems.

1.1.1 Multiple Access. A spread spectrum system allows multiple users to transmit simultaneously with same transmission bandwidth. This is accomplished by assigning to each user a unique code from a set of codes with low cross-correlation properties. Codes with low-cross correlation properties are also termed orthogonal spreading codes. Demodulation by correlating the received signals with the desired code will recover the desired signal. The undesired signals will remain spread over the transmission bandwidth and will contribute noise power to the desired signal-to-noise ratio (S/N). Unlike time division multiple access (TDMA) or

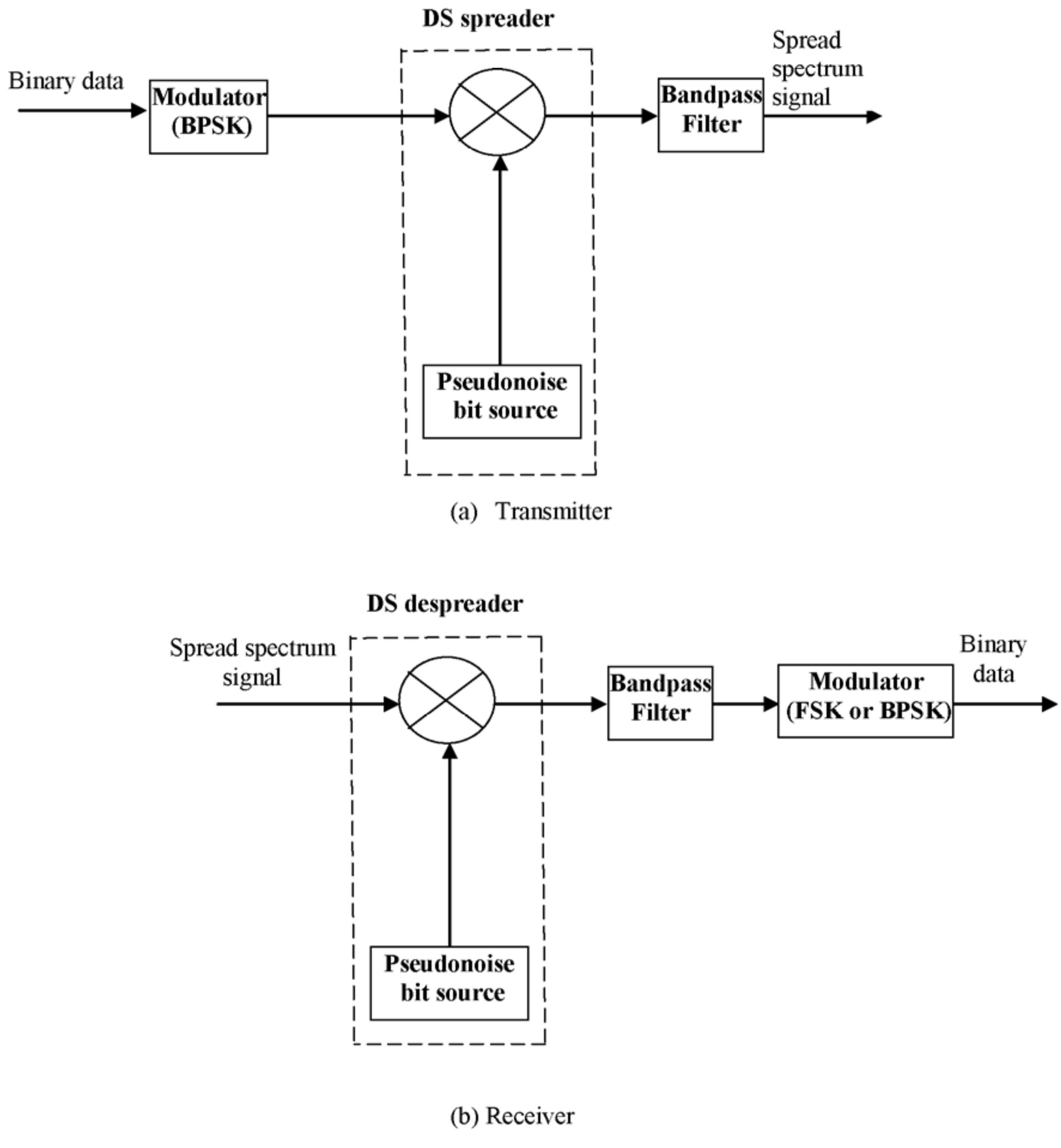


Figure 1.1: Direct Sequence Spreading Spectrum System [1]

frequency division multiple access (FDMA) schemes, spread spectrum multiple access (SSMA) systems do not have a hard limit on the maximum number of users. Increasing the number of users decreases the S/N ratio for existing users, and increases the bit error probability (P_b) for all users.

1.1.2 Multipath interference reduction. A spread spectrum system is capable of reducing multipath interference. Multipath interference occurs when several copies of a transmitted signal arrive at a receiver due to reflection and refraction. The multiple signals arrive with different distortions in amplitude, phase, and time delay, which causes them to combine both constructively and destructively. This interference produces frequency selective fading of the received signal. Multipath interference is reduced by correlating the received signals with the spreading code to recover the desired signal. Multipath signals with large distortions in amplitude, phase or time delay will appear uncorrelated with the spreading code and will remain spread.

1.1.3 Privacy. A spread spectrum system offers some degree of privacy because the spreading code is required to recover the transmitted signal.

1.1.4 Anti-jamming. A spread spectrum system is capable of reducing the effects of narrowband jamming. This is possible because correlating the jamming signal with the spreading code in effect spreads the jamming signal while despreading the desired signal.

1.1.5 Low Probability of Intercept. Spread spectrum signals are difficult to detect, a characteristic termed as low probability of intercept (LPI), because the signal power is dispersed over a wide bandwidth. The transmitted signal is then difficult to distinguish from noise.

The ratio of transmitted bandwidth (B_t) to information bandwidth (B_i) is often used to approximate the performance gain of a spread spectrum system. The performance improvement offered by a spread spectrum system is termed processing gain [2] $G_p = B_t / B_i$.

1.2 DIRECT SEQUENCE SPREAD SPECTRUM (DSSS)

In DSSS, the data signal is modulated by a PN code sequence that effectively spreads the signal power over a wide bandwidth. The block diagram of a DSSS transmitter is shown in Figure 1.1 [1]. The data signal may be analog or digital. The PN spreading code is a digital signal that takes on values of +1and −1, and the number of code bits per second is commonly called the chip rate (R_c). The chip rate is typically much larger than the data symbol rate (R_s), which results in the desired spreading in the frequency domain. The spreading factor (SF) of a DSSS system is the ratio of R_c to R_s. Multiple access is accomplished by assigning each user a unique spreading code from a set of codes with low cross correlation properties. The receiver recovers a desired signal from a group of spread signals by correlating with the correct spreading code. This demodulates the desired signal but not the signals of other users. The capacity of a DSSS system is interference limited. The spread signal of each user has properties similar to additive white gaussian noise (AWGN). Increasing the number of users effectively decreases the carrier to noise ratio (C/N) and increases the bit error probability (P_b), for all users. Multipath interference rejection is possible if the spreading codes have good autocorrelation properties. An ideal autocorrelation function has a value of zero outside the interval $[-T_c, T_c]$, where T_c is the chip duration. A multipath signal that is delayed by greater than $2T_c$, much like a signal modulated with an uncorrelated spreading code is not recovered by correlation with the desired spreading code. DSSS provides privacy because the receiver must know the spreading code to demodulate the signal. DSSS reduces the effects of narrowband jamming because correlating with the spreading codes despreads the desired signal and spreads the jamming signal. Thus only a small amount of the jamming power remains in the signal bandwidth. LPI is provided since the spread signal has properties similar to AWGN and can be hidden in the background noise.

WCDMA radio interface fundamental operation is spreading. The spreading codes in WCDMA are grouped into two types: *channelization* and *scrambling codes* [3-4]. The channels in the forward link and reverse link use theses codes for transmission. Spreading is used in combination with scrambling. Scrambling is basically done on top of spreading needed to separate terminals or base station (BS). Scrambling codes are generated from the stream called pseudo noise sequences [1-2]. They do not affect chip rate or bandwidth. *Channelization* codes are shorter in length and are made from orthogonal function [5-6]. The orthogonality property of channelization codes makes it suitable for WCDMA systems. The use of channelization codes and scrambling codes are different in the uplink and downlink. In downlink, *scrambling* code is used to make differentiation for cells and channelization code is used to make differentiation of service between MS in the same physical channel. In the uplink, scrambling codes are used for user identification and interference mitigation while channelization codes are used for rate matching. For the uplink transmission, the scrambling codes and channelization codes are different. For the downlink transmission, same scrambling code is used for channels corresponding to each mobile station (MS). Hence, in the downlink transmission the efficient use of channelization code becomes important. The channelization codes in WCDMA are OVSF codes. The OVSF codes are generated from the code tree generation given in [7-9]. One of the disadvantage associated with OVSF codes is that when a code is assigned to new call, all of its parents and children codes are blocked from the assignment. This is due to the fact that the codes from root to leaf are orthogonal to each other. This problem leads to blocking of a new call even though the code tree has enough codes to support new call. The code blocking leads to new call blocking. Basically the new call blocking is due to two limitations of OVSF-CDMA termed as internal fragmentation and external fragmentation [10]. The external fragmentation is because of the availability of vacant codes in scattered form in the code tree. The internal fragmentation is due to the quantized nature of the rate handling capability of the OVSF code tree. WCDMA specifies four different traffic classes namely; conversational, streaming, interactive and background with different quality of service (QoS) requirements [11-14]. The typical QoS parameters are throughput, delay, power, and capacity etc. The traffic corresponding to each class needs to be treated differently. Real time calls are always given higher priority compared to the non real time classes. The requirement of different QoS is discussed in [15-18]. The possible data rates for the WCDMA are R, $2R$, …., 128R (where R is 7.5kbps for downlink and 15kbps for uplink). The spreading factor (SF) for OVSF codes in the forward link is $4 \le SF \le 512$ and $4 \le SF \le 256$ in the uplink. The channel chip rate in WCDMA is fixed and is equal to 3.84 Mcps *i.e* $SF \times symbol\ rate = 3.84$ Mcps.

1.3 REVIEW OF OVSF-CDMA

The channelization codes in WCDMA are OVSF codes. The channels in the uplink and downlink use theses codes for transmission. OVSF codes are shorter in length and are made from orthogonal function. The orthogonality property of OVSF codes makes it suitable for WCDMA. The signals from two or more user equipments (UEs) in the reverse link are transmitted to same base station (BS) in the cell from separate locations. This change in the distance gives rise to change in time for the signals to reach at the BS. The orthogonal property of the OVSF codes is disturbed due to different arrival times. Hence, the OVSF codes are not used for calls separation in the downlink. The facility to handle variable call rate is also incorporated in OVSF codes. In contrast to OVSF codes, scrambling codes are quite long (with the exception of the uplink of the short scrambling code). Scrambling codes are generated from the stream called pseudo noise (PN) sequences [4-9]. In WCDMA, the requirement to have codes with high value of autocorrelation and low value of crosscorrelation. High autocorrelation properties are desired to recover the intended signal and to reduce the effect of multipath signals. The low cross correlation properties are required to minimize the effect of interfering signals. The PN codes and orthogonal codes individually do not have both good autocorrelation and good cross correlation

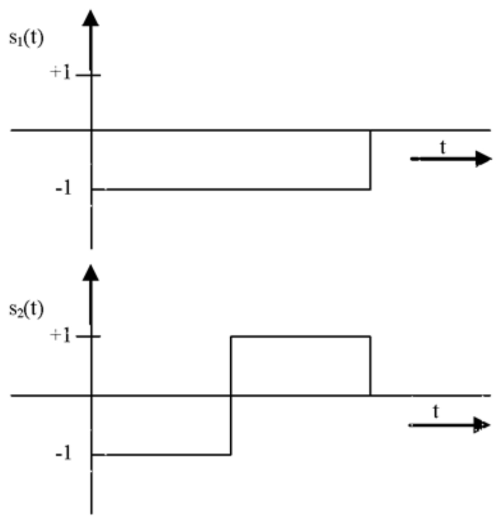


Figure 1.2:. Orthogonal signals

properties [1-2]. Orthogonal codes are sets of binary sequences that have a cross-correlation coefficient equal to zero. A set of periodic signals $s_i(t)$ and $s_j(t)$ is orthogonal if

$$\frac{1}{E}\int_0^T s_i(t)s_j(t)dt = \begin{cases} 0, & i \neq j \\ 1, & i = j \end{cases} \tag{1.1}$$

where, E is the energy of the signal given by

$$E = \int_0^T s_i^2(t)dt \tag{1.2}$$

In Equation (1.1), T is the period of the signals $s_i(t)$ and $s_j(t)$, and E is the signal energy as defined in Equation (1.2). The set of signals representing the orthogonal binary sequences 00 and 01 is shown in Figure 1.2. Walsh codes are orthogonal codes that are generated using a Hadamard matrix. A Hadamard matrix is a square matrix with the first row all zeros, and an equal number of ones and zeros on all other rows. A set of Walsh codes of length n consists of the n rows of an n x n Walsh matrix. That is, there are n codes each of length n. The matrix is defined recursively as follows.

$$W_1 = (0) \qquad W_{2n} = \begin{pmatrix} W_n & W_n \\ W_n & \overline{W_n} \end{pmatrix} \tag{1.3}$$

where n is the dimension of the matrix and the over score denotes the logical NOT of the bits in the matrix. The Walsh matrix has the property that every row is orthogonal to every other row and to the logical NOT of every other row [1-2]. The length of the spreading code (N) is equal to the number of orthogonal codes. The ideal cross-correlation properties of orthogonal codes make them attractive for separating users in a DS-CDMA system. Orthogonal codes are only useful in synchronous systems, as they have extremely poor cross-correlation properties when not synchronized. The downlink of a CDMA cellular network is synchronous and systems such as IS-95 use orthogonal codes to separate users. However, multipath interference on the downlink results in unsynchronized signals at the receiver. In addition, orthogonal codes have poor autocorrelation properties making signal detection difficult.

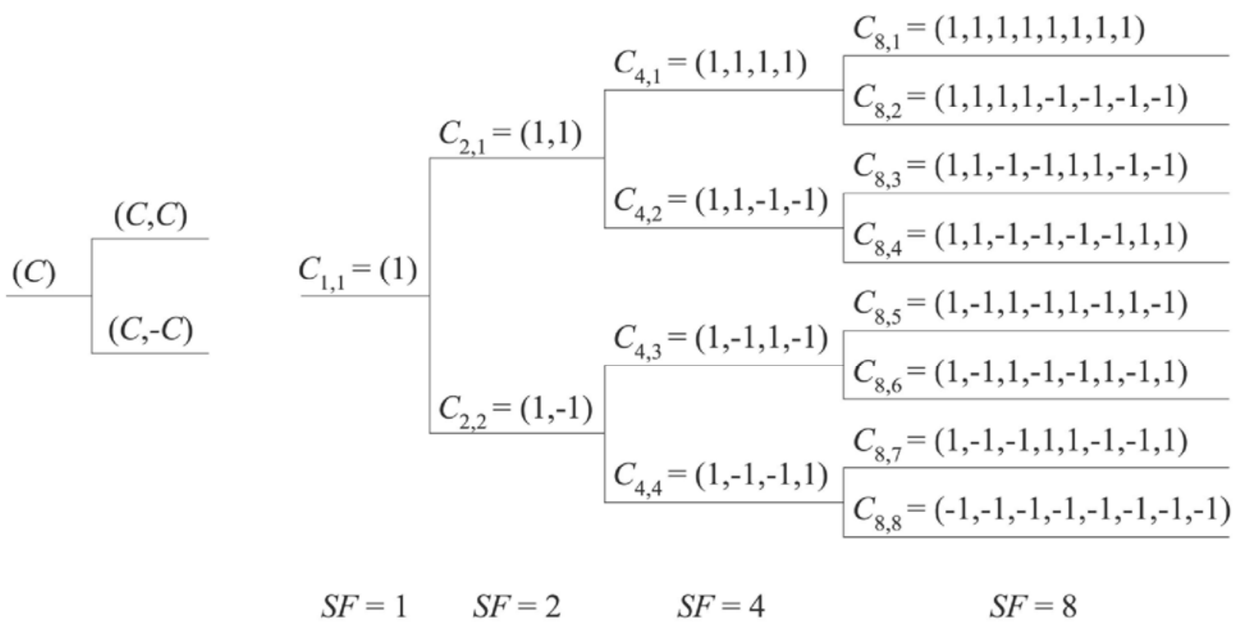


Figure 1.3: OVSF code tree generation.

1.3.1 OVSF CODE TREE

The OVSF code tree generation is explained in Figure 1.3. For simplicity in code generation, we considered root code as layer 1, i.e. layer number increases from top to bottom of the code tree [7-9]. For a code in layer 1, $C_{1,1}$=[1] and its two children codes in layer 2 will be $C_{2,1}$= [$C_{1,1}$, $C_{1,1}$] =[1,1] and $C_{2,1}$= [$C_{1,1}$,-$C_{1,1}$] =[1,-1] respectively. The codes in all the layers will be generated similarly from top to bottom to generate full OVSF code tree. In the assignment schemes utilizing OVSF codes for assignment, a code cannot be assigned to the incoming call if any of its children or parent code is already assigned to ongoing calls. *i.e* only one code can be assigned to a call in the path from root code to leaf code. In Figure 1.3, it is explained how a spreading factor varies in time domain from higher layer to lower layers and vice versa for data rate of a code. As we can see from Figure 1.3, *SF* of children codes is repetition of parent code twice for one children and a combination of parent code and its complement for another children in same time limits to keep orthogonality. Consider an OVSF-CDMA system with L number of layers and the code in layer l is represented by C_{l,n_l}, where $1 \le l \le L$ and $1 \le n_l \le 2^{L-l}$. The maximum capacity of each layer and the system is $2^{L-1}R$. The number of codes in layer l is 2^{L-l}. The spreading factor (*SF*) in layer l is 2^{L-l}. The data rate handled by the code in the layer l is $2^{l-1}R$.

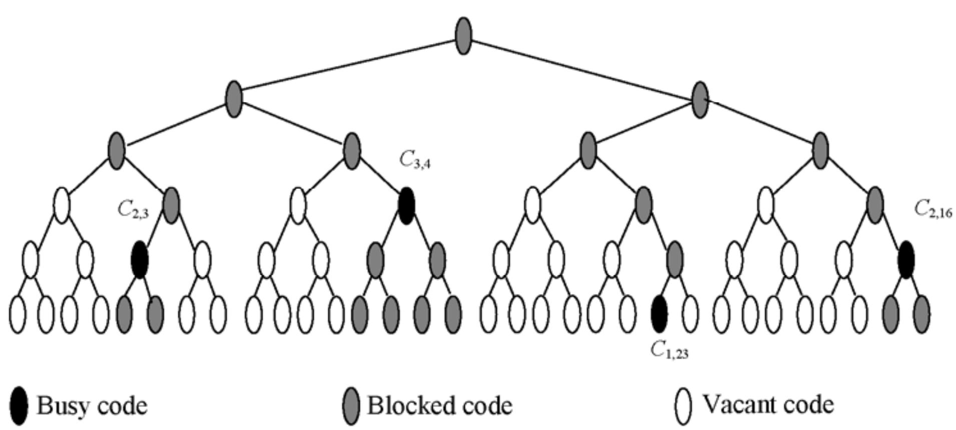
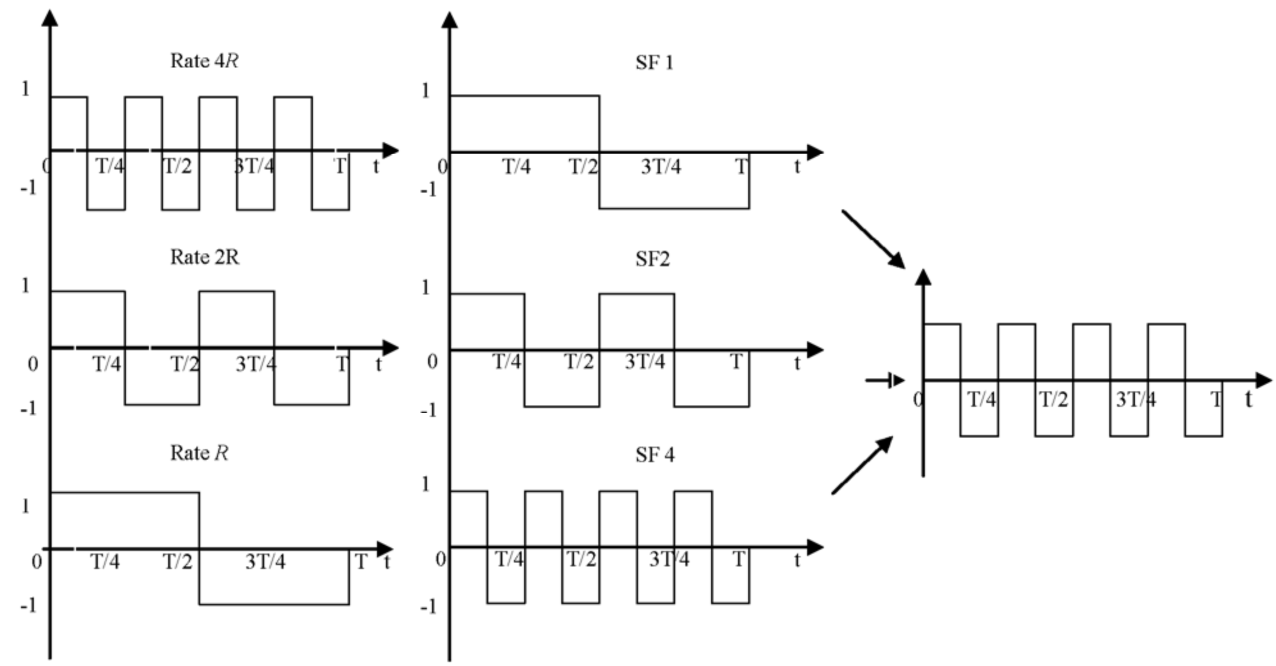


Figure 1.4: OVSF Code tree with six layers

1.3.2 CODE/CALL BLOCKING

The bit rate in OVSF-WCDMA is always quantized and is $2^{l-1}R$. With the increase in layer value from 1 to L, the rate handling capability doubles between two consecutive layers and



(a) Code from OVSF code tree (b) Data rate (c) Channel Data = Data rate × SF

Figure 1.5: The relationship between user data rate, spreading codes and the channel data rate for three users with rates R, $2R$ and $4R$. The spreading factors are normalized w.r.t. 4. [21]

becomes large as the difference between layers increases. This leads to internal fragmentation [10]. The two codes are assignable in the OVSF code tree if they are orthogonal; this leads to code blocking, a condition in which the new call cannot be accepted even though the system has enough capacity to handle new call. The maximum capacity of WCDMA system is always $2^{l-1}R$ (R is 7.5*kbps* in the downlink). We limit our discussion in the downlink transmission only. Let C_{l,n_l} represents a code in layer l, where $1 \le l \le L$ and $1 \le n_l \le 2^{L-l}$. The code blocking is explained using present status of Figure 1.4 where, the maximum capacity of the code tree is $32R$. The used capacity is $9R$ ($2R$ due to $C_{2,3}$, $4R$ due to $C_{3,4}$, R due to $C_{1,23}$ and $2R$ due to $C_{2,16}$). The remaining vacant capacity of the code tree is $(32-9)R = 23R$. If a new call with rate $16R$ arrives, system does not support it because there is no vacant code with rate capacity 16*R*. This is called as *external fragmentation* and produce code blocking. The other cause of code blocking is *internal fragmentation.* This is due to the quantized rate handling capability of OVSF code tree. When a new user with rate kR, $k \neq 2^n$ arrives, the user requires a code with capacity 2^mR, where $k<2^m$ for minimum m. For example, we must assign 16*R* to a call request of 12*R*. The capacity 4*R* (16*R*-12*R*) which is 33% of the required bandwidth is wasted and which increases with increase in difference of requested call rate and assigned code. One of the options to reduce this wastage is to use multiple codes which may increase the complexity and cost of the BS and UE. The relationship between user data rate, spreading codes and the channel data rate for three users with rates *R*, 2*R* and 4*R* is shown in Figure 1.5.

Code blocking is the major drawback of OVSF-CDMA system and can be avoided using efficient assignment schemes, whereas, call blocking is the problem which is constituted by code blocking and blocking of call due to unavailability of vacant codes in code tree as explained in [19-20].

1.4 OVSF CODE ASSIGNMENT SCHEMES

In literature, a number of code assignment schemes are already proposed which aim at reducing code blocking probability. The performance of OVSF-CDMA can be improved using efficient code assignment and reassignment schemes, these schemes can be arranged in any of the following categories [21].

1.4.1 STATIC AND DYNAMIC CODE ASSIGNMENT SCHEMES

The static code assignment schemes rely on efficient placement of the code for the new call in such a way that the available vacant capacity of the tree is better utilized [22-24]. The dynamic code assignment schemes e.g. do code reassignments/replacements to reduce the code blocking [25-30]. This may increase the cost and complexity at the transceiver part.

The schemes available in literature can be broadly classified into two: the single code assignment and multi code assignment. The single code assignment scheme uses only one code from the OVSF code tree to handle new call [31]. The single code usage requires single rake combiner in the BS and UE. The multi-code assignment scheme uses multiple codes to handle quantized or non-quantized data rates [32]. This requires multiple rake combiners equal to the number of codes required to handle new call which leads to increased complexity.

1.4.2 SINGLE CODE SCHEMES

Crowded first assignment (CFA), leftmost code assignment (LCA), fixed set partitioning (FSP) and recursive fewer codes blocked (RFCB) scheme are few popular single code assignment schemes. In CFA, the code assignment is carried out to serve higher rate calls better in future. It has two categories, namely; crowded first code (based upon number of busy children) and crowded first capacity (based upon children used capacity) [22]. In LCA, code assignment is carried out from left side of the OVSF code tree [22],. In the FSP, the code tree is divided into a number of sub trees according to the number of input traffic classes and their distribution [27]. The RFCB scheme works on the top of CFA and the optimum code is the code which makes least number of higher rates codes blocked [33]. It resolves tie by recursively searching for best candidate. The adaptive code assignment (ADA) scheme divides the tree into small portions according to the arrival distribution reducing the number of codes searched for new calls [34]. The dynamic code assignment (DCA) scheme in handles new call using code reassignments [25]. This is the best single code scheme to reduce code blocking but the cost and complexity in reassignments is too high which limits its usage for low to medium traffic conditions. DCA is further improved to reduce complexity and to increase scalability by capacity partitioning and class partitioning methods [35]. The computationally efficient dynamic code assignment with call admission control (DCA-CAC) reduces complexity of traditional DCA in two different ways: (*a*) total resources are divided into number of mutually exclusive groups, with numbers of groups equal to number of call arrival classes; (*b*) by deliberate rejection of those calls which may produce large code blocking for future higher rate calls. The number of codes searched has direct impact on delay or speed of the code assignment and can have significant impact on delay prone services like; speech transmission and video conferencing etc [36-37]. The fast dynamic code assignment (FDCA) reduces the number of code reassignments without causing degradation in the spectral efficiency of system [38]. The rearrangeable (dynamic) approach requires current tree status to reassign some of the existing calls to accommodate new call which is based on a tree partitioning method, which requires additional information of traffic arrival rate (distribution of different data rate users) [23]. Two priority based rearrangeable code assignment schemes were proposed in [39] and [30], respectively. They handle both the real time calls (video streaming, voice calls etc.) and the non real time traffic (file transfer, e-mail). Real time calls are given higher priority. The Fewer Codes Blocked (FCB) scheme [40] selects that vacant code which results in least new parents blocked which were free previously.

1.4.3 MULTI CODE ASSIGNMENT SCHEMES

Multi code assignment schemes [10], the multi code multi rate assignment (MMCA) scheme takes into consideration mobile devices with different multi code transmission capabilities and different quality of service (QoS) parameters [41]. The multi code scheme with code sharing is suggested to reduce wastage capacity or code blocking [42]. It combines scattered capacity (children codes of assigned codes) of already assigned codes to reduce code blocking problem. The multi code scheme formulates the optimum number of codes/rakes required to handle new call [43]. The multi code scheme derives the optimal code with constraints of allocated code amount and maximal resource wastage ratio [44]. It gives superior performance using two and three codes in a multi code with a crowded-group-first strategy. The code utilization and blocking benefits are significant for a resource wastage ratio of 40%.

1.4.4 ALLOCATION BASED ON REMAINING TIME

In most multimedia and real-time applications, such as video on demand, video conferencing, downloading music files, etc., the service time of the requests can be obtained a priori. Thus, the remaining time of each code occupied in the code tree is known. If calls with similar remaining time are allocated to the same subtree, then the whole subtree will be available for higher data-rate requests after the calls are released. As a result, the system is able to support more users and reduce the code blocking probability. The impact of the remaining time factor on the performance of the OVSF code allocation and propose two time-based allocation schemes that take the remaining time of each call as the main factor to assign and reassign codes is investigated.The calls with similar remaining time are allocated to the same sub tree [45].

1.4.5 NOVSF CODES

The non blocking OVSF (NOVSF) codes minimize the code blocking to zero [46-49].

The code usage time is converted into multiple time slots and any one layer is sufficient to handle calls with variable rate requirements. The cost and complexity of the NOVSF codes is very high.

1.4.6 ROVSF CODES

The proposed rotated OVSF codes reduces the code assignment significantly compared to other methods [50]. It also reduces the code blocking significantly. They develop a new

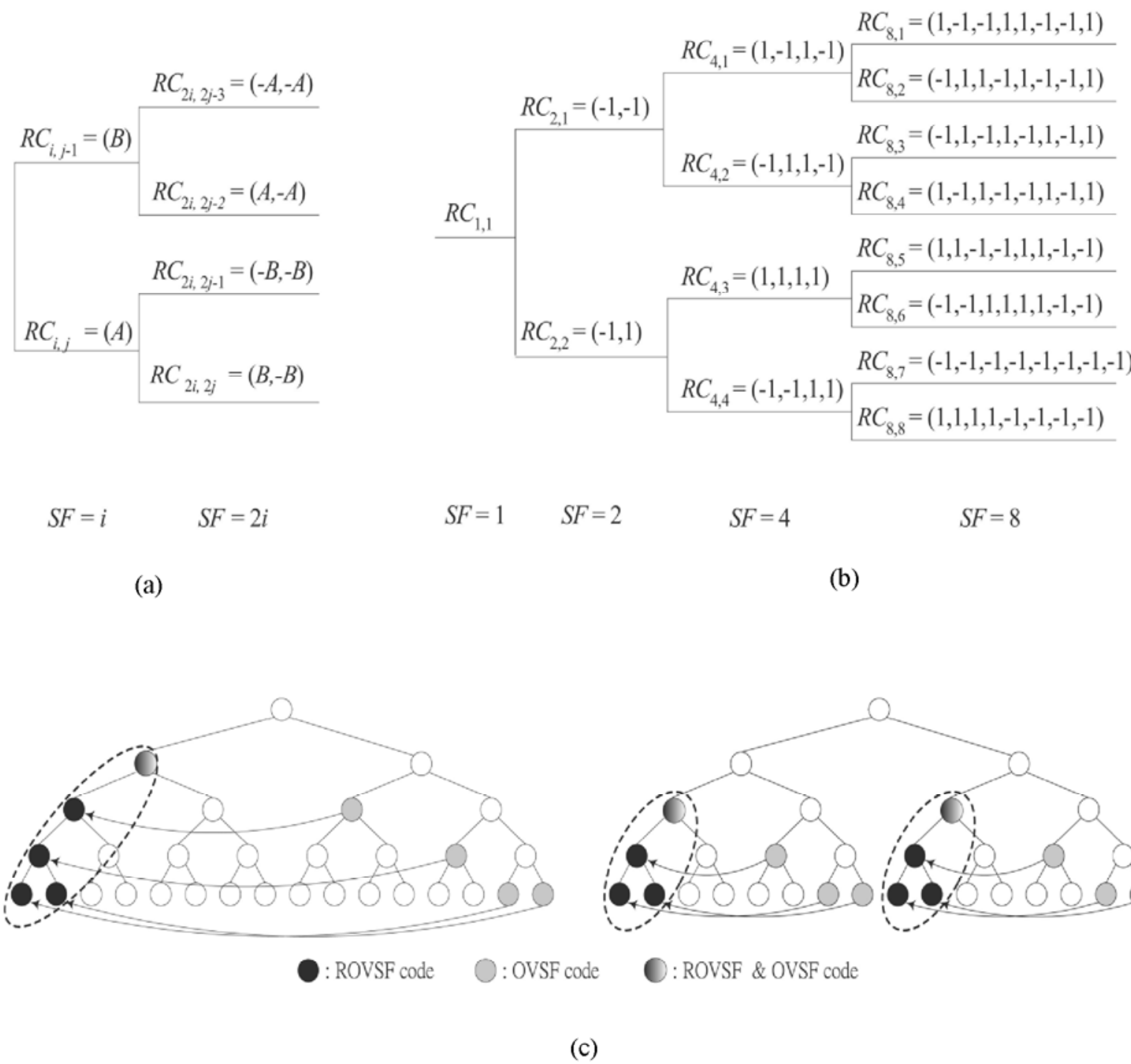


Figure 1.6: (a) ROVSF code generation, (b) ROVSF code generation for four layers, (c) Relationship between ROVSF and OVSF code tree [52].

channelization code tree structure, namely; the ROVSF (rotated-orthogonal variable spreading factor) code tree, as illustrated in Figure 1.6 [51]. The main work of this investigation is to exploit and justify the new properties of the ROVSF code tree. It shows that the ROVSF code tree offers the same code capability as the conventional OVSF code tree, ROVSF code tree has additionally code locality capability. With the code locality capability, a fast code assignment strategy is developed on the ROVSF code tree as compared to the existing code assignment schemes with lower search costs. A new code tree structure, namely an ROVSF (rotated-OVSF) code tree, the code capability of which is same as that of the traditional OVSF code tree is developed and single code placement and replacement schemes are developed for the ROVSF code tree to improve the code blocking probability and their code reassignment cost [50-52]. An OVSF code tree suffers from the *internal* and *external* fragmentation problem, in a similar way code assignment and reassignment schemes for ROVSF code tree suffers from these problems. A new multi code assignment and reassignment scheme for ROVSF code tree is proposed to efficiently reduce the *internal* and *external* fragmentation problems [53].

1.4.7 OTHER ASSIGNMENT AND REASSIGNMENT SCHEMES

More work is available in literature which focuses on different parameters of WCDMA networks using OVSF codes like; rakes utilization, throughput, quality of service (QoS), spatial diversity, spectral density etc.

A quality based assignment method in [54] proposes three assignment and reassignment strategies including fixed service data rates and considering a code limited system capacity. The paper carried out code assignment on the basis of available and guaranteed rate [55]. The OVSF codes support call rates that are powers of two *i.e quantized rate* and do not support many intermediate call rates. This reduces some flexibility in the allocation of code resources, and if *non-quantized rates* are assigned OVSF codes it may result in increased.

An efficient and fast channelization code assignment scheme (FEX scheme), by utilizing the code exchange and garbage collection technique, to reduce the blocking rate of the system [56]. The simulations demonstrated the effectiveness of this scheme by comparing to an "ideal system". The results also revealed the importance of the code exchange and multi-code property to greatly reduce the blocking rate of the system. Orthogonality constraint in the OVSF code tree may prevent efficient resource utilization. Code reassigning process can achieve an average of 20% gain of system throughput. Minimizing the code reassignments is essential in reducing the complexity of the wireless system. The design methodology and implementation techniques proposes an efficient algorithm MIN in optimizing the reassignment cost [57]. The simulation results of the paper indicates that the MIN algorithm can reduce an average 60% of the reassignment cost compared to that of the heuristic method and it is feasible for implementing in BS or Radio Network Controller (RNC). An adaptive grouping code assignment is given to provide a single channelization code for any possible rate of traffic, even though the required rate is not powers of two of the basic rate [58]. It is based on the dynamic programming algorithm, the adaptive grouping approach forms several calls into a group. Then it allocates a subtree to the group and adaptively shares the subtree codes for these calls in the concept of time-sharing of slots during a group cycle time. Therefore, the waste rate and code blocking are thus reduced obviously while using a single rake combiner. Since the delay problem may be occurred in such a time sharing approach, so two schemes of cycle interleaving methods to reduce delay are also investigated. Numerical results indicate that the proposed adaptive grouping approach reduces significantly the waste rate and thus increases the system utilization. The proposed cycle interleaving scheme reduces data delay significantly. Like in every multiservice network, different rate calls in 3G systems will perceive dissimilar system performance if no measures are taken and the channelization code tree is treated as a common pool of resources. A complete sharing, complete partitioning and hybrid partitioning strategies to manage the code tree, and studies the performance in terms of blocking probability per traffic rate class and utilization of codes are introduced [59]. It turns out that fair access to codes by different rate calls and code utilization are conflicting goals, and that hybrid schemes can provide a compromise between these two extremes. The dynamic bandwidth allocation (DBA) scheme is an interesting scheme for future broadband wireless networks, including the 3G and 4G WCDMA systems [29]. A code division generalized processor sharing fair scheduling DBA scheme that exploits the capability of the WCDMA physical layer, reduces the computational complexity in the link layer, and allows channel rates to be dynamically and fairly scheduled in response to the variation of traffic rates. Deterministic delay bounds for heterogeneous packet traffic are derived.

The scheme efficiently utilizes multiple RAKE combiners in user equipments [60-61]. This approach finds in constant time all feasible codewords for any particular request, trying to minimize both rate wastage and code fragments. When working together with an independent code reassignment scheme, this approach has the same code request denial rate as previous work but has lower code management overhead. If code reassignment is not used, our approach still has a bit of improvement on request denial rate. A guard code scheme has been introduced to favour ongoing calls over new calls in WCDMA systems employing OVSF codes as channelization codes [62]. The reservation of the codes takes place at the code management level and a new call(s) is accepted when available capacity of the systems is above threshold value defined for the scheme. The occupied codes state of the system and the transitions between the different states are modelled by a Markov chain. With the integration of an adaptive antenna array (AAA) at the BS, spatial diversity of mobile users can be utilized while assigning OVSF codes [63]. Users at distinct spatial locations probably fall in different downlink beams, and therefore, they can be assigned the same codes as long as the signal power received by the desired user is satisfactorily above the aggregate of interference signal powers. A new OVSF code allocation method is proposed with the AAA so called smart antenna based dynamic OVSF code allocation (SADCA) and robustness of SADCA over CCA and DCA schemes is also demonstrated [64]. The performance of DCA algorithm is evaluated and compared with other schemes [65]. Moreover, they propose a different algorithm for a more restricted setting [66]. It has been evaluated that DCA does not always return an optimal solution and that the problem is *NP*-hard. An exact time algorithm, and a polynomial time greedy algorithm is also given that achieves approximation ratio. A more practically relevant version is the online code assignment problem, where future requests are not known in advance. The flexibility index is defined to measure the capability of an assignable code set in supporting multirate traffic classes. Based on this index, two single-code assignment schemes, nonrearrangeable and rearrangeable compact assignments, are proposed [67]. Both schemes can offer maximal flexibility for the resulting code tree after each code assignment. Two scalable DCA schemes with call admission control for OVSF-CDMA systems are studied [68]. The proposed schemes generate an average data throughput of the system close to that of the optimal scheme while demanding much lower design and implementation complexity than the optimal scheme. The capacity partitioning scheme partitions the capacity of code tree into several mutually exclusive subsets of resource and assigns each subset of resource to a group of users in proportion to the corresponding traffic load. The class partitioning scheme partitions the set of service classes into mutually exclusive groups of classes and assigns a subset of the total resource to the corresponding group of classes. In either case, the call requests that belong to each subset of resource or each group of classes are served independently of others by the optimal DCA.

The proposed an adaptive simulated annealing genetic algorithm (ASAGA) in which population is adaptively constructed according to existing traffic density in the OVSF code-tree [69]. Also, the influences of the ASAGA parameters (selection, crossover and mutation techniques and cooling schedules) were examined on the dynamic OVSF code allocation problem. The proposed scheme outperforms conventional code assignment (CCA) and DCA schemes when compared for code blocking probability and spectral efficiency utilization, genetic algorithm (GA) and simulated annealing (SA) algorithms are also tested with ASAGA [70-71]. The simulation results show that the GA and SA provide reduced code blocking probability and improved spectral efficiency in the system when compared to the CCA and DCA schemes. Finally, the more system performance results, the more computational load. Ad hoc networks are wireless networks without fixed infrastructure. Each mobile node in the network may move arbitrarily, and therefore network topology changes frequently and unpredictably. Since, the OVSF codes are originally used as the channelization codes in the DS-CDMA system of International Mobile Telecommunication (IMT)-2000, previous schemes are centralized and cannot be applied directly to fully distributed systems such as ad hoc networks. Totally, six distributed code management schemes are proposed and three of them heavily exploit two techniques: code reassignment and code tree management [72]. Simulation results show that these schemes, with the help of the techniques, reduce the call blocking rate dramatically. Four different code assignment schemes presents namely; random scheme, left most and crowded first for the OVSF code tree and compared their performance [73]. It also proposes a new technique that selects the best method. The procedure in the selective one is based on using fast algorithm for low traffic loads until a defined load is demanded and selects the best algorithm for it's best performance in higher loads.

A call admission and code allocation schemes are proposed to provide service differentiation in the forward link of wideband code-division multiple-access (WCDMA) systems [74]. In particular, the paper proposes multiple leaf code reservation (MLCR) schemes, where different numbers of OVSF leaf codes (i.e., codes of the lowest layer of the OVSF code tree) are reserved to differentiate users with different bandwidth requirements. Leaf codes are only reserved for as long as the call admission process lasts. Once the decision of whether a new request is admitted or not has been made, a code dereservation procedure is carried out to increase flexibility in the code assignment phase. The performance of these MLCR strategies with/without code reassignments is then evaluated. Analysis shows that MLCR schemes are also useful in improving fair access among different traffic classes. In addition, perfect fair access among requests with different data rates can be achieved when code reassignments are jointly employed with the proposed OVSF-code reservation schemes. The drawbacks are pointed out which are inevitable due to multi code assignment, high complexity of handling multiple codes, and increasing the cost of using more rake combiners at both the base stations and mobile nodes [75]. Therefore, to improve performances on these parameters an adaptive grouping code assignment is proposed to provide a single channelization code for any possible rate of traffic, even though the required rate is not powers of two of the basic rate. Based on the dynamic programming algorithm, the adaptive grouping approach forms several calls into a group. Then it allocates a subtree to the group and adaptively shares the subtree codes for these calls in the concept of time-sharing of slots during a group cycle time. An adaptive grouping code assignment herein to provide a single channelization code for any possible rate of traffic, even though the required rate is not powers of two of the basic rate [76]. Based on the dynamic programming algorithm, the adaptive grouping approach forms several calls into a group. Then it allocates a subtree to the group and adaptively shares the subtree codes for these calls in the concept of time-sharing of slots during a group cycle time. The adaptive grouping approach reduces significantly the waste rate and thus increases the system utilization [76]. To achieve high data rate transmission in personal multimedia communications, 3G wideband CDMA systems (WCDMA) adopt the (OVSF) code tree for channelization codes management. However, it can waste system capacity. One good solution causes two drawbacks, long handoff delay and new call setup delay, which can degrade significantly the performance of 3G cellular systems. An adaptive efficient partition algorithm with the Markov decision process (MDP) analysis approach to provide fast handoff while reducing the waste rate significantly [77]. There are two primary motivations for the proposed MDP approach. First, based on the current state of the OVSF code tree, an adaptive partition algorithm is proposed to determine multiple codes for new connections. After determining these codes, the MDP analysis is adopted to assign least cost codes for them, which results in reducing a large number of reassignments. Second, to support fast handoff processing, the MDP approach assigns a single channelization code for each handoff connection. Thus, the dropping rate and grade of service (GOS) can be reduced. The approach yields several advantages, including the lowest GOS, the least waste rate, and the least number of reassignments. Finally, the optimal number of RAKE combiners is deduced.

An OVSF code management scheme called the *region division assignment* (RDA) scheme that divides total capacity in regions for a call request of particular rate [78]. A new call will check a vacant code in its region only. The performance of the OVSF code assignment schemes based on Universal Mobile Telecommunication Systems (UMTS)/IMT-2000, i.e., the CCA, the DCA), and RDA are compared [79]. It was shown that RDA gives better performance than DCA and CCA in terms of the packet dropping probability under heavy traffic conditions. An extensive performance comparisons among the three proposed OVSF code assignment schemes, that focuses on two performance measures, *i.e.*, the number of simultaneous voice conversations and the calculation time. Hybrid code allocation (HCA) is the scheme that has features of both DCA and RDA, it mitigates the code blocking more than RDA [80]. HCA, at first, performs RDA and then if code blocking happened, reassigns codes in the region which borrows a code and assigns a code to the request. A placement optimality is presented using a novel graph model, constrained independent dominating set problem (CIDP), the algorithm of it addresses both OVSF code placement and replacement issues at the same time and achieves placement optimality in linear time [82]. Chaotic sequence with low cross correlation are generated using phenomenon which is useful for spreading in CDMA. The bit-error-rate (BER) performance of logistic-map based chaos spreading codes with other spreading codes is compared. RAKE receiver and time-invariant multipath channels are used for the simulations. The chaos codes outperform m-sequences and Gold codes. Although the OVSF codes always outperform the chaos codes in AWGN environment, there are cases with time-invariant multipath channels, in which the BER of the chaos codes are better [83]. To improve the system performance while handling bursty internet traffic DSCH and HSDSCH channels are used in current mobile networks. OVSF codes use is limited to data transfer time of the session. Here, through the use of a multiplexer or scheduler, a queue and a buffer, a Traffic Model Scheduler (TMS) in layer 2 / RLC of HSDPA is introduced to reduce the OVSF code usage in HS-DSCH and which could save them [84]. The transform domain code division multiple access (TDCDMA)which synthesizes the excellence of the transform domain communication system (TDCS) and the code division multiple access (CDMA) technology. The TDCDMA has the characteristic of 1ow probability of intercept (LPI) and the capability of multiple access (MA), and it’s great potential to be applied in the cognitive radio (CR) contexts. The channelisation codes used in Universal Mobile Telecommunications System (UMTS) belong to the family of OVSF codes. They are organised according to a code tree structure. OVSF codes are primarily used to preserve orthogonality between different channels. The objective of code assignment is to enhance statistical multiplexing and spectral efficiency of WCDMA systems supporting variable user data rates. The performance of OVSF code assignment schemes, in terms of code blocking probability are compared in [85]. OVSF codes have been used for image spreading before transmission as it solves the synchronization problem and minimizes the effect of unwanted noise during transmission in Multiple Input Multiple Output Orthogonal Frequency Division Multiplexing (MIMOOFDM). Rayleigh fading channel has been used in the proposed scheme [86]. An Orthogonal Frequency and Code Division Multiplexing (OFCDM) technique. OFCDM can be envisaged as an OFDM system with two-dimensional (2D) spreading in time and frequency domains [87]. the performance of OFCDM and OFDM systems as an application in 4G systems. 2-D spreading approach is compared which is based on an adaptive load balancing with Markov decision process (which is denoted by ALM). The ALM approach consists of three phases [88-89]. In a system with Code Division Multiple Access (CDMA), the proposed technique exploits one code channel as the training sequence for channel estimation purposes. In a noisy environment, this technique performs better than the pilot based, thanks to the averaging effect of the noise impairment[90].

CHAPTER 2

VACANT CODES GROUPING

In this chapter, a fast OVSF single code assignment scheme is proposed and extended to multi codes which aims to reduce number of codes searched with optimal/suboptimal code blocking. The code assignment scheme aims to use those vacant codes whose parents are already blocked. This leads to occurrence of more vacant codes in groups, which ultimately leads to less code blocking for higher rate calls. The number of codes searched increases linearly in our scheme compared to most of other novel proposed single code methods like crowded first assignment, where it increases exponentially with increase in user rates. Also, the calculation of vacant codes at one layer will be sufficient to identify the vacant code adjacency for all the layers which reduces complexity. Simulation results are presented to verify the superiority of the scheme. The single code scheme described in section 2.1 reduces code blocking with additional benefit of reduction in number of codes searched prior to assignment of new call. For a particular higher

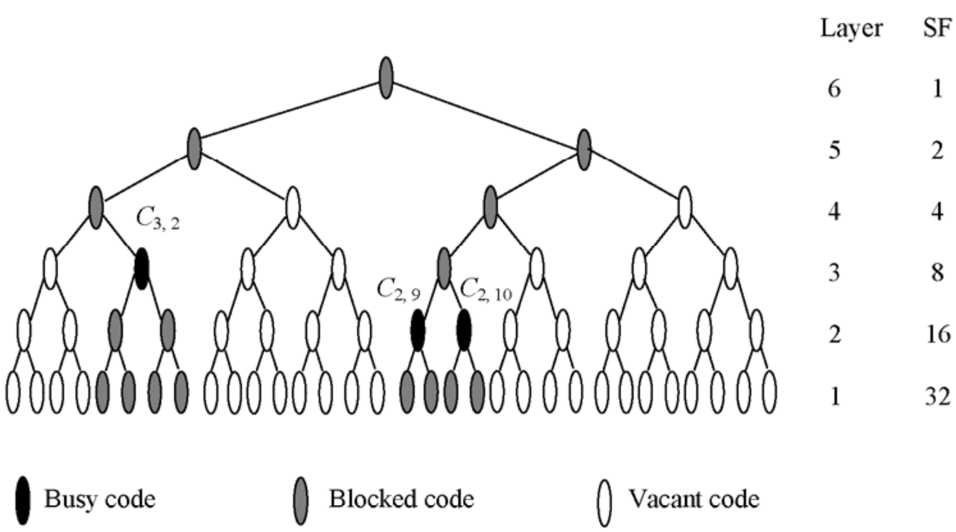


Figure 2.1: A six layer OVSF code tree with maximum capacity of $32R$

** *V. Balyan and D. S. Saini,* **"Vacant codes grouping and fast OVSF code assignment scheme for WCDMA networks"**, *Springer J. Of Telecommun Syst*, **DOI 10.1007/s11235-011-9469-5.**

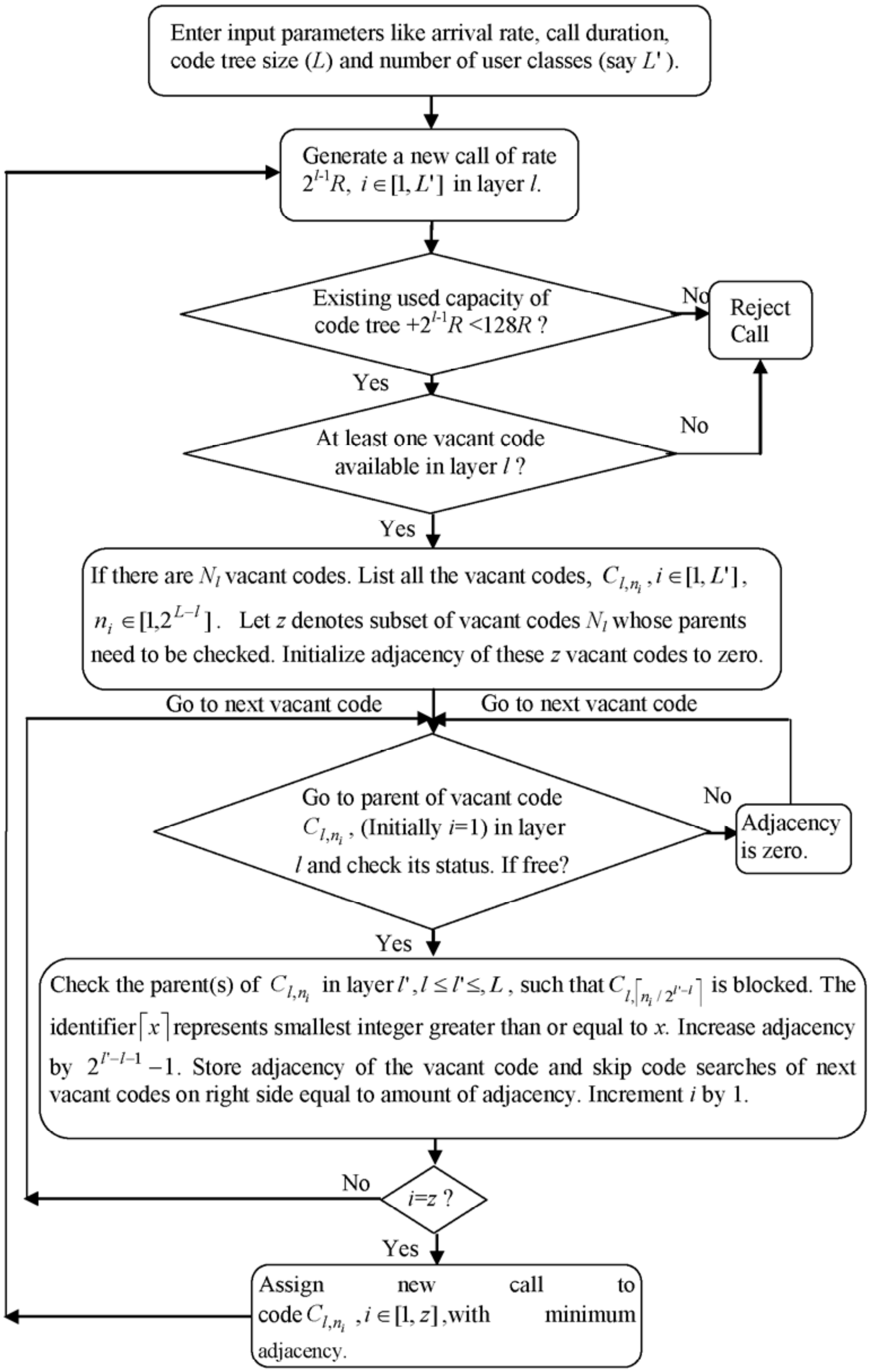


Figure 2.2: Flowchart of proposed AVC scheme

layer code, all the children codes are either vacant or occupied. This provides the occurrence of vacant codes in groups, which can be easily located and the number of code searches are reduced. This further reduces decision time (or call establishment time), complexity and cost. Although, the ADA scheme also provides the reduction in code searches, but the scheme discussed in this chapter provides the additional benefit of reduction in code blocking [34].

A multi code scheme provides lesser code blocking with higher call establishment delay. In this chapter, the proposed multi code assignment schemes in section 2.2 aims to reduce number of codes searched required before assignment of call and number of code searches required by scheme in section 2.1 are reduced. The scheme uses the status of higher layer codes searched in Orthogonal Variable Spreading Factor (OVSF) code tree to identify lower layer codes status. Two schemes variants of our scheme namely scattered and grouped multi code are used for comparison with other schemes. Simulation results are used to verify the superiority of proposed schemes.

The remainder of the chapter is organized as follows. Section 2.1 to 2.2 gives the description of proposed scheme along with flowchart and examples. Simulation results are given in section 2.3. The chapter is concluded in section 2.4.

2. 1 SINGLE CODE ASSIGNMENT: ADJACENT VACANT CODES (AVC)

A fast OVSF code assignment scheme is proposed which aims to reduce number of codes searched with optimal/suboptimal code blocking. The code assignment scheme aims to use those vacant codes whose parents are not free (already blocked). This leads to occurrence of vacant codes in groups (adjacent vacant codes) which ultimately leads to less code blocking for higher

Table 2.1: Relationship between layer 'l' and maximum adjacency

Layer (l)	1	2	…	l	…	L-1	L
Maximum adjacency	2^{L-1}-1	2^{L-2}-1	…	2^{L-l}-1	…	1	0

rate calls. The number of codes searched increases linearly in our scheme compared to most of other proposed (single code) most popular methods like DCA[25] and CFA[23] (where it increases exponentially with increase in user rates). Also, the calculation of vacant codes at one

Table 2.2: Relationship between adjacency of a vacant code in layer *l* and its parents in layer $l', l' > l$.

Layer Number	Vacant Codes	Adjacency
l	C_{l,n_l^i} to $C_{l,n_l^i+A_{l,n_l^i}}$	A_{l,n_l^i}
$l+1$	$C_{l+1,\lfloor n_l^i/2 \rfloor}$ to $C_{l+1,\lfloor n_l^i/2 \rfloor + \lfloor A_{l,n_l^i}/2 \rfloor}$	$\lfloor A_{l,n_l^i} \rfloor$
....	....	....
$l+\log_2[A_{l,n_l^i}+1]$	$C_{l+m,\lfloor n_l^i/2 \rfloor}$ to $C_{l+m,\lfloor n_l^i/2^m \rfloor + \lfloor A_{l,n_l^i}/2^m \rfloor}$	$\lfloor A_{l,n_l^i}/2^m \rfloor$

where $m = \log_2[A_{l,n_l^i}+1]$

layer will be sufficient to identify the vacant code adjacency for all the layers which reduces complexity and call establishment delay of future calls.

Consider an L layer CDMA based OVSF code tree. If a new call with rate $2^{l-1}R$, where $1 \le l \le L$ arrives, the proposed scheme (called as adjacent vacant code (AVC) scheme) lists all the vacant codes in layer *l*. Let layer *l* has $N_l, N_l \le 2^{L-l}$ vacant codes. The *i*th vacant code in the layer *l* is represented by $C_{l,n_l}^{\ i}$ and $1 \le i \le N_l$. For a code $C_{l,n_l}^{\ i}$, define *adjacency* ($A_{l,n_l}^{\ i}$) as the number of vacant codes adjacent to $C_{l,n_l}^{\ i}$ whose parents in all layers are same as the parents of $C_{l,n_l}^{\ i}$. The adjacency of all these vacant codes is the same and it is sufficient to check the status of one of these codes. The scheme checks the status of only left code appearing in code tree from this group and for remaining codes adjacency is same. For a vacant code $C_{l,n_l}^{\ i}$, whose left hand code is busy/blocked or its left hand vacant code has at least one same vacant parent as the parent of $C_{l,n_l}^{\ i}$, the AVC scheme checks the parents of code in layers *l*+1 to L till the blocked parent in layer l' (where $l' > l$ and $l' = \min[l+1], L \mid C_{l',\lfloor n_l^i/2^{l'-l} \rfloor}$ is blocked) appears. The identifier $\lfloor x \rfloor$ represents the smallest integer less than or equal to *x*. The adjacency of code $C_{l,n_l}^{\ i}$ becomes $(2^{(l'-l-1)}-1)$. The vacant codes C_{l,n_l^i} to $C_{l,n_l^i+A_{l,n_l^i}}$ need not to be checked as all these have same value of adjacency. The values of A_{l,n_l^i} for a code C_{l,n_l^i} is equal to $2^{l'-l-1}-1$ although

Table 2.3: Finding number of code searches in Figure 2.1 at the arrival of $2R$ rate call

Code # in layer 2	Whether status checked?	Number of skips	Number of codes searched
$C_{2,1}$	Y	1	3
$C_{2,2}$	N	NA	NA
$C_{2,3}$	Y	0	1
$C_{2,4}$	Y	0	1
$C_{2,5}$	Y	3	4
$C_{2,6}$	N	NA	NA
$C_{2,7}$	N	NA	NA
$C_{2,8}$	N	NA	NA
$C_{2,9}$	Y	0	1
$C_{2,10}$	Y	0	1
$C_{2,11}$	Y	1	3
$C_{2,12}$	N	NA	NA
$C_{2,13}$	Y	3	4
$C_{2,14}$	N	NA	NA
$C_{2,15}$	N	NA	NA
$C_{2,16}$	N	NA	NA

NA: Not applicable, Y: Yes, N: No, the code search is skipped as one of its adjacent vacant codes is already considered for status check.

there may be more than $2^{l'-l-1}-1$ consecutive vacant codes. The maximum possible value of A_{l,n_l^i} depends upon layer number l. The relationship between layer number l, vacant codes location and maximum adjacency is given in Table 2.1. The lesser is the value of adjacency A_{l,n_l^i} for a code C_{l,n_l^i}, lesser is the number of adjacent vacant codes for code C_{l,n_l^i}. For a specified adjacency A_{l,n_l^i} (say), the codes from layer l to (l+log$_2$[A_{l,n_l^i} +1]) are vacant and can be used for new calls as given in Table 2.2.

Hence, if a vacant code C_{l,n_l^i} has higher value of adjacency, the bigger portion of code tree around C_{l,n_l^i} is free, which leads to more higher rate parents free to handle new call(s). The scheme selects the vacant code whose adjacency is least. If a tie occurs for two or more vacant

Table 2.4: Total number of codes searched in layer 'l' at the arrival of $2^{l-1}R$ call

Vacant code	Adjacency (A_{l,n_l^k})	Number of skips	Next location to be searched	Number of codes searched	Total Number of code searched (say $NA_{l,n_l^{x_k}}$)
$C_{l,n_l^{x_1}}$ or $C_{l,n_l^{x_1}}$	$2^{l_{x_1}-1}-1$	$2^{l_{x_1}-1}-1$	$n_l^{x_1}+2^{l_{x_1}-1}+t_{x_1}$	$l_{x_1}+1$	$NA_{l,n_l^{x_1}}=n_l^{x_1}+l_{x_1}$
$C_{l,n_l^{x_2}}$	$2^{l_{x_2}-1}-1$	$2^{l_{x_2}-1}-1$	$n_l^{x_2}+2^{l_{x_2}-1}+t_{x_2}$	$l_{x_2}+1+t_{x_1}$	$NA_{l,n_l^{x_2}}=NA_{l,n_l^{x_1}}+l_{x_2}+1+t_{x_1}$
....	....	....	....	....	....
$C_{l,n_l^{x_z}}$	$2^{l_{x_z}-1}-1$	$2^{l_{x_z}-1}-1$	$n_l^{x_z}+2^{l_{x_z}-1}+t_{x_z}$	$l_{x_z}+1+t_{x_{z-1}}$	$NA_{l,n_l^{x_z}}=NA_{l,n_l^{x_{z-1}}}+l_{x_z}+1+t_{x_{z-1}}$

codes having same adjacency, the optimum code selection can be done using following two ways.

- Pick any code randomly.
- Use a second level criterion like elapsed time, crowded first capacity, crowded first space etc.

It assigns new call to the most crowded portion of the tree. One of the significant benefits of the AVC scheme is to reduce the vacant code searches at the arrival of new call which makes the code assignment faster than single code schemes like; crowded first code and crowded first space schemes. Adjacency leads to less code searches for a vacant code C_{l,n_l^i} in two ways.

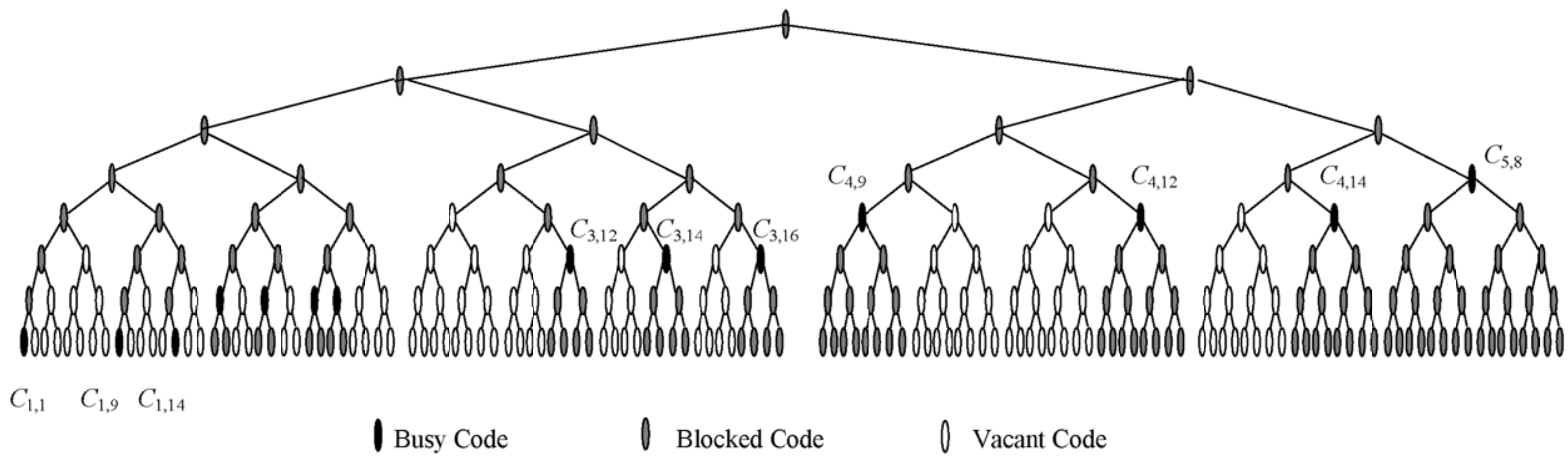


Figure 2. 3: Illustration of code searches in AVC and CFA schemes.

(a) Let adjacency is k for vacant code C_{l,n_l^i}, then for all codes between C_{l,n_l^i} and C_{l,n_l^i+k} has same adjacency k and only one code need to be searched. The next vacant code search will start from C_{l,n_l^i+k+1}. (b) For finding the adjacency of a code C_{l,n_l^i}, we go to its parent codes and check whether they are free or not. If we go up $l'', 1 \le l'' \le (L-l)$ steps above then number of code searches will be $l'' = (L-l)$, (one for each layer) and adjacency or codes vacant preceding C_{l,n_l^i} checked is $(2^{l''}-1)$. Checking of a parent code leads to the search of two children status on code tree and while doing so, code searches will increase linearly (instead of exponentially). For a new $2R$ call arrival, the number of codes searched in the AVC scheme is illustrated in Table 2.3 (corresponding to Figure 2.1). As discussed earlier, all the vacant codes need not be checked. For example, the 1st code $C_{2,1}$ is vacant and status of its parents is checked till the 1st blocked parent appears. The 1st blocked parent is $C_{4,1}$. The parent checking procedure stops and the adjacency of $C_{2,1}$ come out to be 1. Due to the adjacency value 1, code $C_{2,2}$ is not the candidate for status check, as it is already considered in parent check procedure of $C_{2,1}$. The next vacant code whose parent should be checked is $C_{2,5}$. The procedure is repeated for the complete tree as given in Table 2.3. In Table 2.3, column 2 represents whether status of code in layer 2 is checked or not, column 3 represents number of skips before next vacant code to be searched and column 4 represents number of codes searched for finding adjacency. The flow chart of algorithm is given in Figure 2.2. The general outline of the algorithm is given as follows.

1. Input parameters like arrival rate, call duration, code tree size (L) and number of user classes (say L') are entered.
2. Generate new call with rate $2^{l-1}R$.
3. For every new call, capacity check is performed. If capacity is available, go to step 4 otherwise the call is rejected.
4. (a) If at least one vacant code is available, list all the vacant codes (say N_l) and the subset z of these N_l vacant codes whose parent(s) status must be checked.

   (b) Go to the first vacant code from the left in code tree.

   (c) Find the adjacency and skip the number of vacant codes equal to adjacency.

   (d) If all z vacant codes (and their parents) are checked, find the code with the least value of adjacency and assign call to this optimum code. Otherwise, go to the next vacant code and return to step 4(c).

(e) Go to step 2.

Assuming layer l has $N_l, N_l \le 2^{L-l}$ vacant codes and each vacant code in layer l can be denoted by $C_{l,n_l^i}, 1 \le n_l^i \le 2^{L-l}$ and $1 \le i \le N_l$, the number of codes searched in the AVC scheme can be compared with the codes searched in CFA scheme [23].

Table 2.5: Comparison of number of code searches in AVC and CFA scheme for tree in Figure 2.2

Status with Adjacency	Codes #	Code searches in AVC scheme	Number of skips in AVC scheme	Code search in CFA scheme
Busy/Blocked Adjacency:0	$C_{2,1}$, $C_{2,5}$, $C_{2,7}$, $C_{2,9}$, $C_{2,11}$, $C_{2,13}$, $C_{2,14}$, $C_{2,23}$, $C_{2,24}$, $C_{2,27}$, $C_{2,28}$, $C_{2,31}$ to $C_{2,36}$, $C_{2,45}$ to $C_{2,48}$, $C_{2,53}$ to $C_{2,64}$	1 for each code = 31	0	1each= 31
Vacant Adjacency:0	$C_{2,2}$, $C_{2,6}$, $C_{2,8}$, $C_{2,10}$, $C_{2,12}$	2 each = 10	0	5each= 5x5=25
Vacant Adjacency:1	$C_{2,3}$, $C_{2,4}$,	3 (for $C_{2,3}$ only)	1	10each=20
	$C_{2,15}$, $C_{2,16}$	3 (for $C_{2,15}$ only)	1	10each=20
	$C_{2,21}$, $C_{2,22}$	3 (for $C_{2,21}$ only)	1	10each=20
	$C_{2,25}$, $C_{2,26}$	3 (for $C_{2,25}$ only)	1	10each=20
	$C_{2,29}$, $C_{2,30}$	3 (for $C_{2,29}$ only)	1	10each=20
Vacant Adjacency:3	$C_{2,17}$ to $C_{2,20}$	4 (for $C_{2,17}$ only)	3	19each =76
	$C_{2,37}$ to $C_{2,40}$	4 (for $C_{2,37}$ only)	3	19each =76
	$C_{2,41}$ to $C_{2,44}$	4 (for $C_{2,41}$ only)	3	19each =76
	$C_{2,49}$, $C_{2,52}$	4 (for $C_{2,49}$ only)	3	19each =76
	Total number of searches	72		460

2.1.1 CODE SEARCHES FOR AVC SCHEME

For a new call arrival with rate $2^{l-1}R$, let C_{l,n_l^i} be the first vacant code. The next vacant code whose status need to be checked is at the location $n_l^1 + 2^{l_1-1}$ (l_1 is the steps/layers above l where first blocked parent code is found for a vacant code C_{l,n_l^1}). Let the k^{th} vacant code whose status has to be checked (which may be different from C_{l,n_l^k}) is denoted by $C_{l,n_l^{x_k}}, x_k \in [1, N_l], x_k \geq k$. Consider n_l^1 and l_1 is $n_l^{x_1}$ and l_{x_1} respectively. If t_{x_1} denotes number of busy or blocked codes checked, next vacant code $C_{l,n_l^{x_2}}$ searched will be at location $n_l^{x_1} + 2^{l_{x_1}-1} + t_{x_1}$. This result can be generalized to find location of k^{th} vacant code C_{l,n_l^k} and is given by $n_l^{x_{k-1}} + 2^{l_{x_{k-1}}-1} + t_{x_{k-1}}$. The

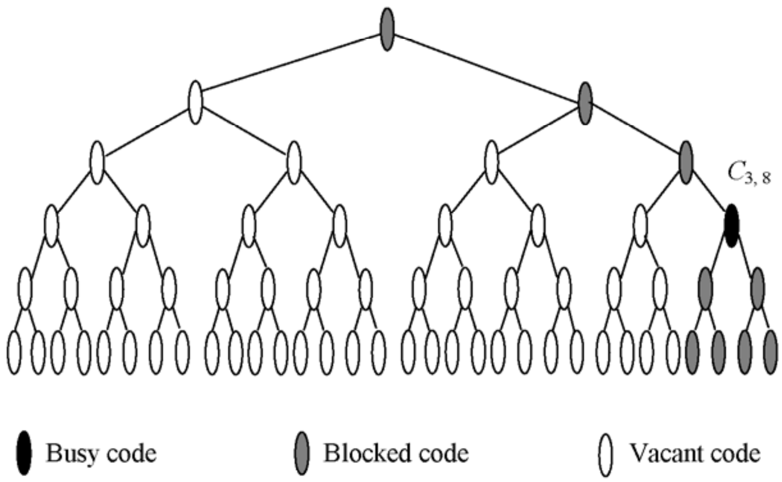


Figure 2.4: Illustration of parent and children code adjacency.

procedure is repeated z times, where z denotes subset of N_l vacant codes which need to be checked for adjacency and the last vacant code checked is $C_{l,n_l^{x_z}}$ as given in Table 2.4.

For the first code C_{l,n_l^1} or $C_{l,n_l^{x_1}}$ number of code searches is $(l_{x_1} + 1)$. For k^{th} vacant code C_{l,n_l^k}, number of code searches is $l_{x_k} + 1 + t_{x_{k-1}}$. The procedure is repeated till last code $C_{l,n_l^{x_z}}$. Number of code searched for a vacant code and total number of code searched till that vacant code is also shown in Table 2.4.

Total number of code searched in AVC scheme is given below:

$$N_{AVC} = n_l^{x_1} - 1 + l_{x_1} + 1 + l_{x_2} + 1 + t_{x_1} + l_{x_3} + 1 + t_{x_2} + \ldots\ldots + l_{x_z} + 1 + t_{x_{z-1}} + t_{x_z} \quad (2.1)$$

$$N_{AVC} = n_l^{x_1} - 1 + \sum_{i=1}^{z} (l_{x_2} + 1 + t_{x_1}) \quad (2.2)$$

$$N_{AVC} = n_l^{x_1} - 1 + z + \sum_{i=1}^{z} (l_{x_i} + t_{x_i}) \quad (2.3)$$

2.1.2 CODE SEARCHES FOR CFA SCHEME

Considering the definition of $C_{l,n_l^{x1}}$, the number of codes searched for $C_{l,n_l^{x1}}$ using CFA [23] scheme is $NC_{l,n_l^{x1}} = l_{x_1} + 2^{l+l_{x1}-1} - 1$. For the i^{th} vacant code, number of code searched is $NC_{l,n_l^{xi}} = l_{x_i} + 2^{l+l_{x_i}-1} - 1$. For all the N_l vacant codes, where $N_l \le 2^{L-l}$, the total number of codes searched becomes

$$NC_{CFA} = 2^{L-l} + l_{x_1} + (2^{l+l_{x_1}-1} - 1) + l_{x_2} + (2^{l+l_{x_2}-1} - 1)\dots + l_{x_{N_l}} + (2^{l+l_{x_{N_l}}-1} - 1) \tag{2.4}$$

$$NC_{CFA} = 2^{L-l} + \sum_{i=1}^{N_l}(l_{x_i} + 2^{l+l_{x_i}-1} - 1) \tag{2.5}$$

$$NC_{CFA} = 2^{L-l} - N_l + \sum_{i=1}^{N_l}(l_{x_i} + 2^{l+l_{x_i}-1}) \tag{2.6}$$

Our proposed scheme always needs less number of code searches as compare to CFA [23], the comparison is carried and is given in Appendix A. In comparison to CFA, where code search is increasing exponentially with every parent search, the AVC scheme leads to a linear code searches which decreases the number of code searched as we go in upper layers from l to L - l. For illustration, consider example in Figure 2.3. Consider a new call arrival with the rate $2R$. CFA scheme searches most crowded portion of the tree. For that it searches the parent codes and their children. It selects a vacant code whose parent/parents have maximum number of busy codes. The comparison of CFA and AVC scheme is illustrated in Table 2.5 in context with Figure 2.3. The adjacencies of all the vacant codes in layer 2 (corresponding to $2R$ arrival) is listed in column 4. The higher value of adjacency (number of skips) corresponds to large free area. The code with the minimum adjacency is the optimum code and hence one of the codes with adjacency 0 is used for $2R$ call.

Periodic code searches can be avoided to a great extent if the current code status remains valid for next k calls. Adjacencies of codes of layer l provide information of adjacency of children codes (layers below l) and parent codes (layers above l). Let the vacant code C_{l,n_l^i} has adjacency P_i. If the layer l - 1, codes $C_{l-1,2n_l^i}$ to $C_{l-1,2n_l^i+l}$ has adjacency

$$A_1 = 2P_i + 1 \tag{2.7}$$

Layer l - 2, codes to has adjacency, $A_2 = 2A_1 + 1$.

Similarly, for vacant codes in l_1 steps below the layer l *i.e* layer l-l_1 codes to has the adjacency

$$A_{l_1} = 2A_{l_1 - 1} + 1 \tag{2.8}$$

For a vacant code C_{l,n_l^i}, parent codes have adjacency given by

Layer $l + 1$, code $C_{l+1,\lfloor n_l^i/2 \rfloor}$ has adjacency

$$P_1 = \lfloor P_i / 2 \rfloor \tag{2.9}$$

Layer l+2, code $C_{l+1,\lfloor n_l^i/2^2 \rfloor}$ has adjacency $P_1 = \lfloor P_i / 2^2 \rfloor$. In a similar way, for a vacant code in l_2 steps above layer l *i.e* layer $l + l_2$ code has adjacency

$$P_1 = \lfloor P_i / 2^{l_2} \rfloor \tag{2.10}$$

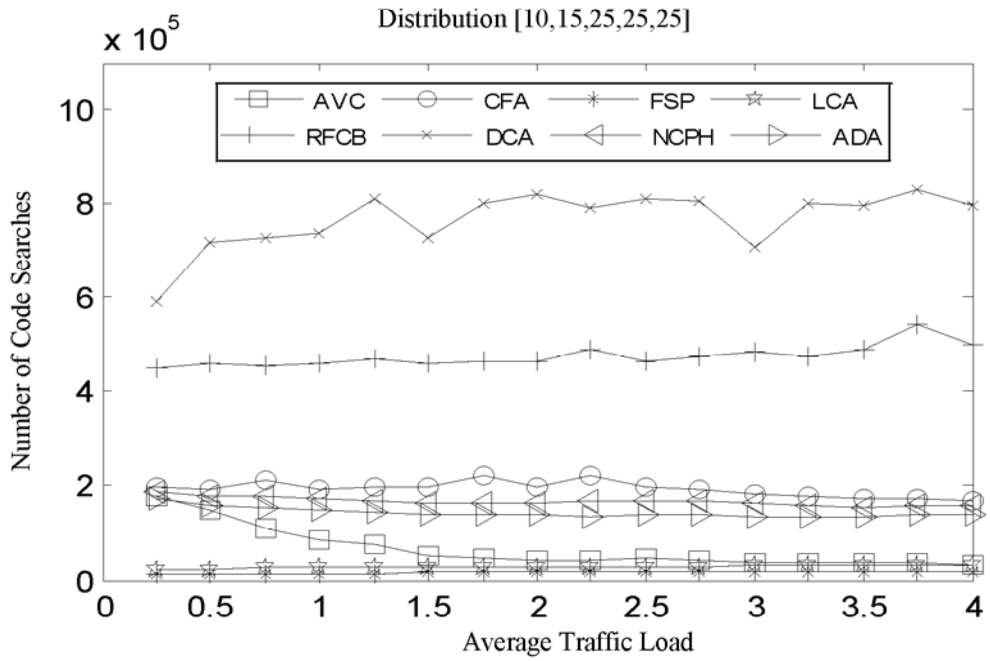
For L^{th} layer adjacency will be $P_L = \lfloor P_i / 2^{L-l} \rfloor$.

For illustration, consider the Figure 2.4. For a call of rate 4R, vacant codes are $C_{3,1}$ to $C_{3,7}$. Adjacency for these codes is 3 from $C_{3,1}$ to $C_{3,4}$,1 for $C_{3,5}$, $C_{3,6}$ and 0 for $C_{3,7}$ respectively. The new call will be assigned to $C_{3,7}$ as adjacency is minimum for $C_{3,7}$. As we know the adjacency of code in layer 3, apparently we know adjacency of their parent and children codes also. From Figure 2.4, it is clear that the adjacency of $C_{2,1}$ to $C_{2,8}$ has adjacency 7 *i.e.* (2×3+1). $C_{2,9}$ to $C_{2,12}$ has adjacency 3 *i.e* (2×1+1). In a similar way, we can find adjacency of parent codes of a vacant code for *e.g* $C_{2,1}$ has adjacency 7, its parent code $C_{3,1}$ has adjacency $\lfloor 7/2 \rfloor$=3.

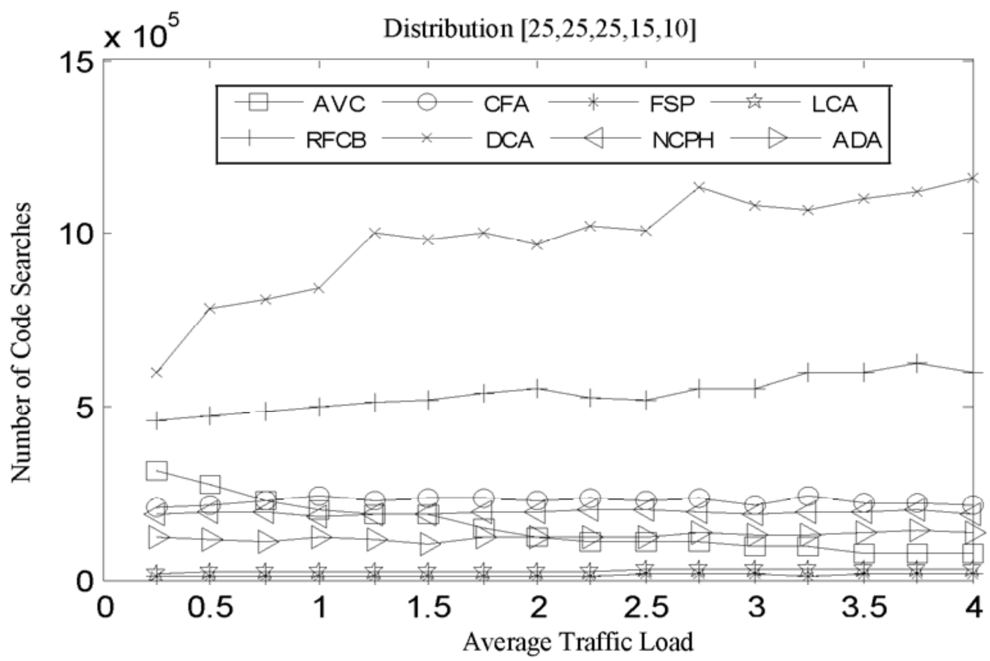
2.1.3 SIMULATION RESULTS

A. SIMULATION PARAMETERS

- Arrival rate 'λ' is Poisson distributed with mean value 0-4 calls/ units of time.
- Call duration '$1/\mu$' (is the service rate) is exponentially distributed with mean value of 3 units of time.
- The maximum capacity of the code tree is 128R (R is 7.5kbps).
- There are 5 classes of users with rates R, 2R, 4R, 8R, 16R.
- Simulation is done for 10000 users and result is average of 10 simulations.



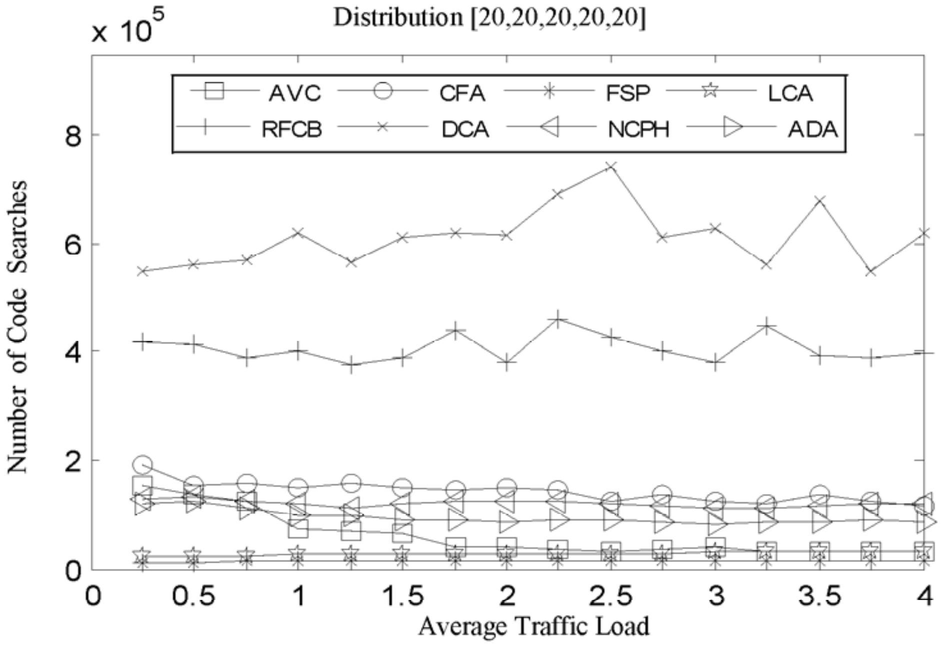
(a)



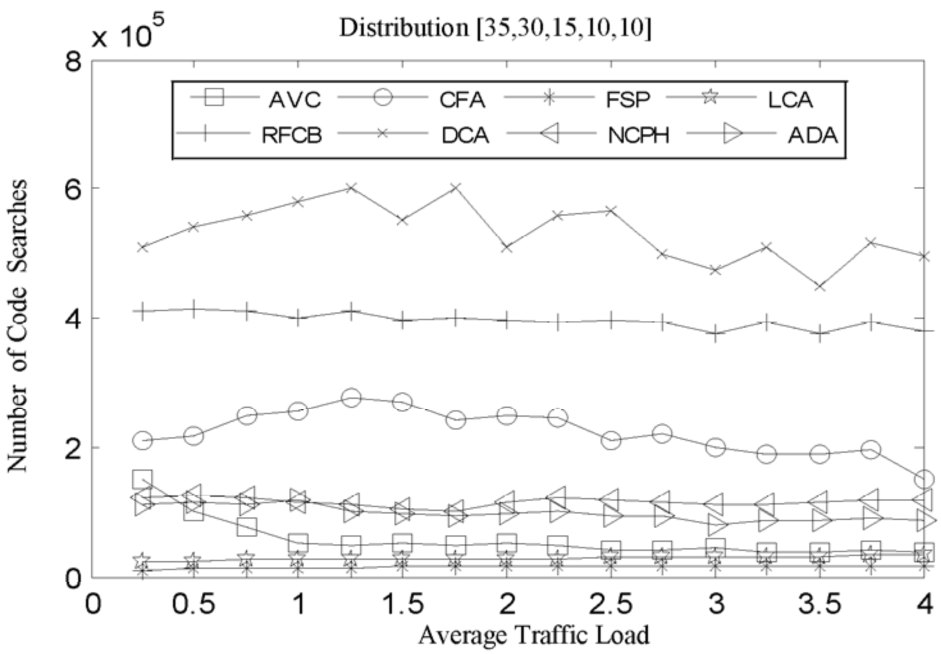
(b)

AVC: Adjacent vacant code, CFA [22]: Crowded first assignment, FSP [27]: Fixed set partitioning, LCA [22]: Left code assignment, RFCB [33]: Recursive fewer codes blocked, DCA[25]: Dynamic code assignment, NCPH [90]: Next Code Precedence High, ADA[34]: Adaptive code assignment.

Figure 2.5: Comparison of number of code searches for distribution (a) [10,15,25,25,25], (b) [25,25,25,15,10]



(a)

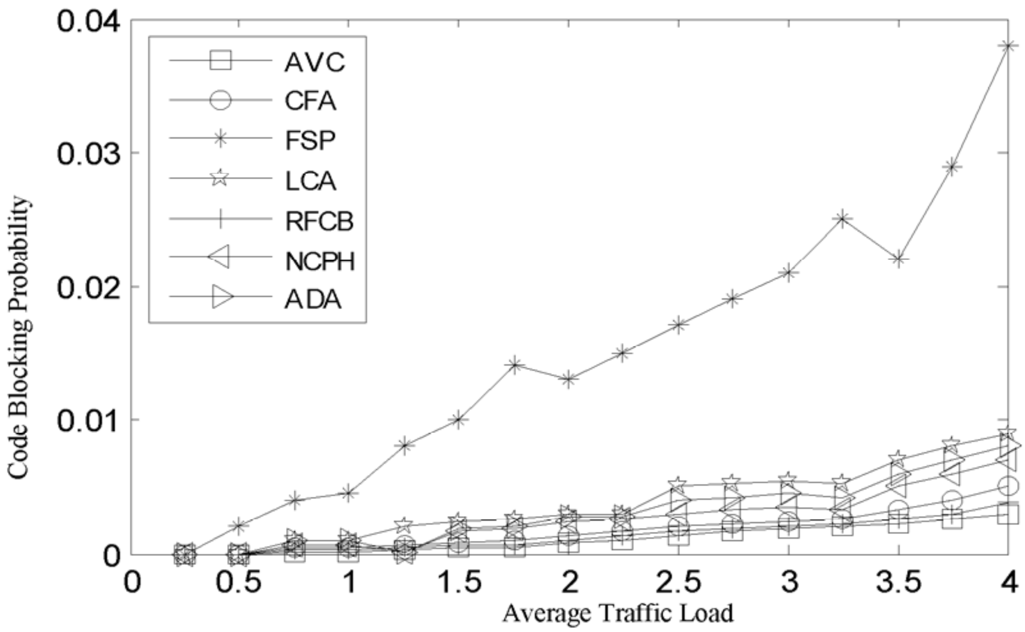


(b)

AVC: Adjacent vacant code, CFA [22]: Crowded first assignment, FSP [27]: Fixed set partitioning, LCA [22]: Left code assignment, RFCB [33]: Recursive fewer codes blocked, DCA [25]: Dynamic code assignment, NCPH [90]: Next Code Precedence High, ADA [34]: Adaptive code assignment.

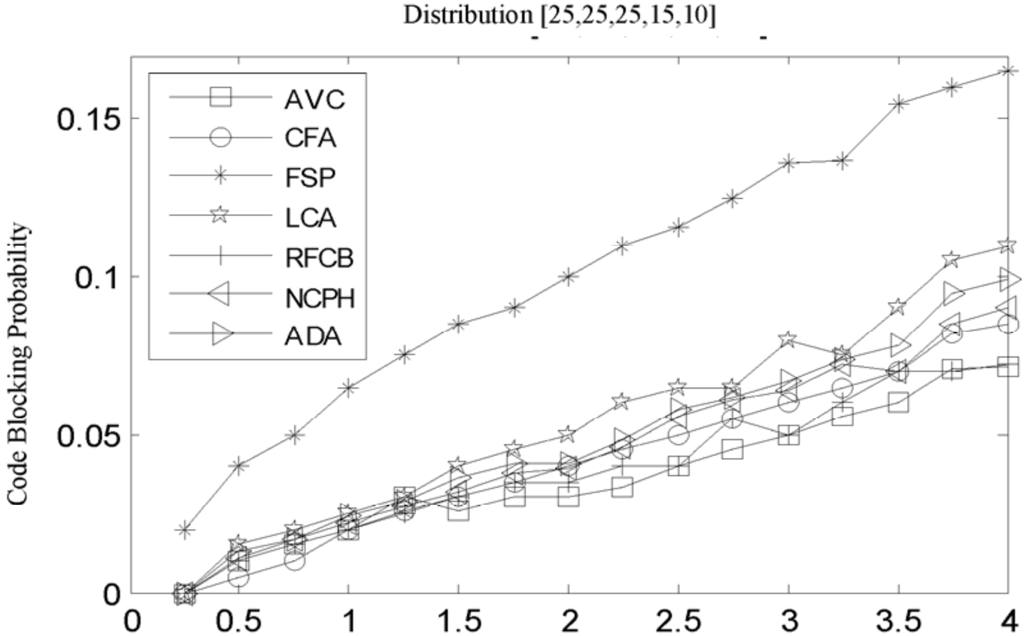
Figure 2.6: Comparison of number of code searches for distribution (a) [20,20,20,20,20], (b) [35,30,15,10,10].

Distribution [10,15,25,25,25]



(a)

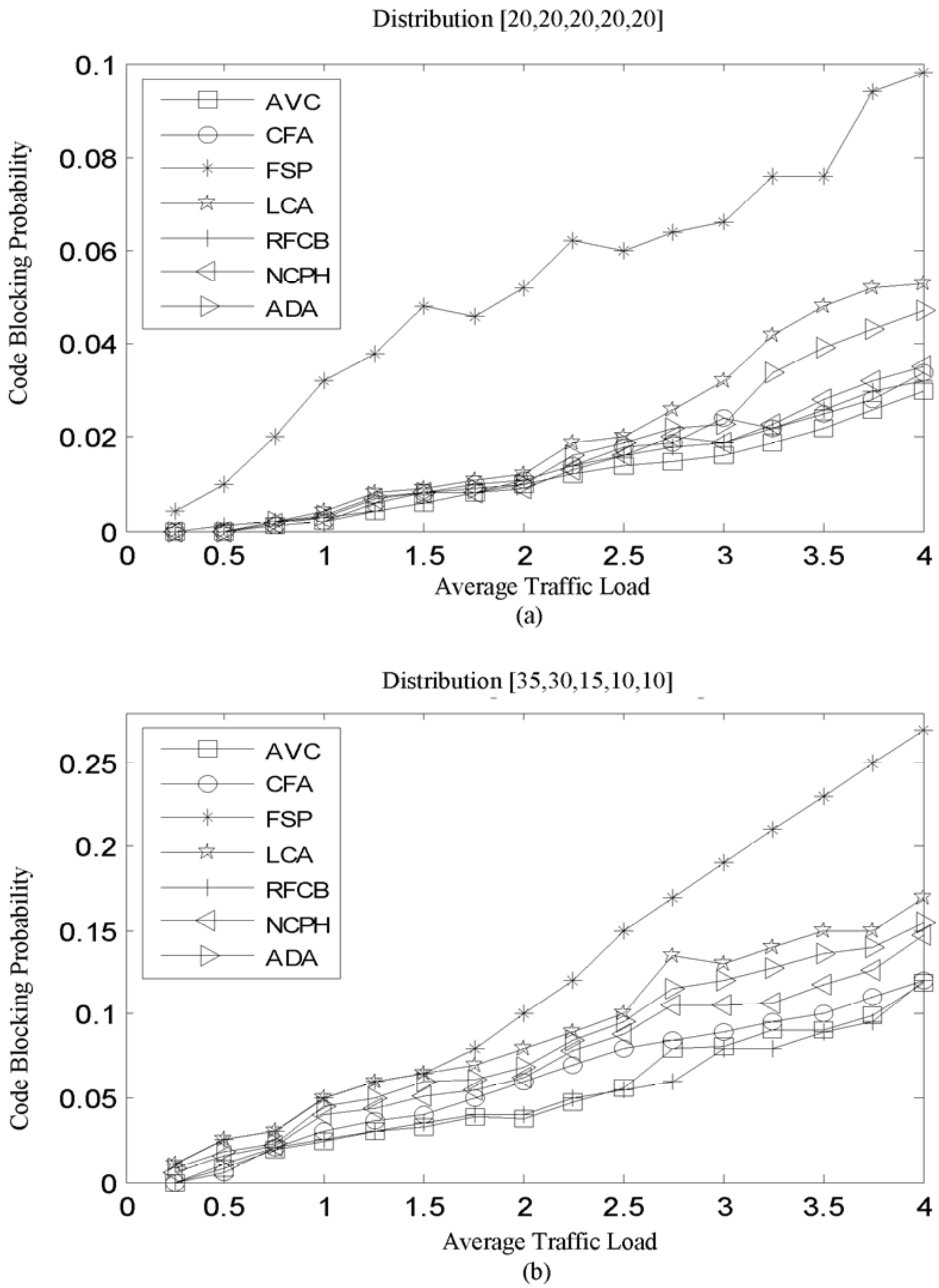
Distribution [25,25,25,15,10]



(b)

AVC: Adjacent vacant code, CFA [22]: Crowded first assignment, FSP [27]: Fixed set partitioning, LCA [22]: Left code assignment, RFCB [33]: Recursive fewer codes blocked, DCA [25]: Dynamic code assignment, NCPH [90]: Next Code Precedence High, ADA[34]: Adaptive code assignment.

Figure 2.7: Comparison of code blocking probability for distribution (a) [10,15,25,25,25], (b) [25,25,25,15,10]



AVC: Adjacent vacant code, CFA [22]: Crowded first assignment, FSP[27]: Fixed set partitioning, LCA [22]: Left code assignment, RFCB[33]: Recursive fewer codes blocked, DCA[25]: Dynamic code assignment, NCPH[90]: Next Code Precedence High, ADA[34]: Adaptive code assignment.

Figure 2.8: Comparison of Code Blocking Probability for distribution (a) [20,20,20,20,20], (b) [35,30,15,10,10]

An event driven simulation is considered where each code in a layer is a server. The number of we consider call duration of all the servers available for layer l is 2^{L-l} which is equal to number of codes in layer l. Let $\lambda_i, i \in [1,5]$ be the load of the i^{th} class users. Also, for 5 class system the average arrival rate and average traffic is $\lambda = \sum_{i=1}^{5} \lambda_i$ and $\rho = \sum_{i=1}^{5} \lambda_i$ respectively. In this simulation, we consider call duration of all the calls equal i.e. $1/\mu_i = 1/\mu$. Therefore, the average traffic load is $\rho = 1/\mu \times \sum_{i=1}^{5} \lambda_i = \lambda/\mu$. If we define $[P_1, P_2, P_3, P_4, P_5]$ as probability distribution matrix, where $P_i, i \in [1,5]$ is the capacity portion used by the i^{th} class users. As mentioned earlier, the code blocking is the major limiting factor in OVSF based networks. The average code blocking for a 5 class system is defined as

$$P_B = \sum_{i=1}^{5} (\lambda_i P_{B_i} / \lambda) \tag{2.11}$$

where P_B is the code blocking of i^{th} class and is given by

$$P_B = \frac{\rho_i^{G_i} / G_i!}{\sum_{n=1}^{G_i} \rho_i^n / n!} \tag{2.12}$$

where $\rho_i = \lambda_i / \mu_i$ is the traffic load for i^{th} class.

B. RESULTS

Four traffic distributions are used for performance results given by

- [10, 15, 25, 25, 25], high rates dominating scenario.
- [25, 25, 25, 15, 10], low rates dominating scenario I.
- [20, 20, 20, 20, 20], uniform distribution scenario.
- [35, 30, 15, 10, 10], low rates dominating scenario II.

Two performance metrics namely number of code searches and blocking probability of the proposed AVC scheme is compared with fixed set partitioning (FSP), crowded first assignment [23](CFA), left code assignment [23](LCA), recursive fewer code blocking [33](RFCB), adaptive code assignment [34](ADA), next code precedence high [90] (NCPH) and dynamic code algorithm [25] (DCA) schemes discussed earlier.

(i) CODE SEARCHES

The comparison of the code searches is shown in Figure 2.5 and Figure 2.6 for four arrival rate distributions. If N_i denotes the number of codes searches required for i^{th} class, the total number of codes searched is given by

$$N = \sum_{i=1}^{5} N_i \tag{2.13}$$

where N_i is the total number of codes searched for i^{th} class of user. Results shows that AVC scheme leads to less number of code searches compared to CFA, RFCB, ADA, NCPH and DCA schemes. Although, CFA and DCA are two most used single code assignment schemes, the AVC scheme requires less code searches compared to both of them which can be very beneficial especially for real time calls. On the other hand, the number of code searches required by AVC scheme are comparable to LCA and more than FSP schemes but the large code blocking probability in LCA and FSP schemes (as discussed in the next subsection) make these schemes less attractive.

(ii) CODE BLOCKING

The code blocking comparison is given in Figure 2.7 and Figure 2.8 for four distributions. The results show that the code blocking performance in the proposed AVC scheme is comparable to CFA, RFCB and is superior to FSP, LCA, ADA and NCPH schemes. The DCA scheme gives almost zero code blocking (hence not plotted) at the expense of higher reassignments which lead to increased complexity. Though the blocking probability is comparable to CFA and RFCB but reduction in code searches (as discussed in the previous subsection) makes the proposed scheme faster, less complex and cheaper.

From above results, it is clear that the AVC scheme gives better aggregate results in terms of number of code searches and blocking probability compared to other novel schemes. As mentioned earlier, the number of code searches is less due to linear increase in code checking with respect to layer number. In all other schemes, this increase is exponential. Therefore, the AVC scheme can be useful especially for real time calls.

2.2 MULTI CODE ASSIGNMENT

2.2.1 SINGLE CODE ASSIGNMENT: MODIFIED

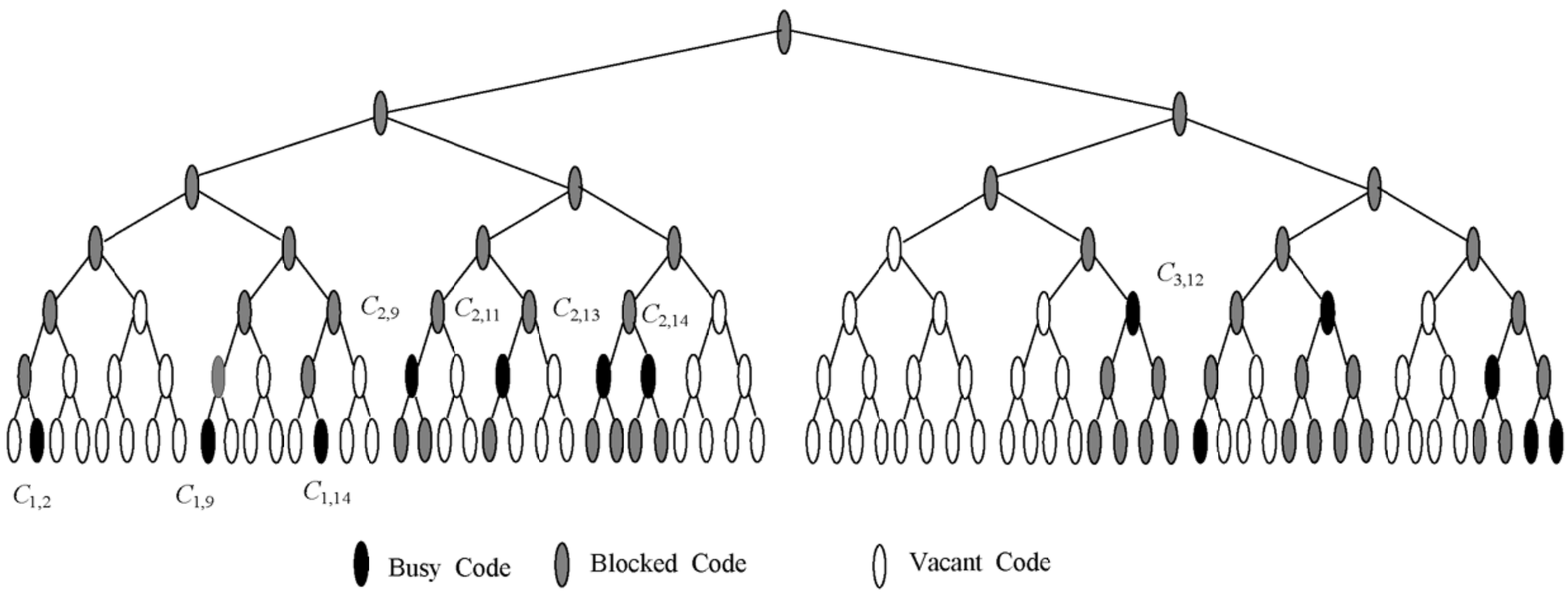


Figure 2.9 : Example illustrating single code and multi code adjacency

For a vacant code, $C_{l,n_l}, 1 \le n_l \le 2^{L-l}$, define vacant code adjacency as the number of vacant codes adjacent to C_{l,n_l} having same vacant ancestor code as for C_{l,n_l} in any layer between l+1 to L. Whenever the system finds a first vacant code it checks for the consecutive other vacant codes in its vicinity on right side of the vacant code. There are two ways to find adjacency when

Table 2.6. Vacant Codes in layers, their adjacency and code search required

Rate	Vacant Codes	Adjacency	Code Searches
R	$C_{1,1}, C_{1,10}, C_{1,13}, C_{1,50}$	0	28
	$C_{1,3\text{-}4}, C_{1,11\text{-}12}, C_{1,15\text{-}16}, C_{1,19\text{-}20}, C_{1,23\text{-}24}, C_{1,51\text{-}52}$	1	
	$C_{1,5\text{-}8}, C_{1,29\text{-}32}, C_{1,41\text{-}44}, C_{1,57\text{-}60}$	3	
	$C_{1,33\text{-}40}$	7	
$2R$	$C_{2,2}, C_{2,6}, C_{2,8}, C_{2,10}, C_{2,12}, C_{2,26}$	0	20
	$C_{2,3\text{-}4}, C_{2,15\text{-}16}, C_{2,21\text{-}22}, C_{2,29\text{-}30}$	1	
	$C_{2,17\text{-}20}$	3	
$4R$	$C_{3,1}, C_{3,8}, C_{3,11}, C_{3,15}$	0	14
	$C_{3,9\text{-}10}$	1	
$8R$	$C_{4,5}$	0	8

a new call with rate $2^{l-1}R$, $1 \le l \le L$ arrives:

1. After locating a vacant code, search adjacent vacant codes one by one, adjacency will be of the form $1,3,5,7\ldots 2^{i-1}-1$, $1 \le l \le L$.
2. Search the parent code(s) of vacant code. If they are vacant, then adjacency increases as :

$1,3,5,7 \quad \ldots 2^{i-1}-1$, where $1 \le l \le L$. If parent is blocked, then the adjacency is 0.

Both the methods lead to the same amount of code blocking. However, later lead to less number of code searches. The searching of a parent code requires search of its two children codes. This reduces number of code searches significantly. For lower requested rates, the second method is preferred. A new call of the rate $2^{l-1}R$ will be assigned to a vacant code which has minimum adjacency *i.e* to that portion of the code tree which has minimum number of vacant codes. For illustration, consider a 7 layer code tree shown in Figure 2.9. Table 2.6 lists all vacant codes in layers 1, 2, 3, 4, 5, along with their respective adjacency and the number of code searches required.

Our scheme further reduces the code searches at the lower layers using adjacency of higher layer vacant codes. Also, if a code is blocked or busy, its parent code is not searched as it will be vacant codes. Also, if a code is blocked or busy, its parent code is not searched as it will be blocked or busy too. If the adjacency of a layer *l* is A_l, the adjacency of layer (*l*-1)

$$A_{l-1} = 2 \times A_l + 1 \tag{2.14}$$

2.2.2 MULTICODE EXTENSION

As discussed earlier, multi code assignment can lead to significantly less code blocking as compared to single code assignment. The multi code extension of our scheme will initially search a vacant code for a call of the rate $2^{l-1}R$, $1 \le l \le L$ as in section 2.2.1 using single code scheme. If the system does not have a vacant code of rate $2^{l-1}R$, the code will be blocked using single code scheme. The multi code assignment is used which divides the incoming call rate. Let the system has *m* rakes. The multi code scheme can have various refinements.

A. MAXIMUM FRACTIONS SCHEME

Find $\max(i) \mid \sum_{j=1}^{i} 2^{l_j - 1} R = 2^{l-1} R, i \le m, 1 \le l_j \le l$. The maximum fraction algorithm converts the rate into maximum fraction, so that all the rakes are utilized for handling the call. For reach of the fractions, the code assignment scheme given in section 2.2.1 is used. Therefore, one call may use up to *m* fractions to handle incoming call. The scheme is complex and costly but leads to very small code/call blocking as the tree fragmentation is reduced due to lower rate fractions.

For illustration, consider Figure 2.9 for a call of the rate 16*R* and 4 rakes, the algorithm will divide it into two 8*R* rates and check total rakes used or not. So, it will further divide one of the 8*R* into two 4*R*s. Still numbers of fractions are not equal to rakes. The algorithm will further divide one of the 4*R* into two 2*R* rates and assign vacant codes to them.

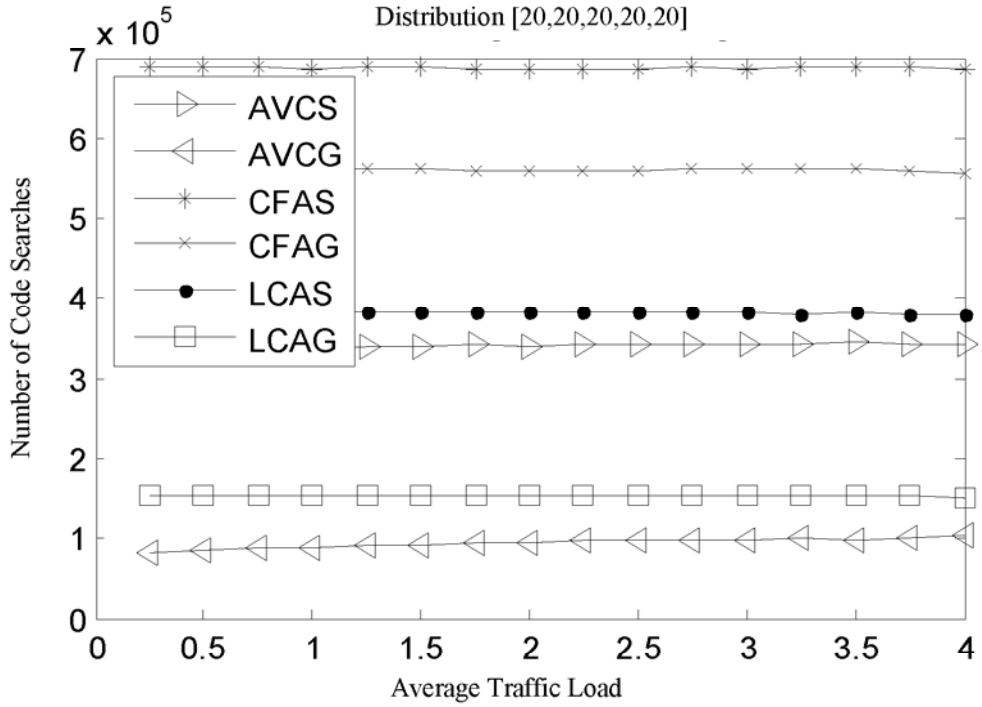
B. MINIMUM FRACTIONS SCHEME

Find $\min(i) \mid \sum_{j=1}^{i} 2^{l_j - 1} R = 2^{l-1} R, i \le m, 1 \le l_j \le l$. The scheme converts rate into least number of fractions. Each fraction is handled by different rake. The algorithm is simple and cost effective, but may lead to higher code blocking as the code fragmentation increases.

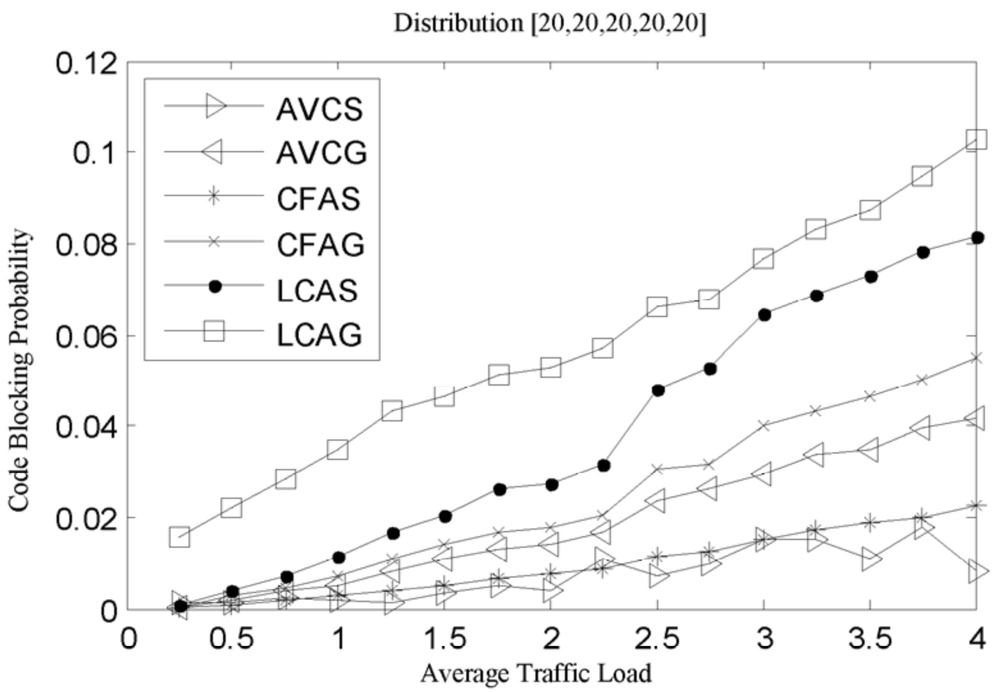
For illustration, consider Figure 2.9 for a call of the rate 16*R* and system of 4 rakes, the algorithm will divide it into two 8*R* rates and will search two vacant 8*R* codes. Code tree has only one vacant 8*R* code. One of the 8*R* will be further divided into two 4*R* fractions and will be assigned to vacant codes. When a call rate is divided into fractions, there are two ways of finding suitable code for rate fractions.

C. SCATTERED MULTICODE SCEHEME

An incoming call is divided into maximum fractions or minimum fractions as defined above. Let *r* be the number of rate fractions. Scattered multi code scheme will treat each rate fraction as a distinct call, and a vacant code for every rate fraction is assigned as discussed in single code scheme. This will lead to higher number of code searches with lesser code blocking. This scheme is best suited for maximum fraction division.



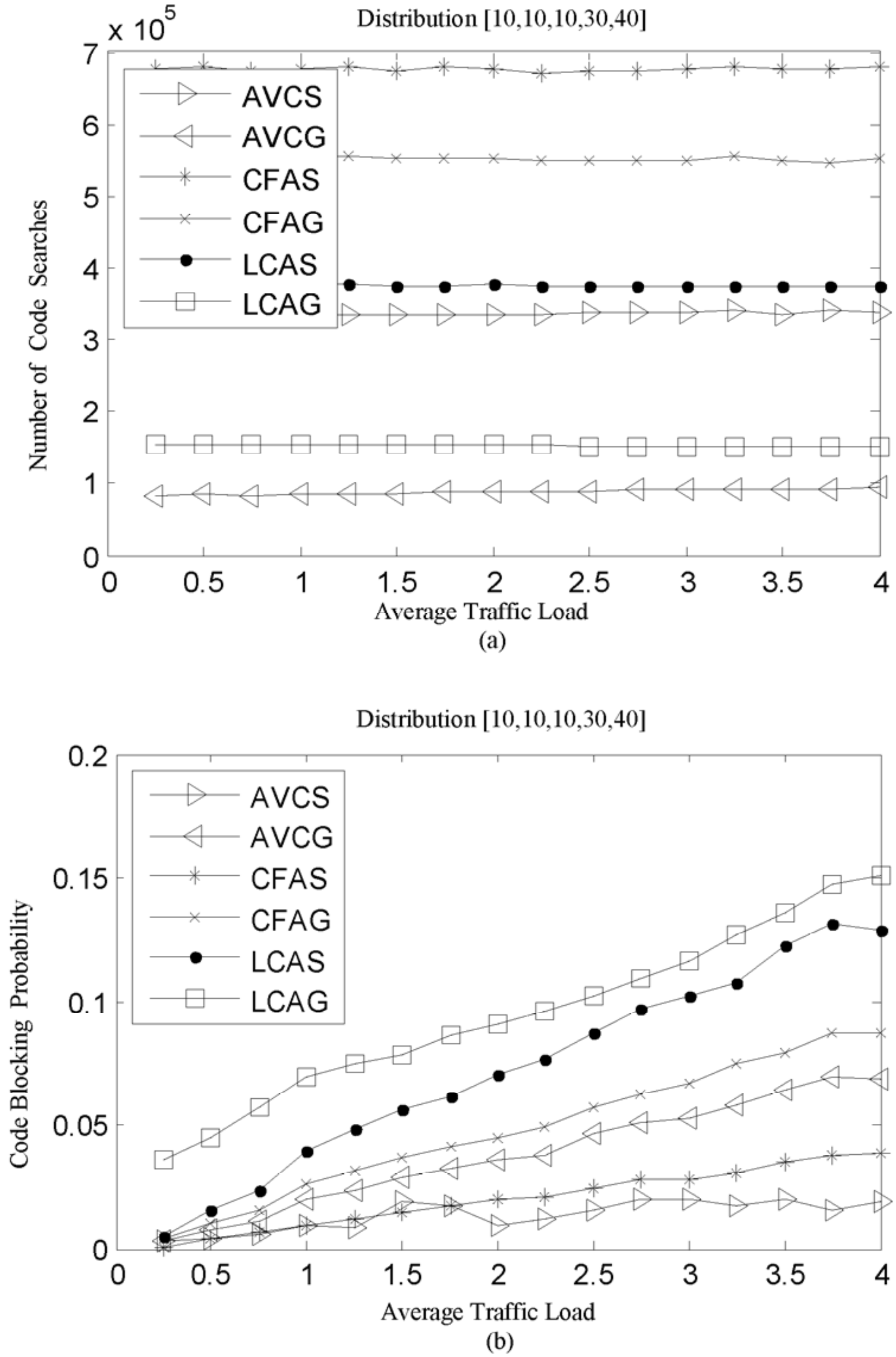
(a)



(b)

AVCS: Adjacent vacant code scattered, AVCG: Adjacent vacant code grouped, CFAS [22]: Crowded first assignment scattered, CFAG [22]: Crowded first assignment grouped, LCAS [22]: Left code assignment scattered, LCAG [22]: Left code assignment grouped.

Figure 2.10: Comparison of number of code searches (a) and code blocking probability (b) for distribution: [20,20,20,20,20].



AVCS: Adjacent vacant code scattered, AVCG: Adjacent vacant code grouped, CFAS [22]: Crowded first assignment scattered, CFAG [22]: Crowded first assignment grouped, LCAS [22]: Left code assignment scattered, LCAG [22]: Left code assignment grouped.

Figure 2.11: Comparison of number of code searches (a) and code blocking probability (b) for distribution: (a) [10,10,10,30,40].

D. GROUPED MULTI CODE SCHEME

An incoming call is divided into maximum fractions or minimum fractions as defined above. Let r be the number of rate fractions. Grouped multi code scheme will assign vacant code to all rate fractions as close as possible. This will lead to lesser code fragmentation when codes are released.

2.2.3 RESULTS

The probability distribution matrix $[p_1, p_2, p_3, p_4, p_5]$, where p_i, $i \in [1,5]$ is the code tree capacity portion used by the i^{th} class users. Three distribution scenarios are analyzed and are given by

- [20,20,20,20,20], uniform distribution.
- [10,10,10,30,40], high rates calls dominating.

The scattered and grouped multi code schemes are compared with CFA and LCA scattered and grouped multi code schemes. The various schemes are denoted as adjacency scattered (AVCS), adjacency grouped (AVCG), CFA scattered (CFAS), CFA grouped (CFAG), LCA scattered (LCAS), and LCA grouped (LCAG). The comparison of number of code searches and code blocking probability for various schemes is given in Figure 2.10 and Figure 2.11. The AVCG scheme requires least number of searches. Further, the comparison of blocking probability for various schemes is also shows clearly that the AVCS version provides least code blocking. Therefore, the adjacency scheme can be incorporated in WCDMA networks.

2.3 CONCLUSION

3G and beyond wireless networks are designed to handle multimedia rates better. Call processing delay and jitter are the significant QoS parameters for most of the real time calls. The chapter proposed a fast, single code, and multi code assignment scheme. The scheme can be even better when the system favors real time calls. The number of codes searched in the AVC scheme is significantly lesser than popular crowded first scheme. The online calculation of codes searched reduces the cost, complexity, and buffer size at the transmitter. The code blocking is also comparable or superior to existing single code schemes. For multi codes, code searches increases with number of rakes, which increases call establishment delay. The adjacency of higher layers can reduce searches at lower layer significantly in our scheme is used to locate optimum codes in multi code assignment. The AVC scheme can be combined with existing FSP to increase speed of call assignment process further.

CHAPTER 3
OPTIMUM VACANT CODE IDENTIFICATION

In this chapter, a single code scheme called as top down scheme is proposed to locate optimum vacant code in OVSF code tree for CDMA wireless networks. The selected code is optimum because the code usage produces least code blocking compared to existing schemes which do not have code reassignment facility. In addition, the codes searched in locating optimum code are significantly less than most popular crowded first scheme and many other schemes. The number of code searches is a significant factor for real time applications. The single code scheme is extended to multi codes. Four categories of multi code scheme are investigated. The first and second multi code schemes use minimum and maximum rakes for a fixed rate system. The third scheme called scattered multi code scheme divide the incoming call into rate fractions equal to number of rakes available in the system, and each rate fraction is handled like single code scheme. The rate fractions may be scattered in the code tree. The fourth multi code scheme, namely grouped multi code scheme allocates codes to all fractions as close as possible. This maximizes future

Table 3.1: Relationship between user rate, SF and transmission rate for WCDMA Downlink

User Rate(*kbps*)	**SF**	**Transmission Rate(*Mcps*)**
7.5	512	3.84
15	256	3.84
30	128	3.84
60	64	3.84
120	32	3.84
240	16	3.84
480	8	3.84
960	4	3.84

****D. S. Saini and V. Balyan, "Top Down Code Search to Locate An Optimum Code and Reduction in Code Blocking for CDMA Networks", *Wireless Personal Communication Springer, in press.***

higher rate vacant codes availability. The relationship between user rates, spreading factor and channel transmission rates is given in Table 3.1 for WCDMA systems.

The rest of the chapter is organized as follows. In section 3, sub section 3.1 and 3.2 discussed the proposed single code top down scheme along with its multi code and dynamic code assignment extensions. Moreover simulation environment and results are given in section 3.3 and finally the chapter is concluded in section 3.4.

3.1 TOP DOWN SCHEME

3.1.1 SING LE CODE TOP DOWN SCHEME

Consider an L layers OVSF code tree with a code C_{l,n_l}, $1 \le l \le L$, and $1 \le n_l \le 2^{L-l}$ representing a code in layer l with branch number n_l. The total number of codes in a layer l is 2^{l-1}. Further, for a new call of rate $2^{l-1}R$, the top down scheme identifies the optimum code in layer *l*, say

Table 3.2: Finding optimum $2R$ code in Figure 3.1

Layer Number	Candidate Codes	1st Candidate Code Parameters	2nd Candidate Code Parameters
6	$^*C_{6,1}$ and $C_{6,2}$	Code $C_{6,1}$ $I_{6,1}$= [1,0,3,5] $N_{6,1}$= 9 $P_{6,1}$= 19R	Code $C_{6,2}$ $I_{6,2}$= [1,1,0,0] $N_{6,2}$= 2 $P_{6,2}$= 12R
5	$C_{5,1}$ and $^*C_{5,2}$	Code $C_{5,1}$ $I_{5,1}$= [2,3] $N_{5,1}$= 5 $P_{5,1}$= 7R	Code $C_{5,2}$ $I_{5,2}$= [1,0,1,2] $N_{5,2}$= 4 $P_{5,2}$= 12R
4	$^*C_{4,3}$ and $C_{4,4}$	Code $C_{4,3}$ $I_{4,3}$= [1,2] $N_{4,3}$= 3 $P_{4,3}$= 4R	Code $C_{4,4}$ $I_{4,4}$= [] No vacant children code
3	$^*C_{3,5}$ and $C_{3,6}$	Code $C_{3,5}$ $I_{3,5}$= [1,0] $N_{4,3}$= 1 $P_{4,3}$= 2R	Code $C_{3,6}$ $I_{3,6}$= [2] $N_{3,6}$= 2 $P_{3,6}$= 2R
2	$^*C_{2,9}$	$C_{2,9}$ is optimum code and procedure stops	

* Optimum code, Path to optimum code: $C_{6,1}$ $C_{5,2} \rightarrow C_{4,3} \rightarrow C_{3,5} \rightarrow C_{2,9}$

$C_{l,n_{l_{opt}}}$ which leads to least code blocking. For a code C_{l,n_l}, define index $I_{l,n_l} = [I_{l,n_l}^L, I_{l,n_l}^{L-1} ... I_{l,n_l}^1]$, where the coefficient $I_{l,n_l}^i, 1 \le i \le L$ has following properties:

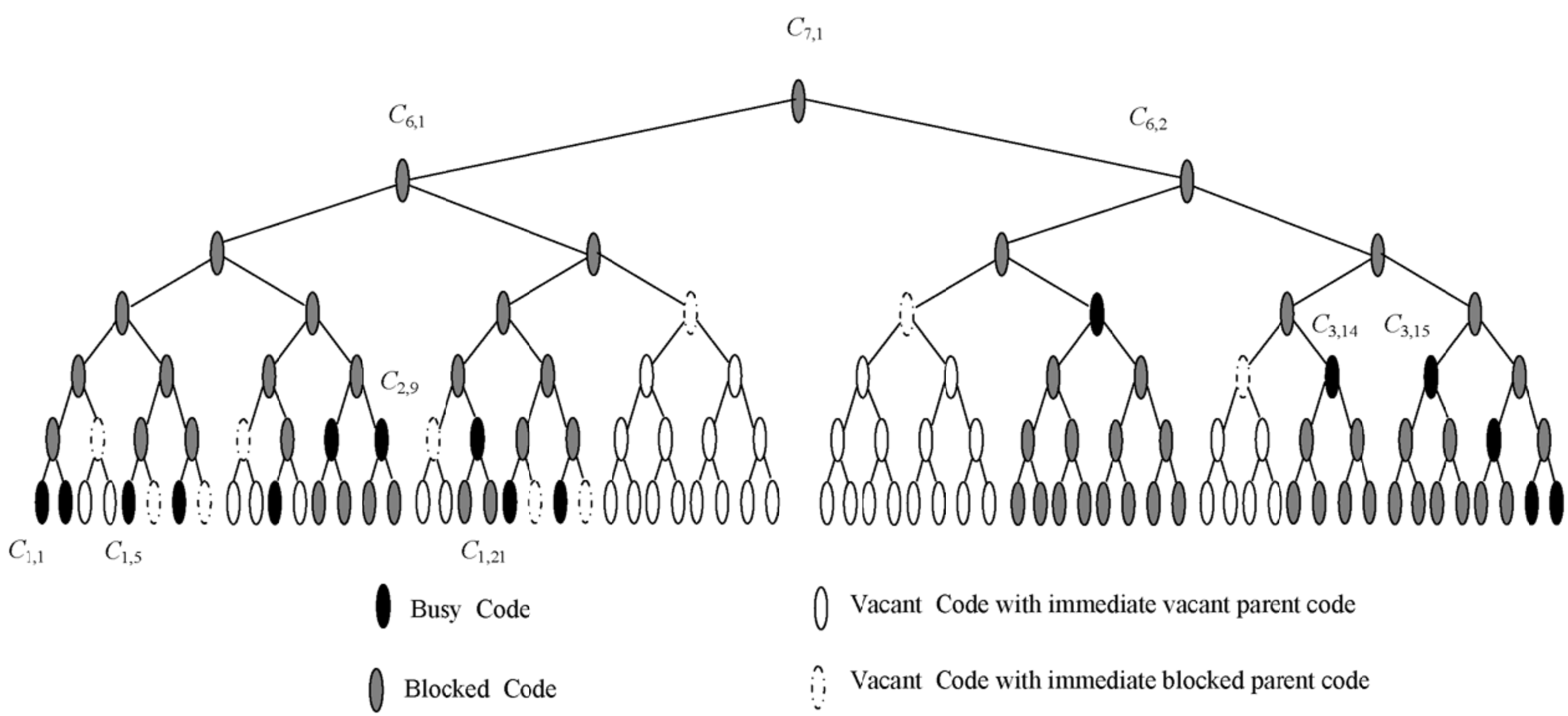


Figure 3.1: A seven layer OVSF code tree.

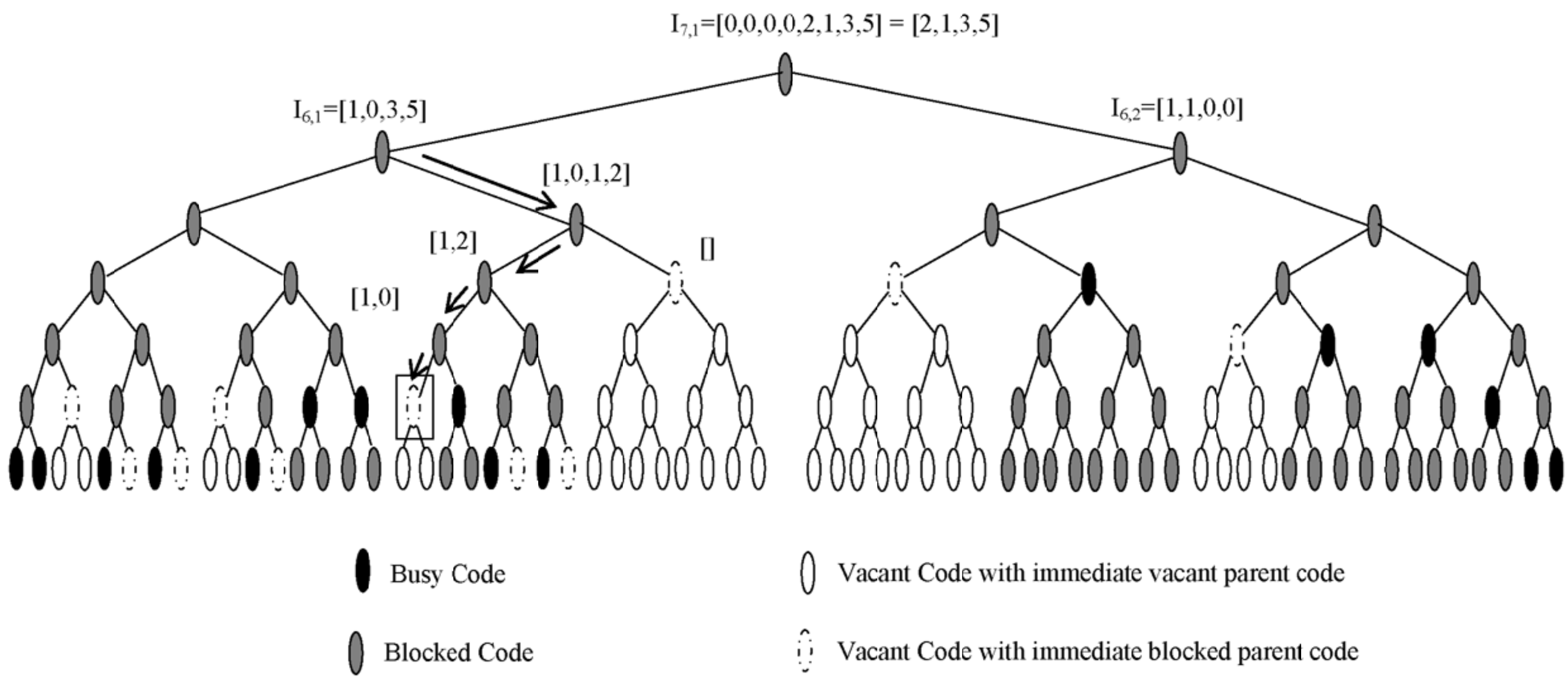


Figure 3.2: Illustration of single code top down scheme.

- $I_{l,n_l}^{l'}, 1 \le l' \le L-1$, represents total vacant children under code C_{l,n_l} in layer l' .
- $I_{l,n_l}^{l'}, l+1 \le l' \le L$, represents the status of parent of C_{l,n_l} in layer l', with $I_{l,n_l}^{l'} = 0$ representing blocked/busy parent.
- I_{l,n_l}^{l} is the status of the code C_{l,n_l} itself (0 for busy/blocked code and 1 if C_{l,n_l} is vacant with the condition that all the parents are blocked).

For a code C_{l,n_l} with index $I_{l,n_l} = [I_{l,n_l}^{L}, I_{l,n_l}^{L-1} ... I_{l,n_l}^{1}]$, the total vacant children under C_{l,n_l} are

$$N_{l,n_l} = \sum_{l'=1}^{l-1} I_{l,n_l}^{l'} \tag{3.1}$$

Also, the vacant capacity in the children of C_{l,n_l} is

$$P_{l,n_l} = \sum_{l'=1}^{l-1} I_{l,n_l}^{l'} \times 2^{l'-1} R \tag{3.2}$$

If for code C_{l,n_l}, the value of coefficient $I_{l,n_l}^{l'}, 1 \le l' \le L-1$ is p, due to p vacant children of C_{l,n_l} in layer l', the indices $I_{l,n_l}^{l'-1} ..... I_{l,n_l}^{1}$ do not include the children of these p vacant codes. In other words, a vacant code whose parent is also vacant is not counted in the definition of code indices. The code index I_{l,n_l} is updated periodically and in addition, at the arrival and completion of new call. For a code C_{l,n_l}, the sibling is C_{l,n_l+1}, if n_l is odd, and C_{l,n_l-1}, if n_l is even. At the arrival of call with rate $2^{l-1}R$, the optimum vacant code in layer l is required. If we represent sibling of $C_{l',n_{l'}}, l \le l' \le L-1$ by $C_{l',n_{l'_s}}$, the code $C_{l',n_{l'}}$ is in the path from root to optimum code if, (a) $I_{l',n_{l'}}^{l}$ is non zero and $\sum_{i=1}^{l'-1} I_{l',n_{l'}}^{i} \le \sum_{i=1}^{l'-1} I_{l',n_{l'_s}}^{i}$ and, or (b) $I_{l',n_{l'}}^{l}$ is zero but $I_{l',n_{l'}}^{l''}, l \le l'' \le l'$ is non zero and, $\sum_{i=1}^{l'-1} I_{l',n_{l'}}^{i} \le \sum_{i=1}^{l'-1} I_{l',n_{l'_s}}^{i}$. If both (a) and (b) fails and $I_{l',n_{l'}}^{l''}, l \le l'' \le l'$ is non zero, the code $C_{l',n_{l'_s}}$ will be in the path from root to optimum code. The procedure is repeated till layer l if for all identified optimum codes at least one vacant code is available in layer l. If on the other hand for an optimum code in layer l'', $l < l'' \le L-1$, there is no vacant child in layer l, the procedure stops at layer, and the optimum code in layer l is the leftmost child of the optimum code in layer

l''. Once all the codes $C_{l',n_{l'opt}}, l \le l' \le L-1$ are identified, the optimum code in layer *l* is assigned to the call. The code indices for codes $C_{l',n_{l'opt}}, l \le l' \le L-1$, are updates as follows,

(*i*) If $I^{l}_{l,n_{lopt}}$ is non zero, *i.e.*, the optimum vacant code exists in layer *l* (say $C_{l,n_{lopt}}$) with all its parents blocked. This vacant code is assigned to the new call and the coefficients $I^{l}_{l',n_{l'opt}}, l \le l' \le L-1$ are decremented by one, (*ii*) If $I^{l}_{l,n_{lopt}}$ is zero (there is no vacant code in layer *l* directly) but $I^{l}_{l',n_{l'opt}}, l+1 \le l' \le L-1$, is non zero, the optimum code in layer *l* is the leftmost child of $C_{l',n_{l'opt}}$ in layer *l*, *i.e.* $C_{l,2^{l'-l}\times(n_{l'}-1)+1}$ code (say $C_{l,n_{lopt}}$). The coefficients incremented are, (*i*) $I^{l'-1}_{l'',\lceil n_{l'opt}/2^{l''-l'}\rceil}$ to $I^{l}_{l'',\lceil n_{l'opt}/2^{l''-l'}\rceil}$, $l \le l'' \le L$ and (*iii*) $I^{l'-1}_{l',n_{l'opt}}$ to $I^{l}_{l',n_{l'opt}}$. Also, the coefficients decremented are $I^{l'}_{l'',\lceil n_{l'opt}/2^{l''-l'}\rceil}$, $l \le l'' \le L$.

The scheme requires significantly fewer code searches to identify optimum code compared to existing alternatives, and to be precise the code searches for code in layer *l* are $2(L-l)+1$. This is due to the fact that for a call $2^{l-1}R$, to find optimum code in layer *l*, two siblings need to be compared in layers *L*-1 to *l* along with the root code. The algorithm for single code top down scheme is described below.

1. Generate new call with rate $2^{l-1}R, 1 \le l \le L$

2. *If current used capacity in the tree* + $2^{l-1}R \le$ *total tree capacity*.

    2.1 Identify the optimum codes $C_{l',n_{l'opt}}, l \le l' \le L-1$.

    2.2 Assign code $C_{l,n_{lopt}}$ to the new call. Do code assignment and blocking. Update code indices $I^{l''}_{l',n_l}$, $C_{l',n_{l'opt}}, l \le l' \le L-1, l' \le l'' \le L-1$.

    2.3 Go to 1.

   Else

    2.1 Reject call.

    2.2 Go to step 1.

   End

The scheme is simple and is useful for medium to high traffic load conditions. For low traffic load conditions, traditional LCA [22] and FSP [27] schemes can be used to provide simplicity and fewer code searches. Considering the 7 layer code tree status as shown in Figure 3.1, if a new call with rate $2R$ arrives, the procedure of optimum path and code selection is explained in Figure 3.2 and Table 3.2. For simplicity, the code index is shown with only m coefficients when there is no vacant code in layer m+1 to L. For example, the code index of $C_{6,1}$ is $I_{6,1}$= [1,0,3,5] instead of [0,0,0,1,0,3,5]. The optimum path is shown by arrows in Figure 3.2 and the optimum code used for $2R$ call is $C_{2,9}$.

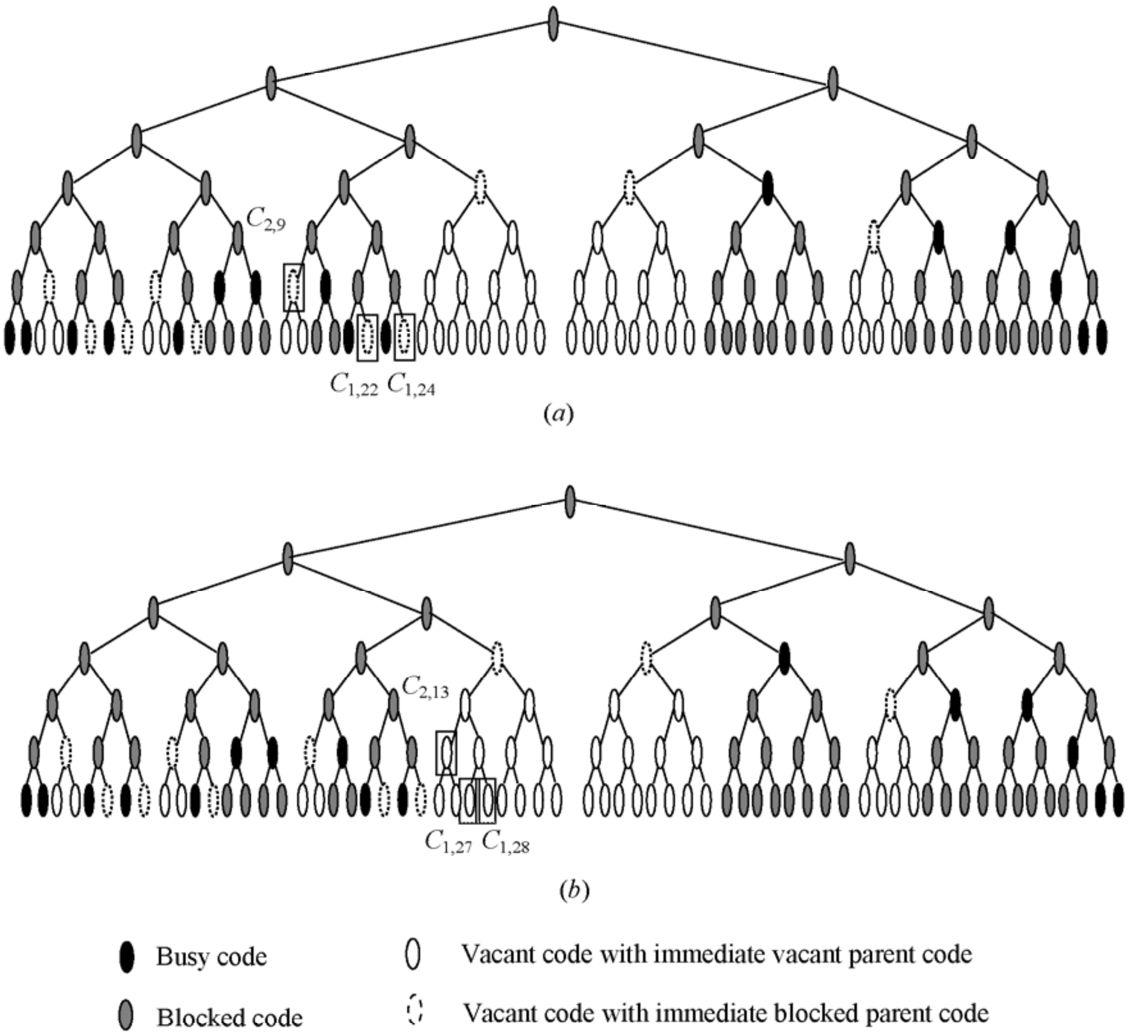


Figure 3.3: Illustration of multi code top down scheme (a) scattered approach (b) grouped approach.

3.1.2 MULTI CODE ENHANCEMENT

When the system has multiple rakes (say *m*), the single code scheme is extended to multi code scheme. The use of multiple rakes provide additional benefit of handling non quantized calls with rate *kR*, $k \neq 2^{l-1}$. In general, there are four variations in multi code schemes.

A. MINIMUM RAKES USAGE

In this scheme, for a new user/call with rate *kR* the minimum possible codes are used to handle new call. The procedure for finding these minimum rakes is given below.

Find $\max(l_1) \mid kR - 2^{l_1-1}R \geq 0, 1 \leq l_1 \leq L-1$ for which condition $kR - 2^{l_1-1}R \geq 0$ is true. If $kR - 2^{l_1-1}R = 0$, a single rake is sufficient to handle new incoming call, otherwise, the wastage capacity for a single rake system is defined as

$$W_1 = kR - 2^{l_1-1}R \tag{3.3}$$

For non zero W_1, find $\max(l_2), 1 \leq l_2 \leq L-2$, for which condition $kR - 2^{l_1-1}R - 2^{l_2-1}R \geq 0$ is true. The result can be extended to maximum of *m* steps for *m* rake system. In general, after $t \mid t < m$ steps, the wastage capacity is given by

$$W_t = kR - \sum_{i=1}^{t} 2^{l_i-1}R \tag{3.4}$$

For *m* = *L*, there is no wastage capacity but the severe complexity in the *BS* and *UE* requires *m* less than *L*.

B. MAXIMUM RAKES USAGE

If efficient resource allocation is the supreme requirement, then all the *m* rakes should be utilized to handle new call (if possible). In OVSF based systems, the resource allocation is efficient if code scattering is smaller, and the code scattering occurs due to the scattered lower rate calls in the code tree along with random arrival and departure time of calls. The scheme breaks the incoming rate into fractions in such a way that the future availability of high rate codes is highest. The incoming rate is divided into appropriate rate fractions so that all the rakes available are utilized. The scheme can have two categories.

LOWER RATE SPLITTING FIRST

The first part of the algorithm is to find minimum number of rake combiners required to handle new call according to equations (3.3) and (3.4). Let $t \mid t < m$ steps leads to zero wastage capacity, $i.e. W_t = kR - \sum_{i=1}^{t} 2^{l_i - 1} R$ and therefore the minimum rakes required are t. Let $2^{l_i^t - 1}, t \le m, i \le t$ represents i^{th} rate fraction of total t fractions. If initially rate fraction vector is represented by $\overline{R} = [2^{l_1^t - 1}, 2^{l_2^t - 1}, ... 2^{l_{j1}^t - 1}, .. 2^{l_t^t - 1}]$, the algorithm identifies the rate fraction j_1 so that $2^{l_i^t - 1}$ is smallest for $i=j_1$. The rate fraction $2^{l_i^t - 1}$ is broken into two rate sub fractions of amount $2^{l_{j1}^t - 1}/2$ and $2^{l_{j1}^t - 1}/2$. The new rate fraction vector becomes

$$\overline{R} = [2^{l_1^t - 1}, 2^{l_2^t - 1}, ..., 2^{l_{j1}^t - 1}/2, 2^{l_{j1}^t - 1}/2, .. 2^{l_t^t - 1}] = [2^{l_1^{t+1} - 1}, 2^{l_2^{t+1} - 1}, ..., 2^{l_{j1}^{t+1}}, 2^{l_{j1+1}^{t+1} - 1}/2, .. 2^{l_{t+1}^{t+1} - 1}] \quad (3.5)$$

The result in equation (3.5) can be extended to identify m_1 fractions and the optimum rate fraction vector becomes

$$\overline{R} = [2^{l_1^m - 1}, 2^{l_2^m - 1}, .., 2^{l_{m1}^m - 1}] \quad (3.6)$$

In equation(3.6), $m_1 = m$ if $k \ge m$, and $m_1 < m$ if $k < m$. All the coefficients of the rate fraction vector are handled by different rakes.

HIGHER RATE SPLITTING FIRST

Considering the definition of $\overline{R}$, the algorithm identify the rate fraction j_1 so that $2^{l_i^t - 1}, 1 \le i \le t$ is largest for $i=j_1$. The procedure can be repeated maximum j_1 times and the rate fraction vector is $\overline{R} = [2^{l_1^m - 1}, 2^{l_2^m - 1}, .., 2^{l_{m1}^m - 1}]$ as given in Equation (3.6). All the coefficients of the rate fraction vector are handled by different rakes.

The top down scheme can be integrated with multi code approach using minimum or maximum rakes as explained in next two subsequent sections.

C. SCATTERED MULTI CODE SCHEME

The incoming rate $2^{l-1}R$ is divided into maximum m_1, m_1=min(k,m) fractions $2^{l_i - 1}, 1 \le i \le m_1, 1 \le l_i \le l$ such that $\sum_{i=1}^{m_1} 2^{j_i} = 2^{l-1}$. The division is performed either by lower rate splitting first or higher rate splitting first depending upon the requirement. For each fraction 2^{j_i}, find the optimum code as discussed in section 3.1.

The algorithm of scattered multi code scheme is described below.

1. Enter number of rakes *m.*

2. Generate new call.

3. *If current used capacity in the tree*$+\ 2^{l-1}R \le$ *total tree capacity.*

3.1 Divide $2^{l-1}R$ into m_1 rate fractions, $2^{l_i-1}, 1 \le i \le m_1, 1 \le l_i \le l$.

For $1 \le i \le m_1$

3.1.1 Choose the optimum code from layer l_i+1.

3.1.2 Assign this code to rate fraction 2^{l_i}.

3.1.3 Update code indices.

End

3.2 Go to step 2.

Else

3.1 Reject call.

3.2 Go to step 1.

End

D. GROUPED MULTI CODE SCHEME

For new call *kR*, find m_1 fractions $\sum_{i=1}^{m_1} 2^{l_i} = k$. Arrange m_1 rate fractions in descending order. Find min(*l*) for which $2^l - k \ge 0$. The scheme works as follows.

If at least one vacant code is available in layer *l*, identify optimum code $C_{l,n_{l_{opt}}}$ in layer *l*. Assign leftmost child of $C_{l,n_{l_{opt}}}$ (*i.e.* code $C_{l_1,2n_{l_{opt}}-1}$) to rate fraction $2^{l_1-1}R$. If code $C_{l_1,2n_{l_{opt}}-1}$ is denoted by $C_{l,n_{l_{opt}}}$, the vacant code used for 2nd fraction is $C_{l_2,2^{l_1-l_2} n_{l_{1opt}}-2^{l_1-l_2}+1}$. The result can be generalized for i^{th} fraction and the code used is $C_{l_i,2^{l_{i-1}-l_i} n_{l_{1(i-1)opt}}-2^{l_{i-1}-l_i}+1}$.

If there is no vacant code in layer l and at least one vacant code is available in layer l-1, identify $C_{l_1,n_{l_{1opt}}}$ and assign this code to the first rate fraction $2^{l-1}R$. The vacant area is checked for the remaining rate $k_1 = (k - 2^{l_1 - 1})R$, *i.e.*, the vacant code is checked in layer $l', 2^{l'} - k_1 \geq 0$, for min(l'). Again starting from layer l', the m_1-1 rate fractions except 1st rate fractions are assigned codes adjacent to each other. The algorithm can be repeated for third fraction if there is no vacant code in layer l-1. The code selection may not be optimum but the future multi code use becomes simple.

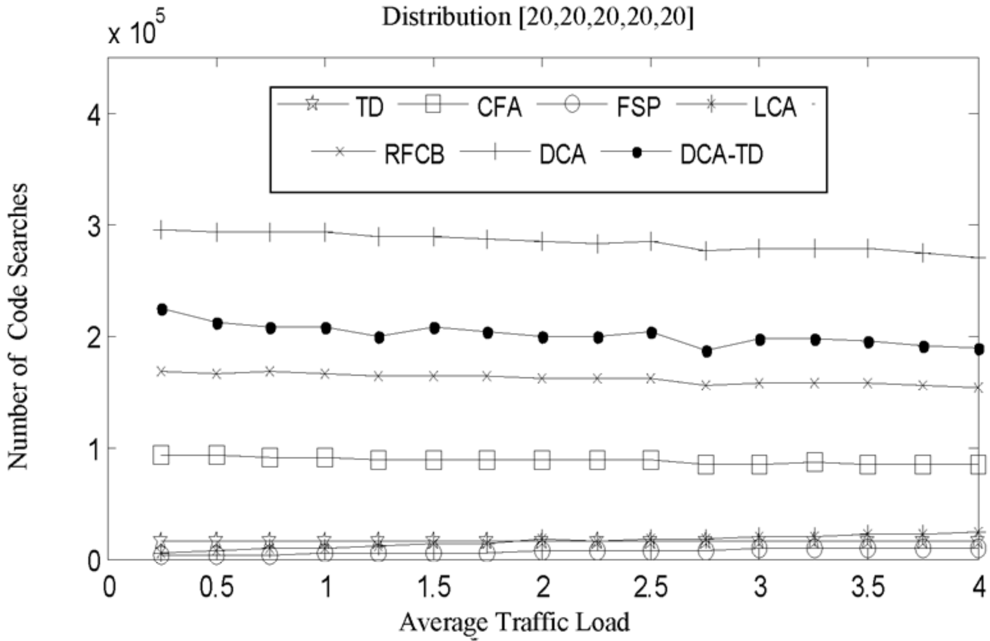
For illustration of grouped multi code scheme, let a 4R call arrives with code tree status in Figure 3.1 and the system is equipped with three rakes. Considering maximum rakes use, the three rate fractions are 2R, R, R. The optimum codes using scattered approach are $C_{2,9}$ (for rate 2R), $C_{1,22}$ and $C_{1,24}$ (for rate R) as shown in Figure 3.3(a). If grouped multi code scheme is used, the optimum codes are $C_{2,13}$ (for rate 2R), $C_{1,27}$ and $C_{1,28}$ (for rate R) as shown in Figure 3.3(b). The scheme requires fewer number of codes searches as compared to most popular CFA[22] scheme as discussed in Appendix B.

3.1.3 DYNAMIC CODE ASSIGNMENT ENHANCEMENT

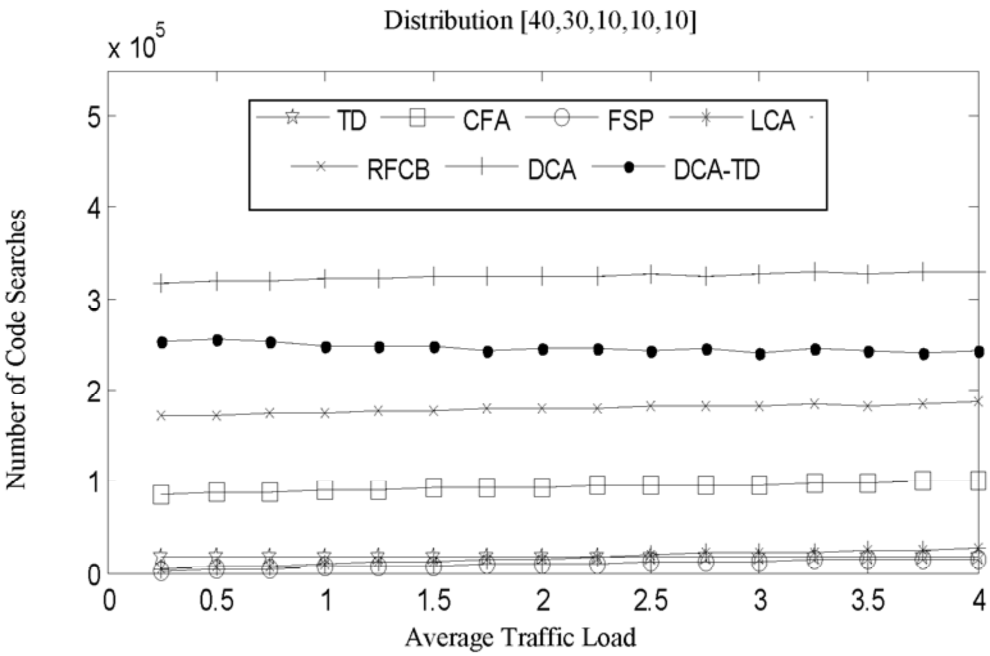
For a new call with rate $2^{l-1}R$, the reassignments are required if $I^{l'}_{l,n_l} = 0, l' \geq l$ and $\sum_{l'=1}^{l-1} I^{l'}_{l,n_l} \times 2^{l'-1} \geq 2^{l-1}$. In DCA [25], the definition of optimum code is different. The code $C_{l',n_{l'}}$ is in the path from root to optimum code if $I^{l''}_{l',n_{l'}} = 0, 1 \leq l' \leq l-1$ is maximum, *i.e*. the number of vacant children in layer 1 to l-1 for code $C_{l',n_{l'}}$ is highest. The algorithm does not intend to find the crowded part rather it finds a blocked code with capacity $2^{l-1}R$, which has least number of busy children. After selecting the best blocked code, all the calls handled by its busy children have to be shifted to appropriate (crowded) locations using top down scheme. When all calls are shifted the code becomes available for $2^{l-1}R$ call. The DCA algorithm alongwith top down scheme works as follows.

1. Generate new call of rate $2^{l-1}R$.

2. *If current used capacity in the tree* + $2^{l-1}R \leq$ *total tree capacity*.



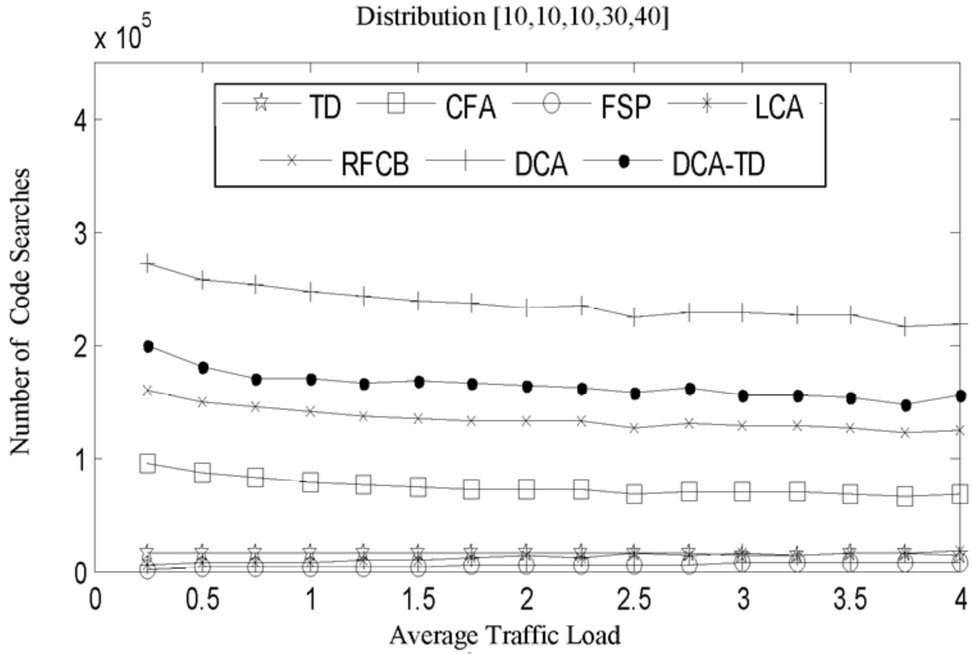
(a)



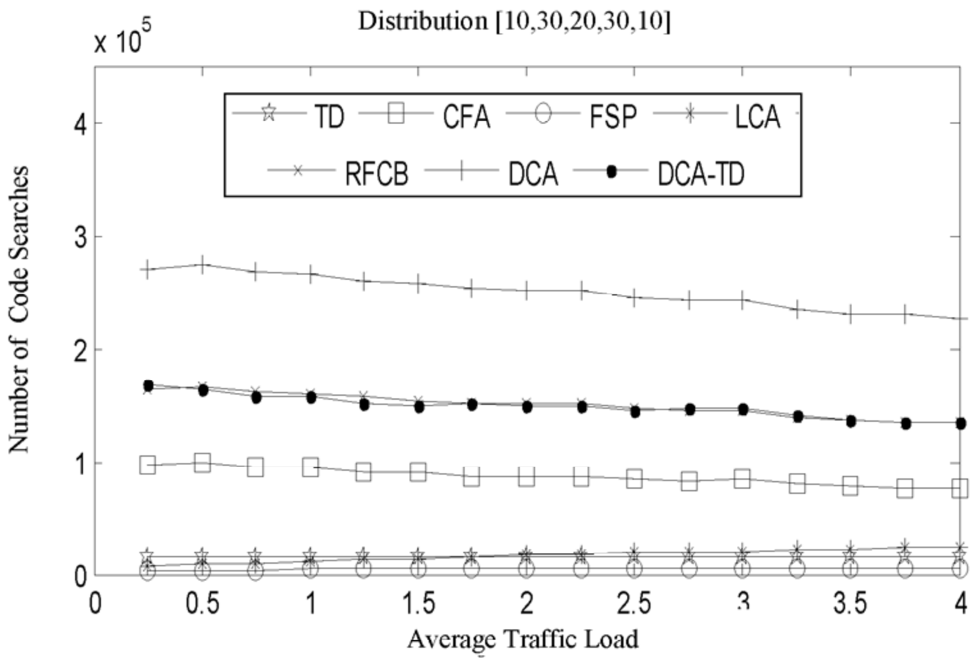
(b)

TD: Top down, CFA [22]: Crowded first assignment, FSP[27]: Fixed set partitioning, LCA[22]: Left code assignment, RFCB[33]: Recursive fewer codes blocked, DCA[25]: Dynamic code assignment, , DCA-TD: Dynamic code assignment with TD.

Figure 3.4: Comparison of number of code searches in single code schemes for distribution: (a) [20,20,20,20,20] (b) [40,30,10,10,10]



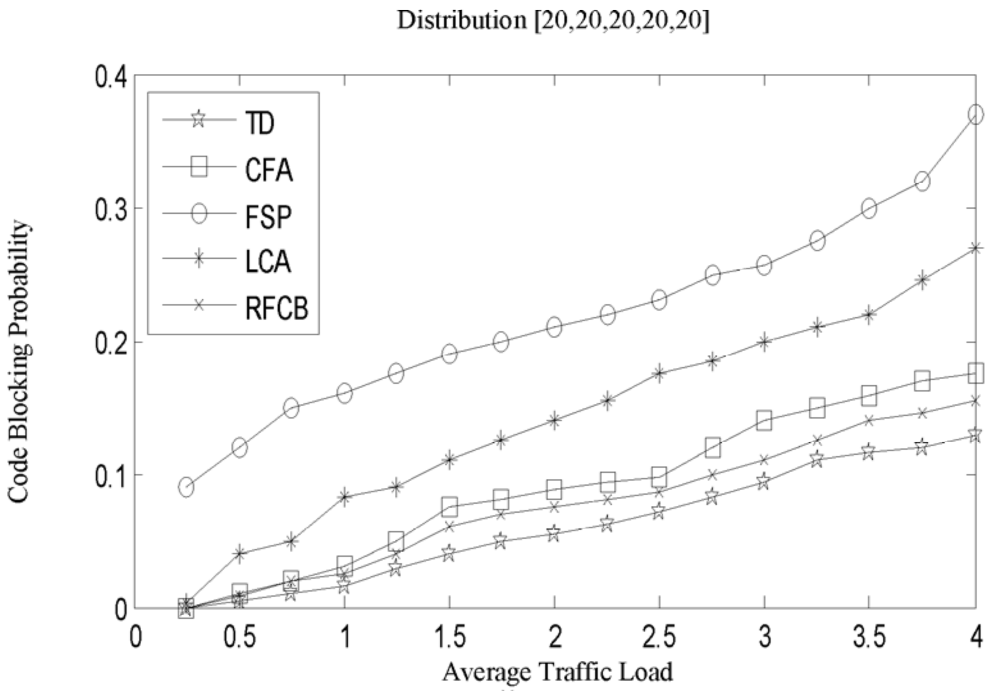
(a)



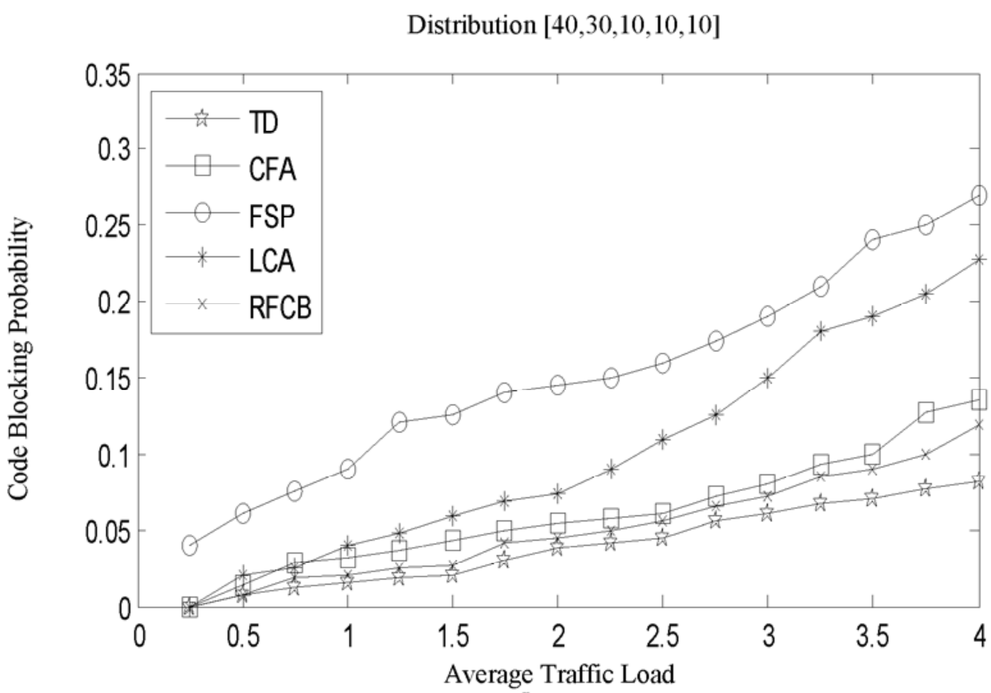
(b)

TD: Top down, CFA [22]: Crowded first assignment, FSP[27]: Fixed set partitioning, LCA[22]: Left code assignment, RFCB[33]: Recursive fewer codes blocked, DCA[25]: Dynamic code assignment, , DCA-TD: Dynamic code assignment with TD.

Figure 3.5: Comparison of number of code searches in single code schemes for distribution: (a) [10,10,10,30,40], (b) [10,30,20,30,10].



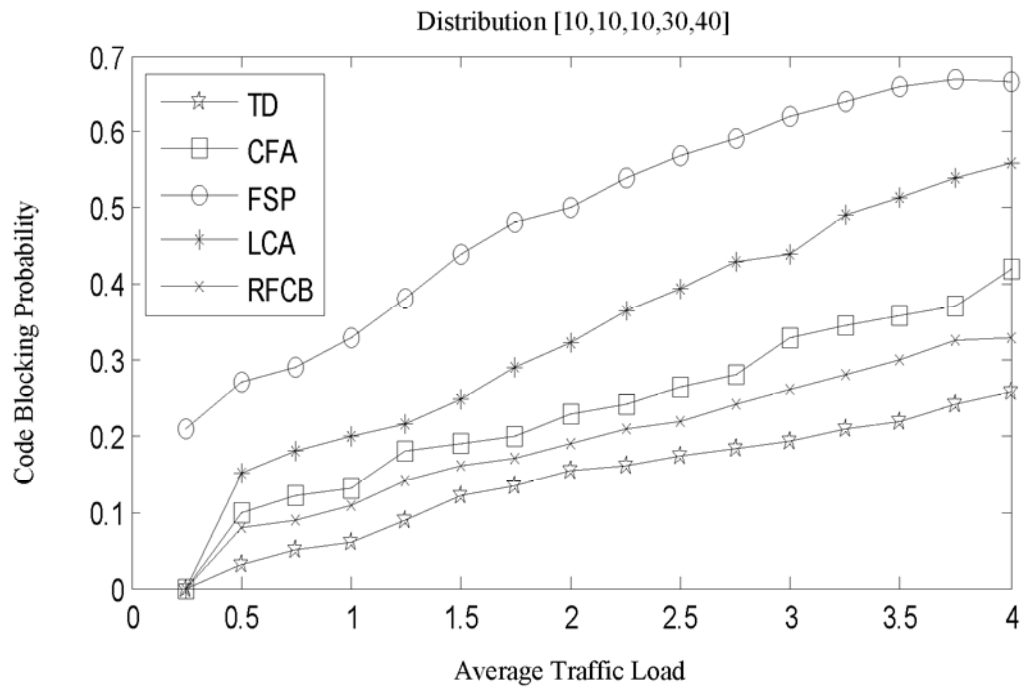
(a)



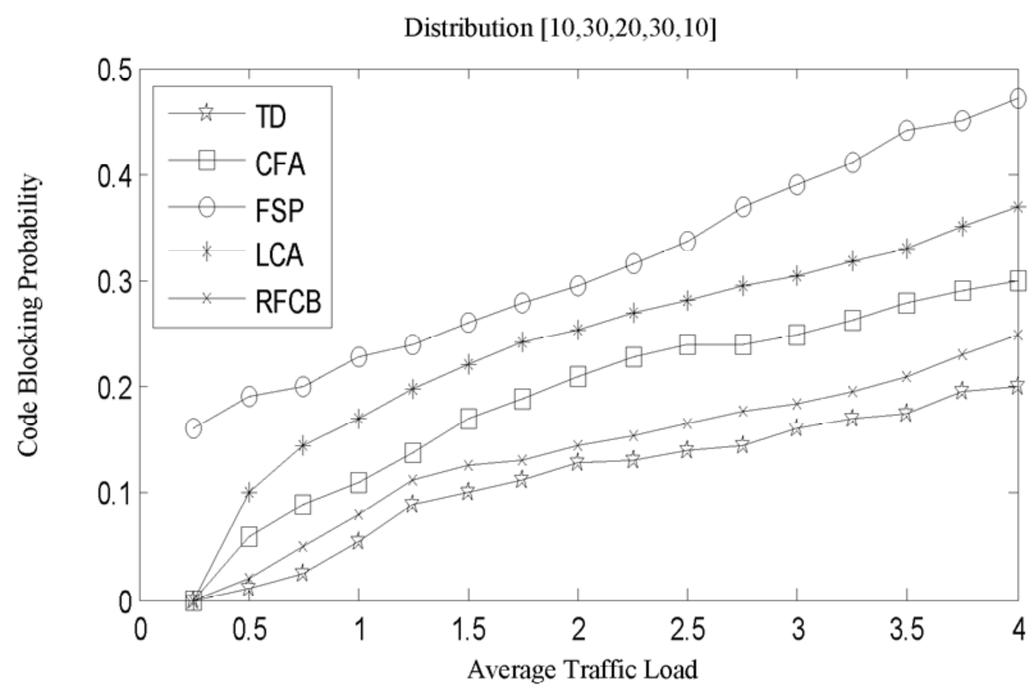
(b)

TD: Top down, CFA [22]: Crowded first assignment, FSP[27]: Fixed set partitioning, LCA[22]: Left code assignment, RFCB[33]: Recursive fewer codes blocked, DCA[25]: Dynamic code assignment, , DCA-TD: Dynamic code assignment with TD.

Figure 3.6: Comparison of Code Blocking Probability in single code schemes for distribution: (a) [20,20,20,20,20] (b) [40,30,10,10,10]



(a)



(b)

TD: Top down, CFA [22]: Crowded first assignment, FSP[27]: Fixed set partitioning, LCA[22]: Left code assignment, RFCB[33]: Recursive fewer codes blocked, DCA[25]: Dynamic code assignment, , DCA-TD: Dynamic code assignment with TD.

Figure 3.7: Comparison of Code Blocking Probability in single code schemes for distribution: (a) [10,10,10,30,40] (b) [10,30,20,30,10]

2.1 choose the optimum code in layer l'', for which $I_{l',n_{l'}}^{l''}, 1 \le l'' \le l-1$ is maximum.

2.2 *if* (the optimum code has at least one blocked child in layer l)

2.2.1 $l'' = l'' - 1$.

2.2.2 *if* $l'' == l$.

- Optimum blocked code is identified.
- Shift busy children of this code to other areas according to top down scheme.
- Allocate this code to the call $2^{l-1} R$.
- Update code indices of allocated code, reassigned codes, their parents and children.

else

- Go to step 2.1.

end

2.2.3 Go to step 2.1.

else

2.2.1 Sibling code is the optimum code.

2.2.2 Go to step 2.2.

end

else

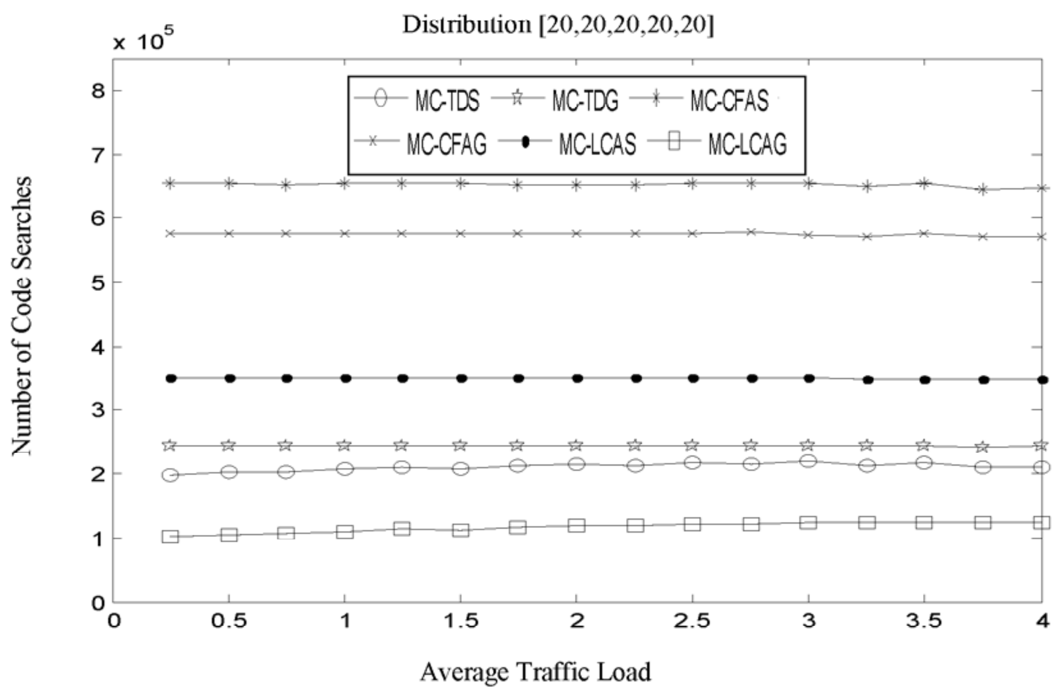
2.1 Reject call.

2.2 Go to step 1.

End

3.2 RESULTS

The codes searched and blocking probability performances of the pure top down and its hybrid schemes are compared with existing good schemes. For simulation, five classes of users are considered with rates R, $2R$, $4R$, $8R$, and $16R$ respectively. For i^{th} class, the arrival rate is represented by $1/\mu$. Call duration $1/\mu_i$ is exponentially distributed with mean value of 1 units of time. If we define $\rho_i = \lambda_i / \mu_i$ as traffic load of the i^{th} class users, then for 5 class system the average arrival rate and average traffic load is $\lambda = \sum_{i=1}^{5} \lambda_i$ and $\rho = \sum_{i=1}^{5} \lambda_i / \mu_i$ respectively. The average arrival rate (or average traffic load as $1/\mu_i$ is 1) is assumed to be Poisson distributed with mean value varying from 0-4 calls per unit of time. In this simulation call duration of all the calls equal $i.e. 1/\mu = 1/\mu_i = 1$ is considered. Therefore, the average traffic load is $\rho = 1/\mu \times \sum_{i=1}^{5} \lambda_i = \lambda / \mu$. The maximum capacity of the tree is $128R$ (R is $7.5 kbps$). Simulation is done for 10000 users and result is average of 10 simulations. Define $[p_1, p_2, p_3, p_4, p_5]$ as capacity distribution matrix, where p_i, $i \in [1,5]$ is the percentage fraction of the total tree capacity used by the i^{th} class users. As the traffic load includes five different rates, where R, $2R$ are low rates and may represent real time calls, and higher rates $4R$, $8R$, $16R$ can be considered as non real time calls. The number of code searches is a good performance parameter and is used in previous works on OVSF code assignments. Two distribution scenarios are analyzed, (*i*) [20,20,20,20,20],



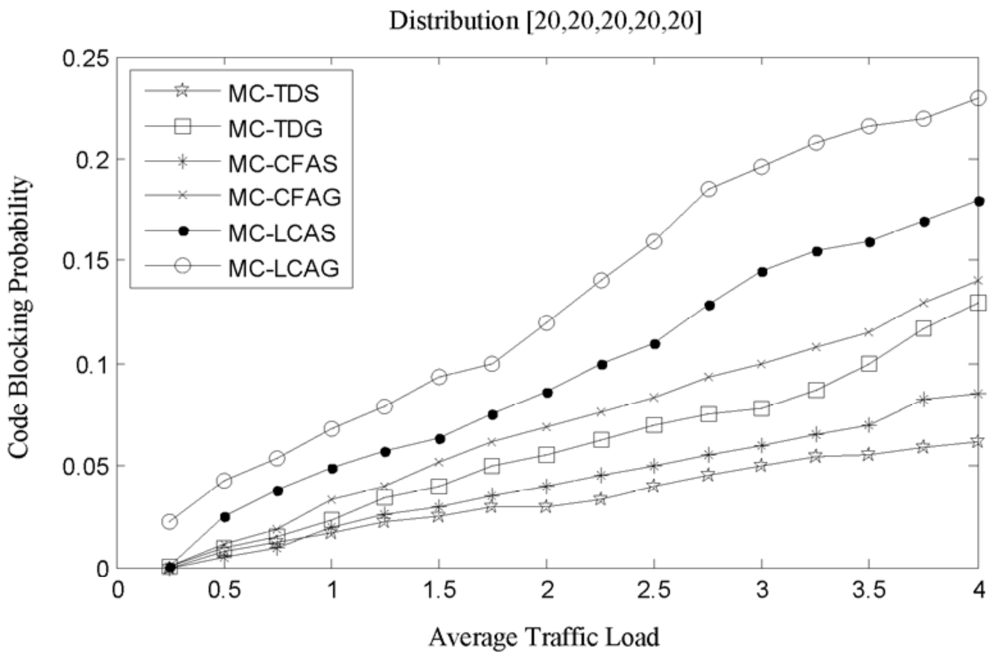
MC-TDS: Multi code top down scattered, MC-TDG: Multi code top down grouped, MC-CFAS[22]: Multi code crowded first assignment scattered, MC-CFAG[22]: Multi code crowded first assignment grouped, MC-LCAS[22]: Multi code left code assignment scattered, MC-LCAG[22]: Multi code left code assignment grouped.

Figure 3.8: Comparison of number of code searches in multi code schemes for uniform distribution

uniform distribution, (*ii*) [40,30,10,10,10], low rates call dominating. The proposed top down (TD) code selection scheme is compared with crowded first assignment [22] (CFA), fixed set partitioning [27] (FSP), leftmost code assignment [22] (LCA), recursive fewer code blocked (RFCB).

3.2.1 SINGLE CODE ASSIGNMENT

The code searches results are plotted in Figure 3.4 and Figure 3.5, code blocking probability



MC-TDS: Multi code top down scattered, MC-TDG: Multi code top down grouped, MC-CFAS[22]: Multi code crowded first assignment scattered, MC-CFAG[22]: Multi code crowded first assignment grouped, MC-LCAS[22]: Multi code left code assignment scattered, MC-LCAG[22]: Multi code left code assignment grouped.

Figure 3.9: Comparison of code blocking probability in multi code schemes for uniform distribution

results in Figure 3.6 and Figure 3.7 respectively. The number of codes searched before assignment are smaller in the top down scheme as shown in Figure 3.4 and Figure 3.5 compare to all other popular approaches except FSP and LCA which suffers from high code blocking probability. The number of codes searched in top down scheme is comparable to LCA and FSP. Since both FSP and LCA produce large code blocking, they are outdated. Hence, top down scheme can be used in the pure form or in the integrated form with other novel single code and multi code methods to improve their performance. The enhanced DCA using top down requires significantly fewer code searches as compare to traditional DCA. The result in Figure 3.6 and Figure 3.7 shows that the top down scheme has significantly less code blocking for all four distributions. The code blocking probability is not plotted for DCA as it always leads to zero code blocking but large reassignment overhead does not encourage the use of DCA.

3.2.2 MULTI CODE ASSIGNMENT

The performance improvements in multi code enhancements namely multi code top down scattered (MC-TDS) and multi code top down grouped (MC-TDG) schemes are compared with multi code scheme equipped with LCA and CFA. Again, two versions of multi code scheme are analyzed for both LCA and CFA, (*i*) Multi code left code assignment with scattered approach (MC-LCAS), (*ii*) Multi code left code assignment with grouped approach (MC-LCAG), (*iii*) Multi code crowded first assignment with scattered approach (MC-CFAS), (*iv*) Multi code crowded first assignment with grouped approach (MC-CFAG). The comparison is done for uniform distribution only. The remaining distributions show similar performance. The MC-TDG scheme requires least number of code searches as compare to all other approaches as shown in Figure 3.8. If CS_x denotes the code searches for scheme x, the various schemes can be arranged as

$$CS_{MC-TDG} < CS_{MC-LCAG} < CS_{MC-TDS} < CS_{MC-LCAS} < CS_{MC-CFAG} < CS_{MC-CFAG} \quad (3.7)$$

Also, MC-TDS scheme produce least blocking compared to other schemes as shown in Figure 3.9. If BP_x denotes the blocking probability for scheme x, the various schemes can be arranged as

$$BP_{MC-TDS} < BP_{MC-CFAS} < BP_{MC-TDG} < BP_{MC-CFAG} < BP_{MC-LCAS} < CS_{MC-LCAG} \quad (3.8)$$

Therefore, the top down scheme is very useful for all traffic conditions and hence can be used for CDMA wireless networks.

3.3 CONCLUSION

The chapter proposes a code assignment scheme which favors real time calls. As we know, real time calls cannot tolerate large call establishment delay. In OVSF based networks, the choice of code assignment has significant impact on call establishment delay. The top down code assignment scheme proposed in this chapter reduces this delay without compromising performance degradation in terms of code blocking. The top scheme is also integrated with multi code and dynamic code assignment schemes. The combination of dynamic code assignment and top down scheme provides best results for both code blocking and code searches (call establishment delay). Also, the requirement of large reassignment overhead encourages the use of multi code scheme with top down integration. Further work can be investigated by making code index updation required for top down scheme adaptive to call arrival rates.

CHAPTER 4
OVSF CODE SLOTS SHARING

In this chapter OVSF (Orthogonal Variable Spreading Factor) codes and non blocking OVSF based WCDMA are used in WCDMA uplink and downlink transmission for assignment in multirate traffic. The orthogonal property of these codes leads to code blocking and new call blocking. Two code (or slot) sharing assignment schemes are proposed to reduce the effect of code blocking using OVSF and NOVSF (non blocking OVSF) codes. The schemes favor the real time calls as they are given higher priority in all 3G and beyond networks. The non blocking OVSF (NOVSF) codes [46-49] provide zero code blocking. There are two categories of NOVSF codes. The first category of codes employ time multiplexing giving different incoming call rates the flexibility of using time slots from a single layer only. The layer used and number of slots per code depends on the type of call and majority. In the second category, the structure of the non blocking OVSF code tree is exactly same as that of OVSF code tree. The codes in different layers are orthogonal to each other (unlike traditional OVSF codes). The SF of the root can be chosen according to the type of wireless network. The chapter aims to provide time slot usage description for two scenarios: (i) when all calls are treated similarly (ii) some calls are given higher priority.

The benefit of the proposed scheme is the better handling of non quantized rates compared to other novel single code and multi code assignment schemes. Both single code and multi code options are analyzed for OVSF as well as NOVSF codes. Simulation results are discussed to show the benefits of the proposed scheme.

Rest of the chapter is organized as follows. Section 4.1 to 4.4 describe NOVSF slot sharing scheme using single code and multiple codes. Section 4.5 explains simulation parameters and results are shown for the amount of calls handled using proposed schemes. The chapter is concluded in section 4.6.

4.1 OVSF BASED TIME SLOTS SHARING

For a layer l, the number of codes in OVSF code tree is 2^{L-l} and number of slots per

****D.S. Saini and V. Balyan, " OVSF code slots sharing and reduction in call blocking for 3G and beyond WCDMA networks ",WSEAS Transactions on Communication, vol. 11, no.4, pp. 135-146, April 2012.**

code is 2^l. It is sufficient to use any one layer from the code tree for time slot sharing with total 2^L vacant slots. For example, if layer 4 is used, there are 8 orthogonal codes in the OVSF code tree with 16 time slots per code making total slots equal to 2^{L-1} (as shown in Figure 4.1). The slots in a layer are denoted by S_x, where x is the slot number. The identifier S_x does not depend on layer number l as any layer can be opted.

4.1.1 SLOTS SHARING WITH SINGLE CODE

Two call scenarios are considered, (*i*) general case: all the calls are treated similarly (ii) some

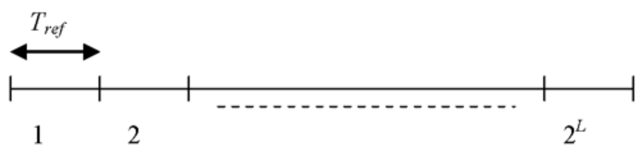


Figure 4.1: Total time slots in any one layer

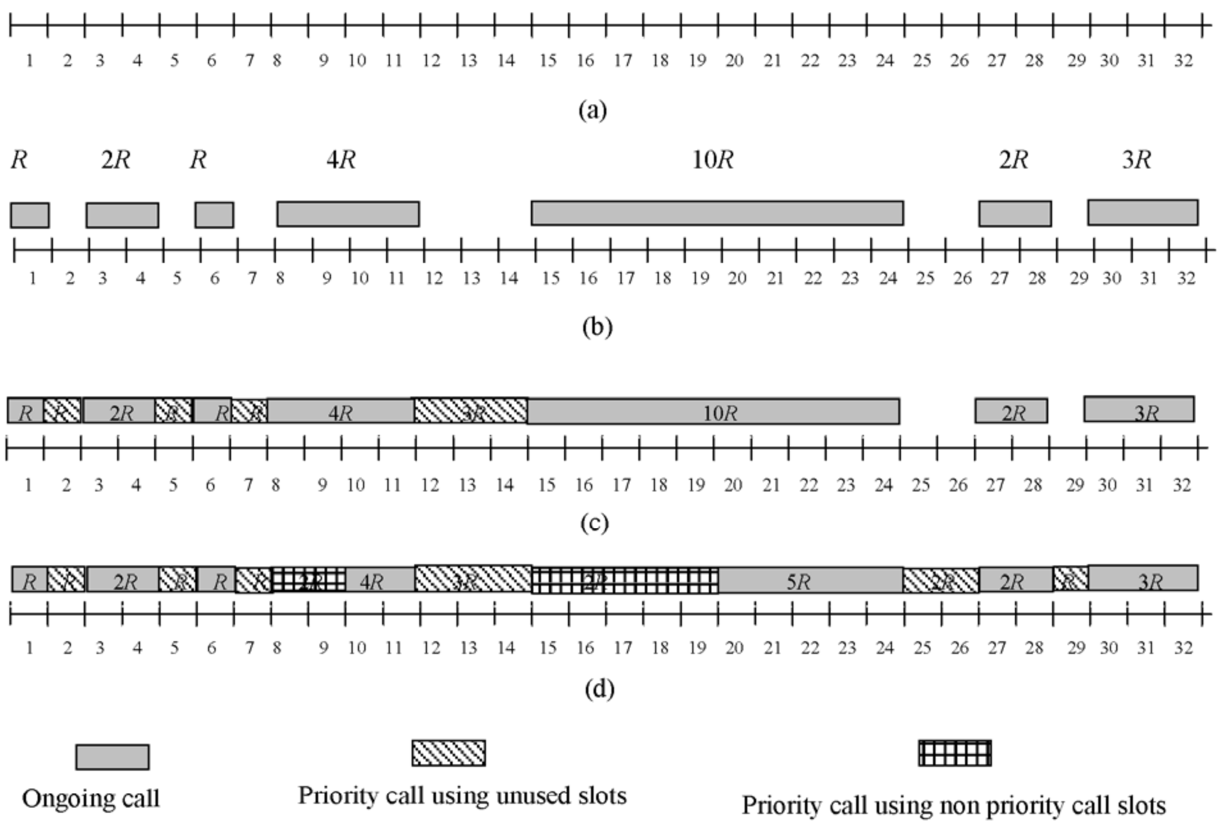


Figure 4.2: Illustration of slot usage in a 32 OVSF CDMA system. All calls except 4*R* and 6*R* are non priority calls. (*a*) total vacant codes available (*b*) Status of the slots at the arrival of 4*R* priority call (*c*) Status of the slots after handling 4*R* call when the system has one rake (*d*) Status of the slots after handling 4*R* call when the system has two rakes (*e*) Status of the slots after handling 6*R* call when the system has two rakes.

layer calls are given higher priority.

A. GENERAL CASE: ALL CALLS ARE TREATED SIMILARLY

For a new call *kR*, if *k* consecutive vacant slots are available, the call is handled. Otherwise the call is rejected. The call rejection is only due to insufficient slots.

B. PRIORITY AND NON PRIORITY CALLS COEXIST

The calls are divided into two categories, namely, priority calls (*P*-calls) and non priority calls (*NP*-calls). When a *NP*-call arrives, the slots are assigned according to the procedure discussed earlier. On the other hand, if *P*-call with rate *kR* arrives, the availability of *k* vacant slots is checked, and if the vacant codes are available call is handled as discussed earlier. If *k* vacant slots are not available, the call can still be handled by allocating some slots currently assigned to *NP*-calls. Thus, the rate of the *NP*-call is reduced at the cost of handling *P*-call. If at the arrival of a new *P*-call of rate *kR*, there are N_p ongoing *P*-calls and N_n ongoing *NP*-calls with corresponding used slots $S_p^i, 1 \le i \le N_p$ and $S_n^j, 1 \le i \le N_n$ respectively, the algorithm identifies the optimum *NP*-call whose some slots can be assigned to the new call. In our scheme we use single level sharing where only one *P*-call can share the slots of a *NP*-call . It may happen that none of existing *NP*-calls has enough slots and therefore, the slots from two or more ongoing calls are required. For a *P*-call of kR rate, if the maximum vacant slots available are $k', k' < k$ and there are some *NP*- ongoing calls with at least $2(k-k')$ slots, the *P*-call call can be handled with level-1 sharing. However, it will increase the time duration of ongoing *NP*-call. If rates of *NP*-calls are denoted by vector $R^n = [R_1^n, R_2^n....R_{N_n}^n]$ from left to right of the code tree, optimum *NP*-call is identified by $R_{i-opt}^n = \max(R_i^n), i \in [1, N_n]$ and R_{i-opt}^n is use to handle *P*-call. The requirement of $2(k-k')$ slots of an optimum *NP*-call guarantee that the existing *NP* calls has at least half the slots available for its own after giving $(k-k')$ slots to *P*-call. This maintains fairness of capacity distribution between *P*-call and *NP*-call. For a i^{th} *NP* call with rate R_i, if there are total S slots out of which S' slots are given to a *P*-call, the new rate of *NP* call becomes $R_i \times (S - S')/S$.

4.1.2 MULTI CODE ASSIGNMENT

A. GENERAL CASE: ALL CALLS ARE TREATED SIMILARLY

Same as discussed in section 4.1.1.

B. ONE OR MORE CALLS ARE GIVEN HIGHER PRIORITY

If a priority call with *kR* rate arrives, find the slot availability as explained in above section. If *r* groups with sufficient vacant slots are not available, the slots from non priority calls are used. Let fraction $k'R \mid k' < k$ is handled by $q \mid q < r$ groups of vacant slots, the rate portion $(k-k')R$ has to be handled by non priority busy call slots. The selection of busy code depends on number of available rakes and number of rakes used will increase level sharing. One rake is used to handle vacant slots group and one rake for each *NP* call whose slots are used.

Level-1 sharing: number of rakes used -2.

Level-2 sharing: number of rakes used -3 and so on.

For remaining $(k-k')$ slots, arrange *NP*-calls in descending order of slots assign to them. If a *NP*-call of maximum slots (say) k_1 with $k_1 \geq 2\times(k-k')$ exists, new call of rate *kR* is handled using k' vacant slots and $(k-k')$ slots of *NP*-call. This is level-1 sharing utilizing two

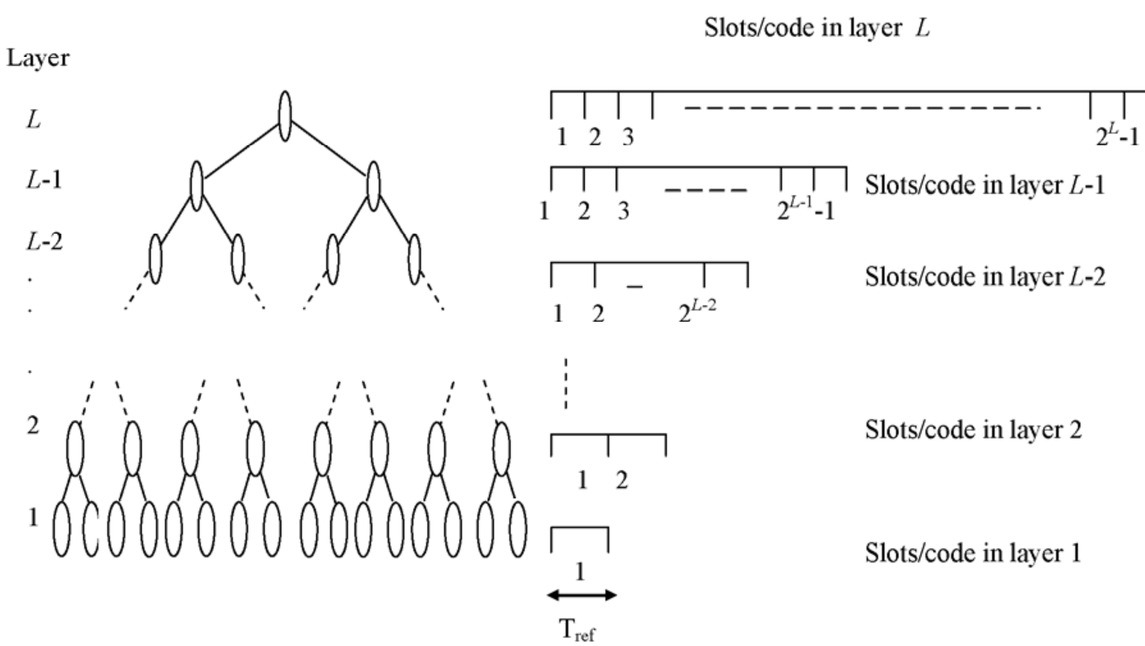


Figure 4.3: (a) *L*-layer N-OVSF code tree (b) Number of slots/code for each layer in part (a)

rakes. If a call rate is not handled and available rakes are more than used rakes, increase level sharing with using more *NP*-calls till $\sum_{i=1}^{m} k_i = (k - k')$, where m denotes *NP*-calls whose slots are used, utilizing m+1 rakes.

Consider a 32 slots OVSF-CDMA system as shown in Figure 4.2 (*a*), where all the users except 4*R* are non priority users. For a particular time, let the status of the code slots is shown in Figure 4.2 (*b*). If a new 6*R* priority call arrives, availability of four vacant slots is checked. As shown in (*c*) there are 3*R* rate vacant slots between 12-14 position and three *R* rate slots available at 2, 5 and 7 positions respectively. This call will be handled without sharing. If a new call of rate 10*R* arrives, vacant slots are not enough to support it. The assigned slots to *NP*-calls are arranged in descending order and half of the slots (required slots) of these calls are assigned to new call. As shown in (*d*), 3*R* available vacant slots and remaining 7*R* is provided by ongoing *NP*-calls slots arranged in descending order 10*R*, 4*R*, 3*R*, 2*R*, 2*R*, *R* and *R*. The call is handled using 3 vacant slots and 5*R*, 2*R* and *R* slots of 10*R*, 4*R* and *R* rate call respectively.

4.1.3 NOVSF BASED TIME SLOTS SHARING

A. SINGLE CODE ASSIGNMENT

Consider a NOVSF CDMA system with L (layer numbers 1 to L) layers in the code tree. The code in layer $l, l \in [1, L]$ is represented by $C_{l,x}, 1 \le x \le 2^{L-l}$ The time frame for code $C_{l,x}$ can be represented as sum of 2^l fixed slots of width T_{ref}, where T_{ref} represents the duration of code time frame in layer 1 known as reference time slot (RTS). For a layer l, the code time frame can be represented as sum of 2^{l-1} RTSs. For a code $C_{l,x}$ the slots are $S_{l,2^{l-1}(x-1)+1}$ to $S_{l,2^{l-1}x}$ with width of each slot T_{ref}. For every layer, the total numbers of slots are 2^{L-l}. For a call of rate $2^{l-1}R$, any of the 2^{l-1} unused slot (consecutive or scattered) in layer l can be used for the new call. This is different from the code allocation without time slot usage where a call can be handled only if there is at least one vacant code (equivalent to 2^{l-1}consecutive vacant slots).

The division of code time frame into slots is illustrated in Figure 4.3 for $l, (l \le L)$ layer OVSF code tree. To illustrate code slots sharing benefits, two scenarios of WCDMA systems are considered (i) general case where all the calls are treated similarly (ii) one (or more) layer calls is (are) given higher priority as compared to others.

A.1 GENERAL CASE: ALL CALLS ARE TREATED SIMILARLY

Consider full vacant code tree. For a new call of rate *kR*, find $\min(l) \mid k \le 2^l$. The call *kR* uses *k* vacant slots of a vacant code $C_{l',x}$, where $1 \le x \le 2^{L-l'}$ and $l' = \min(l, L)$ with $C_{l',x}$ has at least *k* vacant slots. If a code $C_{l',x}$ is used and it has $j, j > k$ vacant slots initially, the vacant slots of code $C_{l',x}$ which can be used by future calls are $j-k$. If layer l' do not have k consecutive vacant slots, find the vacant slots in codes of layer $l'+1$ to L. If we define slot utilization as the ratio of slots used by ongoing calls to the total slots available in the OVSF CDMA system, the code slot sharing increases slot utilization and reduce code blocking significantly.

A.2 ONE OR MORE USERS ARE GIVEN HIGHER PRIORITY

When one or more users (layers) are given higher priority, some of the busy slots of the ongoing non priority calls can be utilized for priority calls. The completion time for non priority calls increases as a consequence. Let layer vectors $\overline{l^p} = [l_1^p, l_2^p \ldots l_P^p]$ and $\overline{l^{np}} = [l_1^{np}, l_2^{np} \ldots l_{L-P}^{np}]$ represents *P* priority and *L-P* non priority layers. For non priority calls a threshold vector $\overline{H^{np}} = [H_1^{np}, H_2^{np} \ldots H_{L-P}^{np}]$ (signifying maximum number of time slots which can be used by priority calls) is used to decide whether or not the incoming priority call can be handled by a non priority layer l_i^{np}. If the number of slots occupied by priority class users in layer l_i^{np} is less than H_i^{np}, some of the busy slots in layer l_i^{np} can be used by the priority layer call. The completion time of the of the non priority call increases (compared to the completion time when there is no slot sharing). For a code $C_{l_i, x_{l_i^{np}}}, 1 \le x_{l_i^{np}} \le 2^{L-l_i^{np}}$, the slot utilization $UC_{l_i, x_{l_i^{np}}}$ is defined as

$$UC_{l_i, x_{l_i^{np}}} = (2^{l_i} - \sum SC_{l_i, x_{l_i^{np}}}) / 2^{l_i} \tag{4.1}$$

In Equation (4.1), $\sum SC_{l_i, x_{l_i^{np}}}$ represents the sum of slots of busy codes $C_{l_i, x_{l_i^{np}}}$, used by the previous priority calls (if any). Slot utilization represents the fraction of the code capacity utilized by ongoing non priority call. The code $C_{l_i, x_{l_i}^{np}}$ with the least value of slot utilization is

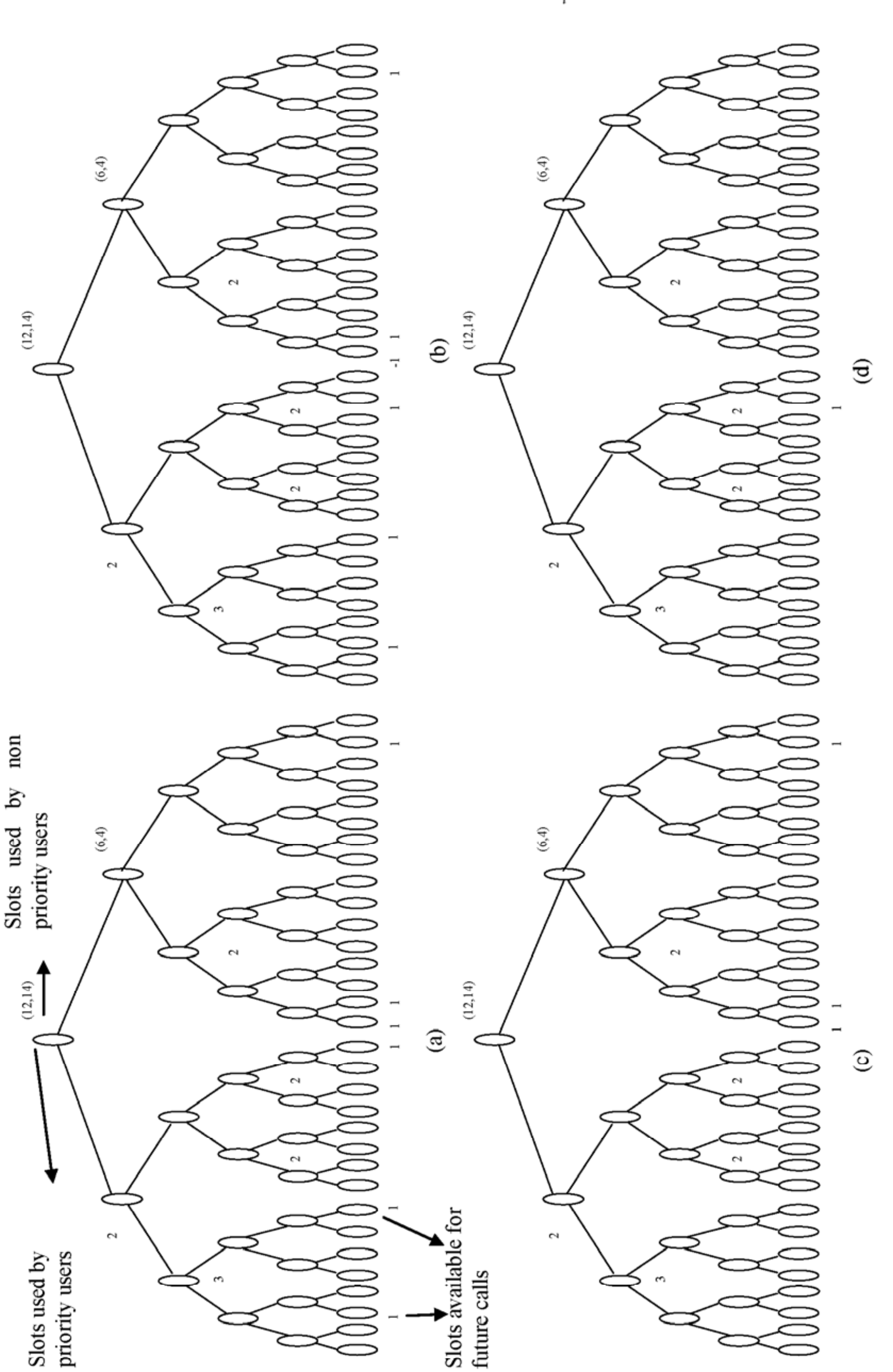


Figure 4.4: Illustration of slot usage in 6 layer NOVSF CDMA systems. All calls except 4*R* and 6*R* are non priority calls. (a) Status of the slots at the arrival of 4*R* priority call (b) Status of the slots after handling 4*R* call when the system has two rakes (c) Status of the slots after handling 4*R* call when the system has one rake (d) Status of the slots after handling 6*R* call when the system has two rakes

the candidate for handling of new call. The aim is not to reduce the number of slots significantly for the ongoing busy call in non priority layer. This maintains the fairness for amount of slots used by ongoing non priority calls. If a tie occurs for slot utilization of two or more busy codes, any one code can be used for priority call handling.

If the priority users can be of non quantized rate nature, consider priority classes represented by vector $\overline{R^p} = [R_1^p, R_2^p....R_p^P]$. All P' priority users fall in P different layers represented by layer vector $\overline{l^p} = [l_1^p, l_2^p....l_p^P]$, where the priority users with rates R_j^p will require time slots from the priority layer l_k^p, if $R_j^p \le 2^{l_k^p} \mid l_k^p = \min(l_k^p)$ for all j. Then for a new call of rate $R_j^p = kR$, the procedure will be same as for quantized rate priority users. This may lead to code capacity wastage and may be avoided by the use of multiple codes as will be discussed in section 4.4.

To illustrate, the delay induced in completion of non priority call by assigning its slots to priority call, assume t_0 is the initial completion time of the call handled by the code $C_{l_i^{np}, x_{l_i^{np}}}, 1 \le x_{l_i^{np}} \le 2^{L-l_i^{np}}$ in non priority layer l_i^{np} with rate $2^{l_i^{np}} R$ whose busy code threshold is not exceeded. The busy code is identified using the slot utilization Equation (4.1). If a new priority call seeking a code in layer $l_{i_1}^p$ arrives and there is no vacant code in layer $l_{i_1}^p$, the initial time t_0 of the code $C_{l_i^{np}, x_{l_i^{np}}}$ is divided into two components $t_0^{'}$ and $(t_0 - t_0^{'})$. The time $t_0^{'}$ signifies the elapsed time of the code $C_{l_i^{np}, x_{l_i^{np}}}$ when the call of $2^{l_{i_1}^p} R$ arrives. The proposed scheme increases remaining time $(t_0 - t_0^{'})$ of the call by assigning $2^{l_i^p} R$ code slots to $2^{l_i^p} R$ priority call. Subsequently, the new remaining time for ongoing non priority call after the arrival of $2^{l_i^p} R$ call becomes

$$t_1 = [(t_0 - t_0^{'}) \times 2^{l_i^{np}}] / (2^{l_i^{np}} - 2^{l_{i_1}^p}) \tag{4.2}$$

Also, the total duration of non priority call becomes $t_0^{'} + t_1$. If some future priority call $l_{i_2}^p$ requires vacant slots and threshold of this non priority call is not exceeded, the $l_{i_2}^p$ number of slots from this call can be utilized by priority call $2^{l_{i_2}^p} R$. The time t_1 of the code $C_{l_i^{np}, x_{l_i^{np}}}$ is divided into two components $t_1^{'}$ and $(t_1 - t_1^{'})$. The new remaining time of the non priority call becomes $(t_0^{'} + t_1^{'} + t_2)$ where t_2 is given as

$$t_2 = [(t_1 - t_1^{'}) \times (2^{l_i^{np}} - 2^{l_{i1}^{p}})] / (2^{l_i^{np}} - 2^{l_{i1}^{p}} - 2^{l_{i2}^{p}}) \quad (4.3)$$

The result can be generalized for m^{th} level sharing of non priority call $2^{l_i^{np}} R$ to handle priority call $2^{l_{im}^{p}} R$. Therefore, the remaining time becomes $\sum_{j=1}^{m-1} t_j^{'} + t_m$, where t_m is defined as

$$t_2 = [(t_{m-1} - t_{m-1}^{'}) \times (2^{l_i^{np}} - \sum_{j=1}^{m-1} 2^{l_{ij}^{p}})] / (2^{l_i^{np}} - \sum_{j=1}^{m} 2^{l_{ij}^{p}}) \quad (4.4)$$

The discussion in the section deals with users with rates in the form of 2^{l-1}. If a new priority call of rate kR, $k \neq 2^l$ requires code, the k consecutive vacant slots are searched for codes in layer $l \mid k \leq 2^l$. If the vacant slots are not available, the codes in layer $l_i^p \mid l_i^p > l$ are searched and k vacant slots in layer l_i^p are utilized to handle the new call. If vacant slots are not present, k number of busy slots used by non priority calls are used as discussed above.

4.1.4 MULTI CODE ASSIGNMENT USING NOVSF

Consider an OVSF system equipped with r rake combiners. We illustrate multi code assignment for two cases as for single code assignment.

A. GENERAL CASE

Consider full vacant code tree. For a new call of rate kR, find $\min(l) \mid k \leq 2^l$. If there are $q_1 \mid q_1 \leq r$ codes in layer l with some vacant slots each, providing total k_1 vacant slots the fraction of kR rate handled by layer l codes is k_1R. The balance rate k_2R=(k-k_1)R is to be handled by codes in the lower layers (l-1,…1). Starting with layer l-1, check the availability of maximum q_2=r-q_1 codes for finding k_2 vacant slots. The procedure can be repeated till layer 1 to find the k_2 vacant slots. Therefore, multi code scheme uses the vacant slots of maximum r different codes to handle new call.

B. ONE OR MORE USERS ARE GIVEN HIGHER PRIORITY

There are two possibilities (i) the call requires code from the non priority layer (ii) the call requires code from the priority layer. If a non priority call kR requires codes (slots) from the non priority layer $l_{i_1}^{np}$ (say) and if there are $q_1 \mid q_1 \le r$ codes (consecutive or scattered) with some vacant slots and all q_1 codes provides $k(\text{or } k_1 \mid k_1 < k)$ total vacant slots, the fraction of *kR* rate handled by layer $l_{i_1}^{np}$ codes is *kR* (or k_1R) as discussed in section 4.2.1. However, if a priority layer call requires slots in priority layer and slots are available then assign new call to them. If vacant slots are insufficient to handle call of rate *kR*, then call will be handled using slots in both priority and non priority layers. If a code in priority layer $C_{l_{i_1}^p, m_1}, 1 \le m_1 \le 2^{L-1-l_{i_1}^p}$ has $MC_{l_{i_1}^p, m_1}$ number of vacant slots available, the total number of vacant slots of codes in priority layer $l_{i_1}^p$ used to handle call of rate *kR* is $\sum_{j=1}^{q_1} MC_{l_{i_1}^p, x_j}, 1 \le x_j \le 2^{L-1-l_{i_1}^p}$. The balance rate k_2R=(k-k_1)R is to be handled by $q_2 \mid q_2 \le (r - q_1)$ codes in the non priority layer $l_{i_2}^{np}$ where $l_{i_2}^{np} = \min(l_{i_z}^{np}) \mid l_{i_z}^{np} < l_{i_1}^{np}$. In lower most non priority layer (say) $l_{i_{L-P}}^{np}$, number of rakes available is. The total number of slots still required (k_{L-P}) can be formulated as

$$k_t = k - \sum_{l=1}^{L-P-1} k_l \tag{4.5}$$

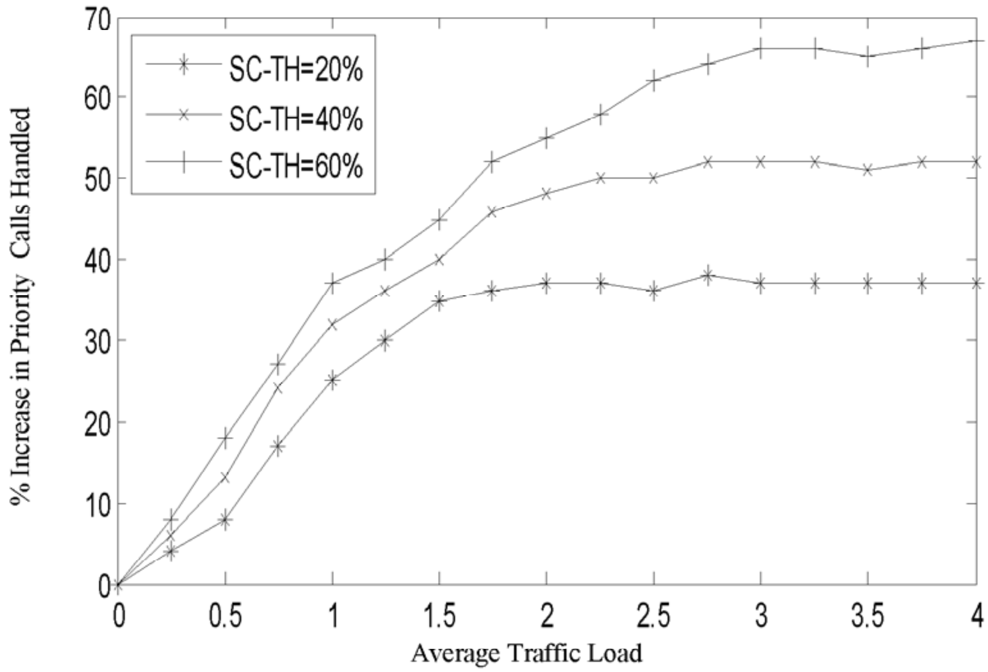
The total numbers of codes which can be used in layer are given by

$$q_{L-P} = r - \sum_{l=1}^{L-P-1} q_l \tag{4.6}$$

If all the non priority lower layers are unable to handle the rate *kR*, the vacant slots are searched in higher layers *i.e.* $\sum_{j=1}^{P} \sum_{m=1}^{q_j} MC_{l_{i_j}^p, x_m^j} \le k$, the vacant slots are searched in higher non priority layers $l_{i_{L-P+1}}^{np} = \min(l_{i_{L-P}}^{np}) \mid l_{i_{L-P+1}}^{np} < l_{i_1}^p$ in ascending order of layer number. If layer $l_{i_{L-P+z}}^{np}$ denotes the uppermost non priority layer, the number of rakes available in layer $l_{i_{t+z}}^{np}$ is $r - \sum_{l=1}^{L-P+z-1} q_l$. The number of slots available at layer $l_{i_{L-P+z}}^{np}$ is $k_{L-P+z} = \sum_{l=1}^{L-P+z-1} k_l$. If the number of slots required is less than k_{L-P+z}, the call is handled. If there are insufficient codes (vacant slots) available even in uppermost non priority layer, the layers $l_{i_{L-P+1}}^{np}$ to $l_{i_{L-P+z}}^{np}$ be checked for the threshold of number of busy codes. The procedure given in section can be used for using few vacant slots for incoming priority call.

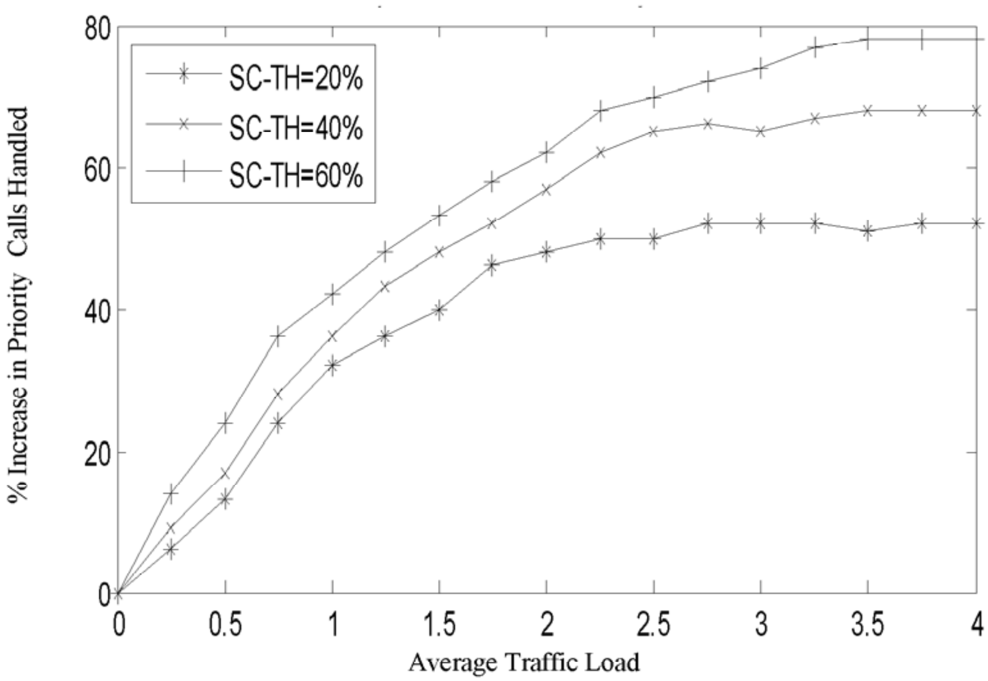
Consider a six layer *NOVSF* code tree as with the status of the codes as shown in Figure 4.4 (*a*). In layers 4 and 5, the variable (*a*,*b*) represents that *a* number of code slots are used by

Priority Calls=20%, Non-Priority Calls=80%



(a)

Priority Calls=80%, Non-Priority Calls=20%

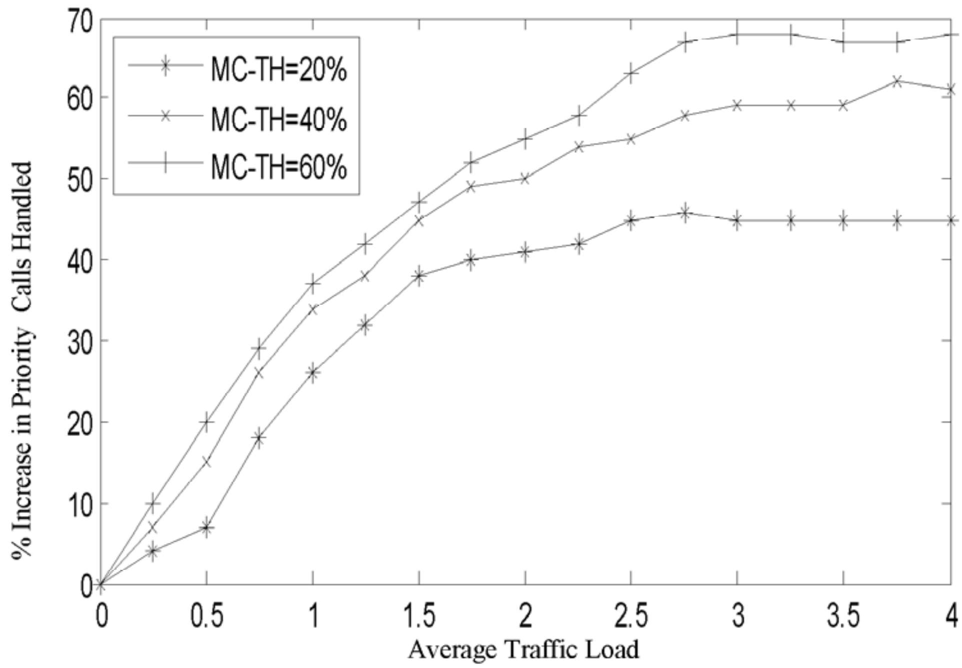


(b)

SC: Single Code, TH: Threshold.

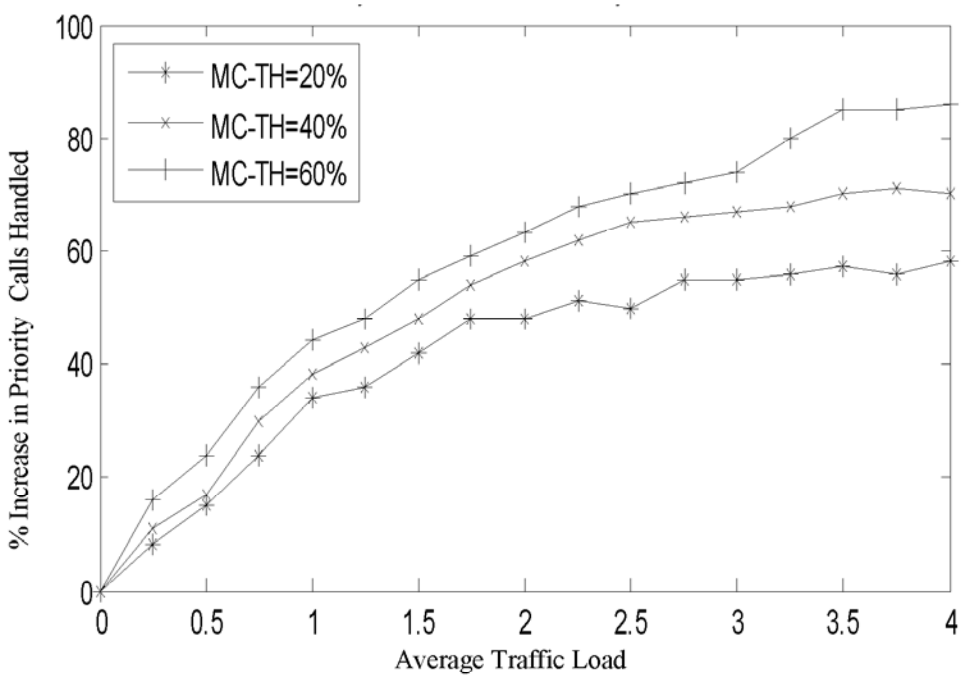
Figure 4.5: Comparison of percentage of priority call handled in single code scheme in (a), (b) for arrival rates distribution: Priority Calls=20%, Non-Priority Calls=80%, and Priority Calls=80%, Non-Priority Calls=20%

Priority Calls=20%, Non-Priority Calls=80%



(a)

Priority Calls=80%, Non-Priority Calls=20%



(b)

MC: Multi code, TH: Threshold

Figure 4.6: Comparison of percentage of priority call handled in multi code scheme in (a), (b) for arrival rates distribution: Priority Calls=20%, Non-Priority Calls=80%, and Priority Calls=80%, Non-Priority Calls=20%

non priority users and b number of code slots are used by priority users. In other layers the variable c on a particular code represents c number of vacant slots for that code which can be used for future calls. Let the users with rates $4R$ and $6R$ are priority users. If a new user with rate $4R$ arrives and the system is equipped with one rake, the call require sharing of slots as there is no group of 4 consecutive vacant slots. If threshold for non priority layer l is 2^{l-1}, 4 vacant slots of code $C_{4,2}$ layer are used and the status of the code tree is shown in Figure 4.4 (*b*). If the system is equipped with two rakes, codes $C_{2,6}$ and $C_{1,5}$ are used as shown in Figure 4.4 (*c*). If a new user with rate $6R$ arrives codes $C_{2,1}$ and $C_{4,2}$ are used as shown in Figure 4.4 (*d*).

4.2 SIMULATION RESULTS

Consider L=8 layer OVSF code tree as in the downlink of OVSF System. For simulation, following classes of users are considered with rates R, $2R$, .. $8R$ respectively. The arrival rate λ is assumed to be Poisson's distributed with mean value varying from 0-4 calls per unit of time. Call duration is exponentially distributed with mean value of 3 units of time. The maximum capacity of the tree is $128R$ (R is 7.5*kbps*). Simulation is done for 10000 users and result is average of ten simulations. There are 8 possibilities for arrival calls rates $2^{i-1}R$,. We assume that there are two sub cases (*a*) layer 1,2,3 users are given higher priority and these users make 20% of the arrival distribution and 80% of the calls are non priority calls (*b*) layer 1,2,3 users are priority users and these make 80% of the arrival distribution along with 50% non priority calls. The lowermost layers (1,2,3 *etc.*) are chosen for priority as real time calls are generally are priority calls. It is assumed that the users with two or more different priority layers are not given priority within themselves (though the provision can be made for the same also). The threshold for number of busy codes in non priority layer i, is 2^{8-i} (threshold for layer 8 is 1 as there is only one code). For quantized users system, let λ_{q_i}, $i \in [1,8]$ is the average arrival rate of i^{th} class calls. The average service time (denoted by $1/\mu$, where μ is average service rate) is assumed to be 1 for all arrival classes. The traffic load for the i^{th} class of users is given by $\rho_{q_i} = \lambda_{q_i} / \mu$. Consider that there are G_i, i=1,2,…,8 servers in the i^{th} layer corresponding to G_i number of vacant codes. The total codes (servers) in the system assuming eight set of classes are the given by vector $G=\{G_1,G_2,G_3,G_4,G_5,G_6,G_7,G_0\}$. The maximum number of servers used to handle new call is equal to the number of rake combiners. The code blocking for the i^{th} class is defined by

$$P_{B_i} = [(\rho_{q_i})^{G_i} / G_i!] / \sum_{n=1}^{G_i} (\rho_{q_i})^n / n! \tag{4.7}$$

Define λ_q as the average arrival rate of all calls in the system and is given by $\lambda_q = \sum_{i=1}^{8} \lambda_{q_i}$.

The average code blocking for quantized users is

$$P_B = \sum_{i=1}^{8} (\lambda_{q_i} / \lambda_{nq}) P_{B_i} \tag{4.8}$$

Three threshold levels are considered for single code and multi codes

- Threshold=20%
- Threshold=40%
- Threshold=60%

As the threshold increases slot utilization increases for single code and multi codes assignment as shown in Figure 4.5 and Figure 4.6 respectively. The NOVSF provides zero code blocking; therefore results are not shown for this scheme.

4.3 CONCLUSION

The code blocking is the major limitation of OVSF based 3G systems which lead to call blocking in both real time and non real time calls. In this chapter the code slots sharing schemes are proposed to reduce the code blocking for real time calls which gives reduction in call blocking. The use of multiple rakes along with the code sharing facility can be used to make real time call blocking close to zero. The code sharing (time sharing) is the complicated task and may require lot of effort. Work can be done to optimize the assignment of code slots for different order of priority within priority users.

CHAPTER 5
CALL INTEGRATING SCHEME

In code division multiple access (CDMA), orthogonal variable spreading factor (OVSF) codes are used to allocate vacant codes when new calls arrive. In this chapter, integration is done for allocation of OVSF codes when a quantized or non-quantized call arrives, and further, the voice calls and data calls are treated differently as former are delay sensitive and later can be stored in buffer. Voice calls are assigned a busy or vacant code when a call arrives. The rate of data calls can increase or decrease according to the availability of free capacity in code tree or according to type of traffic load. Single code and multi code schemes are proposed which minimize the internal fragmentation of busy codes by utilizing their wastage capacity. In addition, the external fragmentation is also reduced. The vacant code is assigned when busy code wastage capacity is insufficient to handle a call. Simulation results prove dominance of our scheme over other novel schemes. OVSF codes support call rates that are powers of two *i.e quantized rate* and do not support many intermediate call rates. This reduces some flexibility in the allocation of code resources, and if *non-quantized rates* are assigned OVSF codes it may result in increased internal fragmentation. In this chapter, we proposed a single code and a multi

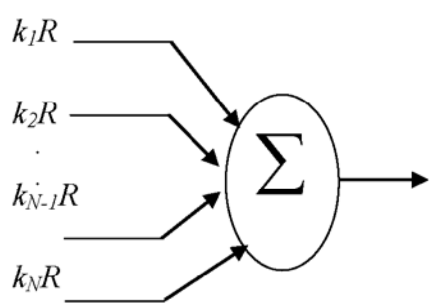


Figure 5.1: Integration of calls at base station.

**** V. Balyan and D.S. Saini, "Integrating new calls and performance improvement in OVSF based CDMA Networks," *International Journal of Computers & Communication*, vol.5, no. 2, pp. 35-42, June 2011.**

code assignment schemes which reduces code blocking by integrating data calls in buffer before assigning them and voice calls are integrated within a code. These schemes reduces internal fragmentation due to ongoing calls by utilizing there wastage capacity. Integration of calls is done as shown in Figure 5.1.

The rest of the chapter is organised as follows. Section 5.1 explains proposed integration scheme. The simulation results are given in section 5.2. Finally, the chapter is concluded in section 5.3.

5.1 PROPOSED INTEGRATION CALLS SCHEME

For an OVSF based CDMA tree of L layer. We define a code C_{l,n_l}, where l denotes the layer number and $n_l, 1 \le n_l \le 2^{L-l}$ denotes the code number in layer $l, 1 \le l \le L$ and rate of a code in layer l is $2^{l-1}R$.

5.1.1 PURE INTEGRATION CALL SCHEME

A. CALL ARRIVAL

If a new call voice or data of rate $kR,\ 1 \le k \le 128$ arrives, search all the busy codes C_{l,n_l} in layer l assigned to ongoing calls with free capacity, where $(l_{\min} \le l \le L)\ \&\ k \le 2^{l_{\min}}$. The proposed scheme uses wastage capacity of busy codes and handles non quantized rate efficiently. The wastage capacity for a code C_{l,n_l} is defined as $W_{l,n_l} = 2^{l-1}R - \sum_{t=1}^{N_{l,n_l}} k_t R$, where N_{l,n_l} denotes the number of calls code C_{l,n_l} is already handling.

1. Find those busy codes for which $W_{l,n_l} \ge kR$.
2. Find code(s) C_{l,n_l^i}, $i \in [1, j_l]$ denotes the number the busy codes which have minimum wastage capacity after handling new call *i.e* $W_{l,n_l^i}^{new} = W_{l,n_l^i} - kR$ is minimum.

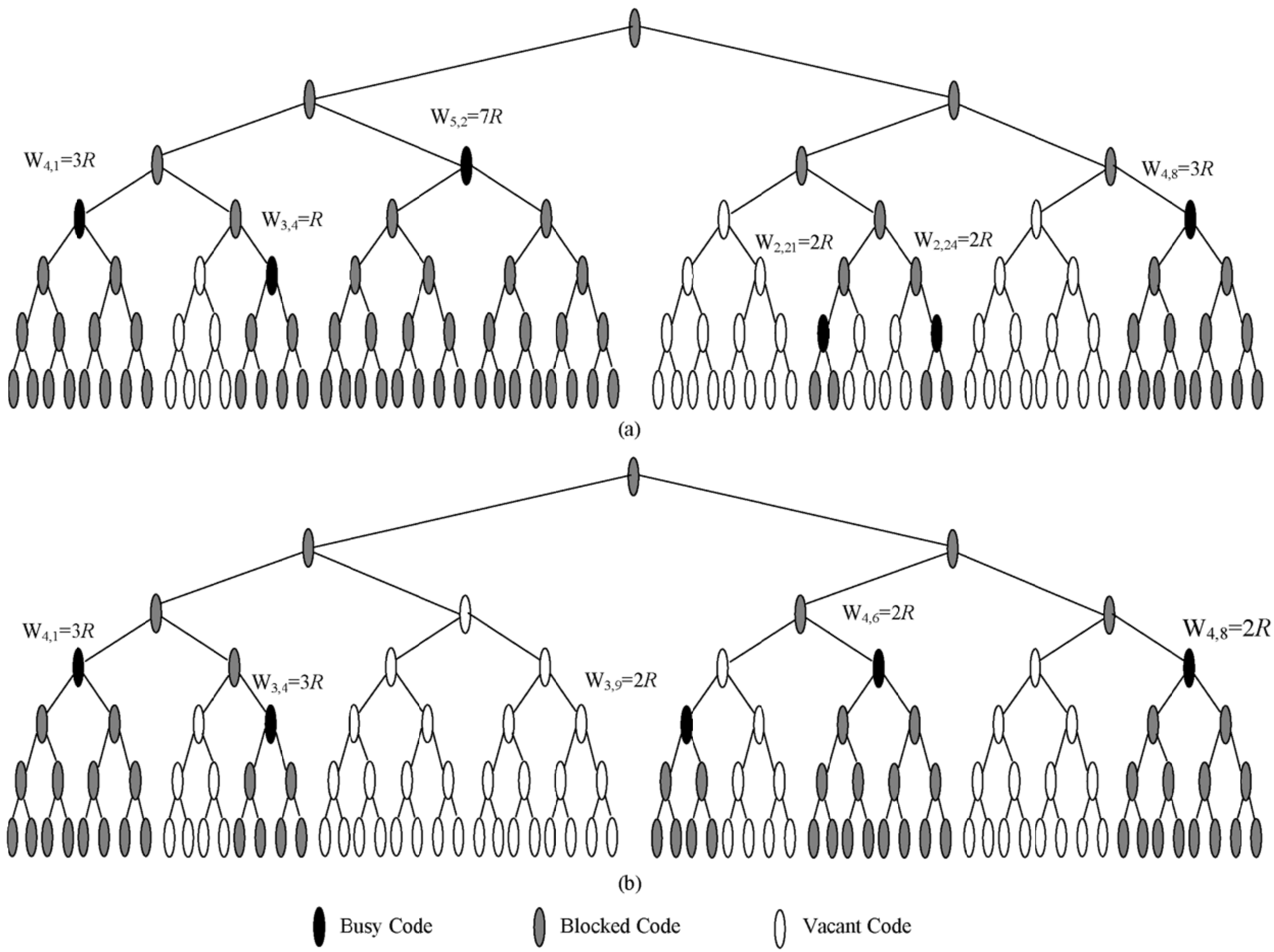


Figure 5.2: Illustration for: a) Pure Integration Scheme. b) Voice Call Priority scheme.

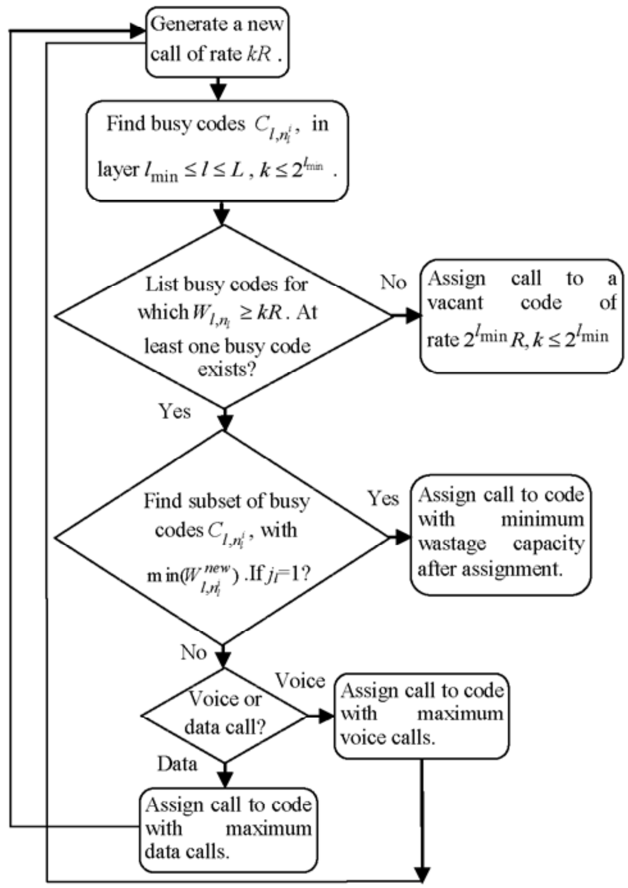


Figure 5.3: Flow chart for Pure Integration Scheme

3. If only one such code exists $j_l = 1$ assign new call to it.
4. Else if, for all j_l having same W_{l,n_l}^{new}, find number of voice calls and data calls they are handling. If new call is a voice call assign it to the code handling maximum voice calls. For a new data call, assign it to the code handling maximum data calls.
5. Else if, no busy code C_{l,n_l} exists which can handle kR rate call, find a vacant code of rate $2^{n_{min}} R$, where $2^{n_{min}} R \ge kR$ and for which $2^{n_{min}-1} R - kR$ is minimum.
6. Else, no busy or vacant code which can handle call of rate kR is available, then block call.

For illustration consider the status of code tree with wastage capacity of codes shown in Figure 5.2 (a). When a voice call of rate $3R$ arrives, this call can be directly assigned to vacant codes of rate $4R$ but it will lead to internal fragmentation of amount R which is avoided in our scheme. The wastage capacity of busy codes is assigned to this call to reduce internal fragmentation, codes with wastage capacity $\geq 3R$ are $C_{4,1}$, $C_{5,2}$ & $C_{4,8}$. Codes $C_{4,1}$ & $C_{4,8}$ have minimum wastage capacity. Tie is resolved by checking the type of call they are handling. If $C_{4,1}$ is handling two data calls only and $C_{4,8}$ is handling one data and one voice call. Then new call will assigned to $C_{4,8}$.

B. CALL TERMINATION

If a call of rate kR terminates of a code C_{l,n_l^i} three scenarios are possible:

1. Code handling only Voice calls

Calculate wastage capacity after a call terminates $W_{l,n_l^i}^{end} = W_{l,n_l^i} + kR$ of code C_{l,n_l^i} and store the information how much more rate this code C_{l,n_l^i} can handle which is equal to $W_{l,n_l^i}^{end}$.

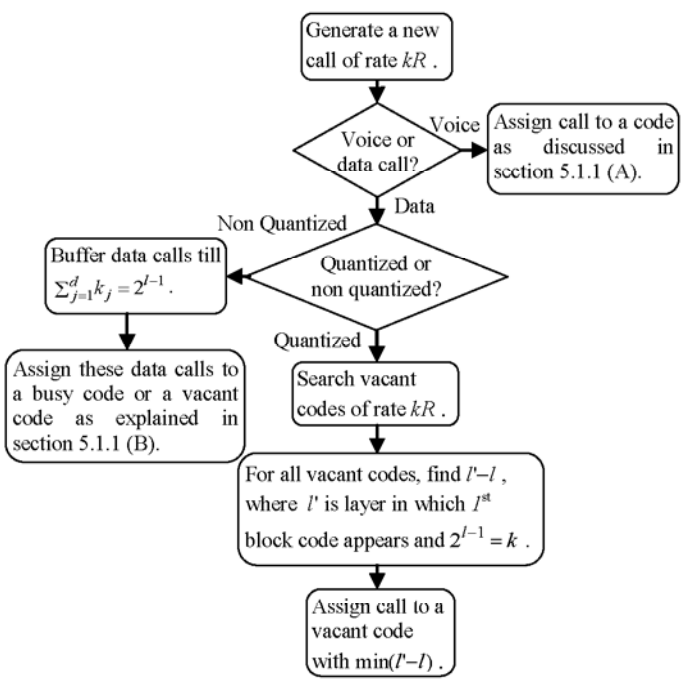


Figure 5.4: Flow chart of Voice Calls Priority

Wait for voice call arrival and assign this code to voice call. However, if no other code C_{l,n_l^i} *i.e* $j_l = 1$ with free capacity to handle data call exists, then this capacity can be can be assigned to data calls.

2. **Code handling only Data calls**

If all the remaining calls of code C_{l,n_l^i} are data calls, then increase rate of remaining data call(s) which will decrease their remaining time. Let the code is handling *x* calls, find the call with maximum remaining time or maximum data rate. If call with maximum remaining time is assigned a rate $k_d R$ and a call of rate *k* ends. It will increase data rate of $k_d R$ to $(k_d + k)R$ and remaining time will decrease by $t_d' \; \alpha \, (1/k_d R) - (1/(k_d + kR))$. If a call arrives and no busy code available to handle it and reducing the rate of data calls to their initial value can handle this call, then data calls are assigned their initial or reduced rate. The reduced rate is the rate subtracted from excess data rate assigned to call with maximum remaining time after handling new call.

3. **Code handling both voice and data calls**

If code is handling both voice and data calls, increase rate of data calls as discussed above.

The flow chart of pure integration scheme, voice call priority scheme and how pure

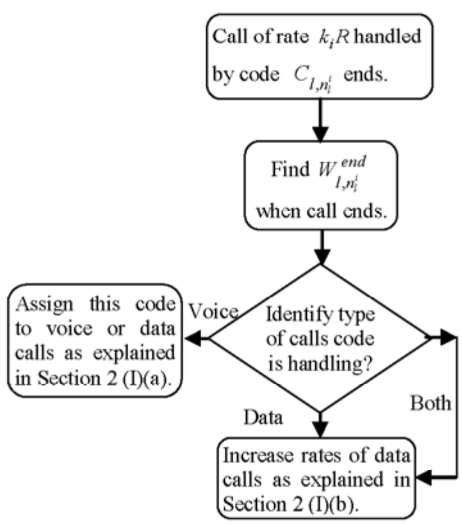


Figure 5.5: Flow chart of Integration Scheme when a call ends.

integration scheme ends a call is shown in Figure 5.3, Figure 5.4 and Figure 5.5 respectively. For illustration consider Figure 5.2 (a). The present status of code tree after assigning 3R rate to $C_{4,8}$ will be $W_{4,8}$=0. $C_{4,1}$ & $C_{5,2}$ wastage capacity is utilized by the data calls assigned to these codes. Let $C_{4,1}$ is handling a call of rate 4R and R & $C_{5,2}$ is handling a call of rate 5R and 4R. If another call of 3R arrives, then $C_{4,1}$ will be the code with minimum wastage capacity, as explained above. Let code $C_{5,2}$ is handling calls of rate 4R, 4R, R and one of the 4R rate call is data call. Wastage capacity $W_{5,2}$=7R can be used by data call. 4R rate data call is then handled by 11R rate. If at the start of 4R rate call its remaining time is $t_{4R}\ \alpha\,(1/4R)$ and time spent before getting 7R rate capacity is assigned to it is t'_{4R}, then its remaining time is $t_{4R} - t'_{4R}$. Now, data will be transferred at a faster rate 11R. Rate of 4R is reduced again to its original or lesser rate when another call of rate $kR, 1 \le k \le 7$ arrives, if elapsed time of data call handled by 11R rate is t'_{11R}, then remaining time of call will be $t_{4R} - t'_{4R} - t'_{11R}$. Let data transferred during t'_{11R} is d, then $t'_{11R} = d/11R$. If same data is to be transferred using 4R rate, then time taken will be $t''_{4R} = d/4R$ which is greater then t'_{11R}. Assigning wastage capacity (unused capacity) is assigned to other data calls it reduces time taken by that call to transfer data and reduces code blocking.

5.1.2 VOICE CALLS PRIORITY SCHEME

When a voice call arrives, it is handled without delay as explained above in pure integration scheme. The data calls which are not delay sensitive can be stored in buffer. For a new data call of rate kR and if code tree is not handling any data call, then

1. If $k = 2^{l-1}, 1 \le l \le 8$, list all vacant code of rate kR and assign call to vacant code which will lead to zero or minimum blocking of higher rate codes. This new code will be of layer l and if first block code above it appears in layer l', than assign new call to vacant code with $\min(l' - l)$.

2. If $k \ne 2^{l-1}, 1 \le l \le L$,

   Then queue data calls in buffer and add all data calls till $\sum_{j=1}^{d} k_j = 2^{l-1}, 1 \le l \le L$, d is the number of data calls added. Assign all data calls in queue to a vacant code C_{l,n_l} which will result in minimum blocking of higher layer code *i.e* $\min(l'-l)$ or to a busy code which can handle all these data calls.

Else

3. Sum data calls after waiting for a particular time in buffer to a vacant code of lower rate which will increase data transfer time. However, it will reduce code blocking significantly for both voice and data calls.

For illustration consider the status of code tree in Figure 5.2 (b). When a data call of rate $6R$ arrives, as $6 \neq 2^{l-1}$, queue it in buffer and wait for data calls till sum of all calls is equal to a quantized value. If another data call of rate $2R$ arrives, sum of both can be handled by $C_{4,7}$.

5.1.3 MULTI CODE INTEGRATION SCHEME

If a call of rate kR cannot be handled using a single code, then multi codes can be used to handle the call utilizing all or required rakes. The multi code integration of voice and data calls uses vacant codes capacity after utilizing total wastage capacity.

1. Find total free capacity T_{l,n_l} *i.e* sum of wastage capacity $\sum_{l=1}^{L} W_{l,n_l}$ of the busy codes or vacant codes free capacity $\sum_{l=1}^{L} V_{l,n_l}$, $T_{l,n_l} = \sum_{l=1}^{L} W_{l,n_l} + \sum_{l=1}^{L} V_{l,n_l}$. Arrange busy codes C_{l,n_l} in decreasing order of their wastage capacity and vacant codes C_{l,n_l} in increasing order of their free capacity.

2. If $\sum_{l=1}^{L} W_{l,n_l} \geq kR$,

   let available busy codes are x and there wastage capacity $\sum_{i=1}^{x} W_{l,n_l}(i) = \sum_{l=1}^{L} W_{l,n_l}$.

   $Sum(W_{l,n_l}) = 0$, i=0;

   While ($Sum(W_{l,n_l}) = kR \| i = x$)

   $Sum(W_{l,n_l}) = Sum(W_{l,n_l}) + W_{l,n_l}(i)$

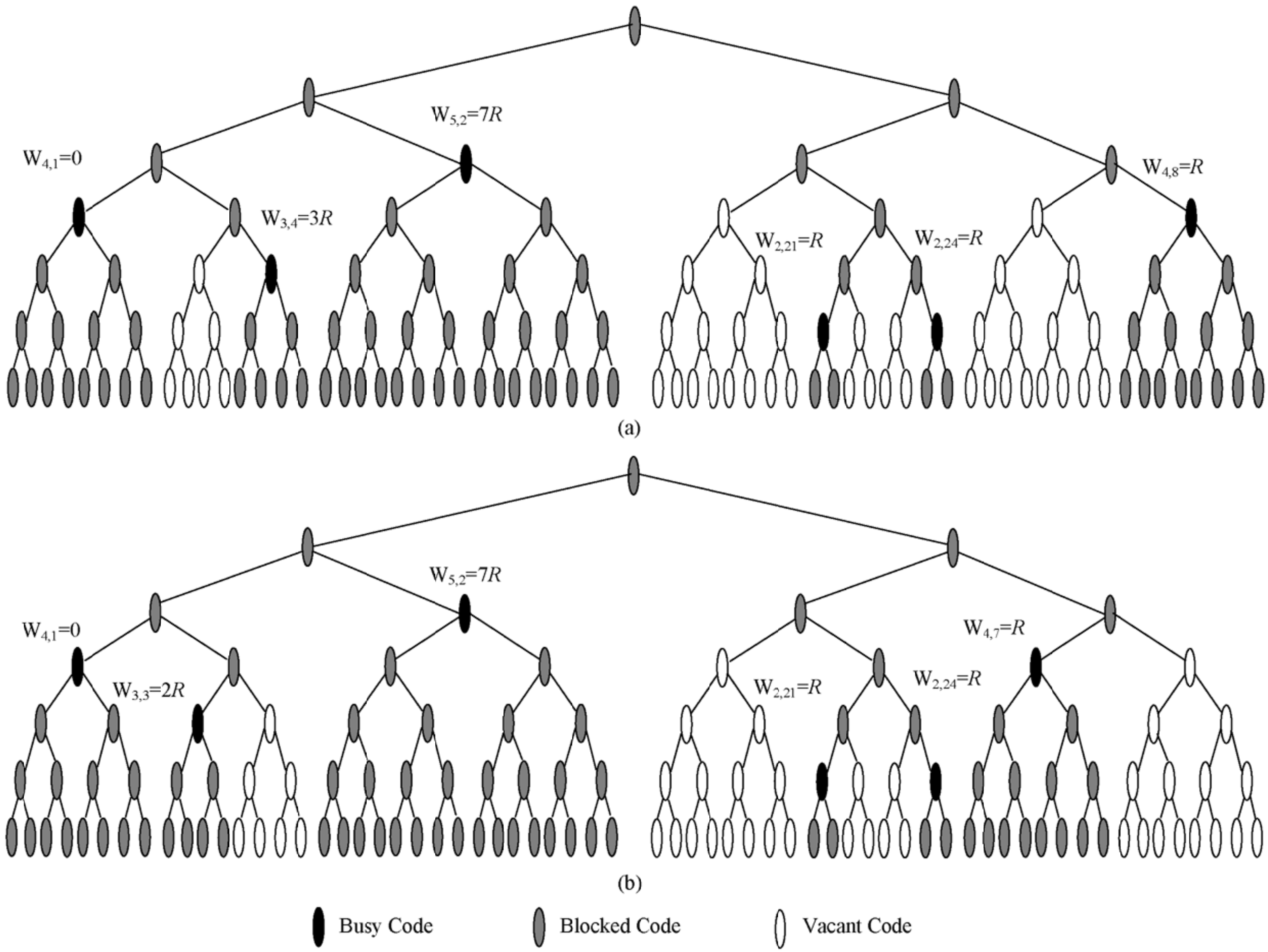


Figure 5.6 : Multi Code Integration Scheme: (a) Using wastage capacity, (b) Using wastage capacity then using free capacity

i=i+1;

End;

Assign call to i codes required using minimum number of rakes or maximum rakes to handle the call. The algorithm picks higher rate busy code(s) with wastage capacity and vacant code of required capacity to handle new call in case of minimum rakes as in Figure. 5.6.

3. Else if

let available busy codes are x and y and there free capacity is

$\sum_{i=1}^{x} W_{l,n_l}(i)$ and $\sum_{i=1}^{y} V_{l,n_l}(j)$.

$Sum(W_{l,n_l}) = 0$, i=0;

While ($Sum\,(W_{l,n_l}) = kR \| \, i = x$)

$Sum\,(W_{l,n_l}) = Sum\,(W_{l,n_l}) + W_{l,n_l}(i)$;

i=i+1;

End

if i=x;

j=0;

$Sum\,(W_{l,n_l}) = Sum\,(W_{l,n_l}) + V_{l,n_l}(j)$;

j=j+1;

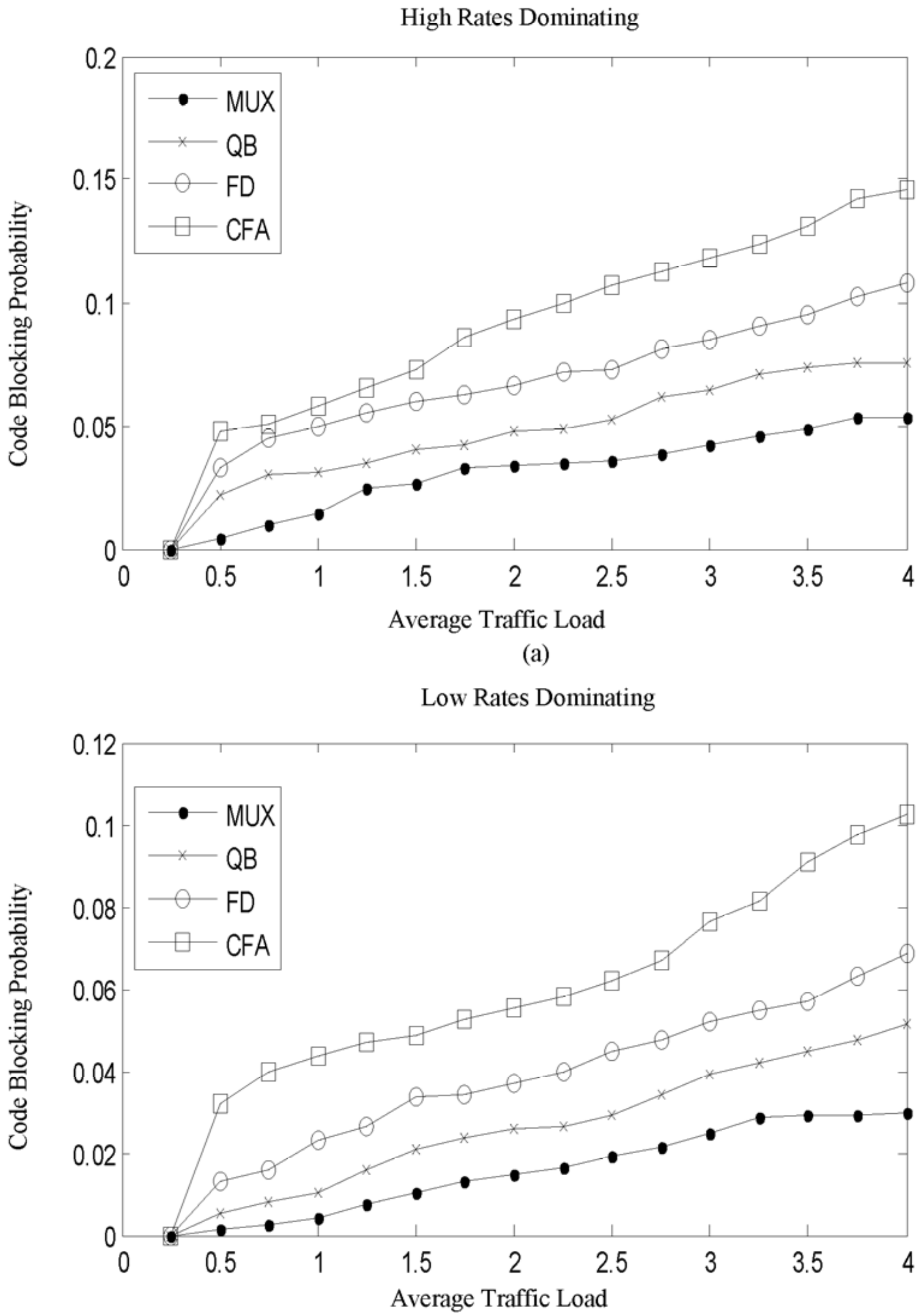
End;

4. Else

Block call.

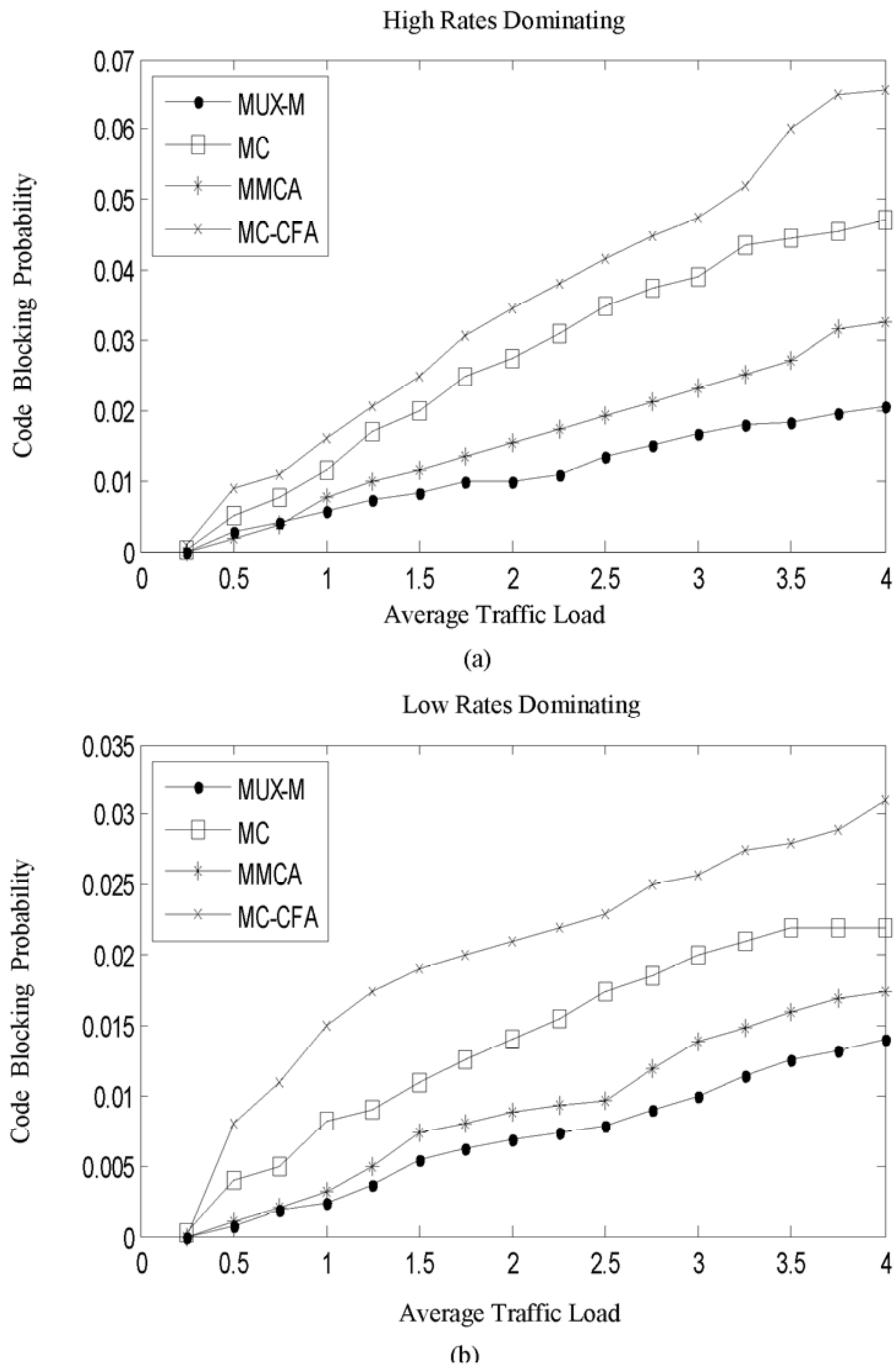
5. End;

For illustration consider the status of code tree in Figure 5.6(a). When a data call of rate 10R arrives, no single code is available which can handle this call. The total wastage capacity is 13R and free capacity is 20R. The vacant code available in code tree will be used when busy code capacity is less then call requested. The algorithm picks busy code $C_{5,2}$ whose wastage capacity



MUX-M: Multiplexing, QB [54]: Quality based, FD [38]: Fixed dynamic, CFA [22]: Crowded first assignment.

Figure 5.7: Comparison of Code blocking probability in single code Schemes for distribution: (a) High Rates Dominating: 2 level sharing, (b) Low Rates Dominating: 3 level sharing.



MUX-M: Multiplexing multi code, MC[10]: Multi code, MMCA[41]: Multi code multi rate assignment, MC-CFA[22]: Multi code crowded first assignment.

Figure 5.8: Comparison of Code blocking probability in multi code Schemes for distribution: (a) High Rates Dominating: 2 rakes, (b) Low Rates Dominating: 3 rakes.

is $7R$ and another busy code $C_{3,4}$ with wastage capacity $3R$ utilizing two rakes. It uses wastage capacity of busy codes. For illustration of wastage capacity and free capacity consider the status of code tree in Figure 5.6 (b), when a new call of $14R$ arrives two cases are possible.

1. **Minimum rakes:** If only three rakes available pick two codes with maximum wastage and one vacant code for remaining required capacity or pick one code with maximum wastage and two vacant codes for remaining required capacity. For example of Figure wastage and two vacant codes for remaining required capacity. For example of Figure 5.6(b) picks codes $C_{5,2}$, $C_{3,3}$ with wastage capacity $5R$,$2R$ respectively and vacant code $C_{4,5}$ with free capacity to handle this call. This will work on number of rakes available and uses maximum possible wastage capacity.
2. **Maximum rakes.** Pick all busy codes with wastage capacity $C_{5,2}$, $C_{3,3}$, $C_{3,21}$, $C_{2,24}$, $C_{4,7}$ whose total free capacity is $10R$ and vacant codes $C_{2,22}$, $C_{2,23}$ whose free capacity is $4R$. This system utilizes total wastage capacity at the cost of more complexity requiring seven rakes to handle same call.

5.2 SIMULATION PARAMETERS AND RESULTS

5.2.1 TRAFFIC CONDITIONS

The codes blocking probability performance of the integration single and multi code schemes are compared with existing schemes in literature. For simulation, following classes of users are considered with rates R, $2R$, .. $15R$ respectively.

- The arrival rate λ is assumed to be Poisson distributed with mean value varying from 0-4 calls per unit of time.
- Call duration is exponentially distributed with mean value of 3 units of time.
- The maximum capacity of the tree is $128R$ (R is $7.5kbps$). Simulation is done for 5000 users and result is average of ten simulations.

5.2.2 RESULTS

Define $[p_1,p_2,..p_{16}]$ as probability distribution matrix where $p_i, i \in [1,16]$, is the capacity portion used by the i^{th} class users. The total codes (servers) in the system for sixteen set of classes are given by set $G = \{G_0, G_1, .., G_{15}\} = \{R, 2R...., 15R\}$ Two distribution scenarios are analyzed and is given by:

- High rates calls dominating
- Low rates calls dominating

The code blocking probability is caused by two reasons: insufficient free capacity in code tree, which causes the so-called capacity blocking and code fragmentation which results in the code blocking probability. Consequently, the blocking probability is composed of the capacity blocking probability and the code blocking probability. The code blocking for a 16 class system is given by

$$P_B = \sum_{i=1}^{16} \frac{\lambda_k P_{B_i}}{\lambda} \qquad (5.1)$$

where P_{B_i} is the code blocking of i^{th} class and is given by

$$P_B = \frac{\rho_i^{G_i} / G_i!}{\sum_{n=1}^{G_k} \rho_i^n / n!} \qquad (5.2)$$

where $\rho_i = \lambda_i / \mu_i$ is the traffic load for i^{th} class.

To decrease the code tree fragments resulting from allocating and releasing codes, allocation of wastage capacity of ongoing calls parents is an imperative procedure to enhance system performance. For single code assignment of integration scheme two and three calls can be assigned to a code *i.e* two or three level sharing of a code in Figure 5.7(a) and Figure 5.7(b) respectively. The two level sharing is used for high rate dominating scenario as number of calls requested will be less as compare to low rate dominating scenario. In Figure 5.7, we compare the proposed schemes with quality based (QB) assignment scheme in [54], fast dynamic scheme in [38] and CFA in [22]. The results shows that for high rates dominating scenario and low rates distribution, the proposed scheme provides significantly less code blocking as compare to rest of the three schemes and high rates dominating scenario provides less external fragmentation compared to low rates dominating scenario.

The use of multi code reduces the code blocking due to reduction in internal and external fragmentation at the cost of complexity. The code tree is better utilized. In general, by the addition of each rake, the code blocking is reduced by half. Also, the multi code scheme is more efficient in handling non-quantized data rates. For multi code assignment of integration scheme two and three rakes are used for high rate dominating scenario Figure 5.8(a) and low rate dominating scenario Figure 5.8(b) respectively. We evaluated the performance of the proposed integration multi code scheme against the multi code assignment schemes presented in [10], MMCA in [41] and multi code CFA [22] (MC-CFA). Integration multi code assignment scheme provides minimum code blocking with less complexity as it uses wastage capacity of busy codes first and then utilizes free capacity of vacant codes which leads to new code blocking when all wastage capacity is utilized and also uses minimum rakes to handle a call.

5.3 CONCLUSION

Capacity of OVSF based WCDMA a precious resource. The wastage capacity increases more for non quantized rates. In this chapter, an integration scheme is proposed which reduces wastage capacity due to non quantized and quantized rates. Two single and one multi code assignment schemes are proposed which handle voice call on priority and multiplex data calls in buffer till a vacant code is available. Work can be done to provide fairness to incoming call rates.

CHAPTER 6
MISCELLANEOUS CALL ASSIGNMENT AND REASSIGNMENT SCHEMES

In this chapter code assignment and reassignment schemes are proposed which aims at reducing complexity, reassignments, number of code searches before code assignment of OVSF codes using single and multi codes for assignment. Code blocking is the major limitation of OVSF based 3G wireless networks. The elapsed time for ongoing calls in the system can be utilized to identify the codes which will be free earlier on average. This leads to efficient code placement and hence vacant code scattering can be minimized. The portion of the code tree which is crowded will remain crowded for longer time and hence the vacant portion has more probability to remain vacant and this handle more high rate users. This is used in section 6.1.

For better utilization of OVSF codes, code assignment must be done so as to reduce number of reassignments in future and reassignments must be done so as to reduce future code blocking. A code assignment and reassignment scheme is proposed in section 6.2 which uses call duration of ongoing calls to decide best candidate for new call and reduce future code blocking. The simulation results show that this scheme can reduce the impact of reassignment process to system efficiently with less call establishment delay due to less number of code searches and maintain low call blocking ratio at the same time.

In section 6.3 an OVSF code assignment scheme is proposed which reduces the complexity involved as compared to crowded first assignment scheme (which is one of the most popular single code assignment schemes). This is because the proposed scheme needs to check the immediate parent of the vacant code for code assignment. The crowded first scheme on the other hand checks all the ancestors of the vacant codes up to the root of the code tree. The code blocking performance can be inferior to the CFA [22] scheme but the number of code searches (which decides the speed of the assignment scheme) is reduced considerably. Further the scheme is investigated using multi codes with and without reassignments which assigns call to a code with immediate busy neighbor or whose immediate neighbor maximum capacity is utilized. Multi code schemes with reassignments require minimum rakes to handle a call and will lead to lesser fragmentation when a call ends. This improves code blocking and requires significant less number of code searches (call processing delay) and complexity. Simulation results are demonstrated to show the superiority of the proposed schemes.

In section 6.4, an efficient flexible code assignment scheme is proposed which utilizes full capacity of the orthogonal variable spreading factor (OVSF) code tree at all time. The scheme favors time bound calls and utilizes scattered capacity when not used by time bound calls for data calls. The call establishment delay for delay sensitive time bound calls is significantly reduced due to less number of code searches, as only a part of code tree is searched for low rates or high rates calls. If there is no vacant code for time bound calls, one of the ongoing data call is stored in buffer and the code released is assigned to time bound call. The data call will get the vacant code later at the completion of some of the ongoing call. Also, the data call rate can be increased or decreased depending upon the vacant code availability in the code tree. Further, the code tree allocates different portions for low rate and high rate calls. Due to the variable rate options, data calls utilize vacant codes scattered in code tree. Simulation results are shown to verify superiority of our scheme.

6.1 CALL ELAPSED TIME SCHEME

6.1.1 PURE ELAPSED TIME BASED APPROACH

Consider an L layer OVSF code tree. If a new call with rate $2^{l-1}R$ arrives, the algorithm lists all the vacant codes in layer l. Let C_{l,n_l} denotes the l^{th} vacant code in layer l. Assume there are m vacant codes in layer l, represented by $C_{l,n_l}, 1 \le n_l \le 2^{L-l}$ and $1 \le l \le L$. For each code C_{l,n_l}, go to immediate parent code $C_{l+1,\lceil n_l/2 \rceil}$. If the code $C_{l+1,\lceil n_l/2 \rceil}$ is blocked, the scheme checks the status of half of its children (which do not include code C_{l,n_l} in its path to code $C_{l+1,\lceil n_l/2 \rceil}$) in layers l to 1. For code $C_{l+1,\lceil n_l/2 \rceil}$, the child in layer l whose status needs to be checked is given by

$$\begin{Bmatrix} C_{l,n+1} \text{ if } n \text{ is odd} \\ C_{l,n-1} \text{ if } n \text{ is even} \end{Bmatrix} \qquad (6.1)$$

If we denote $C_{l,n+1}$ or $C_{l,n+1}$ by $C_{l,z}$, the children of code $C_{l,z}$ in layer l', where $l' < l$, whose status need to be checked are given by

$$C_{l',2^{l-l'}z+m_{l'-1}}, \text{where } 0 \le m_{l'} \le 2^{l-l'} - 1 \qquad (6.2)$$

Let for i^{th}, $(1 \le i \le m)$ vacant code, p_i represents subset of codes given in Equation (6.2) which are busy. For each of the i^{th} vacant code, the algorithm finds sum of elapsed time for all

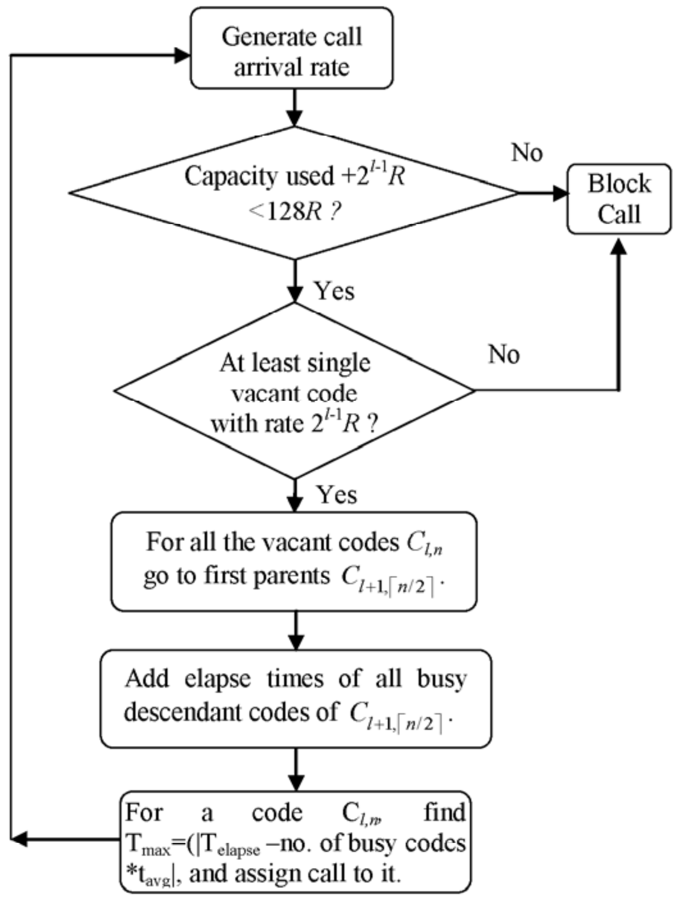


Figure 6.1: Flowchart for the elapsed assignment scheme

the busy children of first parent. If we represent elapsed time of busy code $C_{l',z_j}, 1 \le j \le p_i$, and $(2^{l-l'}z-1) \le z_j \le 2^{l-l'}(z+1)-1$ by t_{l',z_j}, the total elapsed time for the busy codes under parent code $C_{l+1,\lceil n_i/2 \rceil}$ is given by

$$T_i = \sum_{l'=0}^{l} \sum_{j=1}^{p_i} t_{l',z_j} \tag{6.3}$$

If t_{avg} is the average call duration in the system, for all i calculate $|T_i - p_i \times t_{avg}|$. Find $T_j | [|T_i - p_i \times t_{avg}|]$ is maximum for i=j. According to the proposed scheme, to handle a new call with rate $2^{l-1}R$, the most suitable vacant code will be C_{l,n_j}. The fundamental idea is to use the vacant code whose neighbor will be busy for longest duration on average. The crowded portion remains crowded for longer duration. This leads to the most compact code assignment in the OVSF tree. The average call duration t_{avg} can be made adaptive to the traffic type and traffic conditions. One possible way to incorporate this is to find the call duration for N busy

Table 6.1 Selection of optimum code using different code assignment approaches. Call Average time is taken 2 units

Scheme used	New call rate	Vacant codes	Parent code	No. of codes busy under first parent (say N)	Capacity busy under first parent	Elapse Time (T_{elapse})	$\lvert T_{elapse} - N \times t_{avg} \rvert$
Elapse Time	8R	$C_{3,2}$ $C_{3,3}$	$C_{4,1}$ $C_{4,2}$	3 1	R+R+R 4R	12 6	6* 4
Number of Codes Used	8R	$C_{3,2}$ $C_{3,4}$	$C_{4,1}$ $C_{4,2}$	1 2*	4R 2R+2R	3 5	1 1
Hybrid Approach	8R	$C_{3,2}$ $C_{3,4}$	$C_{4,1}$ $C_{4,2}$	2 2	2R+2R 2R+2R	10 11	6 7*
Capacity Used	8R	$C_{3,2}$ $C_{3,3}$	$C_{4,1}$ $C_{4,2}$	2 2	R+R 2R+2R*	11 14	7 10

*indicates selection parameter (columns 6,7 and 9) and the optimum code (column 3)

calls and define $t_{avg} = \sum_{i=1}^{N} t_i / N$, where t_i is the call duration of i^{th} call. The flowchart of the elapsed time scheme is shown in Figure 6.1.

The elapsed call duration scheme is illustrated in Figure 6.2 and Table 6.1. For a new call with rate 8R, there are two vacant candidates $C_{3,2}$, $C_{3,3}$ with first parent $C_{4,1}$, $C_{4,2}$ respectively. Assuming the value of average call duration and elapsed time given in Table 6.1, the optimum code selection will be the code $C_{3,2}$, occurring in the same row in which $\max(|T_{elapsed} - N \times t_{avg}|)$ occurs.

6.1.2 CROWDED FIRST ASSIGNMENT WITH ELAPSED TIME UTILIZATION

A. CROWDED FIRST CODE APPROACH

In this approach, list all the vacant candidate codes C_{l,n_i} in layer l. For each code C_{l,n_i}, list the number of busy children under first parent (in layer l+1). If unique result exists for number of busy codes, the procedure stops and the code with most number of busy codes under first parent will be used to handle new call. Otherwise, traditional CFA goes to parents in layer l+2, l+3,..L till unique result exists. This increases number of code searches exponentially with each additional layer.

For illustration, consider the tree shown in Figure 6.3 and Table 6.1. For a new call with rate 8R, the vacant codes are $C_{3,2}$, $C_{3,4}$ with first parent $C_{4,1}$, $C_{4,2}$. The number of busy codes under $C_{4,1}$, $C_{4,2}$ are 1,2. The new call will be handled by code $C_{3,4}$.

In the hybrid approach, if unique result does not exist in layer l+1, all the codes whose first parents have most number of busy codes are listed. If there are m such codes, for each of these codes the Equations (6.1) to (6.3) can be used (with n_i replaced by m) to identify the best candidate code for new call on the basis of elapsed time. The hybrid approach has benefit of reducing code searches which leads in processing delay reduction. This may be significant criterion for real time calls.

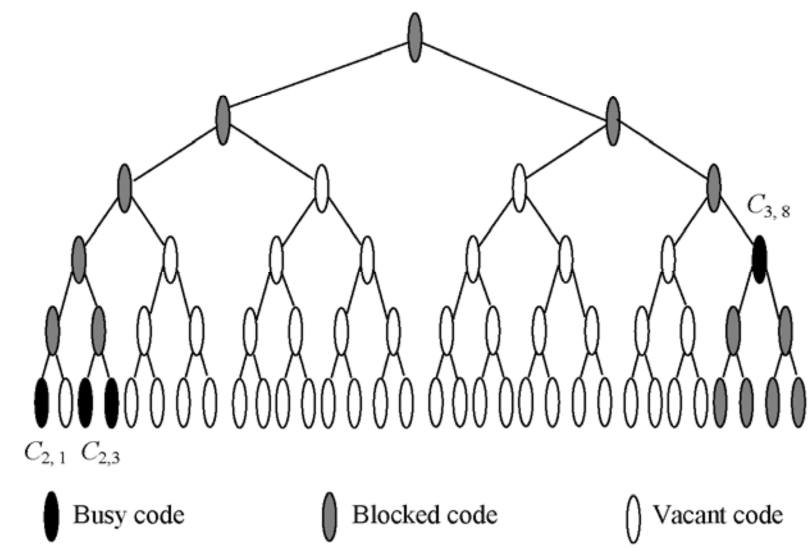


Figure 6.2: Illustration of pure elapse time code assignment design.

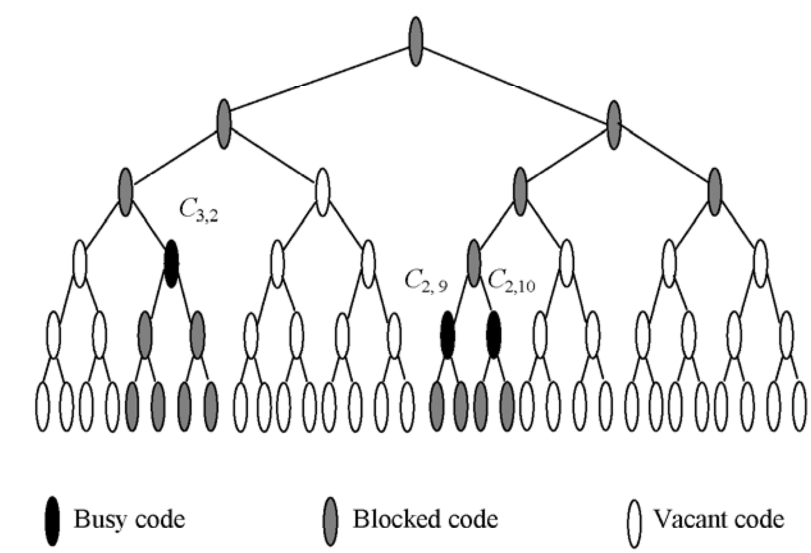


Figure 6.3: Illustration of crowded first code assignment design.

The hybrid approach is illustrated in Figure 6.4 and Table 6.1. There are two candidates $C_{3,2}$, $C_{3,4}$ with first parent $C_{4,1}$, $C_{4,2}$. The number of busy codes and used capacity under each code $C_{4,1}$, $C_{4,2}$ is 2, $4R$ respectively. The optimum selection will be based upon parameter $|T_{elapsed} - N \times t_{avg}|$ and therefore code $C_{3,4}$ appears to be best selection.

B. CROWDED FIRST SPACE APPROACH

The explanation of this hybrid approach is exactly similar to (*i*) except that this scheme checks capacity under the parents in layer *l*+1.

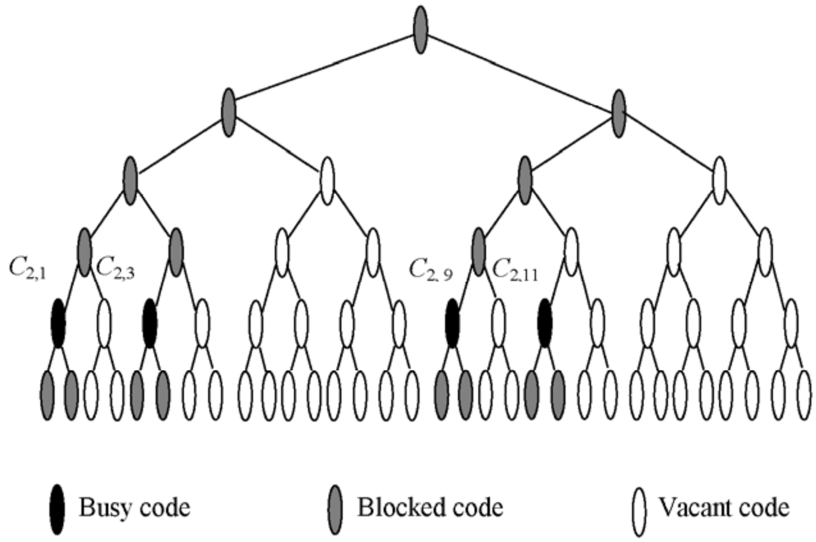


Figure 6.4: Illustration of hybrid approach design

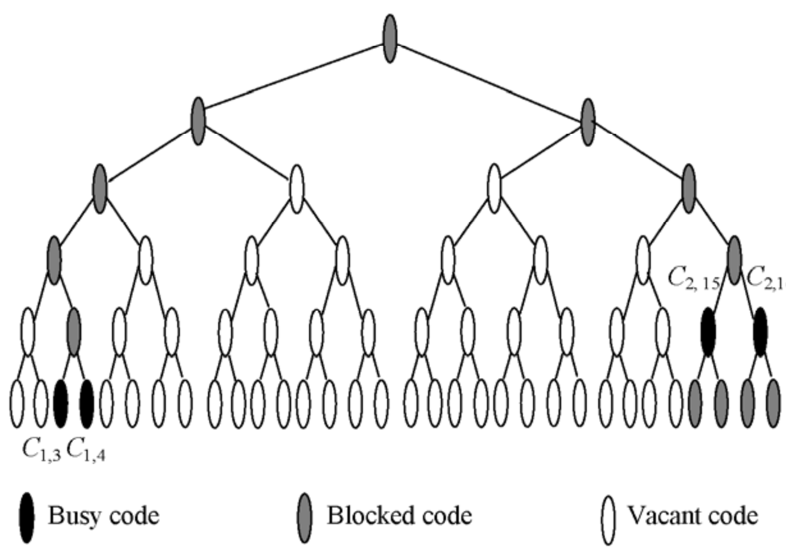
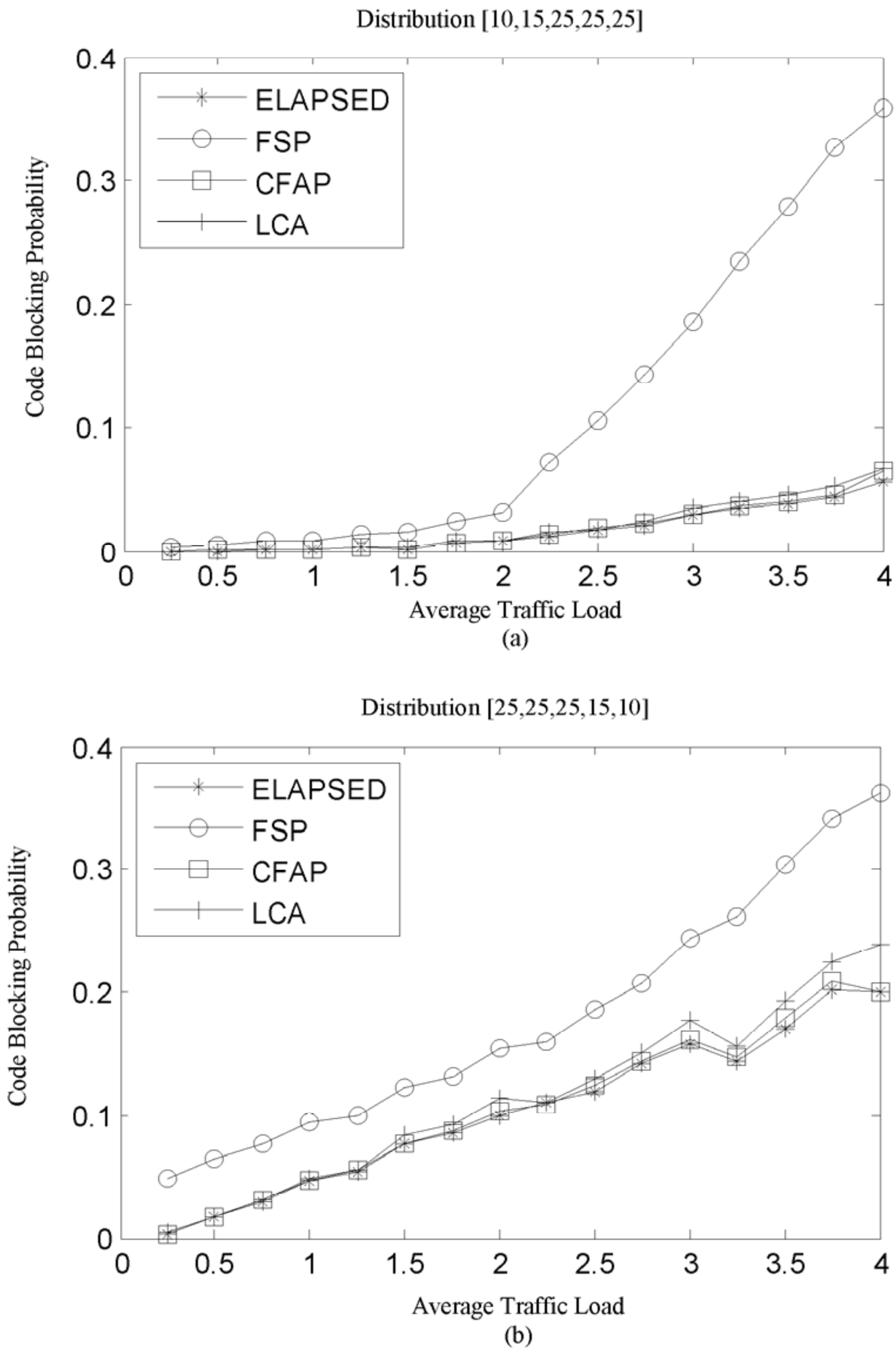


Figure 6.5: Illustration of crowded first space design

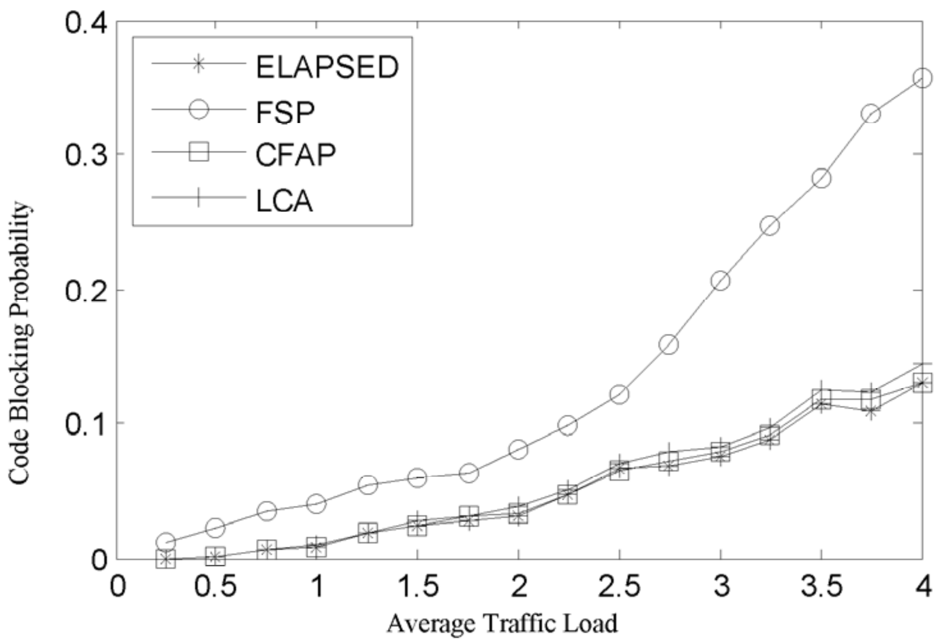
The code assignment can be illustrated with code tree illustrated in Figure 6.5 and Table 6.1. For a new call with rate $8R$, there are two candidates namely $C_{3,2}$, $C_{3,3}$ with first parent $C_{4,1}$, $C_{4,2}$. The capacity utilized under $C_{4,1}$, $C_{4,2}$ is $2R$, $4R$. The new call will be assigned to code $C_{3,3}$.



FSP [27]: Fixed set partitioning, CFA [22]: Crowded first assignment, LCA [22]: Left code assignment.

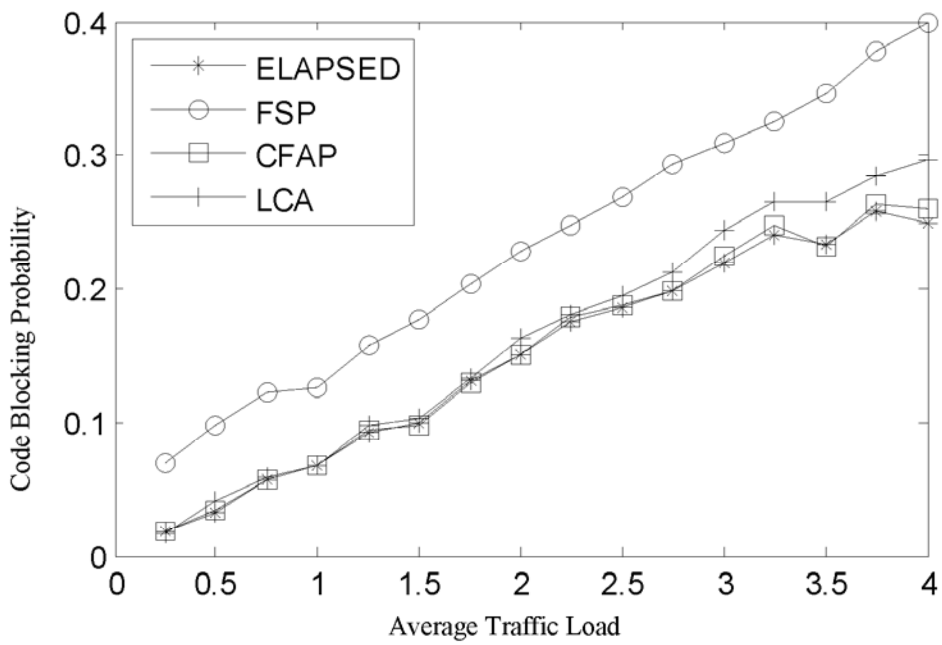
Figure 6.6: Comparison of Code Blocking Probability for distribution: (a) [10,15,25,25,25], (b) [25,25,25,15,10]

Distribution [20,20,20,20,20]



(a)

Distribution [35,30,15,10,10]



(b)

FSP [27]: Fixed set partitioning, CFA [22]: Crowded first assignment, LCA [22]: Left code assignment.

Figure 6.7: Comparison of Code Blocking Probability for distribution: (a) [20,20,20,20,20], (b) [35,30,15,10,10]

6.1.3 SIMULATION AND RESULTS

A. INPUT DATA

- Call arrival process is Poisson with mean arrival rate, λ=0-4 calls/ unit time.
- Call duration is exponentially distributed with a mean value, $1/\mu = 3$ units of time.
- The total capacity of the code tree is 128*R*.
- Possible OVSF code rates are *R*, 2*R*, 4*R*, 8*R* and 16*R*.
- The simulation is performed for 10000 calls and the result is the average of 10 simulations.

B. RESULTS

As mentioned, we consider five classes with quantized arrival rates denoted by rate vector $\lambda \in \{\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5\}$. The total number of codes in the system is, $\{G_1 + G_2 + G_3 + G_4 + G_5\}$, where G_x is the total number of codes corresponding to class x in the system. The service time is $1/\mu$ for all traffic classes. The code blocking probability for the code group $i \in \{1, 2, 3, 4, 5\}$ is given as

$$P_B = (\rho_i^{G_i} / G_i!) / (\textstyle\sum_{n=1}^{G_i} \rho_i^n / n!) \tag{6.4}$$

We compare the code blocking the proposed scheme (denoted by ELAPSED) with leftmost code assignment [23] (LCA), fixed set partitioning [23] (FSP), and crowded first assignment [22] (CFA) schemes discussed earlier. For CFA scheme, crowded first code algorithm is used instead of crowded first space. In CFA scheme we limit the usage of CFA scheme till immediate parent only (denoted by CFAP in result plots). Traditional CFA may go till root of the code tree which can be costly and complex. If C_i denotes percentage of total capacity allotted to rate *iR*, the capacity distribution for rate 16*R*, 8*R*, 4*R*, 2*R* and *R* is given by [C_{16},C_8,C_4,C_2,C_1]. For simulation results we consider following four distributions.

- [10,15,25,25,25], low rates dominating.
- [20,20,20,20,20], uniform distribution.
- [25,25,25,15,10], high rates dominating scenario-I.
- [35,30,15,10,10], high rates dominating scenario-II.

The results in Figure 6.6 and Figure 6.7, shows that the code assignment using elapsed time approach produces less code blocking than FSP and LCA schemes. The code blocking can even becomes lesser than code blocking in CFAP scheme. The elapsed time calculation can be extended to any level of the ancestors. Checking more ancestors reduce the code blocking but may increase the complexity.

6.1.4 CONCLUSION

Code blocking is the major limitation of OVSF based 3G wireless networks. The elapsed time for ongoing calls in the system can be utilized to identify the codes which will be free earlier on average. This leads to efficient code placement and hence vacant code scattering can be minimized. The portion of the code tree which is crowded will remain crowded for longer time and hence the vacant portion has more probability to remain vacant and this handle more high rate users.

6.2 CODE ASSIGNMENT AND REASSIGNMENT BLOCKING

A reactive code reassignment is proposed which assigns and reassigns new call to the children whose parent is having minimum call duration. Consider an OVSF code tree of layer L. Let $C_{l,n_l}, 1 \le n_l \le 2^{L-l}$ denotes a code in layer l with branch number n_l. If a new call of rate $2^{l-1}R$ (R is 7.5*kbps*), $1 \le l \le L$ arrives. Our scheme assigns new call to a code whose parent code has ongoing call(s) of minimum duration. Define call duration T^{CD}_{l,n_l} of a code C_{l,n_l} as time duration since code C_{l,n_l} is assigned to a call or sum of time its children code(s) are assigned to call(s). The total capacity and used capacity of the code tree is denoted by TC and UC, where $TC_{\max} = 2^{L-1}R$.

6.2.1 CODE ASSIGNMENT SCHEME

The proposed scheme assigns new call of rate $2^{l-1}R$ to a code C_{l,n_l} whose parent code of rate $2^{l}R$ has minimum call duration of ongoing children codes calls where $T^{CD}_{l+1,n_{l+1}} \neq 0$ and one available vacant code of rate $2^{l-1}R$. Codes with minimum duration of call will have maximum remaining time; assigning new call to a code with maximum remaining time will use the code for more time. The total call duration of a code of rate $2^{l}R$ is formulated as

$$T^{CD}_{l+1,n_{l+1}} = T^{CD}_{l,n_l} + T^{CD}_{l-1,n_{l-1}} + \ldots + T^{CD}_{1,n_1} \quad (6.5)$$

or $$T^{CD}_{l+1,n_{l+1}} = \sum_{i=1}^{l} T^{CD}_{i,n_i} \quad (6.6)$$

where $C_{l,n_l}, C_{l-1,n_{l-1}}, ..C_{1,n_1}$ denotes the assigned (busy) children codes of parent code of rate $2^l R$. If $T^{CD}_{l+1,n_{l+1}} = 0$ implies that $C_{l+1,n_{l+1}}$, and all its children codes are vacant and will not be assigned to a new call, $T^{CD}_{l+1,n_{l+1}} \neq 0$ exists. For a tie of codes having same call duration, then the code with least capacity wastage is assigned to new call.

This code assignment scheme increases utilization of code tree. Code reassignment is carried out when, $UC \geq (1/2 \times TC)$. Fragmented code tree capacity can be better utilized by using reassignments.

6.2.2 CODE REASSIGNMENT SCHEME

Due to the statistical nature of the arrival and departure of calls, the assigned codes will be randomly scattered across the code tree, even if careful counter measures are taken. This will occur more, if the allocation of codes is performed randomly or even in an ordered fashion. This will result in fragmentation of code tree capacity, more blocking of higher layer codes and reduced spectral efficiency. When a new call of rate $2^{l-1}R$ arrives and no vacant

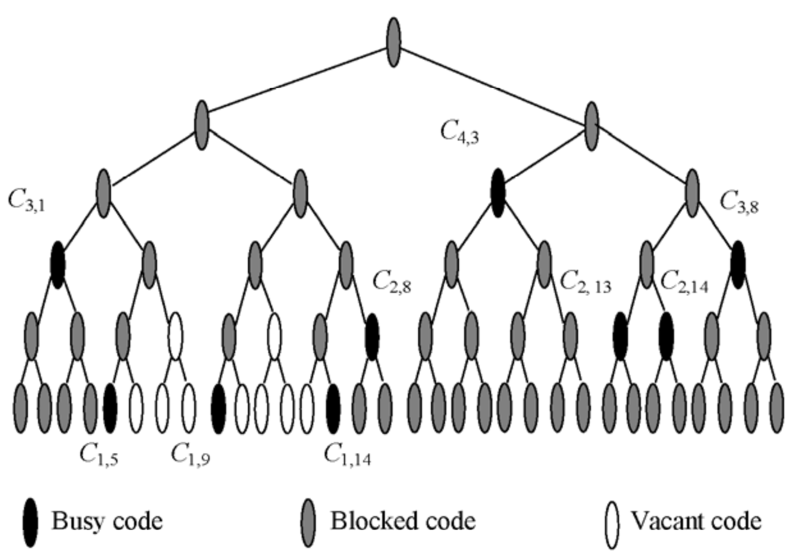


Figure 6.8: Example illustrating call duration reassignments

code of same rate exists. The two possible situations are:

- If $2^{l-1}R+UC \geq TC_{max}$, block the call. This is call blocking, as system don't have enough capacity to support new call.
- If $2^{l-1}R+UC \leq TC_{max}$. Then blocking in this situation is code blocking as code of rate is blocked due to any lower rate code $2^{m-1}R, l-1 \leq m \leq 1$. This condition can be prevented by reassignment of lower rate codes in area where they will not block additional codes which are not blocked earlier.

For reassignment, find all blocked codes which can be assigned to call of rate $2^{l-1}R$ after reassignments, choose the best candidate among them and reassign code(s) blocking it to other code(s). To choose best candidate blocked code with minimum call duration children to be reassigned. The reassignment scheme carried out following steps:

- List all blocked codes of rate $2^{l-1}R$ and choose C_{l,n_l} with $\min(T_{l,n_l}^{CD})$. For a tie pick the code with minimum capacity for reassignment.
- Find UC_{l,n_l},
- *if*

  $UC_{l,n_l} = 2^{l-1}R$, go to step1 and use second best candidate code.
- *elseif* $UC_{l,n_l} < 2^{l-1}R$ but $UC + UC_{l,n_l} > TC_{\max}$,

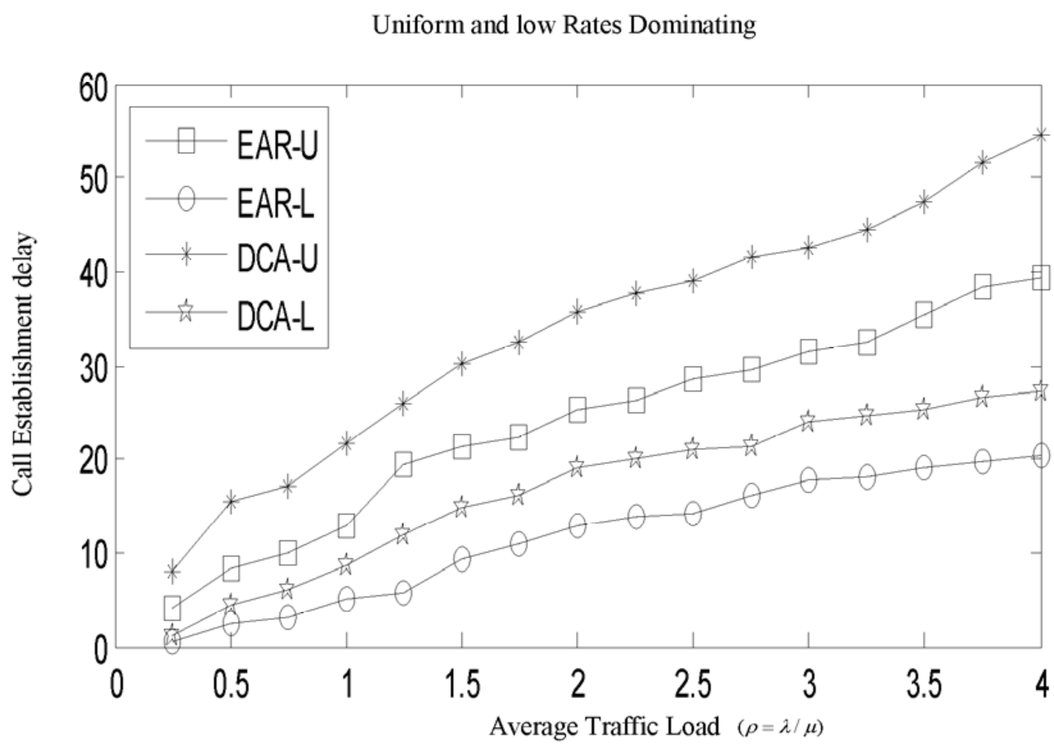
  go to step 1 and use second best candidate code.
- *else*

  The busy children code(s) of best candidate code are reassigned to that portion in code tree with no new code blocking or minimum new codes blocked due to reassignment as will be explained later. Assign new call to best candidate C_{l,n_l}.
- *end.*

Let children code of rate $2^{m-1}R$ is to be reassigned to a parent code of rate $2^{l-1}R$. This children code in our scheme will be reassigned to the code which will block minimum higher layer parent codes. The new code will be of layer m and first block code above it appears in layer m', than $m'-m$ is the number of higher layer parent codes blocked. The children code will be assigned to a code C_{m,n_m} with $\min(m'-m)$. This will lead to lesser code blocking in future. For a tie between codes pick code whose parent have minimum call duration. For illustration consider the status of code tree as shown in Figure.6.8 when a call of rate $4R$ arrive and duration of calls assigned to codes $C_{1,5}$, $C_{1,9}$, $C_{1,14}$, $C_{2,8}$, $C_{2,13}$, $C_{2,14}$, $C_{3,1}$, $C_{3,8}$ and $C_{4,3}$ is 1,5,3,4,6,8,1,6 and 9 time units respectively. There is a tie for minimum call duration between $C_{1,5}$ and $C_{3,1}$ as capacity of $C_{1,5}$ is minimum, this code is reassigned to vacant a code of rate $4R$. This code is assigned to $C_{1,10}$ whose parent code duration is minimum and assigning call to it will lead to minimum number of higher layer codes blocked *i.e* zero.

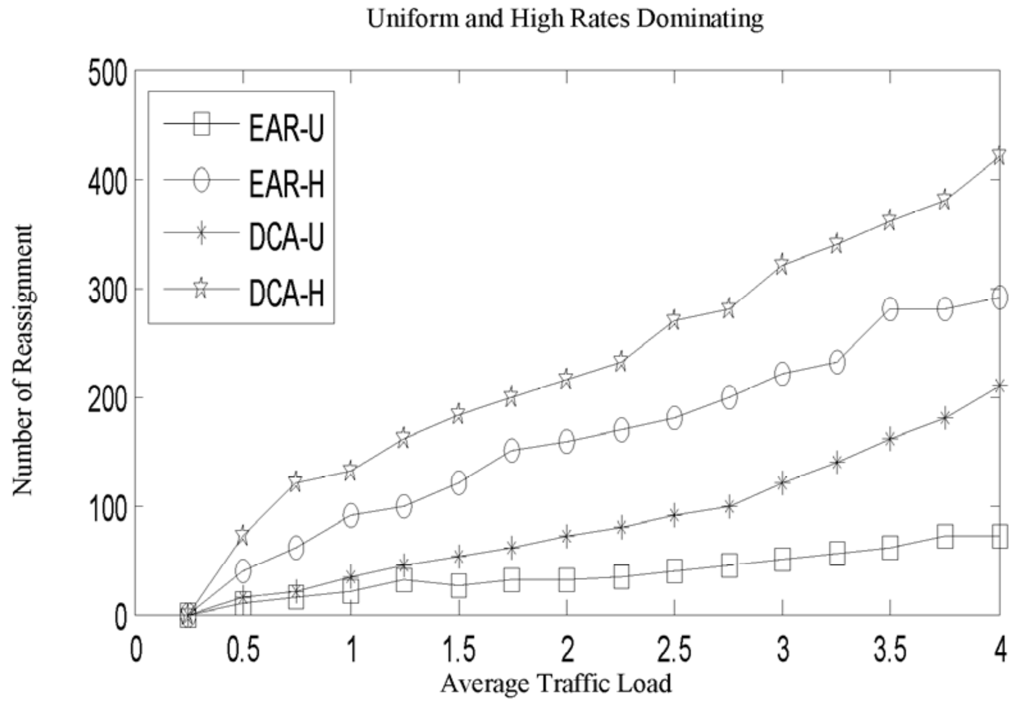
6.2.3 SIMULATION AND RESULTS

Following simulation parameters are considered for comparison of the proposed scheme with other schemes in literature. Call arrival process is Poisson with mean arrival rate λ=0-4 calls/units of time. Call duration is exponentially distributed with a mean value $1/\mu = 3$ units of time.

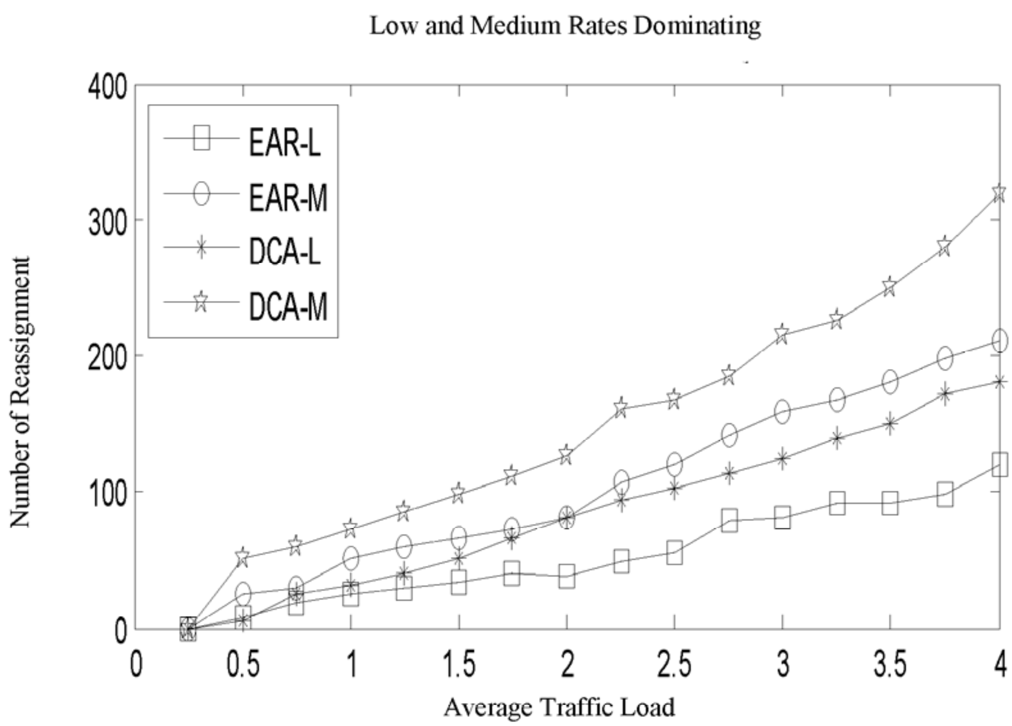


DCA [25]: Dynamic code assignment.

Figure 6.9: Comparison of call establishment delay for uniform and low rates dominating



(a)



(b)

DCA [25]: Dynamic code assignment.

Figure 6.10: Comparison of number of reassignments: (a) Uniform and high rates dominating, (b) Low and medium rates dominating.

- Possible arrival rates are R,$2R$, $4R$, $8R$ and $16R$
- The maximum capacity of the code tree is $128R$ with only five lower layer codes utilized by new users.

Define $\rho_i = \lambda_i / \mu_i$ as traffic load of the i^{th} class users. Also, for 5 class system the average arrival rate and average traffic load is $\lambda = \sum_{i=1}^{5} \lambda_i$ and $\rho_i = \sum_{i=1}^{5} (\lambda_i / \mu_i)$ respectively. In our simulation we consider call duration of all the calls equal *i.e.* $1/\mu = 1/\mu_i$.

The code blocking for a 5 class system is given by

$$P_B = \sum_{i=1}^{5} (\lambda_I P_{B_i} / \lambda) \tag{6.7}$$

where P_{B_i} is the code blocking of i^{th} class and is given by

$$P_{B_i} = (\rho_i^{G_i} / G_i!) / (\sum_{n=1}^{G_i} \rho_i^n / n!) \tag{6.8}$$

where $\rho_i = \lambda_i / \mu_i$ is the traffic load for i^{th} class. The total codes (servers) in the system for five set of classes are the given by set $G = \{G_1, G_2, G_3, G_4, G_5\}$. Define $[p_1, p_2, p_3, p_4, p_5]$ as probability distribution matrix where p_i, $i \in [1,5]$, is the capacity portion required by the four distribution scenarios are analyzed and are given by:

- [20,20,20,20,20], uniform distribution (U)
- [10,10,10,30,40], high rates calls dominating (H)
- [40,30,10,10,10], low rates calls dominating (L)
- [10,20,40,20,10], medium rates calls dominating (M)

The simulation is done for 10000 calls and result is the average of 10 simulations. The scheme represented as EAR in Figure 6.9 and Figure 6.10 is proposed scheme. It requires lesser reassignments as compared to DCA [25] scheme as shown in Figure 6.9. As the number of reassignments increases, call establishment increases in time units significantly as shown in Figure 6.10. The number of reassignment for DCA is almost twice as required by our scheme in all traffic load scenarios. The real time calls cannot tolerate higher call establishment delay, the proposed scheme takes the advantage of less call establishment delay over DCA with same amount of code blocking not shown as both DCA and our scheme provides zero code blocking.

6.2.4 CONCLUSION

3G and beyond generation networks will be handling bursty and time bound calls. Call establishment delay will be a significant parameter and scheme proposed in this section requires less call establishment delay and less complexity as compare to DCA[25]. Call establishment delay is less because this scheme searches minimum duration parent code which reduces complexity too. Additionally, code blocking is same as provided by DCA which is zero. Work can be done to investigate performance using multi codes.

6.3 IMMEDIATE NEIGHBOR ASSIGNMENT SCHEME

6.3.1 SINGLE CODE

Consider an L (L=8 in WCDMA) layer OVSF code tree. If a new call with rate $2^{l-1}R$ arrives,

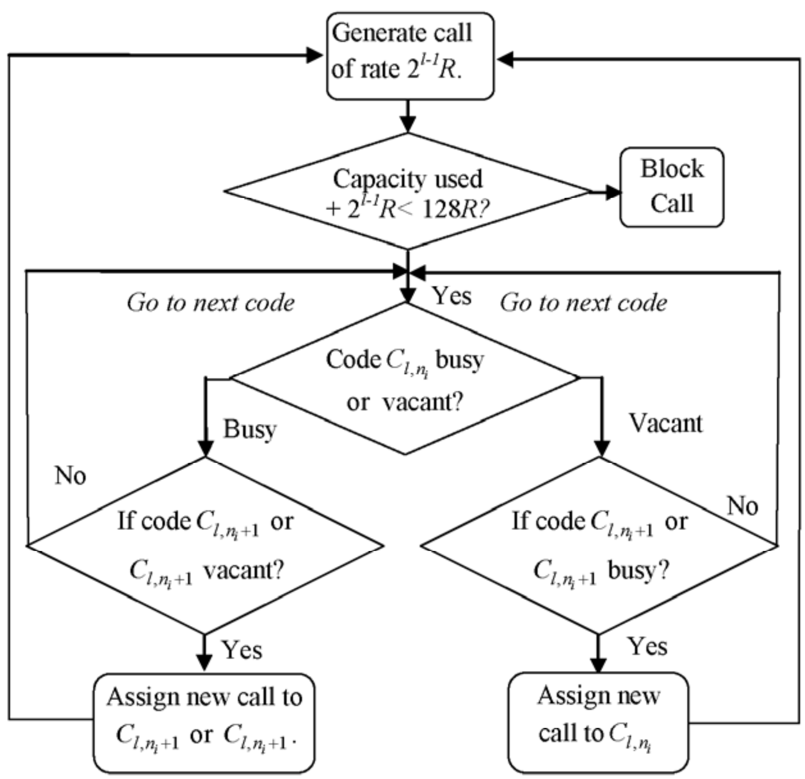


Figure 6.11: Flowchart for Immediate Neighbor design.

the algorithm lists all the vacant codes in layer l till it find a vacant code whose immediate sibling (parent's another child) is busy.

Let C_{l,n_i} denotes the i^{th} vacant code in layer l. Let there are m vacant codes in layer l, represented by $C_{l,n_i}, 1 \le n_i \le 2^{L-l}$ and $1 \le i \le m$. For a vacant code C_{l,n_i}, the immediate neighbor is

C_{l,n_i-1} if n_i mod2=0

C_{l,n_i-1} if $n_i \bmod 2 \neq 0$ (6.9)

If a new call seeking a code from layer l arrives, for each vacant code C_{l,n_i}, check immediate neighbor (vacant code and its neighbor should have same 1st parent) C_{l,n_i+1} or C_{l,n_i-1} (say $C_{l,z}$) according to Equation 6.9. If code $C_{l,z}$ is busy then assign new call to vacant code C_{l,n_i}. Otherwise, check status of next vacant code and its immediate neighbor. The scheme aims to utilize full capacity under its parent(s) codes. In our proposed scheme, comparison with other vacant codes is not carried out as was done in CFA [22] and RFCB [33], as it assigns new call to code C_{l,n_i}, if its immediate neighbor $C_{l,z}$ is busy which reduces complexity and call assignment delay. In worst case if algorithm is not able to find a vacant code whose immediate neighbor is busy then it assigns call to left most vacant code. The fundamental idea is to assign those codes for new call which are close to existing busy codes. This avoids scattering of higher layer vacant (or busy) codes in the code tree which leads to optimum compact code assignment. The flow chart of algorithm is given in Figure 6.11. The proposed scheme gives comparable blocking performance to CFA scheme but it reduces the number of code searches significantly as discussed below.

6.3.2 CODE SEARCHES CALCULATION

This portion find the number of code searches required before suitable code assignment to new call in the proposed scheme and crowded first assignment (CFA) scheme. Let λ_l denotes arrival rate of l^{th} class user.

A. CROWDED FIRST ASSIGNMENT

The algorithm for finding the maximum number of code searches is explained below. For new call with rate $2^{l-1}R$ seeking a code from layer l, number of codes searched in layer l is

$$CFA_l^1 = 2^{L-l} \tag{6.10}$$

For an active system, if there are m_1 vacant codes in layer l, for each C_{l,n_i}, where $1 \leq i \leq m_1$ and $1 \leq n_i \leq 2^{L-l}$, find the number of busy codes under first parent of C_{l,n_i} in layer l+1 (*i.e* code $C_{l+1,\lfloor n_i/2 \rfloor}$). The number of code search for m_1 vacant codes in layer l+1 is given by

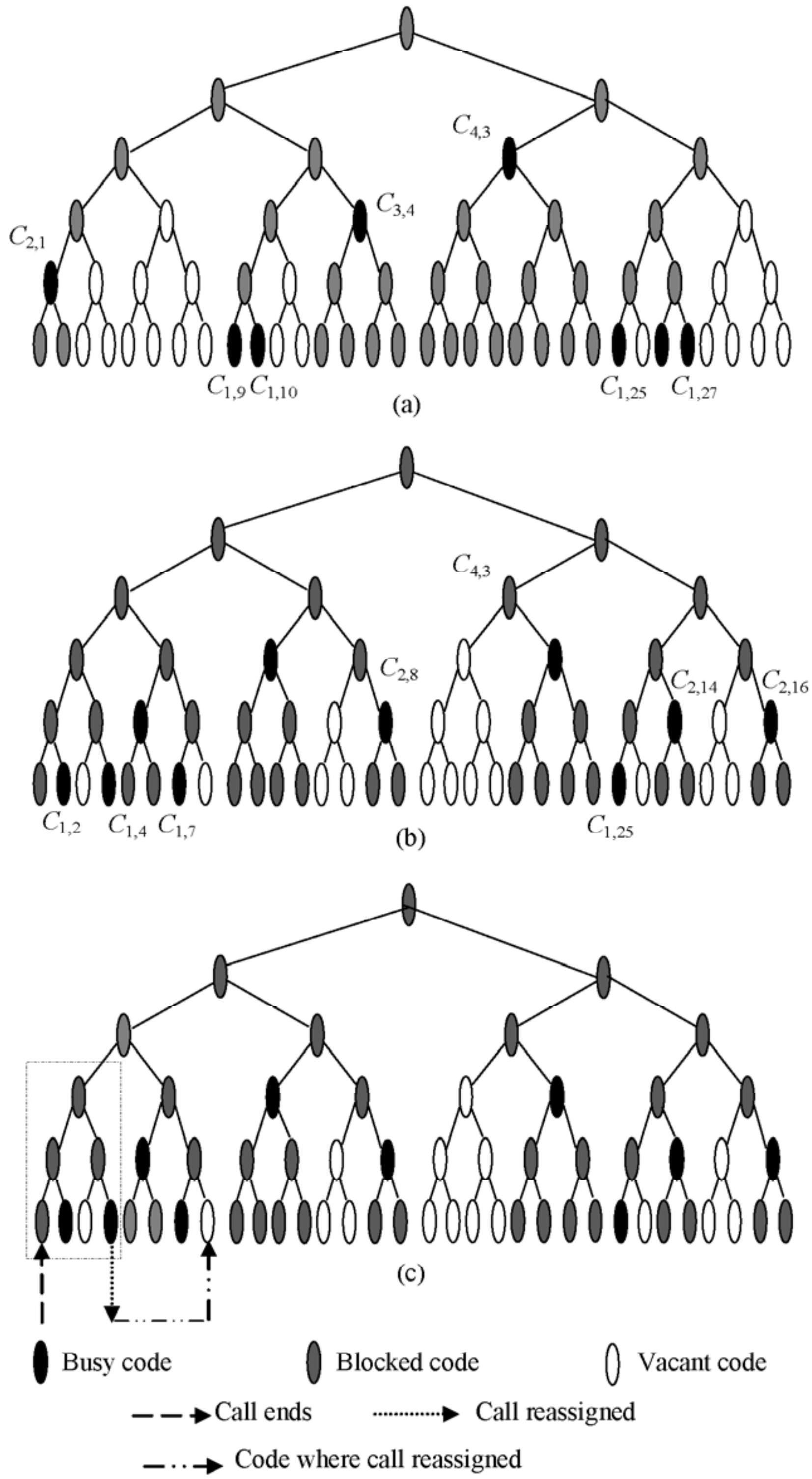


Figure 6.12: Illustration of Immediate parent code assignment (a) Single code, (b) Multi code without reassignments, and (c) Multi code with reassignments.

$$CFA_l^2 = m_1 \times 2^l \qquad (6.11)$$

If code searches in layer l is able to find a unique result *i.e.* a single parent code with maximum number of busy children than algorithm stops. Otherwise it will go to layer l+2. If m_1 parent codes $C_{l+1,\lfloor n_i/2 \rfloor}$ have a tie than check $C_{l+1,\lfloor n_i/2^2 \rfloor}$ parent codes in layer l+2 and searches are

$$CFA_l^3 = m_2 \times CFA_l^2 \times 2^{l+1}) \qquad (6.12)$$

If a unique result does not exist the algorithm can go up to layer L-1. The maximum number of code searched can be formulated as

$$CFA_l^{Total} = \sum_{l=1}^{L} \lambda_l [CFA_l^1 + CFA_l^2 + ... + CFA_l^{L-1}] \qquad (6.13)$$

$$CFA_l^{Total} = \sum_{l=1}^{L} \lambda_l [2^{L-l} + m_1 \times 2^{l+1} + m_2 (m_1 \times 2^l \times 2^{l+1}) ....] \qquad (6.14)$$

B. IMMEDIATE NEIGHBOR ASSIGNMENT

Let identifier n_{i_l,k_l}^l denotes the k_l^{th} vacant code for i_l^{th} new call seeking vacant code in layer l. Also, assume that the first vacant code with busy neighbor is n_{i_l,k_l}^l. If there are N_l total calls corresponding to layer l in the system, the number of codes need to be searched prior to call assignment can be formulated for layer l

$$IMM^l = \sum_{l=1}^{N_l} [n_{i_l,k_l}^l + 1] \qquad (6 .15)$$

The total number of codes searched in the L layer system can be represented as

$$IMM^{total} = \sum_{l=1}^{L} \sum_{l=1}^{N_l} [n_{i_l,k_l}^l + 1] \qquad (6.16)$$

If no vacant code exists with immediate neighbor busy then for a call rate of $2^{l-1}R$, the number of code searches is 2^{L-l} and new call is assigned to left most vacant code and number of code searches for worst case is:

$$IMM_{Worst}^l = 2^{L-l} \qquad (6.17)$$

6.3.3 MULTI CODE EXTENSION

Using multi codes a call can be handled with or without reassignments. However, with reassignments a call can be handled using minimum number of rakes.

A. MULTI CODE ASSIGNMENT WITHOUT REASSIGNMENTS

For an incoming call request of rate $2^{l-1}R$ and free capacity of code tree is $\geq 2^{l-1}R$.

1. Divide $2^{l-1}R$ into two $2^{l-2}R$ rate fractions. Search a vacant code of rate $2^{l-2}R$ whose immediate parent code assigned capacity is $2^{l-2}R$ and store the information of vacant codes whose immediate parent code assigned capacity is less than $2^{l-2}R$ in increasing order of assigned capacity. They will be used when no immediate parent code exists with half capacity assigned.
2. If a vacant code $C_{l-1,n_{l-1}}$ of rate $2^{l-2}R$ available satisfying condition 1, assign one the rate fraction $2^{l-2}R$ to it and repeat step1 for codes in right of $C_{l-1,n_{l-1}}$. It will reduce call establishment delay which is proportional to number of code searches.
3. Else if

   Assign one or both $2^{l-2}R$ rate fractions to vacant codes whose immediate parents have maximum assigned capacity.
4. Else

   Divide one or both $2^{l-2}R$, whichever not assigned to a vacant code into smaller rate fractions and repeat step 1.
5. End

For illustration consider status of six layer code tree in Figure 6.12(b). If a call of rate $8R$ arrives and is not handled by single code assignment, it will be divided into two $4R$ rates and vacant codes of $4R$ rates are searched. One vacant code of $4R$ rate is $C_{3,5}$ and searching for another vacant code of $4R$ will start from $C_{3,5}$ as compare to other schemes which starts from left side again. Second $4R$ is further divided into two $2R$ rates as no other $4R$ vacant code is available and are assigned to $C_{2,7}$ and $C_{2,15}$. It requires 3 rakes to handle this call.

B. MULTI CODE ASSIGNMENT WITH REASSIGNMENTS

To resolve the problem of code blocking, code reassignment can be conducted to squeeze enough space required for the new call. Reassignments of calls increases cost of the system with an advantage of reduced code blocking. However, due to small code tree reassignment cost can be ignored. When a call of rate $2^{l-1}R$ arrives and call is handled without reassignment using multi code will require more number of rakes to handle the call. Therefore, we proposed a multi code immediate reassignment and assignment scheme which reassign scattered capacity of code tree and after reassignment call is handle using multi codes. The reassignment is carried out when a call ends *i.e* offline for most of the times. The algorithm of reassignment work as follows:

1. When a call of $2^{m-1}R$ ends & releases C_{m,n_m}.
2. Find 1st block code $C_{m',n_{m'}}$ above C_{m,n_m} in layer m'.
3. The reassignment of all busy codes under $C_{m',n_{m'}}$ is done as releasing code C_{m,n_m} of rate $2^{m-1}R$ will lead to scattering in sub tree under code $C_{m',n_{m'}}$.
4. Find all busy codes of rate $2^{k-1}R, 1 \le k \le m'-1$ and reassign them using single code immediate assignment in section 6.3.1.

The multi code immediate assignment will assign call using minimum rakes as explained in section 6.3.3 and will lead to minimum scattering when a call utilizing multi codes ends. For illustration consider status of six layer code tree in Figure 6.12(c). If call assigned to $C_{1,1}$ ends before $8R$ rate call arrives and 1st blocked above $C_{1,1}$ is $C_{3,1}$, then busy codes under $C_{3,1}$ *i.e* $C_{1,2}$ will be reassigned to code $C_{1,8}$ satisfying algorithm in section 6.3.1. It will vacant sub tree under $C_{3,1}$ and $8R$ rate call will be handled using two rakes using codes $C_{3,1}$ and $C_{3,5}$.

6.3.4 SIMULATION AND RESULTS

A. SIMULATION PARAMETERS

- There are 5 classes of users with rates R, $2R$, $4R$, $8R$, $16R$.
- Arrival rate is Poisson distributed with mean value varying from 0-4 calls/units of time.
- Call duration is exponentially distributed with mean value $1/\mu_l = 3$ units of time.
- The maximum capacity of the tree is $128R$ (R is $15kbps$).
- Simulation is done for 10000 users and result is average of ten simulations.

B. RESULTS

The single code scheme (IMM) is compared with Fixed Set Partitioning [27] (FSP), Crowded First Assignment[22] (CFA), Leftmost Code Assignment [22] (LCA), Dynamic Code Assignment [25] (DCA) and Recursive Fewer Code Blocked [33] (RFCB) schemes and multi code scheme with reassignments (MCIMM-W) and without reassignments (MCIMM) is compared with multi code CFA(MC-CFA), multi code LCA(MC-LCA) and multi code FSP (MC-FSP). The parameters compared are code searches and code blocking.

Define [p_1,p_2,p_3,p_4,p_5] as probability distribution matrix where p_i , $i \in [1,5]$ is the capacity portion used by the i^{th} class users. Consider four distribution scenarios for single code given by

- [10, 15, 25, 25, 25]; high rates dominating.
- [20, 20, 20, 20, 20]; uniform distribution.
- [25,25,25,15,10]; low rates dominating-I.
- [35,30,15,10,10]; low rates dominating-II.

and two multi code distribution scenarios given by

- [20, 20, 20, 20, 20]; uniform distribution.
- [40,30,10,10,10]; low rates dominating.

C. CODE BLOCKING

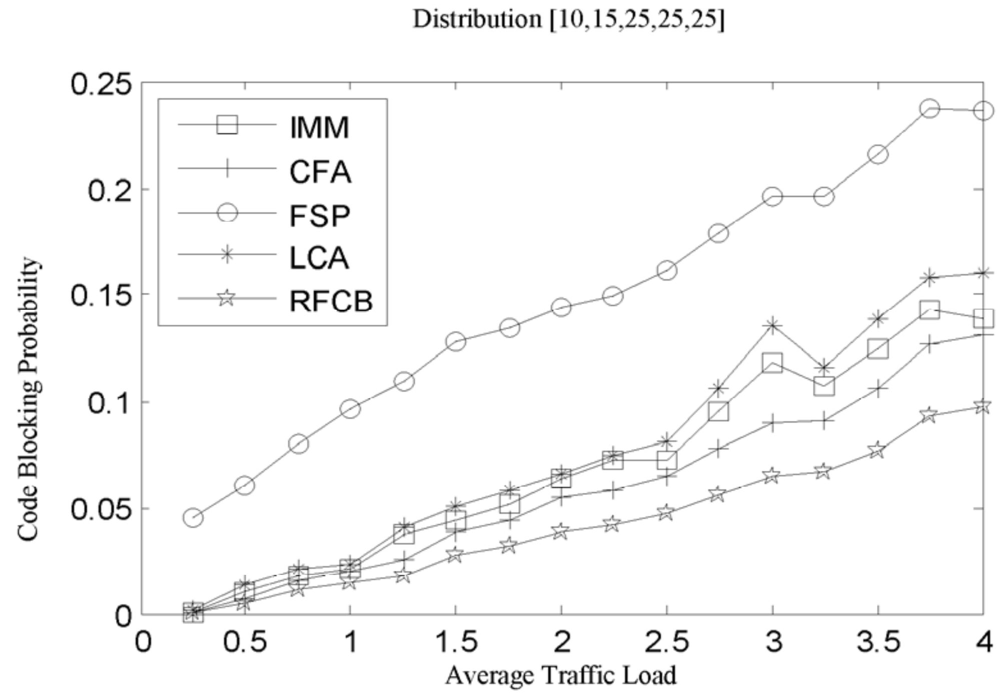
The code blocking for 5 class system is given by

$$P_B = \sum_{i=1}^{5} (\lambda_i P_{B_i} / \lambda) \tag{6.18}$$

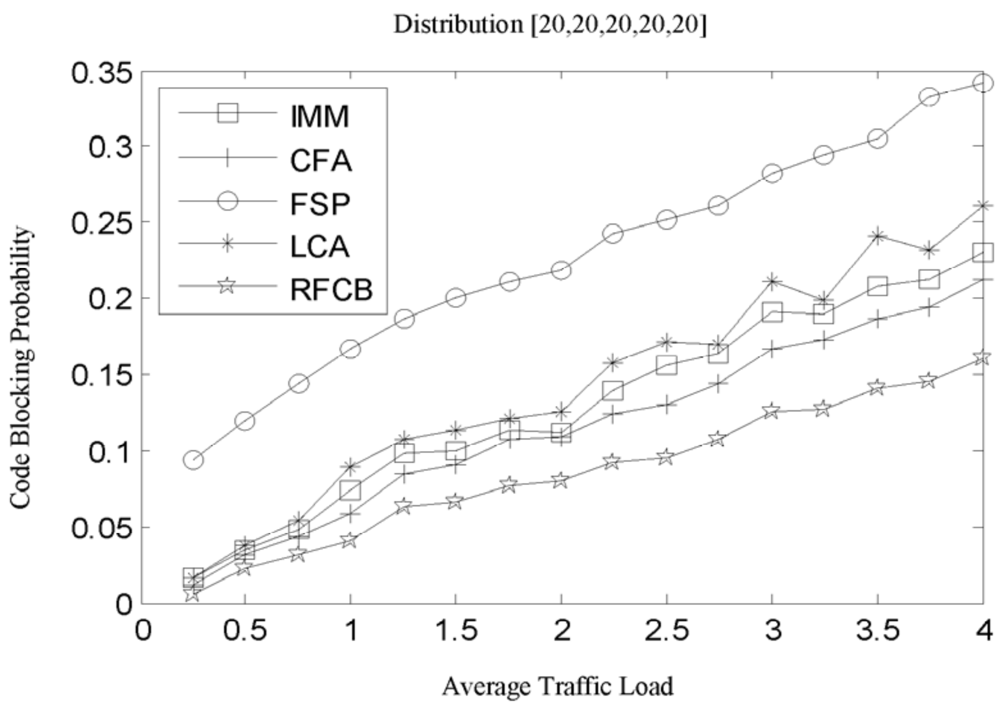
where λ_i is the arrival rate for i^{th} class users and $\lambda = \sum_{i=1}^{5} \lambda_i / 5$ is the average traffic load. P_{B_i} is the code blocking of i^{th} class and is given by

$$P_{B_i} = (\rho_i^{G_i} / G_i!) / \sum_{n=1}^{G_i} \rho_i^n / n! \tag{6.19}$$

where $\rho_i = \lambda_i / \mu_i$ is the traffic load for i^{th} class and G_i is the number of codes (servers) in the i^{th} class. Also, define $\rho = \sum_{i=1}^{5} \rho_i / 5$ as average traffic load. The code blocking comparison is given in Figure 6.13 and Figure 6.14. The results shows that code blocking in the proposed scheme is higher than CFA and RFCB schemes and is superior then FSP and LCA in all distributions. The small code blocking degradation in IMM scheme can be tolerated for significant reduction in codes searched. The average code blocking and fewer code search makes proposed IMM scheme attractive for wireless networks. The code blocking comparison is not shown for DCA as it always produce zero code blocking (at the expense of higher reassignment cost). The code blocking for all multi code schemes decreases and is minimum for MCIMM-W as it tries to utilize total capacity under a parent code as shown in Figure 6.15



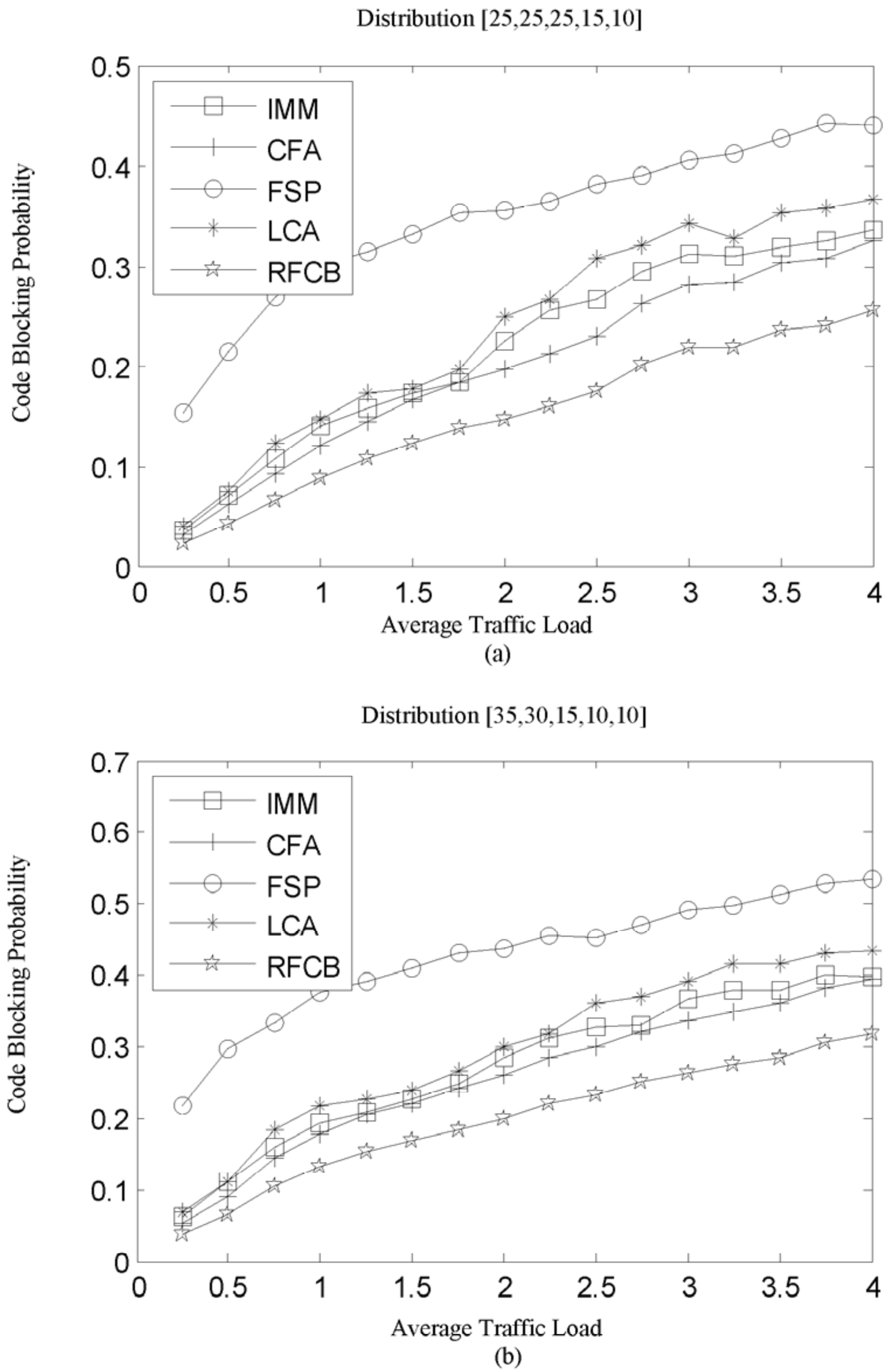
(a)



(b)

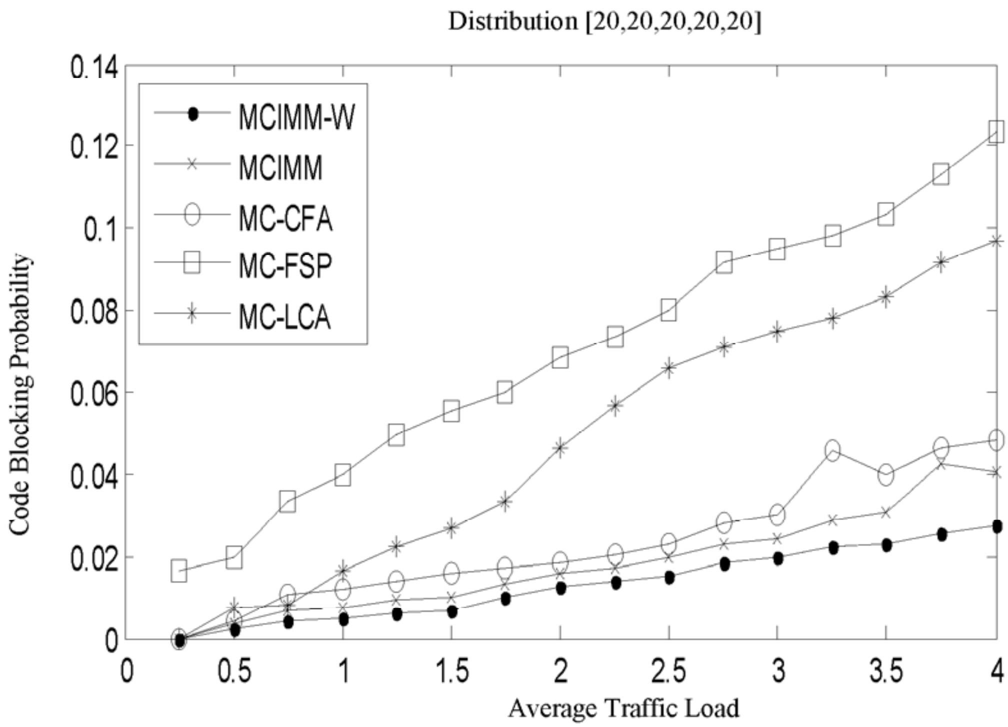
IMM: Immediate neighbor, CFA [22]: Crowded first assignment, FSP [27]: Fixed set partitioning, LCA [22]: Left code assignment, RFCB [33]: Recursive fewer codes blocked.

Figure 6.13: Comparison of Code blocking probability for distribution: (a) [10,15,25,25,25], (b) [20,20,20,20,20].

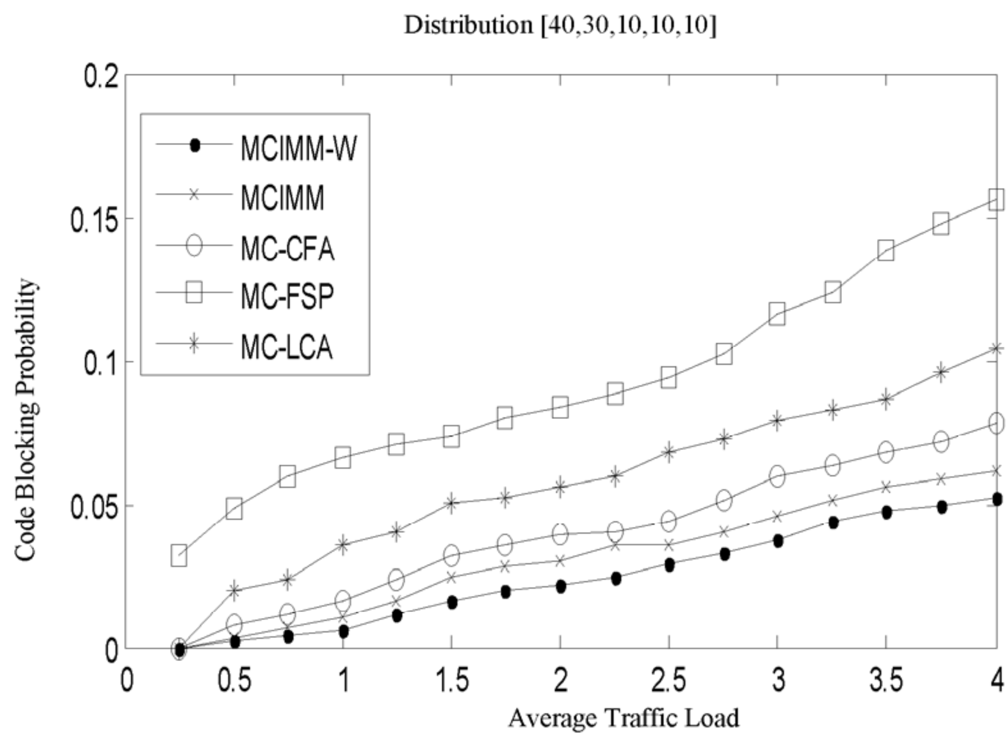


IMM: Immediate neighbor, CFA [22]: Crowded first assignment, FSP [27]: Fixed set partitioning, LCA [22]: Left code assignment, RFCB [33]: Recursive fewer codes blocked.

Figure 6.14: Comparison of Code Blocking Probability for distribution: (a) [25,25,25,15,10], (b) [35,30,15,10,10]



(a)

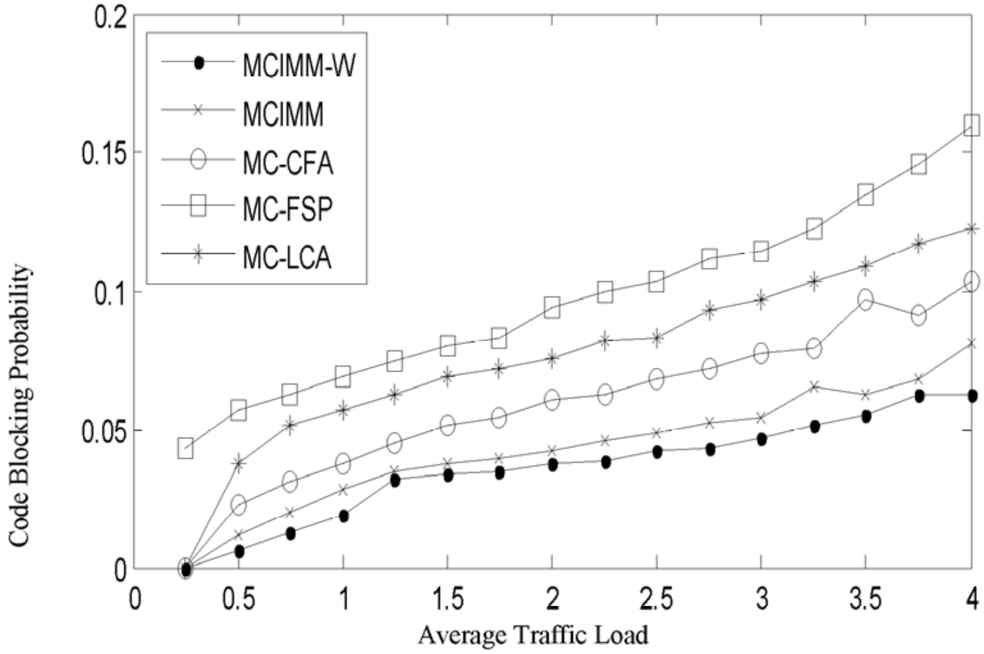


(b)

MC: Multi Code, IMM: Immediate neighbor, CFA [22]: Crowded first assignment, FSP [27]: Fixed set partitioning, LCA [22]: Left code assignment.

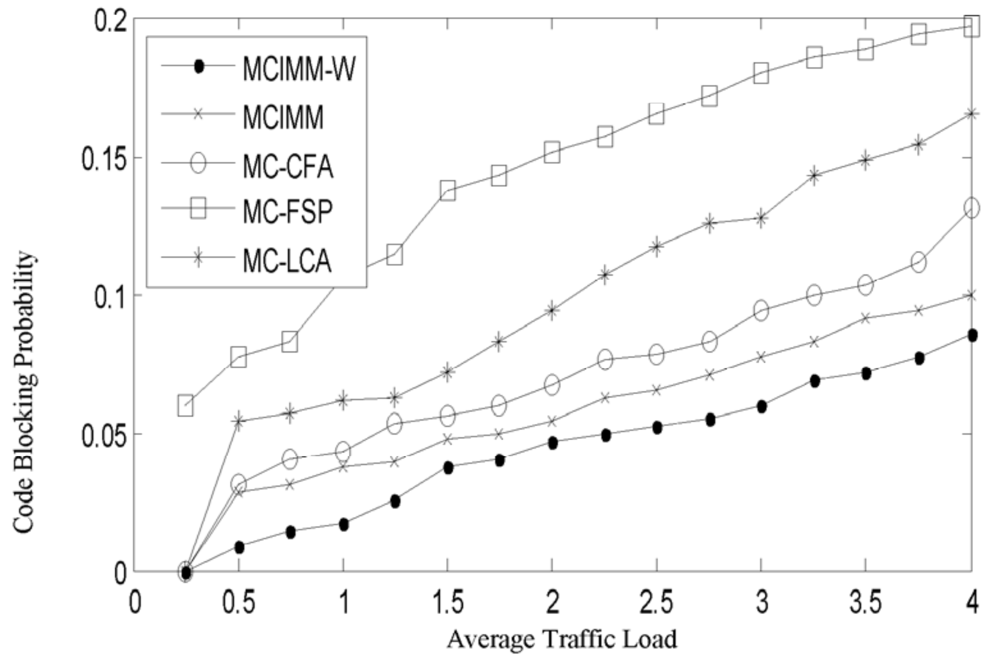
Figure 6.15: Comparison of Code Blocking Probability for distribution: (a) [20,20,20,20,20], (b) [40,30,10,10,10]

Distribution [10,10,10,30,40]



(a)

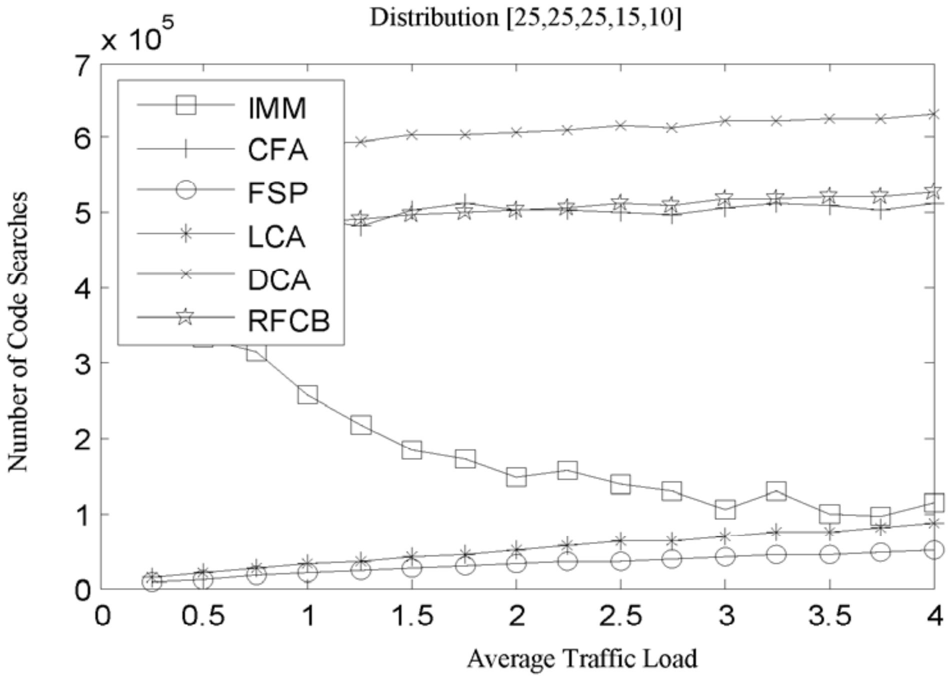
Distribution [10,30,20,30,10]



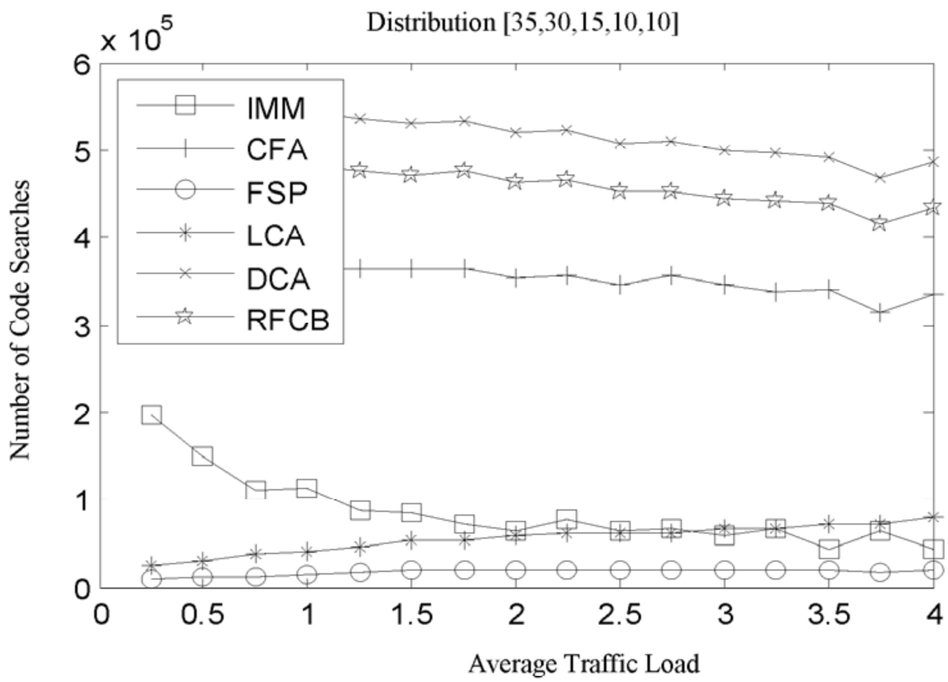
(b)

MC: Multi Code, IMM: Immediate neighbor, CFA [22]: Crowded first assignment, FSP [27]: Fixed set partitioning, LCA [22]: Left code assignment.

Figure 6.16: Comparison of Code Blocking Probability for distribution: (a) [10,10,10,30,40], (b) [10,30,20,30,10]



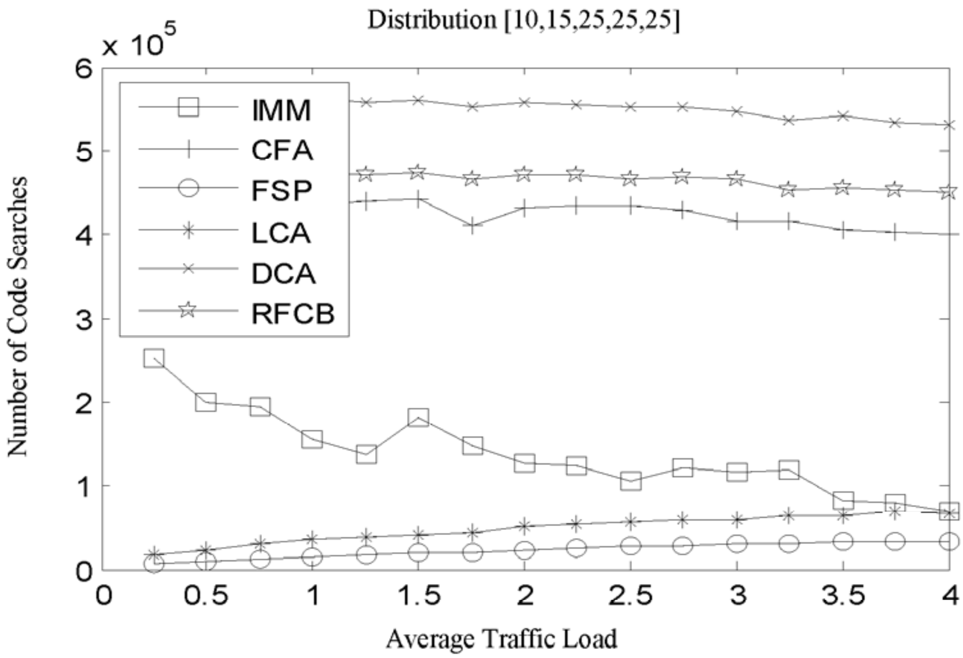
(a)



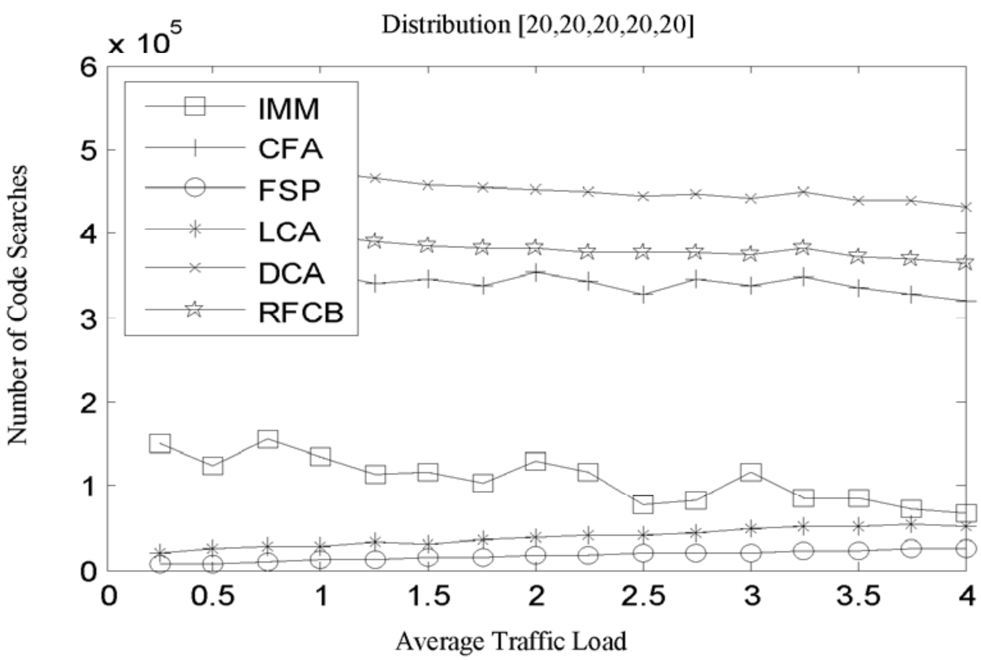
(b)

IMM: Immediate neighbor, CFA[22]: Crowded first assignment, FSP[27]: Fixed set partitioning, LCA[22]: Left code assignment, RFCB[33]: Recursive fewer codes blocked, DCA[25]: Dynamic code assignment

Figure 6.17: Comparison of number of code searches for distribution: (a) [25,25,25,15,10], (b) [35,30,15,10,10]



(a)



(b)

IMM: Immediate neighbor, CFA[22]: Crowded first assignment, FSP[27]: Fixed set partitioning, LCA[22]: Left code assignment, RFCB[33]: Recursive fewer codes blocked, DCA[25]: Dynamic code assignment

Figure 6.18: Comparison of number of code searches for distribution: (a) [10,15,25,25,25], (b) [20,20,20,20,20]

and Figure 6.16. MCIMM-W provides minimum code blocking with additional advantage of using minimum rakes for handling the call.

D. CODE SEARCHES

The code searches comparison for above 2 distributions is given in Figure 6.17 and Figure 6.18. The results shows that the proposed scheme searches lesser number of codes before assigning a code to a call compared to CFA, DCA, and RFCB schemes. FSP and LCA require lesser number of codes to be searched (at the cost of higher code blocking). The multi code scheme searches till first occurrence of vacant code and does not require total code tree to be searched and due to this reason number of code searches will be less as compared to other schemes. The number of code searches leads to delay known as call establishment delay which is directly proportional to number of code searched before handling of a new call.

6.3.5 CONCLUSION

Call processing delay is significant performance parameter for 3G and beyond wireless networks. The requirement becomes more stringent for real time calls. The proposed single code scheme outperforms most of the existing single code schemes for call processing delay (codes searched). The code blocking may not be best for the proposed scheme but simplicity, cost effective features make it suitable for current wireless networks. The multi code scheme provides minimum code blocking with lesser call establishment delay. Also, the numbers of rakes used are fewer which reduce complexity.

6.4 FLEXIBLE ASSIGNMENT SCHEME FOR DATA CALLS

In this section, the code tree capacity is partitioned to handle lower rate calls and higher rate calls separately. This reduces complexity and amount of code searches. Further, the data calls are given flexibility for rate variation according to the vacant codes availability. The scheme focuses on the type of call to be served and rate request. As arrival and termination of calls is random, both low rates and high rate codes may be requested at the same time. Therefore, low rates and high rates calls are handled in different portions of the code tree.

Consider an L layer WCDMA OVSF code tree with call rate $2^{l-1}R$ for a layer l, $1 \le l \le L$. Both data calls and time bound calls can take rates from R to $2^{L-1}R$. If a low rates code in layer

l of WCDMA code tree is assigned to a new call it will block $\max(L-l)$ numbers of higher rate codes which are in non blocking state earlier. If first blocking code appears in layer l' above layer l then number of higher layer codes blocked by l layer code is $(l'-l)$. In the proposed scheme, the ongoing data call facilitates handling of incoming time bound call with requested rate $2^{l-1}R$ in two ways.

1. If there is no vacant code in any layer, the ongoing data call with rate more the $2^{l-1}R$ data call is put in the queue and the code carrying data call is assigned to the time bound call. If there are multiple such data calls ongoing, the optimum data call to be queued is the one with maximum data rate. If in future, a call completes, the released code can handle the queued data call. For multiple data calls in queue, the one with the minimum elapsed time is the candidate to use the vacant code. Further, for a particular data call, there may be many such rate fluctuations.
2. If there are non zero vacant codes in layer $l', l' < l$, the data call in layer l is shifted to the optimum vacant code, where the optimum code is the one which occurs in $l'' \mid l'' = \max(l'), l' < l$. The released code is given to time bound call.
3. For a particular data call with requested rate $2^{l'-1}R$, let the n rate fluctuations are $2^{l_i-1}R, 1 \le i \le L$. If t_i denotes the time elapsed while transmitting at rate $2^{l_i-1}R$, the total call duration becomes

$$\Sigma_{i=1}^{n}(2^{l-1}/2^{l_i-1}) \times t_i + t_{off} \tag{6.22}$$

where t_{off} is the time when the call is in queue. If the data call duration without any rate fluctuation is t, the overall data call rate is more than requested rate if

$$\Sigma_{i=1}^{n}(2^{l-1}/2^{l_i-1}) \times t_i + t_{off} < t \tag{6.23}$$

If (6.23) is not true the overall data call rate is reduced.

To avoid blocking of higher rate calls due to low rates calls, tree is partitioned in two parts. The first part handles low rate calls (time bound or data) and second part handles high rates calls. Define data capacity index $D_i = [R_{L-1}^i, R_{L-2}^i, ...R_1^i]$, where R_l^i is the amount of data calls handled in layer l in i^{th} portion $i \in [1,2]$. The index D_i is a measure of number of data calls queued due to ongoing time bound calls. Depending upon the type of call arrival, the description of the scheme is as follows.

6.4.1 TIME BOUND CALLS

A. LOW RATES CALLS

For a time bound call request of rate $2^{l-1}R$, where $1 \le l \le (L/2)$, the optimum code selection will start from first portion of the code tree. The optimum code search algorithm is described as follows.

- Search all codes with capacity $2^{l-1}R$ with code id ranging from $C_{l,1}$ to C_{l,n_i}, where $1 \le n_i \le 2^{L-l-1}$.
- If there are more than one vacant codes of rate $2^{l-1}R$, assign code to the most crowded portion (considering only time bound calls) of the code tree, where the crowded portion is the one with maximum number of codes already assigned to time bound applications.
- If no vacant codes of rate $2^{l-1}R$ available, assign new call to the optimum code already assigned to data calls with rate at least $2^{l-1}R$. The data call is either put in queue or assigned lower rate code depending upon the availability.
- If there is no vacant code with rate $2^{l-1}R$ in 1st portion and if status of $D_i \ne 0$ in 2nd portion with $\sum_{j=1}^{L-1} R_j^i \ge 2^{L-2}$, search a code assigned to data calls in 2nd portion. The search is started from right side of 2nd portion as probability of finding such code on right side is higher. This reduces code searches significantly. Release the code assigned to data calls, which is now assigned to time bound call. Again the data call is queued or assigned lower rate code. The scheme periodically search vacant code in 1st portion in offline mode and on availability the low rate time bound call from 2nd portion is shifted to 1st portion. The data call queued in buffer previously can now use the vacant code in 2nd portion.

B. HIGH RATES CALLS

For a time bound call request of rate $2^{l-1}R$, where, $(L/2+1) \le l \le (L-1)$ the optimum code selection is done from right half of the 1st portion. To select an optimum code, following algorithm is followed

1. Search all codes with rate $2^{l-1}R$ in right half of the 1st portion, with code id ranging from $C_{l,k}$ to C_{l,n_i} where, $k \le n_i \le 2^{L-l}$ and $k = 2^{L-l-1} + 1$.

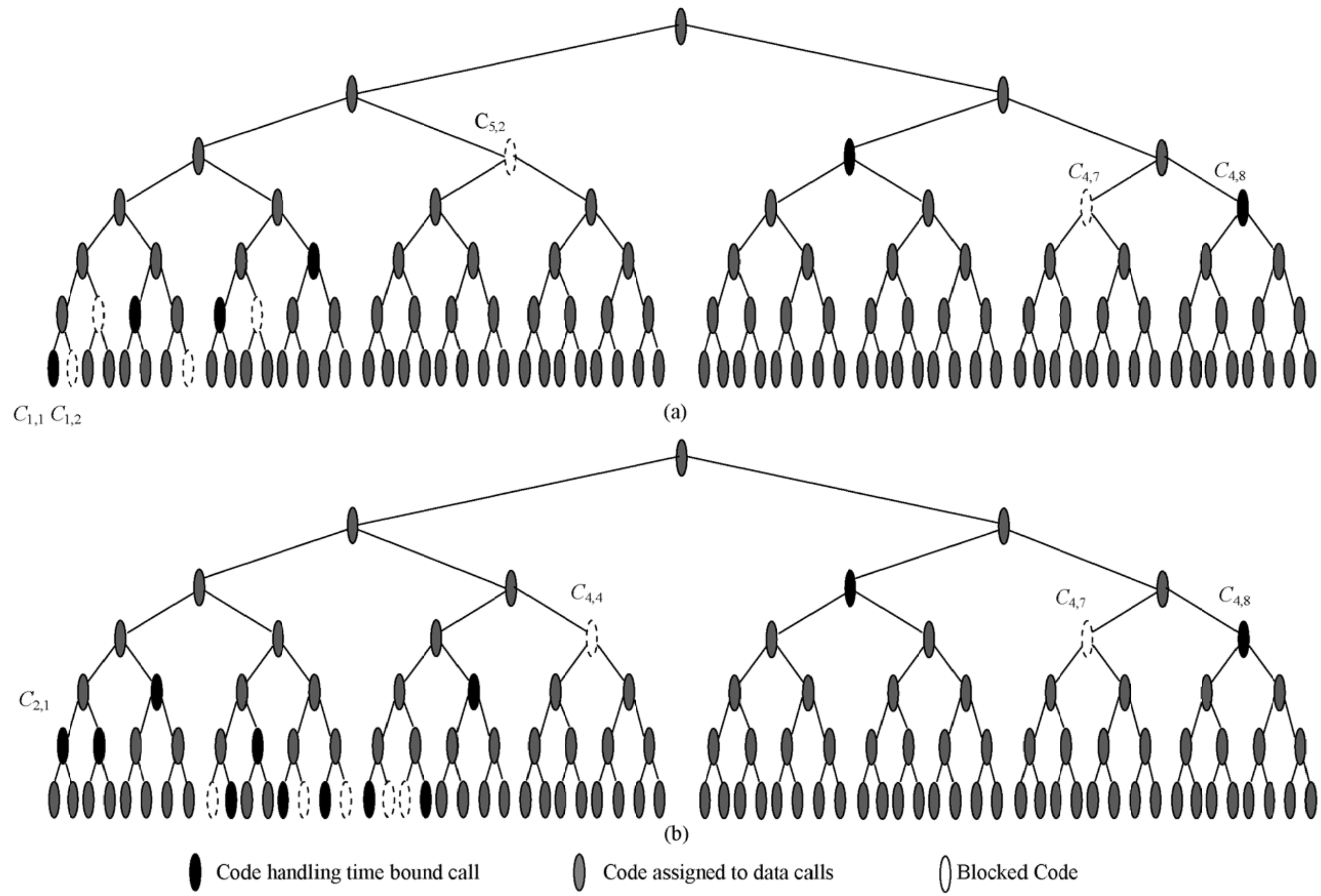
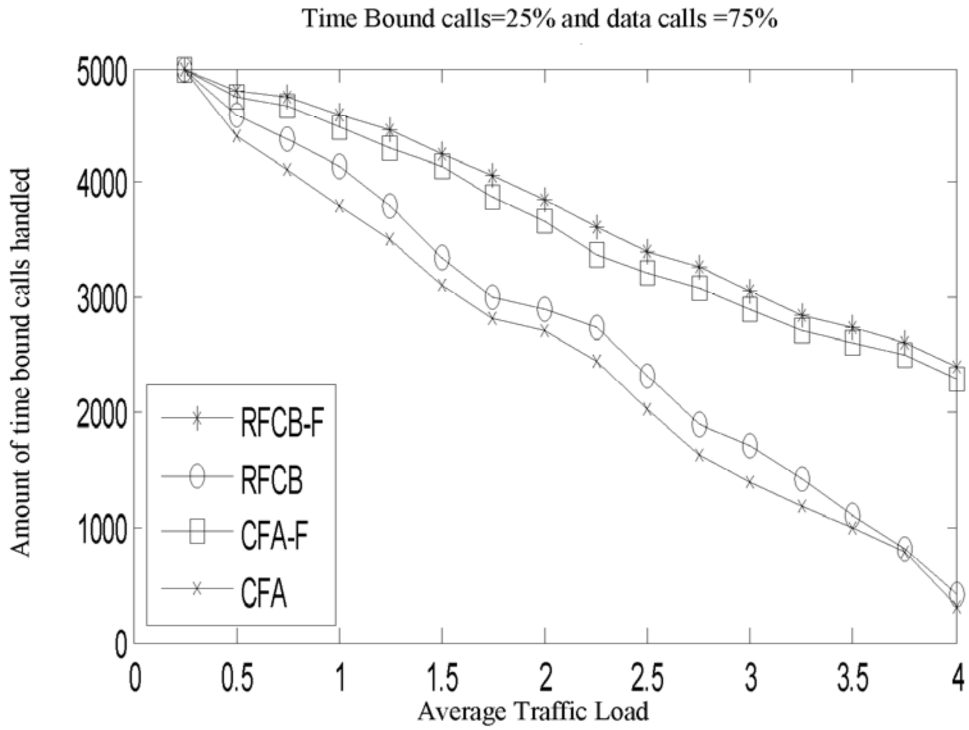


Figure 6.19: Illustration of the flexible assignment design

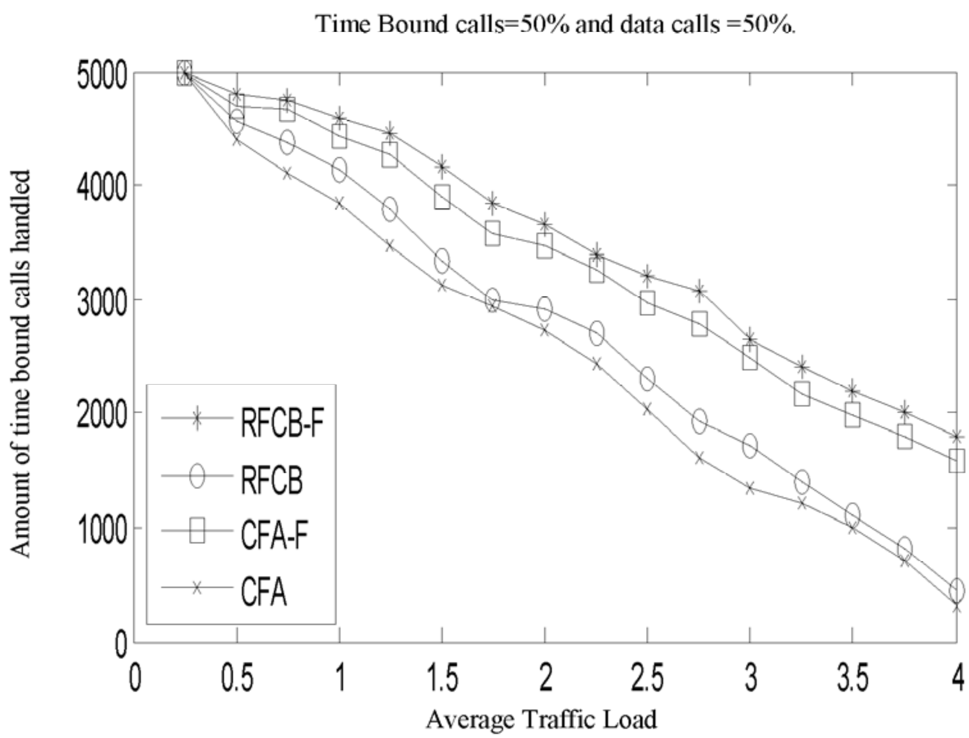
2. If there are more than one vacant codes of rate $2^{l-1}R$ assign code to most crowded area of the 1st portion.
3. If no vacant codes of rate $2^{l-1}R$ available, assign new call to the code already assigned to data calls. Put data call in queue or assign it a lower rate code depending upon the code tree status.
4. If all above conditions do not provide a vacant code, D_i in the 1st portion. If $D_i \neq 0, (L/2+1) \leq i \leq L-1$ start searching codes assigned to data call from right side of 1st portion. This reduces code searches and release code blocking or utilizing call of rate $2^{l-1}R$ and assigns this code to new time bound call. In offline mode search 2nd portion for a vacant code and does code shifting when a vacancy is available.
5. If there is no code handling data call in 1st portion, the call is rejected.

C. DATA CALLS

If a new data call with rate $2^{l-1}R$ arrives, a vacant code is searched in portion 1 and 2. If a vacant code is available, the call is accepted and assign to it. Considering that the vacant code is available in a lower layer, $l', l' < l$, the data call rate is reduced and the vacant code is assigned to the data call. Otherwise, the call is put in the queue. At a specific time there may be number of data calls whose rate is reduced or which are waiting in the queue. If we define T_Q as the elapsed time of a data call, for multiple data calls with reduced rates or waiting in queue, when vacant code become available (due to some call completion), the code with maximum elapsed time is the candidate to get vacant code. The code tree utilization becomes close to 100%. The variation in data call rates and storage in queue adds complexity to the code assignment scheme. In general, the rate fluctuations depend upon the traffic load. For low to medium traffic loads, the data call rates need not to be reduced, and the assignment is simple. For high traffic load conditions the data rate fluctuations are incorporated, and specifically to get 100% utilization, there should be the provision for infinite rate fluctuations. Consider the status of code tree in Figure 6.19 (a) with $D_1 = [0, 16R, 0, 0, 4R, 2R]$ and $D_2 = [0, 0, 8R, 0, 0, 0]$. It is assumed that the rates R, $2R$, $4R$ use portion 1 and rates $8R$, $16R$, $32R$ use portion 2. If a new time bound call with rate $16R$ arrives, the algorithm search 2nd portion of the code tree. As vacant code (or codes assigned to data calls) of rate $16R$ is not available, and $R_l^1 = 0, l \geq 5$, the call can be handled by code currently given to some data call



(a)

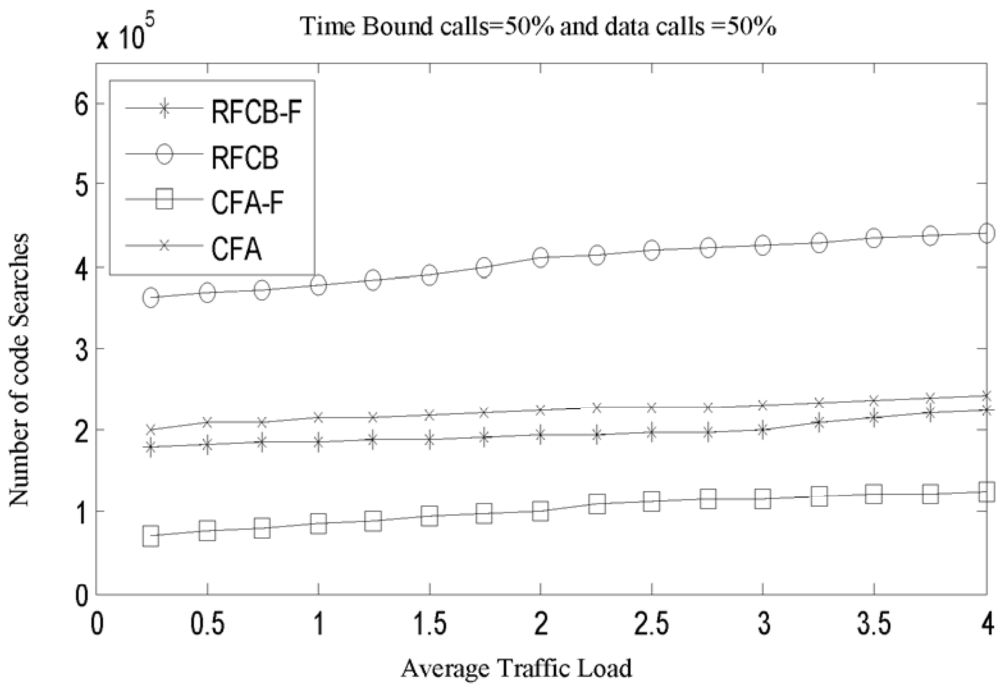


(b)

RFCB-F [33]: Recursive fewer blocked codes-Flexible, CFA [22]: Crowded first assignment-Flexible.

Figure 6.20: Comparison of amount of time bound calls handled for distribution (a) Time Bound calls=25% and data calls =75%, (b) Time Bound calls=50% and data calls =50%.

in portion 1. For the code tree status in Figure 6.19 (*a*), the code $C_{4,2}$ assigned to data calls will be released and assigned to 16*R* time bound call. For data call, the amount of data already transferred and remaining data will be placed in queue/buffer with infinite T_Q. In offline mode, the scheme keep searching 2nd portion for a vacant code for this stored data call. The codes searched for assignment of 16*R* call will be 3 only. If a time bound call of rate 4*R* arrives after 16*R*, 1st portion will be searched. As no vacant code (or code handling data call) is available, the algorithm switches to 2nd portion. Searching in 2nd portion will be considerably reduced as the only few codes need to be searched. The number of code searches



FSP [27]: Fixed set partitioning, CFA [22]: Crowded first assignment, LCA [22]: Left code assignment.

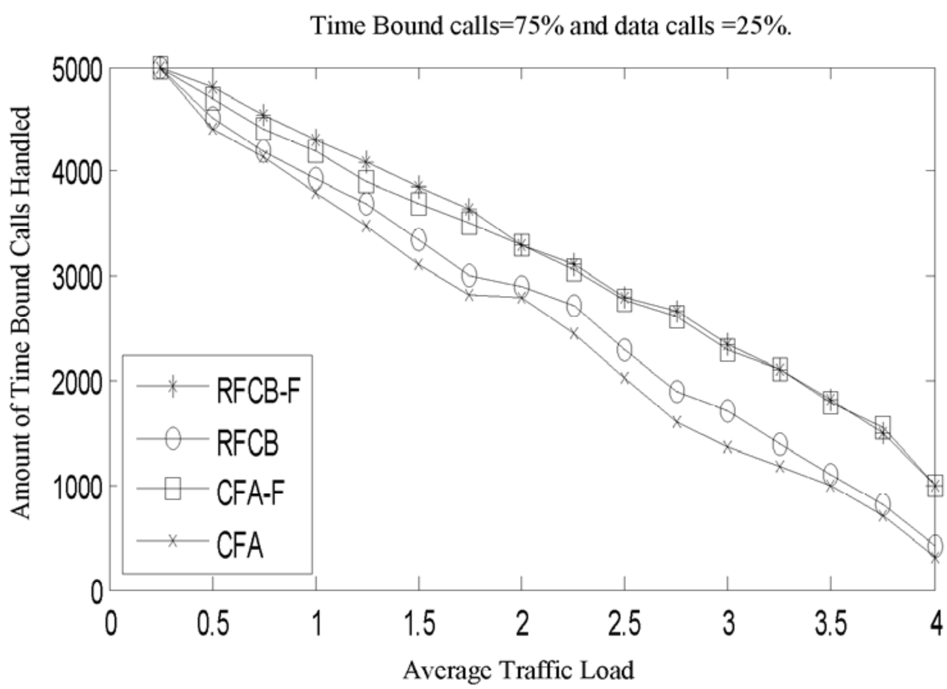
Figure 6.21: Comparison of number of code searches for equal arrival of time bound and data call.

before assignment are 10 only. In offline mode, 1st portion will be searched for a code that can be assigned to 4*R* rate call.

The amount of data calls handled can be improved by releasing only amount of capacity required by time bound call of the code handling data call. Let a time bound call of rate $2^{l-1}R$ arrives and optimum data call ongoing is $2^{k-1}R$, the code handling $2^{k-1}R$ is released and two of its children are used for incoming time bound call and ongoing data call. Any child with capacity $2^{l-1}R$ is used for time bound call and ongoing data call is handled by optimum code selected after following steps.

1. Find $2^{k-1} - 2^{l-1}$.
2. Calculate $m = \log_2(2^{k-1} - 2^{l-1})$.
3. Find $2^{\lfloor m \rfloor}$, $\lfloor m \rfloor$ denotes integer value lesser than or equal to m.
4. The optimum code used for the data call is the one with capacity $2^{\lfloor m \rfloor}$.

For illustration, consider example in Figure 6.19 (b). For an incoming time bound call of rate $4R$, the optimum code to be used is $C_{3,7}$. However, its parent code $C_{4,4}$ is assigned to data



FSP [27]: Fixed set partitioning, CFA [22]: Crowded first assignment, LCA [22]: Left code assignment.

Figure 6.22: Comparison of amount of time bound calls handled for distribution: Time Bound calls=75% and data calls =25%

call. As the total capacity of code $C_{4,4}$ is $8R$, it is divided into two $4R$ rate fractions. One fraction is assigned to time bound call of rate $4R$ (code $C_{3,7}$) and the other (code $C_{3,8}$) will be used to handle the data call with reduced rate.

6.5 SIMULATION AND RESULTS

To compare the performance of the proposed scheme with other schemes, following simulation parameters are considered.

- There are 5 classes of users with rates *R*, 2*R*, 4*R*, 8*R*, 16*R*.
- Both time bound calls and data calls both can request rate varying from *R* to 16*R*.
- Arrival rate λ is Poisson distributed with mean value varying from 0-4 calls/units of time.
- Call duration is exponentially distributed with mean value of 3 units of time.
- The maximum capacity of the tree is 128*R* (*R* is 7.5kbps).
- Simulation is done for 10000 users and result is average of 10 simulations.

Let $\lambda_i, i \in [1,5]$ is the arrival rate and $\mu_i, i \in [1,5]$ is service rate for i^{th} class users. Define $\rho_i = \lambda_i / \mu_i$ as traffic load of the i^{th} class users. Also, for 5 class system the average arrival rate and average traffic load is $\lambda = \sum_{i=1}^{5} \lambda_i$ and $\rho = \sum_{i=1}^{5} (\lambda_i / \mu_i)$ respectively. In our simulation we consider call duration of all the calls equal i.e. $1/\mu = 1/\mu_i$. Define $[p_1, p_2, p_3, p_4, p_5]$ as probability distribution matrix where, $p_i, i \in [1,5]$, is the capacity portion required by the i^{th} class calls on average. Three distribution scenarios are considered

- Time Bound calls=25% and data calls =75%.
- Time Bound calls=50% and data calls =50%.
- Time Bound calls=75% and data calls =25%.

The number of time bound calls handled for various scenarios of our scheme is compared with popular CFA [22] and RFCB [33] scheme in Figure 6.20 and Figure 6.21. CFA-F and RFCB-F denotes the hybrid of our flexible scheme CFA and RFCB scheme which treats time bound calls and data calls differently. It increases amount of time bound calls handled by code tree. Results show that both CFA and RFCB schemes handles more time bound calls compared to existing CFA and RFCB schemes. This is because in pure CFA and RFCB schemes, all calls are treated in a similar manner. Therefore, they handle almost same amount of time bound calls. The variation in their plots is due to the lower or higher rate request dominating. The number of data calls handled does not suffer as the time bound call use the unused capacity of the code tree.

If N_i denotes the number of codes searches required for i^{th} class, the total number of codes searched is given by

$$N = \sum_{i=1}^{5} N_i \tag{6.24}$$

where N_i is the total number of codes searched for i^{th} class of user. Our scheme search half portion of code tree for most of the time bound calls and does not require code tree to be searched for data calls as data calls are queued in buffer. They are assigned to a vacant code in offline mode. When any time bound call ends, data call 1st in queue is handled by that code or its parent code (if vacant). This decreases code searches significantly. The number of the code searches required before a time bound call is handled is shown in Figure 6.22 for equal arrival of time bound and data calls. As expected, the CFA-F and RFCB-F schemes requires less number of code searches as compare to traditional CFA and RFCB.

6.6 CONCLUSION

In 3G and beyond wireless networks, time bound calls cannot tolerate large call establishment delay and hence require least number of code searches. Our scheme searches half portion of code tree prior to code assignment almost every time and hence reduces code searches significantly. Data calls utilize the available scattered capacity of code tree. Further the scheme is more suitable for bursty data, where data density fluctuates considerably. In future, work can be done to make code assignment fair in terms of call rates. Further, the cost and complexity issues can be investigated.

CHAPTER 7
CONCLUSION AND FUTURE WORK

7.1 CONCLUSION

3G and beyond wireless networks are designed to handle multimedia rates. The call establishment delay and jitter are the significant QoS parameters for most of the real time calls. The schemes in thesis are fast, efficient and can be used for both single and multi code assignment. Most of the schemes in thesis outperforms existing schemes in either code blocking or call establishment delay or both. For most of the proposed single code assignment schemes the vacant code used for assignment occurs at a location where future calls rejection is minimum. On the other hand, for multi codes a lower number of rakes used are preferred to reduce complexity. Both AVC and TD schemes reduces call establishment delay in locating optimum code while maintaining required QoS. The scheme that assigns slots for OVSF and NOVSF codes reduces code blocking along with call differentiation on the basis of priority. Again, a code is assigned, which reduces code blocking for real time calls.

The calls integration scheme reduces wastage capacity (internal fragmentation) as it favours non quantized rates. Two single and one multi code assignment schemes are described. Voice calls are given priority over non real time calls. The multiplexed data call can be buffered if required till a vacant code is available. The QoS of data calls is affected as different calls are integrated on same code.

The call elapsed time scheme which handles data calls with almost zero code blocking but higher waiting time in queue, utilizes elapsed time of ongoing calls for new call assignment. Reassignment scheme is also given which reassigns calls to other parts of code tree requiring lesser code searches for reassignment as compared to DCA [25]. An immediate neighbour assignment scheme which checks only immediate code before assignment is also given and is used for multiple codes. The results and simulations verifies the superiority of proposed schemes.

7.1 FUTURE WORK

In this thesis, we introduced the notion of those schemes which reduces call establishment delay while maintaining optimal and sub-optimal code blocking probability which leads to efficient utilization of OVSF codes available capacity at BS. The effective capacity assigned to users at a time should meet QoS requirements requested by the users which limits further code assignment. The resource reservation and admission control can be designed, based on the effective capacity that can be assigned without affecting QoS requirements of ongoing calls. Also, the fast AVC and TD scheme can be combined with existing FSP to increase speed of call assignment process further. Index updation of TD scheme can be made adaptive to call arrival rates. The use of multiple rakes along with the code sharing facility can be used to make real time call blocking close to zero. The code sharing (time sharing) is the complicated task and may require lot of effort. To optimize the assignment of code slots for different order of priority within priority users the present work can be improved. Work can also be done to bring fairness in all schemes, *i.e* all call rates share the total capacity of code tree equally.

4G systems require transmission of multimedia services with higher capacity requirement which inevitably implies an increase in data rate. Orthogonal frequency and code division multiplexing (OFCDM) technique has shown promising results in achieving a high data rate while simultaneously combating multipath fading. OFCDM comprises of orthogonal frequency division multiplexing and two-dimensional (2D) spreading. 2D spreading helps to achieve diversity gains in both time and frequency domains. The 1 dimensional OVSF codes correlation properties are poor then 2D OVSF codes. Work can be done to design assignment schemes for 2D OVSF codes.

.

Appendix A

The proposed scheme is better if *number of code searches for AVC≤ Number of code searches for CFA i.e.*

$$z+\sum_{i=1}^{z} l_{x_i} +\sum_{i=1}^{z+1} t_{x_i} -1 \le 2^{L-l} - N_l + \sum_{i=1}^{N_l} (l_{x_i} + 2^{l+l_{x_i}-1}) \quad \text{A.1}$$

$$z+\sum_{i=1}^{z} l_{x_i} +\sum_{i=1}^{z+1} t_{x_i} -1 \le 2^{L-l} - N_l + \sum_{i=1}^{N_l} l_{x_i} + \sum_{i=1}^{N_l} 2^{l+l_{x_i}-1} \quad \text{A.2}$$

We can represent $\sum_{i=1}^{N_l} l_{x_i}$ as

$$\sum_{i=1}^{N_l} l_{x_i} = \sum_{i=1}^{z} l_{x_i} + \sum_{i=z+1}^{N_l} l_{x_i} \quad \text{A.3}$$

using Equation A.3 in Equation A.2

$$z+\sum_{i=1}^{z} l_{x_i} +\sum_{i=1}^{z+1} t_{x_i} -1 \le 2^{L-l} - N_l + \sum_{i=1}^{z} l_{x_i} + \sum_{i=z+1}^{N_l} l_{x_i} + \sum_{i=1}^{N_l} 2^{l+l_{x_i}-1} \quad \text{A.4}$$

$$z+\sum_{i=1}^{z+1} t_{x_i} -1 \le 2^{L-l} - N_l + \sum_{i=z+1}^{N_l} l_{x_i} + \sum_{i=1}^{N_l} 2^{l+l_{x_i}-1} \quad \text{A.5}$$

$$z+\sum_{i=1}^{z+1} t_{x_i} + N_l -1 \le 2^{L-l} + \sum_{i=z+1}^{N_l} l_{x_i} + \sum_{i=1}^{N_l} 2^{l+l_{x_i}-1} \quad \text{A.6}$$

It can be easily find out that

$$z \le \sum_{i=1}^{N_l} 2^{l+l_{x_i}-1} \quad \text{A.7}$$

where z is number of vacant code searched for adjacency.
Using Equation A.7 in Equation A.6, we

$$\sum_{i=1}^{z+1} t_{x_i} + N_l -1 \le 2^{L-l} + \sum_{i=z+1}^{N_l} l_{x_i} \quad \text{A.8}$$

Also,

$$\sum_{i=1}^{z+1} t_{x_i} + N_l -1 \le 2^{L-l} \quad \text{A.9}$$

As 2^{L-l} is total number of codes in layer l, we can write

$$0 \le \sum_{i=z+1}^{N_l} l_{x_i} \quad \text{A.10}$$

which is always true.

Appendix B

B.1 Pure top down code searches

For a new call with rate $2^{l-1}R$, the codes searched are

$$N_{TD} = 2(L-l)+1 \qquad \text{B.1}$$

B.2 CFA code searches

For a call of rate $2^{l-1}R$, the maximum number of code searched to find vacant code in layer l is 2^{L-l}. If there are z vacant codes in layer l, for each code, C_{l,y_i}, $1 \le i \le z$, $1 \le x_i \le 2^{l-1}$, CFA scheme finds number of busy codes under immediate parent of each C_{l,y_i}. The total number of code searches for parents of each vacant code in layer $l+1$ are $z \times 2^{l+1}$. If a unique immediate parent code (say $C_{l+1,\lceil y_i/2 \rceil}$) with maximum number of busy codes exists, new call will be assigned to its children and code searching stops. Otherwise, let z_1 number of parent codes in layer l+1 that leads to tie for maximum number of busy children. The number of code searches for layer l+2 are $z_1 \times 2^{l+2}$. If a unique result does not exist the procedure is repeated till layer L giving maximum code searches. The total number of code searches for CFA becomes

$$N_{CFA} = 2^{(L-l)} + z \times 2^{l+1} + z_1 \times 2^{l+2} + ... + z_{L-2} \times 2^{l+L-1} \qquad \text{B.2}$$

B.3 Top down and dynamic code assignment (DCA)

DCA circumvent code blocking problem by providing zero code blocking at the cost of increased number of code searches which makes it unsuitable for real time applications. The use of top down scheme can significantly reduce code searches in DCA scheme as follows.

B.3.1 Conventional DCA

For new $2^{l-1}R$ call arrival, the maximum number of codes searched to find a vacant code are given by

$$NC_{DCA}^{1} = 2^{L-l} \qquad \text{B.3}$$

If vacant code is available, procedure stops. Otherwise, let k_1 is the number of blocked codes denoted by C_{l,x_i}, where $1 \le i \le k_1$ and $1 \le x_i \le 2^{L-l}$ in the layer l. For each of the blocked codes, the codes in layer l_1, $l_1 \in [1,2,...l-1]$ are checked to count number of busy children. The number of codes searched in layer l_1, $l_1 \in [l-1, l-2,...,1]$ are 2^{l-l_1}. The total code searches in layer 1 to l-1 is $NC_{DCA}^{2} = k_1 \times (2+4+....2^{l-1})$ B.4

Total number of codes searched in DCA becomes

$$NC_{DCA}^{'} = NC_{DCA}^{1} + NC_{DCA}^{2} = 2^{L-l} + k_1 \times \sum_{i=1}^{l-1} 2^{i} \quad \text{B.5}$$

A code with minimum number of children codes is selected, reassignments needs to be carried out for it. Let at least one code in layer $p \mid p < l$ is busy, the number of codes need to be check for code availability is

$$N_{DCA}^{''} = \sum_{p=1}^{l-1} a_p \times 2^{L-p} \quad \text{B.6}$$

where a_p=0 if layer p do not have a busy code and a_p=1 if layer p has a busy code.

So, total number searches becomes

$$N_{DCA} = \sum_{l=1}^{L} \lambda_l \times [NC_{DCA}^{'} + NC_{DCA}^{''}] \quad \text{B.7}$$

$$N_{DCA} = \sum_{l=1}^{L} \lambda_l \times [NC_{DCA}^{'} + NC_{DCA}^{''}] \quad \text{B.8}$$

B.3.2 DCA top down

The number of code searches required to identify suitable blocked code are

$$N_{TD-DCA} = 2(L-l)+1 \quad \text{B.9}$$

If suitable code is C_{l,n_l}, let there are $p_{l'}, 1 \le l' \le l-1$ busy children of C_{l,n_l} in layer l', the total busy children who need reassignments are $\sum_{l'=1}^{l-1} p_{l'}$. The maximum number of searches required to identify $\sum_{l'=1}^{l-1} p_{l'}$ vacant codes are

$$N_{TD-DCA_2} = \sum_{l'=1}^{l-1} 2^{l'} \quad \text{B.10}$$

The number of code searches required to shift all $p_{l'}, 1 \le l' \le l-1$ busy codes are

$$N_{TD-DCA_3} = \sum_{l'=1}^{l-1} p_{l'} \times (2 \times (L-l')+1) \quad \text{B.11}$$

The total code searches in top down DCA are

$$N_{TD-DCA} = N_{TD-DCA_1} + N_{TD-DCA_2} + N_{TD-DCA_3} \quad \text{B.12}$$

$$N_{TD-DCA} = 2(L-l)+1+\sum_{l'=1}^{l-1} (2^{l'} + p_{l'} \times (2 \times (L-l')+1)) \quad \text{B.13}$$

REFERENCES

[1] Stallings, W. *Wireless communications and networks*. Pearson 2nd edition, 2005.

[2] Rao, G. S. *Mobile cellular communication*. Pearson 1st edition, 2012.

[3] Viterbi, A. J. *CDMA: Principles of spread spectrum communication*. Addison-Wesley, New York, 1995.

[4] Kohno, R., Meidan, R., & Milstain, B. *Spread spectrum access method for wireless communications*. IEEE Commun. Magazine, pp 58-67, 1995.

[5] Dahlman, E., Beming, P., Knutsson, J., Ovesj¨o, F., Persson, M., and Roobol, C. *WCDMA-The Radio Interface for Future Mobile Multimedia Communications*. IEEE Trans. on Vehicular Technology, vol. 47, no. 4, pp 1105-1118, 1998.

[6] Dinan, E. H. & Jabbari, B. *Spreading codes for direct sequence CDMA and Wideband CDMA Cellular Networks*. IEEE Commun. Magazine, pp 48-54, 1998.

[7] Adachi, F., Sawahashi, M., & Okawa, K. *Tree-structured generation of orthogonal spreading codes with different lengths for forward link of DS-CDMA mobile radio*. Electronic Letters, vol. 33, no. 1, pp 27- 28, 1997.

[8] Okawa, K. & Adachi, F. *Orthogonal forward link using orthogonal multi spreading factor codes for coherent DS-CDMA mobile radio*. IEICE Trans. on Commun., vol. 8, no. 4, pp 778–779, 1998.

[9] Kam, A. C., Minn, T. & Siu, K.Y. *Reconstruction methods of tree structure of orthogonal spreading codes for DS-CDMA*. IEICE Trans. on Fundamentals E83-A (11), pp 2078–2084, 2000.

[10] Chao, C.M., Tseng, Y. C. & Wang, L. C. *Reducing Internal and External Fragmentation of OVSF Codes in WCDMA Systems With Multiple Codes*. IEEE Trans. on Wireless Commun., vol. 4, pp 1516-1526, 2005.

[11] Malik, S. A & Zeghlache, D. *Resource allocation for multimedia services on the UMTS downlink*. ICC 2002, vol. 5, pp 3076-3080, 2002.

[12] Malik, S. A., Akhtar, S. & Zeghlache, D. *Performance of prioritised resource control for mixed services in UMTS W-CDMA networks*. in Proc. VTC Fall 2001, vol. 2, pp 1000-1004, 2001.

[13] Akhtar, S., Malik, S. A., & Zeghlache, D. *Prioritised admission control for mixed services in UMTS WCDMA networks*. in Proc. 12th IEEE International Symposium on Personal, Indoor and Mobile Radio Commun., vol. 1, pp B-133 - B-137, 2001.

[14] Malik, S. A & Zeghlache, D. *Downlink capacity and performance issues in mixed services UMTS WCDMA networks*, VTC Spring 2002, vol. 4, pp 1824-1828, 2002.

[15] Garg, V. K. & Yu, O. T.W. *Integrated QoS support in 3G UMTS network*. IEEE Wireless Communications and Networking Conference, vol. 3, pp 1187-1192, 2000.

[16] Cordier, S & Ortega, S. *On WCDMA downlink multiservice coverage and capacity*. IEEE VTC Fall 2001, vol. 4, pp 2754-2758, 2001.

[17] Ruijun, F & Junde, S. *Some QoS issues in 3G wireless networks*. IEEE Conference on Computers, Communications, Control and Power Engineering, vol. 2, pp 724-727, 2002.

[18] Cordier, S. & Ortega, S. *On WCDMA downlink multiservice coverage and capacity*. IEEE VTC Fall 2001, VTS 54th, vol. 4, pp 2754-2758, 2001.

[19] Zafer, M. & Modiano, E. *Blocking probability and channel assignment in wireless networks*. IEEE Trans. On Wireless Commun., vol. 5, no. 4, pp 869-879, 2006.

[20] Kaufman, J. S. *Blocking in a shared resource environment*. IEEE Trans. Commun., vol. 29, no. 10, pp 1474-1481, 1981.

[21] Saini, D.S. & Bhoosan, S.V. *Code Tree Extension and Performance Improvement in OVSF-CDMA Systems*. IEEE ICSCN 2007, pp 316-319, Feb. 22-24, 2007.

[22] Tseng, Y. C., Chao, C. M. & Wu, S. L. *Code placement and replacement strategies for wideband CDMA OVSF code tree management*. IEEE Global Telecommunications Conference (GLOBECOM'01), vol. 1, pp 562-566, 2001.

[23] Amico, M. D., Merani, M. L. & Maffioli, F. *Efficient algorithms for the assignment of OVSF codes in wideband CDMA*. IEEE International Conference on Communications (ICC 2002), vol. 5, pp 3055-3060, 2002.

[24] Tsai, S., Khaleghi, F., Oh, S. J. & Vanghi, V. *Allocation of Walsh codes and quasi-orthogonal functions in cdma2000 forward link*. IEEE Vehicular Technology Conference (VTC 2001), vol. 2, pp 747-751, 2001.

[25] Minn, T. & Siu, K.Y. *Dynamic assignment of orthogonal variable-spreading-factor codes in WCDMA*. IEEE Journal on Selected Areas in Communications, vol. 18, no. 8, pp 1429-1440, 2000.

[26] Park, J. S. & Lee, D.C. *On static and dynamic code assignment policies in the OVSF code tree for CDMA networks*. in Proc. IEEE Military Communications Conf. (MILCOM'02), vol. 2, pp 785-789, 2002.

[27] Park, J. S. & Lee, D.C. *Enhanced fixed and dynamic code assignment policies for OVSF-CDMA systems.* in Proc. of ICWN 2003, pp 620-625, 2003.

[28] Chao, C. M., Tseng, Y. C. & Wang, L. C. *Dynamic bandwidth allocation for multimedia traffic with rate guarantee and fair access in WCDMA systems.* IEEE Trans. on Mobile Computing, vol. 4, no. 5, pp 420-429, 2005.

[29] Xu, L., Shen, X. & Mark, J. W. *Dynamic bandwidth allocation with fair scheduling for WCDMA systems.* Journal IEEE Wireless Communications, vol. 9, no. 2, pp 26-32, 2002.

[30] Fossa, C. E. Jr. & Davis, N. J. IV. *Dynamic code assignment improves channel utilization for bursty traffic in third-generation wireless networks.* in Proc. IEEE International Conference Communications (ICC'02), vol. 5, pp 3061-3065, 2002.

[31] Hwang, R. H., Chang, B. J., Chen, M. X. & Tsai, K. C. *An efficient adaptive grouping for single code assignment in WCDMA mobile networks.* Springer Wireless Personal Communication, vol. 39, no. 1, pp 41- 61, 2006.

[32] Chang, B. J. & Chang, P.S. *Multicode-based WCDMA for reducing waste rate and reassignments in mobile cellular communications*. Computer Communication Journal, vol. 29, no.11, pp 1948-1958, 2006.

[33] Rouskas, A. N. & Skoutas, D.N. *Management of channelization codes at the forward link of WCDMA.* IEEE Communication Letter, vol. 9, pp 679-681, 2005.

[34] Saini, D.S & Bhoosan, S.V. *Adaptive assignment scheme for OVSF codes in WCDMA*. in Proc. of the IEEE ICWMC, pp 65, 2006.

[35] Park, J. S., Huang, L. & Kuo, C.C.J. *Computationally efficient dynamic code assignment schemes with call admission control (DCA-CAC) for OVSF-CDMA systems*, IEEE Trans. on Vehicular Technology, vol. 57, pp 286-296, 2009.

[36] Askari, M., Saadat, R. & Nakhkash, M. *Comparison of Various Code Assignment Schemes in Wideband CDMA*. in Proc. of IEEE on Computer and Comm. Engineering, pp 956-959, 2008.

[37] Saini, D. S, Kanwar, S., Parikh, P. & Kumar, U. *Vacant code searching and selection of optimum code assignment in WCDMA wireless network*, in Proc. of the IEEE NGMAST, vol. 3, pp 224-229, 2009.

[38] Wan, C.S., Shih, W. K. & Chang, R. C. *Fast dynamic code assignment in next generation wireless access network*. Computer Communication Journal, vol. 26, pp 1634-1643, 2003.

[39] Fantacci, R. & Nannicini, S. *Multiple access protocol for integration of variable bit rate multimedia traffic in UMTS/IMT-2000 based on wideband CDMA*. IEEE J. Select. Areas Commun., vol. 18, pp 1441-1454, 2000.

[40] Rouskas, A. & Skoutas, D. *OVSF code assignment and reassignment at the forward link of W-CDMA 3G systems.* in Proc. of the IEEE PIMRC, vol. 13, pp 2404-2408, 2002.

[41] Yang, Y. & Yum, T. S. P. *Multicode multirate compact assignment of OVSF codes for QoS differentiated terminals*. IEEE Trans. on Vehicular Technology, vol. 54, pp 2114-2124, 2005.

[42] Saini, D.S & Bhoosan, S.V. *OVSF Code Sharing and Reducing the Code Wastage Capacity in WCDMA*. Springer J. of Wireless Personal Comm., pp 521-529, vol. 48, 2009.

[43] Saini, D. S. & Upadhyay, M. *Multiple rake combiners and performance improvement in WCDMA systems*. IEEE Trans. on Vehicular Technology, pp 3361-3370, vol. 58, no.7, 2009.

[44] Chen, M. X. *Efficient integration OVSF code management architecture in UMTS*. Computer Communication Journal, vol. 31, pp 3103-3112, 2008.

[45] Cheng, S. T. & Hsieh, M. T. *Design and analysis of time-based code allocation schemes in W-CDMA systems*. IEEE Trans. on Mobile Computing, vol. 4, pp 3103-3112, 2005.

[46] Vadde, K. & Qam, H. *A Code Assignment Algorithm for Nonblocking OVSF Codes in WCDMA*. Springer Journal of Telecommunication Systems, vol. 25, no. 3, pp 417-431, 2004.

[47] Qam, H. *Non-blocking OVSF codes and enhancing network capacity for 3G wireless and beyond systems*, Elsevier Journal of Computer Communication, vol. 26, no. 17, pp 1907-1917, 2003.

[48] Cam, H. & Vadde, K. *Performance analysis of nonblocking OVSF codes in WCDMA*, in Proc. of International Conference on Wireless Networks, pp 204-215, 2002.

[49] Qam, H. *Nonblocking OVSF codes for 3G wireless and beyond systems*, Patent, Publication No. WO/2003/073665, 2003.

[50] Chen, Y. S. & Lin, T.L. *Code Placement and Replacement Schemes for W-CDMA Rotated-OVSF Code Tree Management.* in Proc. of International Conference on Information Networking (ICOIN 2004), pp 224-239, 2004.

[51] Chen, Y. S., Lin, T.H. & Chien, J. Y. *A Fast Code Assignment Strategy for a WCDMA Rotated-OVSF Tree with Code Locality Property*. Springer Telecommunication Systems, vol. 29, no. 3, pp 199-218, 2005.

[52] Chen, Y. S & Lin, T.H. *Code Placement and Replacement Schemes for WCDMA Rotated OVSF Code Tree Management*. IEEE Trans. Mobile Computing, vol. 5, no. 3, pp 224-239, 2006.

[53] Chen, Y. S & Chang, H.C. *Multi-Code Placement and Replacement Schemes for W-CDMA Rotated-OVSF Code Tree*. in Proc. of the IEEE 6th Circuits and Systems, pp 345 - 348, 2004.

[54] Tsai, Y. R. & Lin, L. C. *Quality-Based OVSF code assignment and reassignment strategies for WCDMA Systems*. IEEE Trans. on Veh. Technol., vol. 58, no. 2, pp 1027-1031, 2009.

[55] Khedr, M. E., Roshdy, A., Abdel, R. & Youssef, M. M. *Efficient Utilization of Orthogonal Variable Spreading Factor Trees Using Two levels Of Hierarchies and Adaptive Rate Control Mechanism*. in Proc. of IEEE Conference Rec. 2007 IIT, pp 277-281, 2007.

[56] Shueh, F & Chen, W. S. E. *Code assignment for IMT-2000 on forward radio link*. in Proc. IEEE Vehicular Technology Conference, vol. 2, pp 906-910, 2001.

[57] Shueh, F & Chen, W. S. E. *Optimum OVSF code reassignment in Wideband CDMA forward radio link*. International Journal of Information Science, vol. 174, no. 1-2, pp 81-101, 2005.

[58] Hwang, R. H., Chang, B. J., Chen, M. X & Tsai, K.C. *An Efficient Adaptive Grouping for Single Code assignment in WCDMA Mobile Networks*. Wireless Personal Communications, vol. 39, no. 1, pp 41-61, 2006.

[59] A. Rouskas and D.N. Skoutas, *Comparison of code reservation schemes at the forward link in WCDMA*. in Proc. 4th IEEE International Conference on Mobile and Wireless Communications Networks 2002, pp 191-195, 2002.

[60] Yen, L. H. & Tsou, M. C. *An OVSF code assignment scheme utilizing multiple rake combiners for W-CDMA*. in Proc. IEEE International Conference on Communications (ICC'03), vol. 5, pp 3312-3316, May 2003.

[61] Yen, L. H. & Tsou, M. C. *An OVSF code assignment scheme utilizing multiple RAKE combiners for W-CDMA*, Elsevier Journal of Computer Communications, vol. 27, no. 16, pp 1617-1623, 2004.

[62] Skoutas, D. N., Saied M., & Rouskas, A. N. *A Guard Code Scheme for Handover Traffic Management in WCDMA Systems*, International Journal of Wireless Information Networks, vol. 15, pp 98-104, 2008.

[63] Kavak, A. & Karakoc, M. *Smart Antenna Based OVSF Code Assignment for WCDMA Networks*. Wireless Personal Communication, vol. 43, pp 1761-1766, 2007.

[64] Kavak, A., Karakoc, M., Cleveland, J. R., & Demiray, H. E. *Dynamic allocation of OVSF codes to access terminals with and adaptive antenna array*. in Proc. IEEE PIMRC, pp 21-25, 2005.

[65] Assarut, R., Husada, M. G., Yamamoto, U. & Onozato, Y. *Data rate improvement with dynamic reassignment of spreading codes for DS-CDMA*. Computer Communications, vol. 25, no. 17, pp 1575-1583, 2002.

[66] Erlebach, T., Jacob, R.,Mihal´ak, M., Nunkesser, M., Szab´o, G. & Widmayer, P. *An Algorithmic View on OVSF Code Assignment*. Algorithmica, vol. 47, pp 269-298, 2007.

[67] Yang, Y. & Yum, T. S. P. *Maximally flexible assignment of orthogonal variable spreading factor codes for multirate traffic*. IEEE Trans. Wireless Commun., vol. 3, no. 3, pp 781-792, May 2004.

[68] Park, J. S., Huang, L. & Kuo, C. C. J. *Scalable Dynamic Code Assignment for OVSF-CDMA System.* in proc. of IEEE 2004 Vehicular Technology Conference, vol. 1, pp 744-748, 2004.

[69] Karakoc, M & Kavak, A. *A New Dynamic OVSF Code Allocation Method based on Adaptive Simulated Annealing Genetic Algorithm (ASAGA)*. IEEE 18th International Symposium PIMRC 2007, pp 1-5, 2007.

[70] Feng, L., Fan, P. & Tang, X. *A General Construction of OVSF Codes With Zero Correlation Zone.* IEEE Signal Processing Letter, vol. 14 , no. 12, pp 908-911, 2007.

[71] Karakoc, M & Kavak, A. *A Genetic Algorithm and Simulated Annealing for Dynamic OVSF Code Allocation in WCDMA Networks.* in Proc. of IEEE 15th International Conference SIU 2007, pp 1-4, 2007.

[72] Chang W. Y., Rong Q. H. *OVSF code management schemes on ad hoc networks.* IEEE International Conference on Communications, vol. 7, pp 4152- 4156, 2004.

[73] Rahimi, S. *Selective Technique in Code Assignment Strategies for Wideband CDMA OVSF Code Tree Management.* Wireless and Microwave Technology Conference, WAMICON '06, pp 1- 4, 2006 .

[74] Perez, F. A. C., Avila, J. L. V., Jimenez, A. S. & Guerrero, L.O. *Call Admission and Code Allocation Strategies for WCDMA Systems with Multirate Traffic*. IEEE Journal of Selected Areas in Communications, vol. 24 , no. 1, pp 26-35, 2006.

[75] Chang, B. J. *Optimal number of RAKE combiners for multiple codes assignment with fast handoff in UMTS mobile networks.* in Proc. of Vehicular Technology Conference, vol. 1, pp 401- 405, 2005.

[76] Chang, B. J., Chen, M. X., Hwang, R. H. & Tsai, K. C. *Adaptive Time-sharing Based Grouping Code Assignment in Mobile WCDMA Networks*. in Proc. of IEEE 16th International Symposium on computer networks, vol. 1, pp 341- 345, 2005.

[77] Askari, M., Saadat, R. & Nakhkash, M. *Assignment of OVSF Codes in Wideband CDMA*. Advances in Computer Science and Engineering Communications in Computer and Information Science, vol. 6, no. 2, pp 723-727, 2009.

[78] Assarut, R., Kawanishi, K., Yamamoto, U. , Onozato, Y & Matsushita, M. *Region division assignment of orthogonal variable spreading-factor codes in W-CDMA*. in Proc. of IEEE Vehicular Technology Conference, vol. 3, pp. 1884-1888, Fall 2001.

[79] Assarut, R., Kawanishi, K., Deshpande, R., Yamamoto, U., & Onozato, Y. *Performance evaluation of orthogonal variable spreading factor code assignment schemes in W-CDMA*. in Proc. of IEEE International Conference on Communications (ICC), pp 3050-3054, 2002.

[80] Sekine, Y., Kawanishi, K., Yamamoto, U., & Onozato, Y. *Hybrid OVSF code assignment scheme in W-CDMA*. in Proc. IEEE Pacific Rim Conf. Communications, Computers and Signal Processing (PACRIM'03), vol. 1, pp 384-387, 2003.

[81] Chang W. Y., Wen O., Chien -Yeh W., Meng-Ti L. & Yu-Wei C. *Graph Model for OVSF Code Placement* . in Proc. Future Information Technology (FutureTech), pp 1-6, 2010.

[82] Ihan M., Philip, Andi S., Eddy W., Herman K. & Kuniwati G. *Chaos Codes vs. Orthogonal Codes for CDMAISSSTA2010*, pp 189-193, 2010.

[83] Chimeh, J. D., Mousavinejad, S.M & Mahmoodi, A. An Algorithm for Restraining the OVSF Codes Shortage in HSDPA. *An Algorithm for Restraining the OVSF Codes Shortage in HSPDA*. in Proc. 4^{th} IFIP International New Technologies, Mobility and Security (NTMS), pp 1-4, 2011.

[84] Kasapovic, S., Mujacic, S. , Ceke, D. & Hadzimehmedovic. *A Comparison ways of assignment of codes in UMTS networks*. in Proc. MIPRO, pp 552- 556, 2012.

[85] Sheikh, J.A., Parah, S.A., Akhtar, U.S., Hafiz, A.M. & Bhat, G.M. *Orthogonal Variable Spreading Factor (OVSF) based image Transmission using Multiple Input Multiple Output Orthogonal Frequency Division Multiplexing (MIMO-OFDM) Communications System*, in Proc. International Conference on Devices and Intelligent Systems (CODIS), pp 45-48, 2012.

[86] Atarashi, H., Abeta, S. & Sawahashi, M. *Variable spreading factor orthogonal frequency and code division multiplexing (VSF-OFCDM) for broadband packet wireless access*, IEICE Trans. Commun., pp 291-299, 2003.

[87] Shah, S.M., Umrani, A.W. & Memon, A. *A Performance comparison of OFDM, MC-CDMA and OFCDM for 4G wireless broadband access and beyond*. in Proc. PIERS, pp 1396-1399, 2011.

[88] Chang, B.J. & Wu, C. H. *Adaptive Load Balancing MDP-Based Approach of Two-Dimensional Spreading for VSF-OFCDM in 4G Next-Generation Cellular Communications*. IEEE Trans. on Vehicular Technology, vol. 58, no. 3, pp 1143-1155, 2009.

[89] Fedra, Z. & Blumenstein, J. *Code Based Channel Estimation in VSF-OFCDM Systems*. 20^{th} Telecommunications Forum (TELFOR), pp 468-470, 2012.

[90] Saini, D. S., Hasthir, V., & Sood, M. *Vacant code precedence and call blocking reduction in WCDMA systems*. in Proc. of IEEE IACC, pp 812-815, 2009.

Publications in International Journals

1. D. S. Saini and V. Balyan, "Top Down Code Search to Locate An Optimum Code and Reduction in Code Blocking for CDMA Networks", *Wireless Personal Communication Springer, in press.*
2. V. Balyan, D.S. Saini and G. Gupta, "OVSF based Fair and Multiplexed Priority Calls Assignment CDMA Networks", WSEAS Transactions on Communication, no. 1, vol. 12, Jan. 2013.
3. D. S. Saini and V. Balyan., " OVSF code slots sharing and reduction in call blocking for 3G and beyond WCDMA networks ",WSEAS Transactions on Communication, vol. 11, no.4, pp. 135-146, April 2012.
4. V. Balyan and D.S. Saini , "Integrating new calls and performance improvement in OVSF based CDMA Networks," *International Journal of Computers & Communication*, vol.5, no. 2, pp. 35-42, June 2011.
5. V. Balyan and D.S. Saini , "Vacant codes grouping and fast OVSF code assignment scheme for WCDMA networks," *Springer J. Of Telecommun Syst*, DOI 10.1007/s11235-011-9469-5, June

Publications in International Conference

1. V. Balyan and D.S. Saini, "Call Elapsed Time and Reduction in Code Blocking for WCDMA Networks, *IEEE International Conference on Software Telecommunication and Computer Networks*, SoftCom 2009, pp. 141-145, Croatia, Sept. 2009.
2. V. Balyan and D.S. Saini, "A fair multi code OVSF design for 3G and beyond wireless networks, *IEEE International Conference on Software Telecommunication and Computer Networks*, IEEE, Softcom 2009, pp. 146-150, Croatia, Sept. 2009.
3. D.S. Saini, V. Balyan, K. Mittal, M. Saini, and M. Kishore, "Fast OVSF Code Assignment Scheme for WCDMA Wireless Networks", Springer international Conference, BAIP 2010, pp. 66-70, Feb 2010, kerala, INDIA.
4. V. Balyan and D.S. Saini, "Immediate Neighbor Assignment and Reduction in Code Blocking for OVSF-WCDMA, *IEEE International Conference on Software Telecommunication and Computer Networks*, SoftCom 2010, Sep 23-25, 2010, Croatia, Print ISBN: 978-1-4244-8663-2.
5. V. Balyan and D.S. Saini, "An efficient multi code assignment scheme to reduce call establishment delay for WCDMA *networks IEEE international conference on parallel, distributed and grid computing (PDGC),* pp. 263-268, oct 28-30, Solan, INDIA, DOI: 10.1109/PDGC.2010.5679907.
6. V. Balyan and D.S. Saini, "Flexible assignment of OVSF Codes for data calls in CDMA Wireless Networks", *IEEE International Conference on Communication Systems and Network Technologies*, pp. 5-9, 3-5 June, 2011, Katra, Jammu.
7. V. Balyan and D.S. Saini, "Multi Code Assignment with Minimum Number of Rakes for OVSF CDMA", *accepted for publication IEEE International Conference on Software Telecommunication and Computer Networks*, SoftCom 2011, Sep 15-17, 2011, Croatia.
8. V. Balyan and D.S. Saini, "Code Assignment and Reassignment to reduce new code blocking in WCDMA networks", *accepted for publication IEEE International Conference on Software Telecommunication and Computer Networks*, SoftCom 2011, Sep 15-17, 2011, Croatia.
9. V. Balyan, D. S. Saini, Alok K. Singh, Paras Agarwal, Pranjal Agarwal," Neighbour Code Capacity and Reduction in Number of Code Searches",IEEE Conference on Information & Communication Technologies (ICT),2013, pp. 589-593.

VIPIN