# OVSF CODE ALLOCATION IN THE FORWARD LINK OF CDMA WIRELESS NETWORKS

by

**NEERU SHARMA**

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

MARCH 2013

# CERTIFICATE

This is to certify that the thesis entitled, "**OVSF Code Allocation in the Forward Link of CDMA Wireless Networks**" which is being submitted by **Neeru Sharma** in fulfillment for the award of degree of Doctor of Philosophy in Electronics and Communication Engineering by the Jaypee University of Information Technology, is the record of candidate's own work carried out by her under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma.

Dr. Davinder Singh Saini
Associate Professor
Department of Electronics and Communication Engineering
Jaypee University of Information Technology
Waknaghat.

# ACKNOWLEDGEMENT

"I bow in adoration unto the Great Spirit, the Principle of principles, the One who has no beginning, no middle, no end, the unborn, the One who knows no growth and no decay, the immortal."

Now, I have come to the end of my journey of Ph.D. and finally take this opportunity to express my gratitude to friends, colleagues, collaborators, relatives, etc. who have provided me the moral and scientific support for completion of this thesis.

His actions have inspired me to dream more, learn more, do more and become more. He holds this capacity to define oneself to others in a way that clarifies and expands the vision for future. He is my guiding force, **Dr. Davinder Singh Saini,** whose repertoire of scientific skills, talent and knowledge has enabled me to attain the level where I find myself today. I am extremely grateful for his unfailing patience, readiness to help throughout by affectionate behavior, constructive criticism, useful advice and astute guidance.

I would like to express my sincere thanks to **Dr. Y Medury**, COO of Jaypee Education System. I am also thankful to our Vice Chancellor **Prof. Ravi Prakash** and Director, **Brig.(Retd.) Balbir Singh** for their invaluable support. Also I would like to offer my deep gratitude to **Prof. S.V Bhooshan**, HOD (ECE) for approving my endeavor and imparting their beneficial suggestions. I am also thankful to **Prof. T.S. Lamba** for his ever challenging guidance to explore new ideas.

I am short of words in expressing my gratitude towards my parents who have been and will be guiding stars in my life. It is their inspirations and prayers which have always been a driving force behind all this. I am indebted to my father and mother for their unfailing patience, great tolerance, and lots of love and care. It is primarily because of my parents' guidance and support that I have been able to complete this arduous journey. It would have been too tough a journey without the constant encouragement and selfless attitude of my younger brother Rohit.

A word of thanks to the library staff of our university.

**Neeru Sharma**

# TABLE OF CONTENTS

**1. Introduction**

**2. Reduction in code blocking using scattered vacant codes**

**3. Compact Code Assignment**

**4. Flexible assignment of data calls**

**5. Some other novel code assignment schemes**

# LIST OF FIGURES

# LIST OF TABLES

# CHAPTER 1

# INTRODUCTION

## 1.1 Introduction

Mobile communications have come a long way till this date. It all started with the first generation of mobiles which provided analog voice communication. The popular standards of first generation were AMPS, TACS and NMT [1]. Eventually2G was introduced, which provided digital voice communication and also data services of 9.6 kbps. All the 2G cellular communication systems (for example PDC/GSM/IS-54 and IS-94) adopted digital technology. The major services provided by 2G were limited to basic services as voice, facsimile and low bit rate data. The need was felt for wider services which could provide high speed internet access and video/high quality image transmission. Then the third generation was evolved. The evolution of 2G systems to 3G was a smooth process with 2.5G as a bridge between both the generations of mobiles. There was a rapid progress in the standardization of 3G. The services provided by 2G were extended with high rate data capabilities. The third generation systems, called Universal Mobile Telecommunications System (UMTS)/International Mobile Telecommunications-2000 (IMT-2000) by ITU were designed to support wideband services at data rates as high as 2 Mbps and provide quality as that of fixed networks [1]. A new wireless access technology was chosen to realize UMTS/IMT-2000. UTRA is based on wideband 4.096 Mcps DS-CDMA technology. UTRA included both the frequency division duplex (FDD) mode and time division duplex (TDD) mode. The main features of UTRA were: (*i*) For paired bands 1920-1980 MHz and 2110-2170 MHz wideband CDMA (WCDMA) was to be used in FDD operation, (*ii*) For unpaired bands of total 35 MHz time division code division multiple access (TD-CDMA) was to be used in TDD operation [2,3].

WCDMA was chosen as basic radio-access technology for UMTS/IMT-2000 in all major areas of the world. UTRA based on WCDMA fully supported the UMTS/IMT-2000 requirements, for example support of 384kbps with wide area coverage and 2Mbps with local coverage. Compared to 2G narrowband CDMA, the WCDMA radio interface offers significant improvements, in addition to the support of higher rate services [4]. These included improved coverage and capacity due to a higher bandwidth and coherent uplink detection; support of inter-frequency handover necessary for Hierarchical Cell Structure (HCS), support for capacity, improving technologies such as adaptive antennas, multi user detection and a fast efficient packet access protocol [3].

The main characteristics of WCDMA are:

1) Improved performance as compared to 2G systems, like improved capacity, improved coverage.
2) Support wide range of services with maximum bit rate above 2Mbps and making multiple parallel services possible in one connection.
3) High degree of operator flexibility and support of evolutionary technologies such as adaptive antenna arrays and multi user detection.
4) An air interface based on DS-CDMA and operation at a wide bandwidth gave the opportunity to design a system with properties fulfilling the 3G requirements. One of the important characteristics of WCDMA is that the power is the common shared resource for users. In the downlink, the total transmitted power of an RF carrier is shared between the users transmitting from the base station using code division multiplexing.

WCDMA is flexible in handling mixed services and services with variable bit rate demands. Various specifications of WCDMA are given in Table 1.1. Power is allocated to each user depending upon the factor that maximum interference is not exceeded and as a result radio resource management is achieved. To provide higher and variable data rates, two schemes are proposed in the 3G wireless standard: multi code CDMA (MC-CDMA) and orthogonal variable spreading factor CDMA (OVSF-CDMA). In MC-CDMA, multiple orthogonal constant spreading factor (OCSF) codes can be assigned to user. The maximum data rate a user can receive depends on the number of transceivers in the device. Therefore, a higher rate implies higher cost. In OVSF-CDMA, a single OVSF code is assigned to each user. Higher data rates are provided using lower spreading factors. This flexibility is supported in WCDMA by the use of OVSF codes for channelization of different users. These OVSF codes have the characteristic of maintaining downlink transmit orthogonality between users even if they operate at different bit rates. Prior to OVSF codes, only OCSF codes were used. All OCSF codes have same number of chips (spreading factor). For higher data rate, more number of

**Table 1.1:** WCDMA Specifications

| **Parameters** | **WCDMA Specifications** |
|---|---|
| Channel Bandwidth | 5, 10, 20 MHz |
| Downlink RF channel structure | Direct Spread |
| Chip rate | 4.096/8.192/16.984 Mcps |
| Roll-off factor | 0.22 |
| Frame length | 10ms/ 20ms |
| Spreading Modulation | Balanced QPSK (downlink), Dual channel QPSK (uplink), Complex spreading circuit |
| Data modulation | QPSK (downlink) |
| Coherent detection | User-dedicated time-multiplexed pilot (downlink and uplink), common pilot in downlink |
| Channel multiplexing in uplink | Control and pilot channel time-multiplexed I and Q multiplexing for data and control channel |
| Multirate | Variable spreading and multicode |
| Spreading factors | 4-256 (FDD), 1-16 (TDD) |
| Power Control | Open and fast closed loop (1.6kHz) |
| Spreading (downlink) | Variable-length orthogonal sequences for channel separation<br>Gold sequences for cell and user separation |
| Spreading (uplink) | Variable-length orthogonal sequences for channel separation, Gold sequences $2^{41}$ for user and user separation (different time shifts in I and Q channel, cycle$2^{16}$ 10ms radio frame) |
| Handover | Soft handover, Interfrequency handover |

codes were used. Usage of OCSF codes lead to increased hardware complexity because the number of transceivers required were equal to the number of codes used.

UTRA/FDD is based on 5MHz WCDMA with a basic chip rate of 4.096Mchips/s, corresponding to a bandwidth of approximately 5MHz [3,4]. Higher chip rates (8.192 and

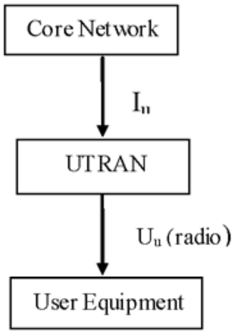


**Fig1.1:** UMTS (IMT-2000) general system architecture

16.384Mchips/s) were also specified for future evolution of the WCDMA air interface toward even higher data rates (> 2 Mbps). For low delay speech and fast control messages, the basic radio frame length is 10ms.

WCDMA carriers are located on a 200 kHz carrier grid with typical carrier spacing in the range 4.2-5.0MHz. High degree of service flexibility of WCDMA air interface is supported by OVSF codes. The OVSF codes preserve the orthogonality between downlink physical channels even if they use different spreading factors and thus offer different bit rates.

## 1.2 UMTS/IMT-2000 System Architecture

The general system architecture includes user equipment (UE), UMTS terrestrial radio access network (UTRAN) and a core network as shown in Figure 1.1. Two interfaces are included in general architecture - the $I_u$ interface between UTRAN and the core network and the $U_u$ (radio) interface between UTRAN and the UE [1]. A functional layering of UMTS/IMT-2000 has been agreed upon in ETSI SMG12, as shown in Figure 1.2. The functional layering introduces the concepts of access stratum and non-access stratum. The access stratum contains all radio-access-specific functionality. It offers services in the UE and the core network to the non access stratum.

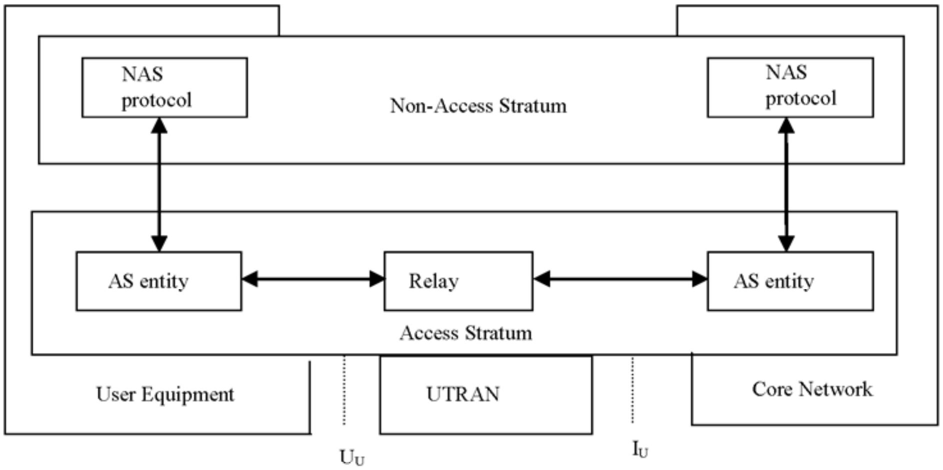


**Figure 1.2:** Access and non - access stratum

The non-access stratum offers the UMTS/IMT-2000 services to the users [3]. The functional layering of the UMTS/IMT-2000 system into the access and non-access stratums implies a functional division between UTRAN and the core network. UTRAN handles all the radio-specific procedures whereas the core network handles the service-specific procedures, including mobility management and call control.

## 1.3 Radio interface protocol architecture for WCDMA

Figure 1.3 shows the protocol architecture for WCDMA. Layer 1 comprises the WCDMA physical layer. Layer 2 comprises the medium access (MAC), radio link control (RLC-C for the control plane and the RLC-U for the user plane) protocols, as well as link access control (LAC) protocol [3]. MAC and RLC belong to the access stratum and terminate within UTRAN whereas LAC belongs to the non-access stratum and terminates in the core network. The network layer of the control plane is spilt into radio resource control (RRC) sublayer and the MM and connection management (CM) sublayers. CM and MM belong to the non-access stratum while RRC belongs to access stratum [11]. The codec layer shown can either belong to the access or non-access stratums.

### *1.3.1 WCDMA Physical Layer*

WCDMA is based on wide-band direct sequence CDMA (DSCDMA) technology with a basic chip rate of 4.096 Mcps [1]. The chip rate can be expanded to 8.192 and 16.384 Mcps in order to accommodate user bit rates above 2 Mbps. WCDMA uses FDD and has a flexible carrier spacing of 4.4-5.0 MHz with a carrier raster of 200 kHz.

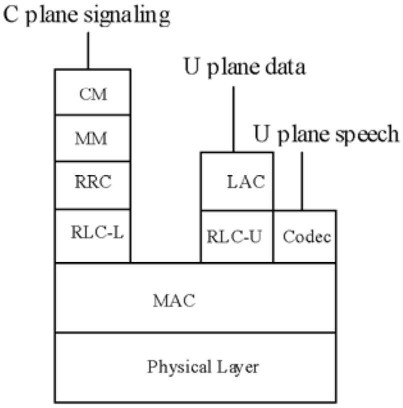


**Figure 1.3:** Protocol stack of WCDMA

### *A. Physical Channels*

Two types of dedicated physical channels are defined in WCDMA [2]:

1) Dedicated physical data channel (DPDCH): This is used to carry dedicated data generated at layer 2 and above.

2) Dedicated physical control channel (DPCCH): This is used to carry layer 1 control information.

Each connection is allocated one DPCCH and zero, one or two several DPDCH's.

Additionally, common physical channels are defined:

- Primary and secondary common control physical channel (CCPCH) used to carry downlink common channels.
- Synchronization channel (SCH) used for cell search.
- Physical Random Access Channel (PRACH).

*Uplink DPDCH and DPCCH*

In the uplink, the DPDCH and DPCCH are code and IQ multiplexed within each radio frame.

The uplink DPDCH carries layer 2 data, while the DPCCH carries pilot bits, transmit power control (TPC) commands and an optional transport format indicator (TFI). A certain TF defines how the layer 2 data carried on the DPDCH(s) is multiplexed and coded and what SF is used. The TFI informs the receiver side when TF is used in the current data frame in order to simplify detection, decoding and demultiplexing.

In Figure 1.4, each frame of length 10 ms is divided into 16 slots of length 0.625 ms, each corresponding to one power-control period. Hence the power control frequency is 1600Hz. Within each slot, the DPDCH and DPCCH are transmitted in parallel on the I (in-phase) and Q (quadarature phase) respectively using different codes. The spreading factors for the DPDCH and DPCCH can vary from 4 to 256, SF = $256/2^k$, $k = 0, 1, 2.....6$, carrying $10\times 2^k$ bits per slot each. The DPDCH and DPCCH use different codes and can be of different rates. Hence the spreading factor will differ between the two channels. The relative power between the DPDCH and DPCCH can be varied to control the amount of overhead. Typically values for the relative power difference are 3 and 10 dB for speech and 384 kbps data respectively. Spreading and modulation of the uplink dedicated

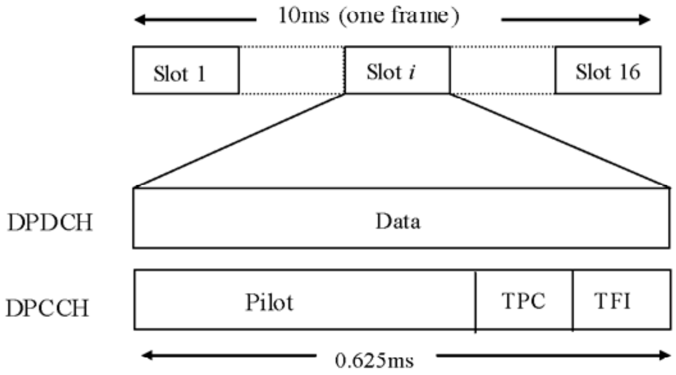


**Figure 1.4:** Uplink DPDCH/DPCCH frame structure

physical channels is shown in Figure 1.5. The DPDCH and DPCCH are mapped to the I and Q branch respectively, and spread to the chip rate with two different channelization codes. The resulting complex signal is scrambled, and QPSK modulation with root-raised cosine pulse shaping with a rolloff factor of 0.22 in the frequency domain is applied. When multi code transmission is used, additional DPDCH's are mapped to either the I or Q branch. For each branch, each additional DPDCH is assigned a new channelization code.

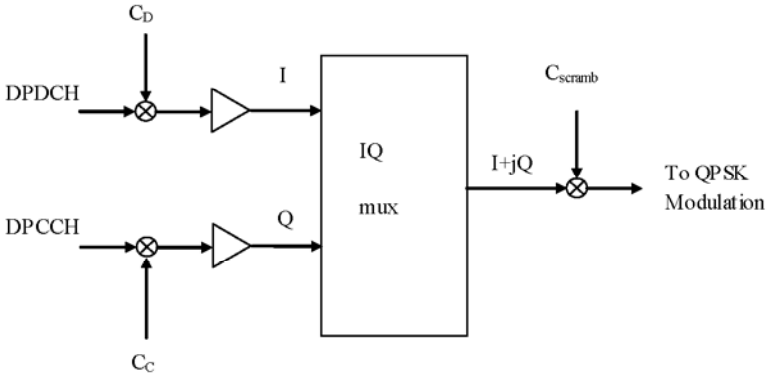


**Figure 1.5:** Uplink channelization and scrambling. Channelization codes are $C_C$ and $C_D$ and scrambling code is $C_{scramb}$.

The channelization codes are used to spread the data to the chip rate, preserving orthogonality between physical channels with different rates and spreading factors. These OVSF codes are used as channelization codes. Each level in the code tree corresponds to certain spreading factor. A physical channel spread by the code *A* is orthogonal to another physical channel spread by code *B* if and only if *B* is not on the path to the root of the tree from *A* or in the subtree below *A*. Hence the number of available codes depends on the rate and spreading factor of each physical channel. The uplink scrambling code can be either short or long. The short scrambling code is a complex code built of two 256-chips-long extended codes from the VL-Kasami set of length 255. The long scrambling code is a 40960-chips segment of a Gold code of length $2^{41}$-1. Both channelization codes and UE-specific scrambling codes are assigned by the network. The set of channelization codes used may be changed during the connection. Cells using advanced receivers, e.g., multiuser detection will typically use the short scrambling code to lower the complexity of the receiver algorithm. When short codes are used, the cross correlation properties are maintained between symbols making the updating of a cross-correlation matrix less complex. However, short codes have worse interference averaging properties than long codes. Hence, in cells where an ordinary rake receiver is used, the long scrambling code is used [2]. The IQ multiplexing of control and data is used to ensure that electromagnetic compatibility (EMC) problems are minimized in the UE. To minimize interference and maximizing capacity, during speech silent periods no data is transmitted. However, pilot bits and power-control commands are needed to keep the link synchronized and power controlled. The IQ multiplexing avoids pulsing the power with a given frequency. If time multiplexing of control and data was used instead, a 1600-Hz tone would be emitted during silent periods.

*Downlink DPDCH and DPCCH*

In the downlink, the DPDCH and DPCCH are time multiplexed within each radio frame. As in the uplink, the downlink DPDCH contains layer 2 data, while the DPCCH carries pilot bits, TPC commands, and an optional TFI as shown in Figure 1.6. Similar to the uplink, each frame of length 10ms is divided into 16 slots of length 0.625 ms, each corresponding to one power-control period. Within each slot, the DPCCH and DPDCH

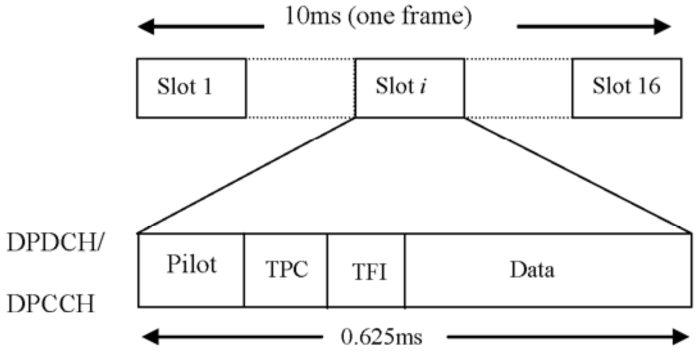


**Figure 1.6:** Downlink DPDCH/DPCCH frame structure

are time multiplexed and transmitted with the same code on both the I and Q branches. The spreading factor for the DPDCH and DPCCH can vary between 4–512, SF=512/2k, k=0,1,…7. Figure 1.7 shows the spreading and modulation of the downlink dedicated physical channels. The DPCCH/DPDCH bits are mapped in pairs to the I and Q branches, and spreading to the chip rate is done with the same channelization code on both I and Q branches. Subsequent scrambling is then performed before QPSK modulating the complex signal. Root raised cosine pulse shaping with a rolloff factor of 0.22 in the frequency domain is used. Channelization is done using the same type of OVSF codes as for the uplink dedicated physical channels, and the set of codes used can be changed by the network during a connection. The downlink scrambling code is a 40960 chips segment of a Gold code of length $2^{18}$ -1. There are 512 different segments used for downlink scrambling. These are divided into 16 groups of 32 codes each in order to simplify the cell-search procedure. Each cell is assigned a specific downlink scrambling code at initial deployment. For multi code transmission, each additional DPCCH/DPDCH is spread and scrambled in a similar way using a channelization code that keeps the physical channels orthogonal. Taking into account the fact that all users share the channelization codes in the downlink, the IQ multiplexing scheme where a whole code is needed for the DPCCH only will use unnecessarily many codes. Hence, time multiplexing is a logical choice in the downlink. The use of pilot bits on the WCDMA

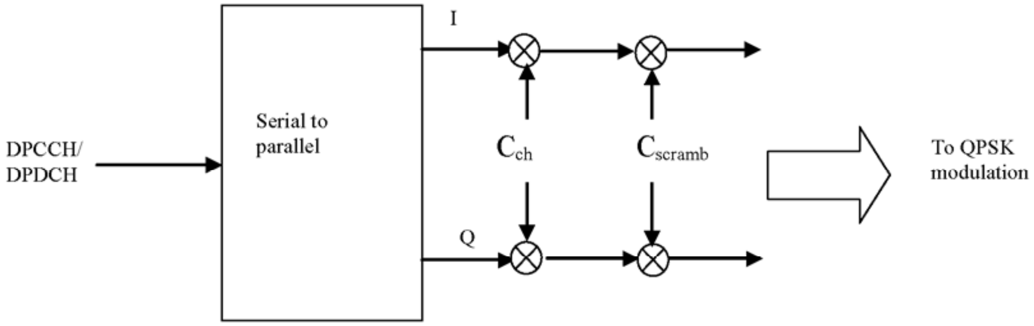


**Figure 1.7:** Downlink channelization and scrambling. Channelization codes are $C_{ch}$ and scrambling code is $C_{scramb}$.

dedicated physical channels ensures that adaptive antennas can be introduced in the downlink. If a common downlink pilot signal is used for coherent detection, like in IS-95, that pilot must have the same antenna diagram as the traffic channel. This prohibits the use of downlink beam forming, where the traffic channels are transmitted in narrow beams.

### *B. Transport Channels*

The transport channels in WCDMA are classified into the dedicated and common categories [3]. The dedicated transport channels are dedicated to specific UE in which the UE is identified by the physical channel (that is, code and frequency for FDD and code, time slot, and frequency for TDD). In the common transport channels, particular UEs are addressed when there is a need for in-band identification of a UE. The transport channels may carry user plane information or Control plane information. To each transport channel, there is an associated Transport Format set. The dedicated transport channel types are:

1) Dedicated Channel (DCH): A channel dedicated to one UE used in uplink or downlink. The DCH channel can be used for the transmission of packet or circuit data. Since the set up time for this channel is of the order of about 250ms, it is suitable for transmission of data on long sessions.
2) Enhanced Dedicated Channel (E-DCH): A channel dedicated to one UE used in uplink only.

The common transport channel types for the FDD mode are:

1) Random Access Channel (RACH): A contention based uplink channel used for transmission of relatively small amounts of data.
2) Common Packet Channel (Uplink): This is an extension to the RACH channel and is intended for uplink transmission of packet-based user data. The CPCH transmission may last several frames in contrast to one or two frames for the RACH. The main differences of this channel with RACH are the use of fast power control and a physical layer-based collision detection mechanism.
3) Forward Access Channel (FACH): This is a common downlink channel without closed loop power control used for transmission of relatively small

amount of data. In addition, FACH is used to carry broadcast and multicast data.

4) Broadcast Channel (BCH): A downlink channel used for broadcast of system and cell-specific information.
5) Paging Channel (PCH): A downlink channel used for broadcast of control information into an entire cell allowing efficient UE sleep mode procedures.
6) High Speed Downlink Shared Channel (HS-DSCH): A downlink channel shared between UEs by allocation of individual codes, from a common pool of codes is assigned for the channel.

## 1.4 OVSF Code Tree

High data rate and variable bit rate transmission is provided by UMTS using OVSF codes are used as channelization codes in the forward link of WCDMA systems. WCDMA uses a fixed transmission chip rate of 3.84Mcps in order to approximately use the 5MHz frequency bandwidth of the channel. WCDMA can use the channelization codes to transmit information at different bit rates, transmitting every bit of information as a code at 3.84Mcps, the bit rate will depend on the length of the code. The shorter the code the higher the information bit rate. If every bit of information is multiplied by the spreading code with a chip rate of 3.84Mcps, it means that the bandwidth of the information signal is spread along the bandwidth used by the chip rate (approximately 5MHz). OVSF codes can be represented by a binary code tree. In general 8 layer OVSF code tree is used for WCDMA system. A six layer OVSF code tree is shown in Figure1.8. The code tree generation takes place according to the Walsh procedure which produces orthogonal codes. If a code A is there, it will have two children [A,-A] (say code B). Now this will again have two children [B,-B] and so on. In WCDMA systems, the spreading factor (SF) is varied and the chip rate is fixed to 3.84Mcps. The SF is chosen in such a way that the product of SF and incoming data rate is always equal to 3.84Mcps. The code tree is numbered from layer 1 to layer 8, with the eighth layer being the top most layer or the root node.

In general, consider an $L$ layer OVSF code tree. For a layer '$l$', SF is $2^{7-l}$ and the capacity is $2^{l}$R. The possible rates for $R$, $2R$, $4R$.......$128R$ (where $R$= 7.5kbps for forward and 15kbps for reverse link). The spreading factor varies from 4, 8…..512 for the forward link and 4, 8, …..256 for the reverse link. Each code is represented by $C_{l,n}$ where $l$ is the layer number and $n$ is the code id. The relationship of the spreading factor and the data rate is shown in Table 1.2. In WCDMA, only quantized rates are possible, that is, the rates in $2^nR$ format are the quantized rates and rates not in this format are the non quantized rates. The OVSF code tree exhibits two important properties:

1) Two codes are said to be orthogonal if and only if none of the two is ancestor or the descendant of the other.
2) If a code is used to handle any call, then neither its parent nor its children can be used to handle any other call, that is, they are blocked.

### *1.4.1 Code Blocking*

The main drawback of OVSF codes is that they exhibit a property known as code blocking. At any time instant, any code tree has three status of each code:

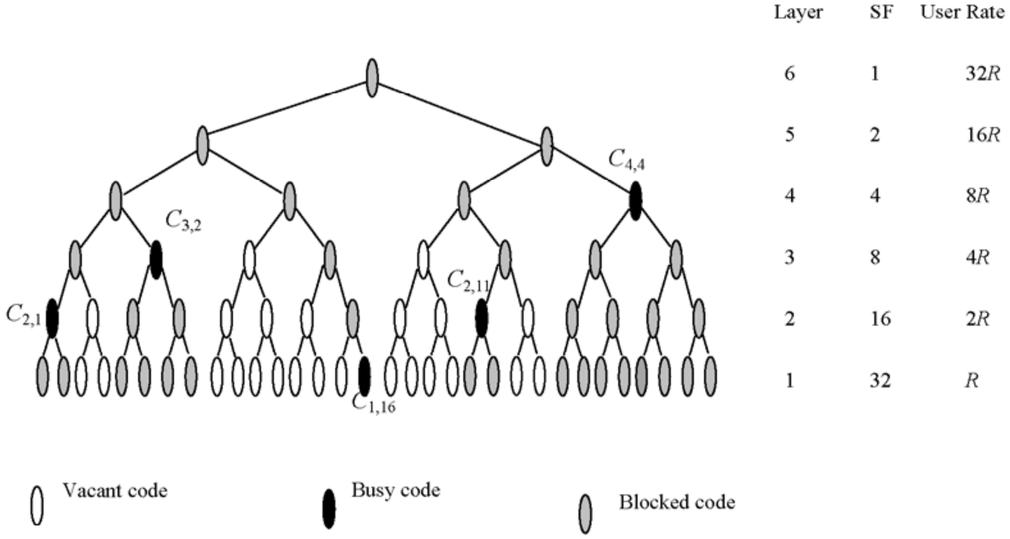


**Figure 1.8:** 6 layer OVSF code tree

- Vacant code: This is the unused code which is available with its full capacity to handle a new incoming call.
- Busy code: This is the code which is handling the call at present.

- Blocked code: These are the parent or the children of the busy code, which cannot be used to handle any new call. These are called blocked because they are unable to handle the call themselves because of the busy code.

Code blocking is the inefficiency of the code tree to handle a new incoming call even if it has enough capacity to handle the call. The code blocking property is illustrated using a six layer code tree as shown in Figure. 1.8. The tree has five busy codes with code id $C_{1,16}$,$C_{2,1}$, $C_{2,11}$, $C_{3,2}$, and $C_{4,4}$. The total capacity of the tree is $32R$, out of which $17R$ is currently used (due to five busy codes) and $32R$-$17R$=$15R$ is free. Despite of having $15R$ free capacity, if a new call with rate $8R$ comes, it will be rejected by the code tree as there is no $8R$ capacity vacant code available. Code blocking can be reduced by using assignment and reassignment schemes.

### *1.4.2 Internal and External Fragmentation*

An OVSF code tree can handle quantized rates efficiently. If a non quantized call comes, then the problem of internal fragmentation occurs. A call is "overserved", that is, the data rate allocated to the incoming call is more than requested. The reason being is that the codes available in OVSF code tree are always in power of 2 (quantized). If a new call of

**Table1.2**: Relationship between data rate and spreading factor in the forward link of WCDMA

| DataRate (kbps) | Spreading Factor (SF) | Chip 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 |

rate 24R arrives, and is to be handled using a single code only, and the code assigned to handle this call is 32*R* which results in 8/32= 25% wastage of bandwidth[5].

To avoid internal fragmentation, one solution is to break the incoming call into quantized fractions so that multiple codes can be used to handle a single call. As in the above case, 24*R* call can be broken into 16*R* and 8*R* which can be handled efficiently using two different codes. As the number of calls arrive and leave the code tree, eventually the code tree becomes so fragmented that it cannot handle any new calls even if it has enough capacity. This is known as external fragmentation [6]. The solution to this problem is code assignment(s) and reassignment(s) [7,8]. Code assignment refers how to handle the new call by utilizing the best possible methods whereas code reassignment refers to the process of reassigning the new codes to already ongoing calls so that the new call can be handled efficiently reducing the external fragmentation [10]. The goal of all the schemes available in literature is to reduce external fragmentation so that the code tree can be efficiently utilized.

### 1.5 Existing Schemes

The schemes can be divided into two categories, single code and multi code depending on the codes used. The single code assignment schemes are simpler, cost effective and require single rake combiner at the BS/UE.

These schemes lie in the following categories:

1) *Static code assignment schemes*: These schemes rely on efficient placement of the code for the new call such that the available capacity of the tree is better utilized [9,31]. Codes should be selected in such a way that the code tree fragmentation is minimum. As a result there is less code scattering and the number of vacant codes for higher rate calls increases.
2) *Dynamic code assignment schemes*: The main aim of these schemes is to reduce code blocking [12-15]. The criterion to choose reassignments depends upon the cost of reassignments. The reassignments increase the cost and complexity of the transceivers. The vacant code is searched and once a minimum cost branch is found, the root code of the branch is assigned to the new call. If the branch is vacant, the root code is assigned to the call and the process of handling the call is complete. It

minimizes the number of OVSF codes that must be reassigned to support a new call. The basic principle is “if a vacant code is not available and the net capacity is within the maximum capacity, the call can be handled by reassignments of the busy codes” [12]. To make code assignment optimal, the cost is checked for every blocked code with rate equal to rate of incoming call. The code with minimum cost is picked. One of the parameters of cost can be number of codes reassigned. All the children of the minimum cost code are reassigned to other branched and the code becomes vacant. Lesser is the number of code reassigned, less is the amount of overhead required to send the information of the reassignments [34]. This vacant code can be assigned to the incoming call. Other parameters are also considered while assigning a code to the new call on the basis of cost.

3) *Single code assignment schemes*: These schemes use only one code from the OVSF code tree, the use of single code requires single rake combiner in the BS and the UE. Some such schemes are as follows:

    a. Leftmost Code Assignment (LCA) scheme: This is one of the simplest schemes in which availability of free code is checked from left side of the code tree [16]. As soon as a free code is found, the call is handled. If no free code is available after thorough search, the call is rejected. The aim is to keep the right side of the code tree vacant for the future calls. This schemes works well when traffic is limited. If traffic is more, there is large code/call blocking.

    b. Random Assignment (RA) scheme [16]: The vacant code which is capable of handling the call is picked randomly from anywhere in the code tree. This scheme also suffers results in larger code/call blocking.

    c. Crowded First Assignment (CFA) scheme: The free code is picked from the crowded portion of the OVSF code tree [16]. When a new call arrives and a number of codes are available, the code whose ancestor has least free capacity, that is, is most crowded is chosen. This assignment scheme produces less code blocking as compared to LCA and RA schemes because the tree is less fragmented. But this scheme is more complex than other two schemes discussed above because the status of the code tree is to be monitored every time with the arrival of new call. This scheme can be used for medium to high traffic load conditions.

d. Class Partition Assignment (CPA): In CPA scheme, the code tree is divided into $L$, $L<8$ number of groups. Each of the $L$ groups is assigned to one of the arrival rate classes. The number of codes in each group depends on the data rate of the classes. Main advantage of CPA scheme is that less number of codes is searched for new calls [12-15]. Limitation is large code blocking and smaller throughput. CPA scheme is also called Fixed Set Partitioning (FSP). Such similar scheme is also given in [39].

e. Fast Dynamic Code Assignment (FDCA): In this scheme, cost is calculated for the assignment of a single code to the incoming call.The cost is calculated by keeping a track of the occupied and vacant codes of the code to be assigned to the call. Cost is reduced by making reassignments of the children of the code to be used.

f. Maximally Flexible Assignment (MFA): In this scheme, flexibility index is defined to measure the capability of the set of codes to be assigned [19]. Two schemes are there – non rearrangable and rearrangable. The goal of first one is to keep the assignable codes in the most compact state after each code assignment without rearranging the codes, that is, to maximize tree's flexibility index. Second is also called blocking triggered scheme because when the call cannot be handled by the first scheme, and code blocking occurs, then this is applied. Codes are rearranged in such a way that all the busy codes are moved to one side of the code tree. Then the codes to be assigned are aggregated to one side and the tree is maximally flexible.

There is also a variant to DCA, DCA-CAC (dynamic code assignment with call admission control) [20]. Three different variants are there, first is resource partitioning, second is optimal DCA-CAC and third is the hybrid of the both, which is done to get a good tradeoff between throughput and complexity.

Multi code schemes are proposed in literature to reduce internal and external fragmentation. The use of multiple codes for single call increases hardware complexity [32, 50]. In addition to the internal fragmentation problem, while connections are arriving and leaving the system, an OVSF code tree may become too fragmented to support newly arrived calls even if there are sufficient spaces in the code tree. In multi code schemes, multiple codes are used to handle the calls [33,36-38, 41]. The non quantized calls are handled by breaking them into quantized fractions. Different types of multi code schemes are:

a. Multicode Multirate Compact Assignment (MMCA): In this scheme compact index is calculated for the traffic with different QoS requirements [17]. It has following important properties:
    - MMCA does not perform code rearrangement and is therefore simple.
    - MMCA provides priority differentiation between real time calls and data packets.
    - MMCA supports mobile terminals with different multi-code transmission capabilities.
    - MMCA balances transmission qualities among the multiple codes assigned to the same user.
    - MMCA supports multi-rate real time calls and keeps the code tree as flexible as possible in accepting new multi-rate calls

b. Time Based Code Allocation: In this scheme, service time is considered and based upon that remaining time is calculated [18]. Then the remaining time of parent nodes are compared and the maximum one is picked for assignment. Cost consideration is also done while doing reassignments.

Along with the above code assignment and reassignment schemes, some more assignment and reassignment strategies are given in literature. In the rotated single code assignment scheme [40], linear code chains (LCCs) and non-linear code trees (NCTs) are identified and the code assignment gives lesser blocking probability and reassignment cost. It uses the unsequence property of linear code chains to design a new code assignment and reassignment algorithm. The scheme initially attempts to allocate request code to LCCs and then tries to allocate them to NCTs. The code blocking is reduced along with the reassignment cost. In recursive fewer codes blocked (RFCB) [21] design, code blocking can be reduced with careful selection of optimum code among possible candidate codes during the assignment process. It works on the top of CFA design, and the criterion for the selection of a candidate code is number of parents blocked which were free earlier. In fast dynamic code assignment (FDCA) [22], the code assignments are done considering the cost for an OVSF code allocation, where the cost is determined by keeping a track on the number of available and occupied descendant codes of the code. The code is assigned by reassigning occupied descendant codes for a requested data rate with least cost function. To improve the system throughput, a scheduling design is given in [23, 30] that dynamically assign OVSF codes to mobile users on a timeslot basis maximizing system throughput. Also, the average data rate guarantee is provided to each mobile user. There is no need for a mobile user to overbook its required rate. In [24], a code selection design is given that can be used in the assignment process with or without reassignments. The selection is made using a simple measure which differentiates each code irrespective of the incoming or reassigned call rate. It simply counts the number of new codes that will be blocked due to a potential allocation of the candidate code. In [25], the code assignment and reassignment is based on QoS requests. If there are multiple options for the vacant code, the optimum code is one whose ancestor code has the maximum free capacity. The used codes are scattered in the code tree, and therefore, there is space for a call to raise its data rate without the need of a code reassignment.

Fairness Issues are also considered in [31, 35]. The rotated single code assignment scheme is extended to multi-code rotated code assignment in [41]. The non-blocking OVSF codes [42-45] reduce the code blocking to zero. There are three categories of NOVSF codes with the properties given below.

1. Time multiplexing is used to divide the code usage time into slots. The slots of the code time can be used by one or more channels.
2. The OVSF codes are reorganized such that all the codes are orthogonal. The OVSF code trees with initial 4 or 8 codes are generated.
3. In the third category, the OVSF codes are generated such that there is no limit on the upper bound of SF.

The performance of fixed and dynamic code assignment schemes with blocking probability constraint is given in [46, 52-58]. The throughput performance is proved to be better. Non rearrangeable compact assignment (NCA) [47] makes the code assignment compact so that the remaining assignable codes are most flexible to accommodate future multirate calls.

The code assignment scheme proposed in [48], the code with the least number of parents blocked for a candidate code allocation is used for assignment. The recent advancements in OVSF are given in [59-66].

## 1.6 Problem Formulation

All the schemes proposed in literature aim in efficient utilization of the OVSF code tree. But the code blocking is not reduced in maximum cases. The thesis aims in reducing the code blocking and handling maximum number of calls efficiently by utilizing the code tree to its full capacity. The thesis is organized as follows:

Chapter 2 discusses single code as well as multi code approach to handle the calls. A parameter called "scattering index" is defined for both the cases which gives the information about the availability of vacant codes in a particular region of the code tree. For multi code assignment, two algorithms are proposed based on the number of available rake receivers and the particular layer in which the new code to be assigned lies. Simulation is done using multi code schemes and varying the number of RAKE receivers [28]. At the end the results are compared with existing single code and multi code assignment schemes on the basis of traffic load and blocking probability and blocking probability of the proposed schemes is found minimum.

Chapter 3 discusses compact code assignment schemes. Here we have reserved some area for the incoming call depending on the priority number, which is called code reservation scheme. Another scheme is proposed where assignment is done by choosing the group leaders. The third compact code assignment scheme we have used the aggregate capacity under the ancestors to assign a new code to the incoming call. Various proposed schemes are compared with the existing schemes in terms of traffic load and blocking capacity and the blocking capacity is reduced in all the proposed schemes.

Chapter 4 discussed the division of calls on the basis of type of call, i.e, real time calls or data calls. Based upon the type of call, bandwidth is taken into account to assign a new code to the incoming call. Guaranteed capacity and the available capacity of the code tree are calculated and Code index is found for each code. Depending on the status of the code index, code is assigned to the incoming call. Also, we have proposed a scheme here to divide the code tree into three different regions – one for real time calls, second for data calls and the third one for the mixed calls. Algorithms are explained with the help of flow charts and the results are calculated taking into account the average traffic load and code tree utilization. The proposed schemes utilize the code tree most efficiently as compared to existing schemes. Also, the results are found calculating the number of calls handled and the average traffic load available. The number of calls handled is maximum by the proposed schemes as compared to other existing schemes.

Chapter 5 discusses some more single code and multi code assignment schemes.

The thesis is concluded in Chapter VI. The aim of all the schemes proposed till date is the efficient utilization of the code tree by handling maximum number of calls. Also the blocking probability should be minimized. The proposed work takes into account various such parameters and proves that the code tree can be efficiently utilized by reducing the blocking probability.

# CHAPTER 2

# REDUCTION IN CODE BLOCKING USING SCATTERED VACANT CODES

## 2.1 Introduction

OVSF codes are depicted in the form of a tree with the properties discussed in Chapter I. This code tree should be efficiently utilized so that maximum number of calls can be handled. Any new call can use either a single code from this code tree of more than one code. In this chapter we have proposed two compact code assignment schemes (a) Single code (b) multi code. Multi code scheme is further divided into two categories

- Code/Rake limited scenario
- Minimum code blocking scenario, which uses maximum number of codes

Definition: Scattered vacant codes are the ones which lie in the neighborhood of occupied codes in same layer. They are called scattered because they are distributed throughout the code tree, and block more ancestors, which is undesirable for future calls. These scattered vacant codes decrease efficiency and throughput of OVSF based CDMA systems. The code scattering can be reduced using three different ways: 1) efficient code management, 2) reassignment of busy codes, and 3) using multiple codes for a call. The code management reduces the code scattering by assigning optimum code from a set of candidate codes at call arrival. Our compact single code assignment scheme allocate the codes in such a way that congested part of tree becomes more congested, facilitating some vacant area for future rate calls. The compact assignment is based on *scattering level* and *elapsed time* of the already occupied codes. The design is similar to crowded first scheme [16] but with a difference that while crowded first scheme is based on branch wise compactness, our scheme is based on compactness at specific level. In reassignment schemes, the reduction in scattering is obtained at the cost of more overhead and complexity. The probability of reassignment is quite high for low to medium traffic load conditions. The use of multiple codes for single call requires multiple rake combiners at base station (BS) and user equipment (UE) which increases to cost and complexity.

### 2.2 Single code assignment

At a specific time instant the busy codes and vacant codes are known in the code tree. A code $C_{l,n}$ is part of group with $b = 2^{l'}, (l' < l)$ consecutive vacant codes if following is true.

Condition 1: Code $C_{l+l',\lceil n/2^{l'}\rceil}$ and all its children in layer *l* are vacant.

**Table 2.1:** Deriving consecutive vacant codes groups in layer 2 to 8 from consecutive vacant code groups in layer 1

| **Layer (*l*)** | | | | | | | |
|---|---|---|---|---|---|---|---|
| *l*=1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| $N_{1,1}$ | $N_{2,1}= N_{1,2}$ | $N_{3,1}= N_{1,4}$ | $N_{4,1}= N_{1,8}$ | $N_{5,1}= N_{1,16}$ | $N_{6,1}= N_{1,32}$ | $N_{7,1}= N_{1,64}$ | $N_{8,1}= N_{1,128}$ |
| $N_{1,2}$ | $N_{2,2}= N_{1,4}$ | $N_{3,2}= N_{1,8}$ | $N_{4,2}= N_{1,16}$ | $N_{5,2}= N_{1,32}$ | $N_{6,2}= N_{1,64}$ | $N_{7,2}= N_{1,128}$ | |
| $N_{1,4}$ | $N_{2,4}= N_{1,8}$ | $N_{3,4}= N_{1,16}$ | $N_{4,4}= N_{1,32}$ | $N_{5,4}= N_{1,64}$ | $N_{6,4}= N_{1,128}$ | | |
| $N_{1,8}$ | $N_{2,8}= N_{1,16}$ | $N_{3,8}= N_{1,32}$ | $N_{4,8}= N_{1,64}$ | $N_{5,8} = N_{1,128}$ | | | |
| $N_{1,16}$ | $N_{2,16}= N_{1,32}$ | $N_{3,16}= N_{1,64}$ | $N_{4,16}= N_{1,128}$ | | | | |
| $N_{1,32}$ | $N_{2,32}= N_{1,64}$ | $N_{3,32}= N_{1,128}$ | | | | | |
| $N_{1,64}$ | $N_{2,64}= N_{1,128}$ | | | | | | |
| $N_{1,128}$ | | | | | | | |

Condition 2: Code $C_{l+l'+1,\lceil n/2^{l'+1}\rceil}$ is blocked.

**Table 2.2:** Listing codes in Figure 2.1 into groups according to number of consecutive vacant codes

| **Layer ($l$)** | **Codes in group $N_{l,1}$** | **Codes in group $N_{l,2}$** | **Codes in group $N_{l,4}$** | **Codes in group $N_{l,8}$** | **Codes in group $N_{l,16}$** | **Codes in group $N_{l,32}$** | **Codes in group $N_{l,64}$** |
|---|---|---|---|---|---|---|---|
| 1 | $C_{1,6}$,$C_{1,8}$,$C_{1,14}$,<br>$C_{1,29}$, $C_{1,31}$ | $C_{1,9}$,$C_{1,10}$,<br>$C_{1,25}$,$C_{1,26}$ | 0 | 0 | $C_{1,49}$…..$C_{1,64}$ | 0 | 0 |
| 2 | $C_{2,5}$,$C_{2,13}$ | 0 | 0 | $C_{2,25}$,…..$C_{2,32}$ | 0 | 0 | N A |
| 3 | 0 | 0 | $C_{3,13}$,… …$C_{3,16}$ | 0 | 0 | N A | N A |
| 4 | 0 | $C_{4,7}$,$C_{4,8}$ | 0 | 0 | N A | N A | N A |
| 5 | $C_{5,4}$ | 0 | 0 | N A | N A | N A | N A |
| 6 | 0 | 0 | N A | N A | N A | N A | N A |
| 7 | 0 | N A | N A | N A | N A | N A | N A |

Let scattering index $N_{l,b}$ denotes the number of the codes in layer $l$ within a group of $b$

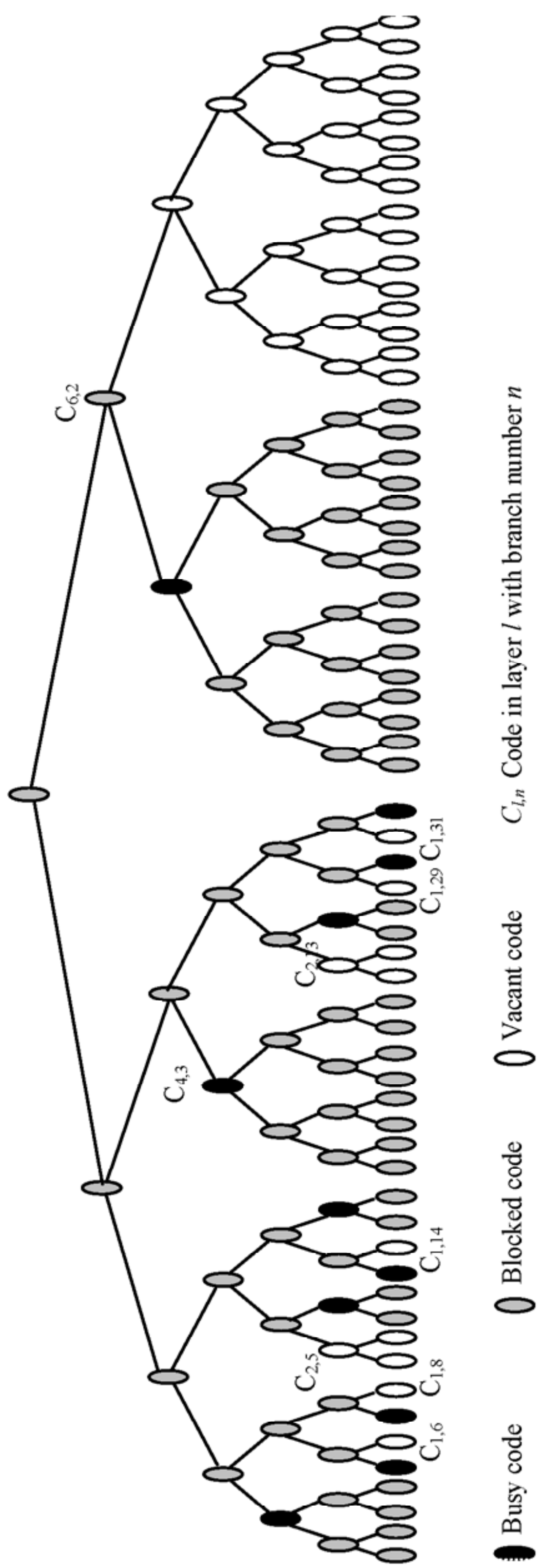


**Figure 2.1:** Illustration of the single code and multi code sharing scheme for handling quantized and non quantized data rates

consecutive vacant codes whose ancestors in layers $l+\log_2(b)$ to $L$ are same. For a new quantized call $2^{l-1}R$ the design identifies the vacant code $C_{l,n}$ in layer $l$ that is within the group $N_{l,b}$ vacant code group, where $b=min(1,2^{l-1})|N_{l,b}\neq 0$, i.e. the code with the minimum consecutive vacant codes is the candidate for handling new call with rate $2^{l-1}R$. The index $N_{l,b}$ is a measure of scattering in the OVSF code tree. The use of vacant code from optimum $N_{l,b}$ group guarantees the code usage from the most congested portion. This maximizes the probability of handling high rate calls in future. If $N_{l,b}$ is known for layer $l$, it can be shown that for a layer $l'$

$$N_{l+l',b/2^{l'}} = N_{l,b} \text{for}\, l' \leq L-l \tag{2.1}$$

Therefore it is sufficient to find the scattering index in layer 1, i.e., $N_{1,b}$ as higher layer indices can be derived from it. The relationship between $N_{1,b}$ and $N_{l,b}, l\in[2, L=8]$ is given in Table 2.1 for WCDMA system with L=8 layers. Also, as the multiple candidate codes are available in optimum vacant code group, the most appropriate vacant code can be found by the elapsed time information of busy neighbor of codes within the minimum consecutive vacant group. For a vacant code $C_{l,n}$ in group $N_{l,b}$, define *neighbor* codes as those codes in the layer $l$, which are the children of the code $C_{l+\log_2(b),\lceil l/2^{\log(b)}\rceil}$. Find the number of busy children of the code $c_{l+\log_2 b,\lceil n/2^{\log(b)+1}\rceil}$ (say $N$). For all $N$ busy codes, find the average elapsed time $\sum_{i=1}^{N} t_i / N$, where $t_i$ represents the elapsed time of the call handled by $i^{th}$ busy code. Repeat the procedure for all the codes in the group $N_{l,b}$. The children of the parent code of $C_{l,n}$ with minimum average elapsed time is the one used for handling new call. Hence the code whose sibling(s) has the latest call arrival will be used so that all these codes become vacant at similar time. Therefore the crowded portion remains more crowded, increasing code utilization and better handling of high rate calls. To illustrate the code assignment scheme, consider a 7 layer code tree in the Figure 2.1. The consecutive vacant codes groups are given in the Table 2.2. If a $2R$ user arrives, the vacant code from layer 2 is required. The vacant codes availability is with groups $N_{2,1}$ and $N_{2,8}$. The assignment scheme picks any one code from the group $N_{2,1}$ (either code $C_{2,5}$ or code $C_{2,13}$). Further, the new calls with rates $32R$ and $64R$ will not be handled due to absence of vacant codes.

For a new non quantized call *kR*, $k \ne 2^{l-1}$, there is no code with the code capacity *kR* in the OVSF tree. The optimum code lies in the layer $l'+1, l' = \min(l'') \mid k \le 2^{l''}$. The code wastage capacity is given by

$$WC = (2^{l'} - k)R \tag{2.2}$$

For *k* close to $2^{l-1}$, the single code assignment produces large wastage capacity and hence multi code assignment is to be used to avoid this wastage.

## 2.3 Multi code assignment

The unused capacity in $l^{th}$ layer of the code tree (say$A_l$) can be defined as

$$A_l = \sum_{i=1}^{l} N_{l,2^{i-1}} \times 2^{i-1} R \tag{2.3}$$

If at the arrival of new call with rate $2^{l-1}R$ there is no vacant code in layer *l*, and $A_l \ge 2^{l-1} R$, the single code assignment fails and the call can be handled only with multiple codes. There are two ways to handle call with multiple codes, (*i*) use of minimum codes, (*ii*) use of maximum codes.

### *2.3.1 Call handling with minimum number of codes*

At the arrival of quantized call $2^{l-1}R$ in an *m* rake system, there are two possibilities to handle new call, (*i*) $l \le m$, in this case the codes with capacities $2^i R, i \in [0, l-1]$ are used, (*ii*) $l > m$ where the codes with capacities $2^i R, i \in [l-m, l-1]$ are eligible candidates. The code assignment is done using successive capacity reduction as follows. Let $p_{l-i}$ is the number of vacant codes in layer *l-i* to handle full/partial $2^{l-1}R$ call capacity. Also, let $P_{l-i}, \max(P_{l-i}) = 2^{i-l}$ represents layer *l-i* codes used to handle full/partial call capacity. Starting with layer *l* the fraction of call capacity handled upto layer *l-x* is $\sum_{i=0}^{x} P_{l-i} \times 2^{l-i} R$, and layer *l-x* needs to be checked if, $\sum_{i=0}^{x} P_{l-i} < m$. The remaining capacity to be handled by layers 1 to *x*-1 (say$Q_{x-1}$) is

$$Q_{x-1} = (2^{l-1} - \sum_{i=0}^{x} P_{l-i} \times 2^{l-i})R \quad (2.4)$$

Further, the codes used in different layers should be the ones which belong to minimum consecutive vacant code groups. For a layer *l-i*, if there are *j* consecutive vacant code groups $N_{l-i,a_k}, k \in [1, j]$ and $a_k = 2^k \mid 2^k \leq 2^{L-l+i}$, $P_{l-i}$ number of codes in layer *l-i* should be used from the consecutive vacant code group $N_{l-i,a_{k1}}$ where $a_{k_1} = \min(a_k)$. If $P_{l-i} > N_{l-i,a_{k1}}$, some vacant codes are required from second optimum consecutive vacant codes group $N_{l-i,a_{k2}}$ where $a_{k_2} = \min(a_k)$ and $a_{k_2} > a_{k_1}$. The procedure is repeated for

**Table 2.3:** Relationship between number of codes, capacity handled, and remaining capacity for new call $2^{l-1}R$ in various layers

| **Layer (*l-i*) *i*=[0..*l*]** | **Number of codes used ($P_{l-i}$)** | **Capacity handled by layers *l-i*** | **Capacity handled by layers 1 to *l-i*-1** |
|---|---|---|---|
| *l* | 0,1 | $P_l \times 2^{l-1}$ | $2^n - P_l \times 2^{l-1}$ |
| *l*-1 | 0,1,2 | $P_{l-1} \times 2^{l-2}$ | $2^n - \sum_{i=0}^{1} P_{l-i} \times 2^{l-(i+1)}$ |
| *l*-2 | 0,1,2,3,4 | $P_{l-2} \times 2^{l-3}$ | $2^n - \sum_{i=0}^{2} P_{l-i} \times 2^{l-(i+1)}$ |
| …. | …. | …. | …. |
| 1 | 0,1,2… $2^{l-1}$ | $P_1$ | Nil |

maximum $P_{l-i}$ steps. For any layer *l-i,* the relationship of number of codes, capacity handled and remaining capacity is given in Table 2.3.

The non quantized rates in the form of $k_1R, k_1 \neq 2^{l-1}$ ($k_1R$ is used instead of $kR$ for uniform notation) are converted to quantized as follows. Find $\min(l_1) \mid k_1 \geq 2^{l_1} R$. Calculate $k_2 = k_1 - 2^{l_1}$. Starting with $l_1$ and $k_2$, the procedure can be extended to find $\min(l_i) \mid k_i \geq 2^{l_i} R$ and $k_{i+1} = k_i - 2^{l_i}$ till $k_{i+1}=2^n$. The quantized rates $2^{l_1} R, 2^{l_2} R, \ldots 2^{l_{i+1}} R$ are handled as discussed earlier. Assume that the number of rakes required to handle rate components $2^{l_1} R, 2^{l_2} R, \ldots\ldots 2^{l_{i+1}} R$ are $m_1, m_2, \ldots, m_{i+1}$. The new call can be handled only if $\sum_{j=1}^{i+1} m_j \leq m$. The algorithm of the design is given as

```
1. Rejected calls=0;
2. Enter input parameters like user rate kR, number of rakes 'm', number of layers 'L' etc.
3. If (rate is quantized, i.e. $kR=2^{l-1}R$)
       3.1 Find the number of codes required $(P_i), 1 \leq i \leq l-1$ total codes available($p_i$)
       3.2 Use the vacant codes $P_i$ in layer l-i from least consecutive vacant code groups
       3.3 Go to step 2
   Else
       3.1 Convert non quantized rate into quantized rate fractions $2^{l_1} R, 2^{l_2} R, \ldots 2^{l_{i+1}} R$
           requiring rakes $m_1, m_2, .. m_{i+1}$.
       3.2 If $\sum_{j=1}^{i+1} m_j \leq m$
               3.2.1 Use codes from layers $l_1$-1, $l_2$-1, ...$l_i$ to handle new call. All the codes
                     must be from the minimum consecutive vacant code group.
               Else
               3.2.1 Rejected calls=Rejected calls+1
               End
       3.3 Go to step 2
```

*End*

The flow chart for call handling using minimum number of codes is shown in Figure 2.2.

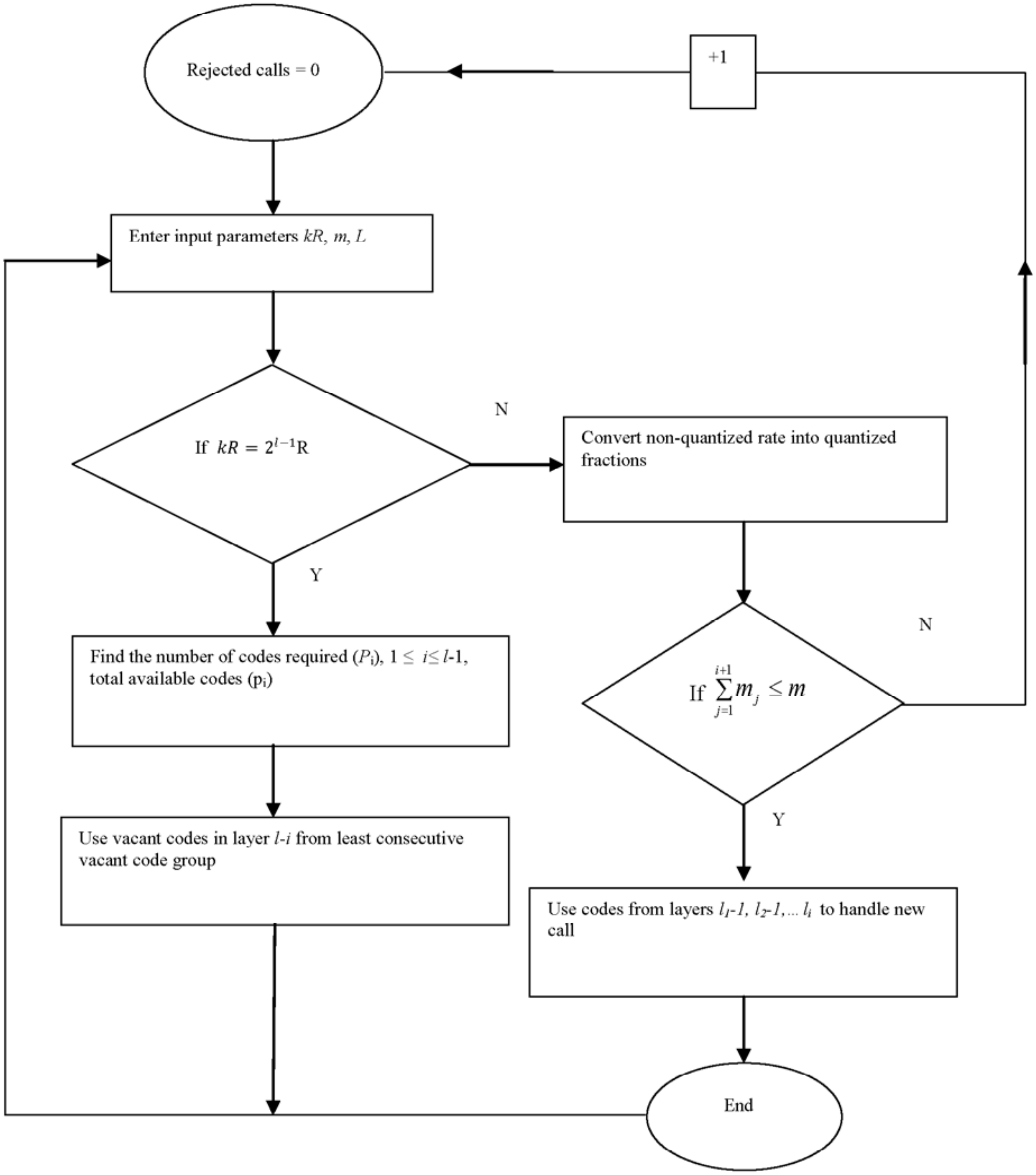


**Figure 2.2** Flowchart for call handling using minimum number of codes.

### *2.3.2 Call handling with maximum number of codes*

At the arrival of new call $2^{l-1}R$ in $m$ rakes system the essential requirements to reduce code blocking are

- Maximum number of rakes should be utilized.
- Maximum low rate codes should be used.
- Each code selected in the multi code, is used from the consecutive vacant code group $N_{l,b}$ with minimum possible $b$.

For a new quantized call $2^{l-1}R$, if $2^{l-1}R \le A_1$ and $m < 2^{l-1}$, the code tree has enough capacity to handle new call. There are two possibilities, (*i*) $l > m$ and (*ii*) $l \le m$

(*i*) If $l > m$, the codes with capacities $2^{l-1}R, i \in [1,m]$ are the candidates to handle new call. Construct a vector $Z = [z_1, z_2, \ldots ..z_m]$, where $z_i$ represents the capacity fraction handled by $i^{\text{th}}$ rake in units of $R$ kbps. Initially put the value of each coefficient in $Z$ equal to $2^{l-m}$. Define $B_1$ as

$$B_1 = 2^{l-1} - (m-1) \times 2^{l-m} \tag{2.5}$$

For integer $P_1$, find $p_1 = \max(P_1) \mid 2^{P_1} \le B_1$. In vector $Z$, coefficient $z_1$ is assigned the value equal to $2^{p_1}$. The vector $Z$ becomes $Z=[2^{p_1}, 2^{l-m}, \ldots ..2^{l-m}]$. For $2 \le i \le m-2$, calculate $B_i = 2^{p_{i-1}} - (m-i) \times 2^{l-m}$ and $z_i = 2^{p_i}$ and $p_i \max(P_i) \mid 2^{P_i} \le B_i$. The vector Z is redefined as $Z = [2^{p_1}, 2^{p_2}, \ldots 2^{p_{m-2}}, 2^{l-m}, 2^{l-m_1}]$. Therefore in maximum code scattering design, one code is used from each of the layer $l$-1, $l$-2,…$l$-($m$-2) and 2 codes are used from the layer $l$-($m$-1), and no vacant code is used from layers $l$-$m$ to 1. Further, the vacant codes in each layer should be used from least consecutive vacant code group(s).

(*ii*) For $l \le m$, the codes with capacities $2^iR, 0 \le i \le l-1$ are the candidates to handle new call. Considering $Z = [z_1, z_2, \ldots . z_m]$, find all $j, 1 \le j \le m-2$ for which $(2^{l-1} - \sum_{i=1}^{j} a_j) > m - j$ where $a_j = 2^{l-1}/2^j$. The vector Z becomes $Z = [2^{l-1}/2, 2^{l-1}/2^1, \ldots ..2^{l-1}/2^j, z_{j+1}, \ldots . z_m]$. Find $j$, $j = \min[1, m-2]$ for which $2^{l-1} - \sum_{i=1}^{j} a_i = m - j$, where $a_i$ is in the form of $2^n$, $n \in [0, l-2]$. The coefficients $z_i$ in vector $Z$ becomes

$$z_i = \begin{cases} a_i, & i = 1..j \\ 1, & i = j+1..m \end{cases} \qquad (2.6)$$

The vector $z_i$ represents the capacity fraction handled by $i^{th}$ rake.

If $m \geq 2^{l-1}$, maximum $2^{l-1}$ codes of rate $R$ are used to handle new call. The non quantized rates are converted into quantized rates as discussed earlier. The algorithm of the design is given as

```
1. Rejected calls=0;
2. Enter input parameters like user rate kR, number of rakes 'm', number of layers 'L' etc.
3. If (rate is quantized, i.e. $kR=2^{l-1}R$)
    3.1 If $m<2^{l-1}$
        3.1.1 If $l>m$
                3.1.1.1 Construct vector $Z=[z_1,z_2,.....z_m]$, where $z_i = 2^{p_i}$
                3.1.1.2 For rate fraction $2^i$, $i \in [1,m]$, assign codes from minimum consecutive vacant group
            Else
                3.1.1.1 Construct $Z=[z_1,z_2,.....z_m]= [2^{l-1}/2, 2^{l-1}/2^1, 2^{l-1}/2^2,..., 2^{l-1}/2^j, z_{j+1}, ...., z_{j+}1]$, where $z_i = 2^{p_i}$
                3.1.1.2 Assign codes from minimum consecutive vacant group
            End
        Else
        3.1.1 Use m codes of rate R to handle the call
        End
    3.2 Go to step 2
   Else
    3.1 Convert non quantized rate into quantized fractions $2^{l_1}, 2^{l_2}, .....2^{l_{i+1}}$ requiring rakes $m_1, m_2,..m_{i+1}$.
    3.2 If $\sum_{j=1}^{i+1} m_j \leq m$
```

*3.2.1 For each $m_i$ use step 3.*

*Else*

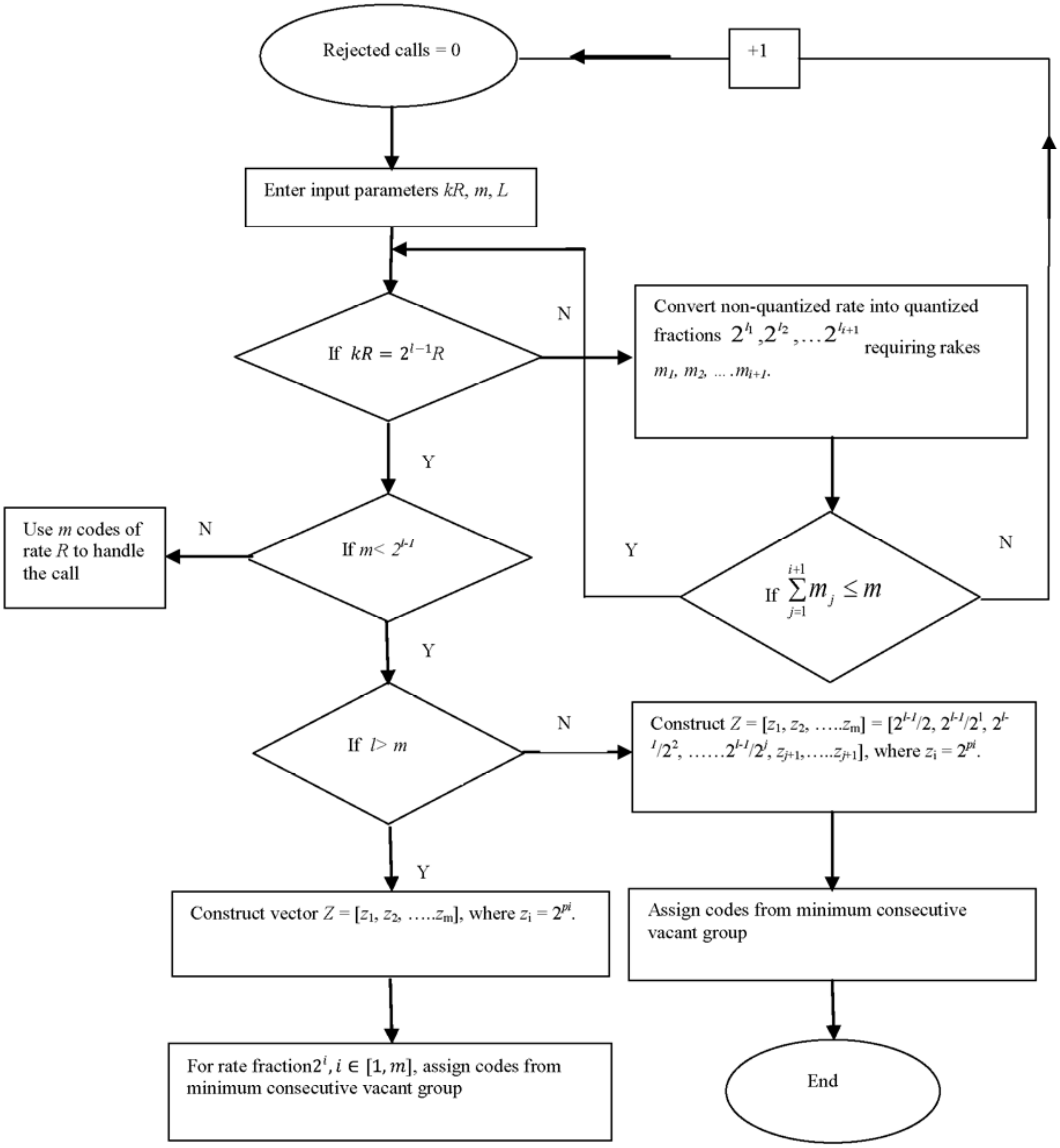


**Figure 2.3:** Flowchart for handling the calls using maximum number of codes

*3.2.1 Rejected calls=Rejected calls+1*

*End*

*3.3 Go to step 2*

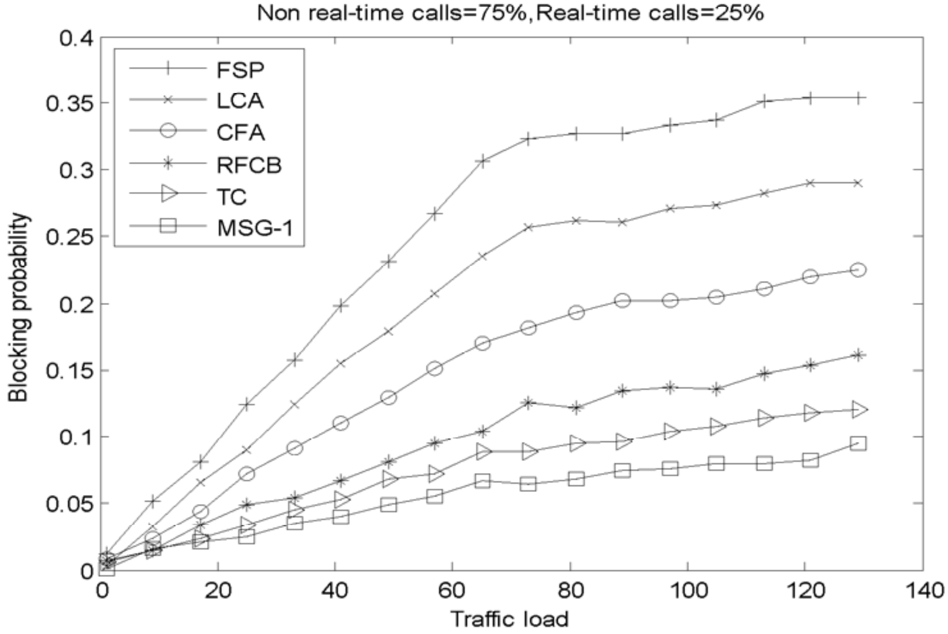
*3.4 End*

To illustrate the multi code assignment scheme consider the 7 layer tree of Figure 2.1 assuming that the system is equipped with 4 rakes. If a new call with rate 16$R$ arrives, i.e., $l$ is equal to 5, and $l>m$, for use of maximum scattered codes, the initial value assigned to the four rakes are $Z$=[2,2,2,2].As per section 2.3.2, the value of $p_1$, $p_2$, $p_3$ and $p_4$ becomes 3, 2, 1, and 1 respectively. The vector $Z$ becomes [8,4,2,2] and the codes $C_{4,7}$, $C_{3,15}$, $C_{2,5}$ and $C_{2,13}$ are used to handle capacity portion 8$R$, 4$R$, 2$R$ and 2$R$ respectively. If instead of 16$R$, the new call of rate 8$R$arrives, i.e., $l$=4, then as $l \le m$ and $2^{l-1} > m$, the vector Z becomes [4,2,1,1] as per section 2.3.2. The codes used are $C_{3,13}$, $C_{2,5}$, $C_{1,6}$, and $C_{1,8}$ respectively. In the above two examples the effect of elapsed time of busy siblings is not considered, otherwise the selected code may be different. The flow chart to handle calls using maximum number of codes is given in Figure 2.3.

## 2.4 Simulation and Results

For simulation, 8 layer OVSF code tree (as per WCDMA specifications) is considered. Only quantized rates $2^{l-1}R, l \in [1,8]$ are considered with rates $R$, 2$R$, 4$R$, 8$R$ are treated as real time calls and rates 16$R$, 32$R$, 64$R$, 128$R$are treated as non-real time (best effort calls). Let $\lambda_l, (l \in [1,8])$ is the arrival rate of $2^l R$ calls. The total arrival rate is of the system is $\lambda = \sum_{l=1}^{8} \lambda_l$. The service time is assumed to be exponentially distributed with average value $1/\mu$ (service rate for all classes is assumed to be same equal to $\mu$ ). For simulation, total traffic load $\rho$ (equal to $\lambda/\mu$) is varied from 1 to 128 calls per unit of time. The arrival rate is assumed to be varying between 1 to 128 calls per unit of time and service time is assumed to be 1 unit of time and service time is 3 units of time. In the single code assignment if rate $kR$, $2^{l-2} < k \le 2^{l-1}$, arrives the vacant code is required from layer $l$. Also the current $l^{th}$ layer $\lambda_l$ is updated as

$$\lambda_l = \lambda_l + 1 \tag{2.7}$$



FSP- Fixed Set Partitioning, LCA- Left Code Assignment, CFA- Crowded First Assignment, RFCB- Recursive Fewer Codes Blocked, TC- Time Based Code Scheme, MSG-1- Maximum Scattered Group(ours)

**Figure 2.4:** Single code assignment, more non real time calls

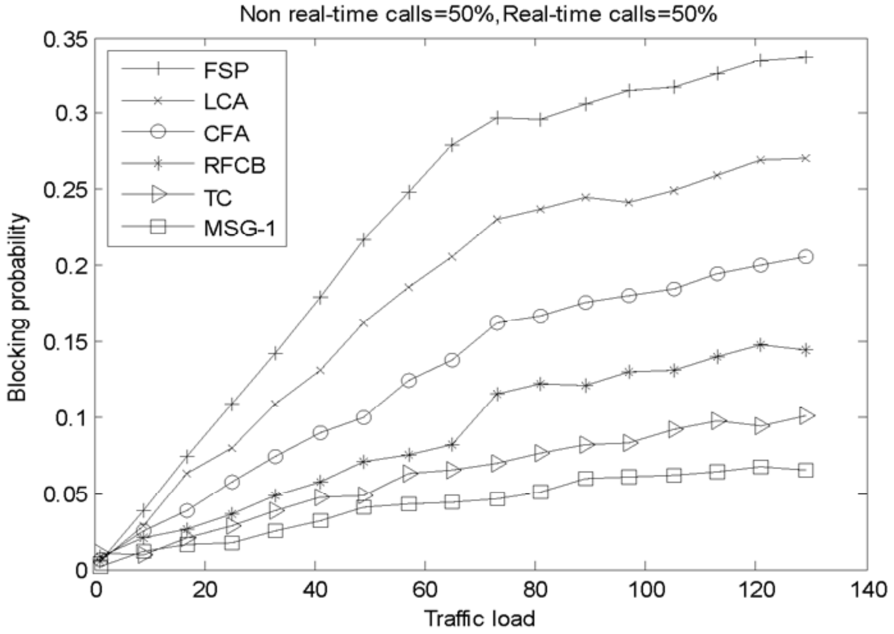
In multi code assignment with $m$ rakes, if rate $k_1R$, $2^{l-2} < k_1 \le 2^{l-1}$ arrives, finds $k_2$, $k_3, \ldots k_{m+1}$ as in section 3.2.1. If within $m$ steps, $\sum_{j=2}^{p} k_j = k \mid p \le m+1$, use of least codes will update the $\lambda_{l-1}, \lambda_{l-2}, \ldots \lambda_{l-m}$ to value given by

$$\lambda_{l-1}, \lambda_{l-2}, \ldots \lambda_{l-m} = \lambda_{l-1} + a_1, \lambda_{l-2} + a_2, \ldots . \lambda_{l-m} + a_m \tag{2.8}$$

In Equation (2.8) the coefficient $a_j, 1 \le j \le m$ can take values 0 or 1. The value of $a_j$ is 0 if $k_j$ is 0 else $a_j$ is 1.

If the multi code assignment with minimum future scattering is used, the codes within the minimum consecutive vacant codes in each layer are picked. If the vector $Z = [z_1, z_2, \ldots z_t], t < m$, is calculated as per section 3.2.2, find $l_i = \log_2(z_i) + 1$ for each $z_i$. The arrival rate is updated as

$$\lambda_{l_i} = \lambda_{l_i} + 1, 1 \le i \le t \tag{2.9}$$

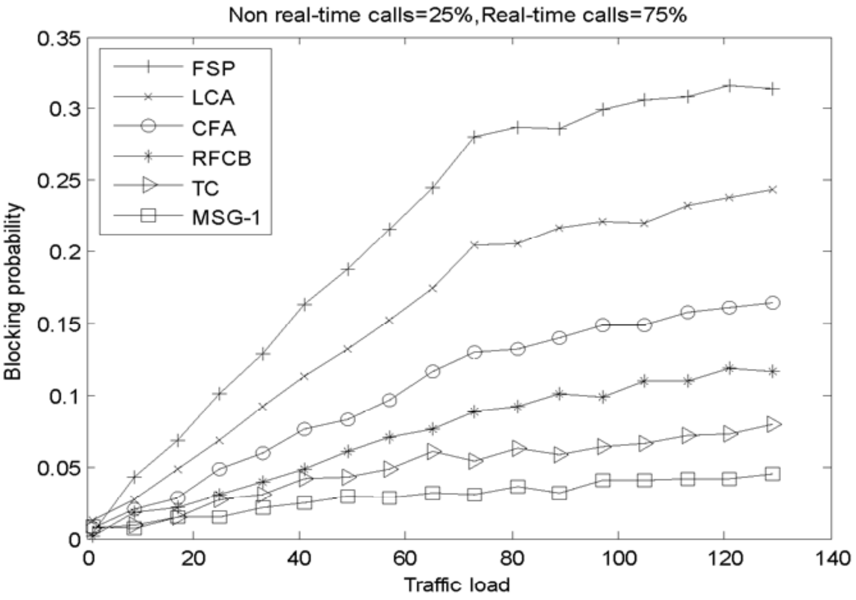


FSP- Fixed Set Partitioning, LCA- Left Code Assignment, RFCB- Recursive Fewer Codes Blocked, TC- Time Based Code Scheme, MSG-1- Maximum Scattered Group(ours)

**Figure 2.5:** Single code assignment uniform distribution

Also for 8 layer code tree the number of servers (codes) in layer $l$ are $G_l = 2^{8-l}, l = [1,2...8]$. The total codes (servers) in the system assuming eight set of classes are represented by vector $G=[G_1,G_2,G_3,G_4,G_5,G_6,G_7,G_8]$. Also, the maximum number of servers used per calls is equal to the number of rakes '$m$'. If the traffic load for the $l^{th}$ class is denoted by $\rho_l = \lambda_l / \mu$, and the average traffic load is given by $\rho = \sum_{l=1}^{8} \rho_l$. The code blocking for the $l^{th}$ class is defined by

$$P_{B_l} = \frac{\rho_l^{G_l} / G_l!}{\sum_{n=1}^{G_l} \rho_l^n / n!} \tag{2.10}$$



FSP- Fixed Set Partitioning, LCA- Left Code Assignment,CFA- Crowded First Assignment, RFCB- Recursive Fewer Codes Blocked TC- Time Based Code Scheme, MSG-1- Maximum Scattered Group(ours)

**Figure 2.6:** Single code assignment, more real time calls

The average code blocking for 8 class system is

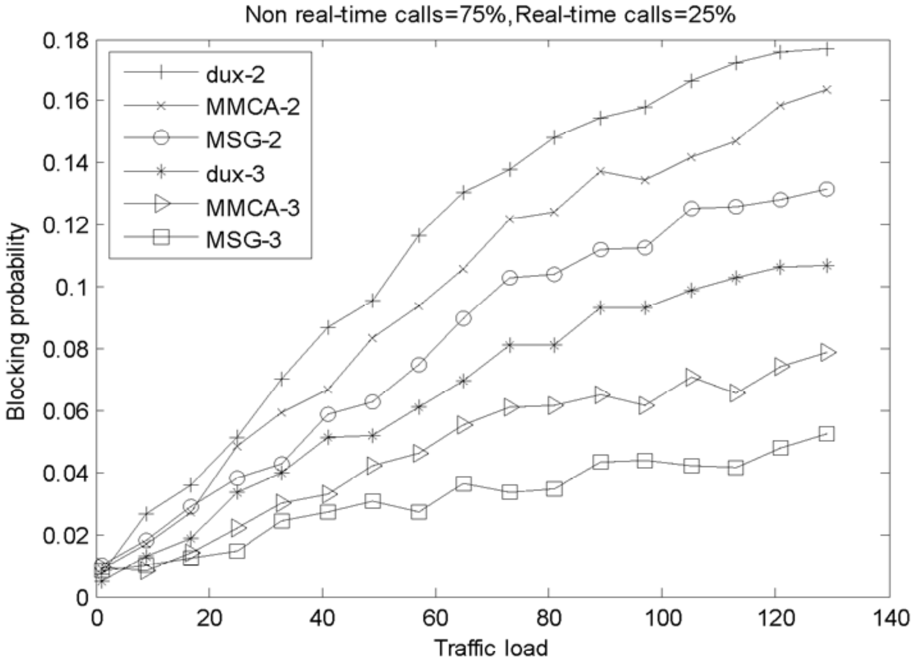
$$P_B = \sum_{l=1}^{8} (\lambda_l / \lambda) P_{B_l} \tag{2.11}$$

Let the arrival distribution for the eight classes is given by [$p_1$, $p_2$], where $p_1$ and $p_2$ is the probability of arrival of real time and non real time calls. Three arrival distributions are considered as given below

- [0.75,0.25], real time calls dominates the traffic
- [0.5,0.5], uniform distribution of real time and non real time calls
- [0.25,0.75], non real time calls dominates the traffic

The single code assignment blocking comparison for the proposed maximum scattered group (MSG-1, and 1 indicates single code facility for new call) is done with fixed set partitioning scheme(FSP)[16], leftmost code assignment(LCA)[16], crowded first assignment (CFA)[16], fewer codes blocked scheme [20] and time code (TC) [18] schemes. Results in Figure 2.4, show the blocking in the proposed scheme with more non real time calls and less real time calls.

Figure 2.5 shows the comparison of the proposed design with other single code

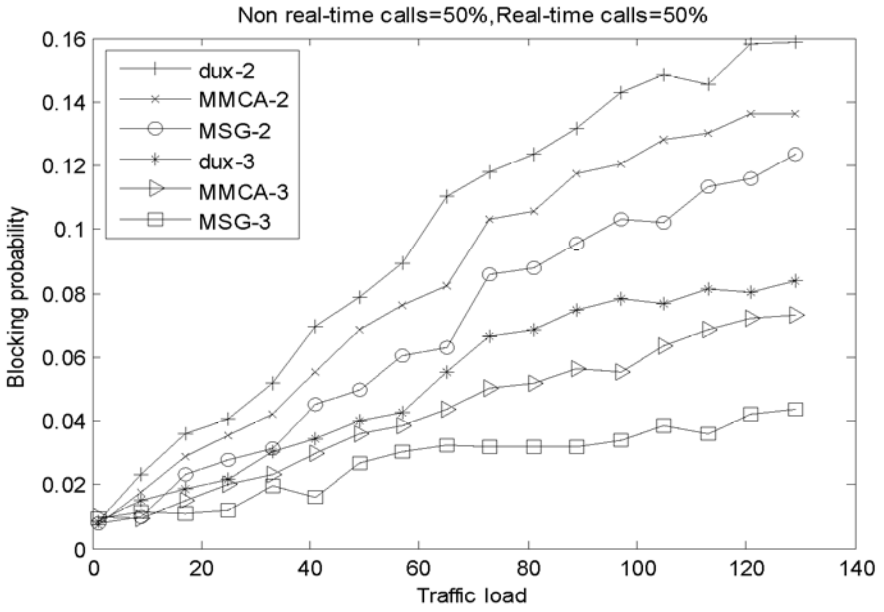


dux-*n*- Decreasing , United, No reassignments (using *n* codes), MSG-*n*- Maximum Scattered Group (using *n* codes)(ours)
MMCA-*n*- Multicode Multirate Code Assignment (using *n* codes)

**Figure 2.7:** Multi code assignment, more non real time calls

schemes. In this there is uniform distribution of real time as well as non real time calls.

In Figure 2.6 the real time calls dominate and non real time calls are less. The multi code assignment blocking comparison of the maximum scattered group (MSG-*n*), where *n* indicates *n*OVSF codes facility for handling new call is done with multi code scheme given in [6] and [17]. The multi code assignment in [6] is represented by five dux-*n*, where *d*, *u* and *x* stands for decreasing strategy, united strategy and no reassignment used. Also it uses CFA for code assignments. The multi code assignment in [17] called multicode multirate compact assignment is represented by MMCA-*n* where *n* denotes number of codes in the multi code. The results show the reduction in code blocking in the proposed single code assignment and reassignment schemes. The complexity of the proposed design is less because using maximum scattered code groups in the first layer, it is possible to derive the maximum scattered vacant codes groups in all remaining layers. The results in Figure 2.7, 2.8 and 2.9 shows the reduction of blocking probability in the



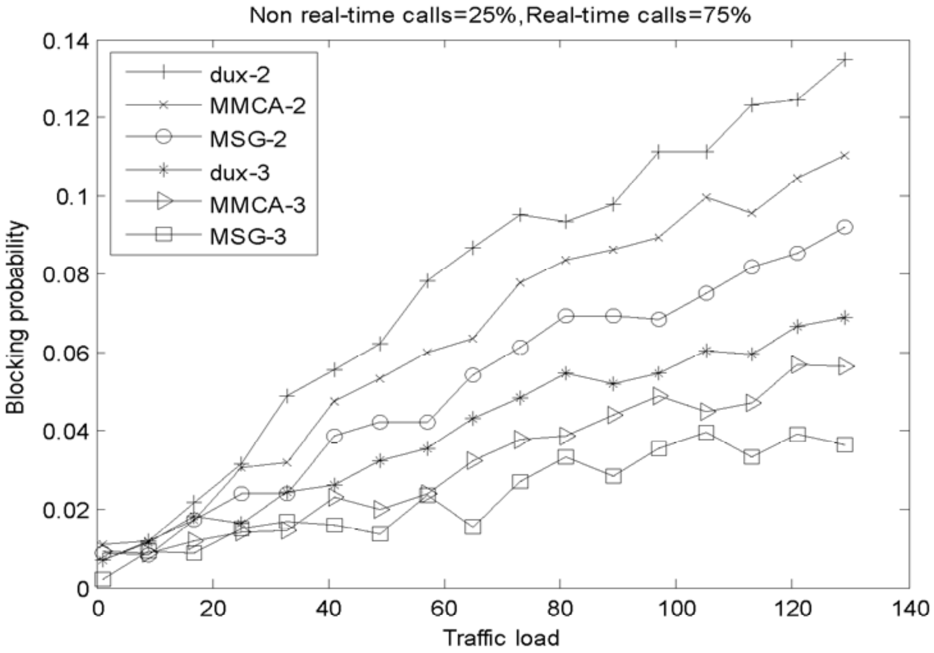
dux-*n*- Decreasing , United, No reassignments (using *n* codes), MSG-*n*- Maximum Scattered Group (using *n* codes)(ours)
MMCA-*n*- Multicode Multirate Code Assignment (using *n* codes)

**Figure 2.8:** Multi code assignment, uniform distribution

proposed multi code design compared to existing schemes.

## 2.5 <u>Conclusion</u>

OVSF codes are limited resources in 3G and beyond WCDMA wireless networks. The occurrence of scattered vacant codes in the code tree due to random call arrival and departure, increase future high rate calls blocking. The compact single code and multi code assignment schemes discussed in this chapter use the most scattered vacant code(s) reducing code/call blocking. The multi code design gives the added benefit of handling non quantized rates and reducing the code wastage capacity. The frequent call arrival and completion requires regular update for the most scattered vacant group, which may increase the computation/decision time for new calls. The work can be done to formulate decision time (due to online calculation of most scattered codes) and finding the optimum vacant code based on decision time and blocking probability. For real time calls the vacant code search can be made offline but it requires large buffer size at the BS and UE.



dux-*n*- Decreasing , United, No reassignments (using *n* codes), MSG-*n*- Maximum Scattered Group (using *n* codes)(ours)
MMCA-*n*- Multicode Multirate Code Assignment (using *n* codes)

**Figure 2.9:** Multi code assignment, more real time calls

# CHAPTER 3

## COMPACT CODEASSIGNMENT

### 3.1 Introduction

As discussed earlier, the presence of scattered vacant codes in OVSF codes leads to code blocking which leads to call blocking. As discussed in previous chapters, various assignment and reassignment schemes are proposed in literature. In this chapter, four single code assignment schemes are proposed. The use of assignment scheme depends upon the type (data or real time) of input calls. The code reservation assignment (CRA) is used to efficiently handle higher rate class calls. It assigns priority number to the children of priority class codes such that the future availability of vacant codes in the priority layer is the highest. The remaining three code assignment schemes favors low to medium calls. The code assignment using group leaders uses busy codes (capacity) under predefined leaders to handle future calls. The code blocking in group leader approach can be reduced further if the used capacity of all the parents of the eligible vacant codes is examined and the code whose parent has maximum used capacity is used for new call. In adjacent vacant codes grouping scheme, the eligible vacant codes are listed to find the code with least adjacent vacant codes. If unique result does not exists, the code (among the codes producing same adjacent vacant codes) with the least elapsed time of the busy neighbors is used for incoming call.

### 3.2 Code reservation assignment (CRA) scheme

Define $P_{l,n}$ as the priority number signifying priority of the code $n$ in layer $l$. Let the users with rate $2^l R, l \in [1, L-2]$ be given higher priority. The value of $l$ depends on the application network (e.g. real time video conferencing may require layers 5 or 6 and internet data may also require one or more layers close to root). The CRA schemes assign priority to all the vacant codes in the layers 1 to $l$-1 (with rates $R$ to $2^l R$). When a new user with rate $2^p R, p \in [1, l-1]$ arrives, the vacant code is assigned to user and some of its relatives will be assigned priority such that future probability of priority class codes is highest. For the layers 1 to $l$-1, the vacant codes with highest priority number are the candidates to handle new call. For a newly occupied code $C_{p,l}$, group of blocked codes ($G_B$) in the OVSF code tree are given by

$$G_B = \begin{cases} C_{i,2^{p-i}(l-1)+1} \ldots . C_{i,2^{p-i}l}; & 1 \le i \le p \\ C_{i,\left\lceil \frac{l}{2^{i-p}} \right\rceil}; & p \le i \le L \end{cases} \tag{3.1}$$

Define $y = \left\lceil \frac{x}{2^{m-p}} \right\rceil$. The group of total codes ($G_T$) which are the children of code $C_{m,n}$ in the layer *m* ($C_{m,n}$ should also be the parent of $C_{p,l}$) are given by

$$G_T = C_{j,2^{m-j}(y-1)+1} \ldots C_{j,2^{m-j}y}; 1 \le j \le m \tag{3.2}$$

If the entire children codes under the code $C_{m,n}$ are vacant prior to new call, the group of codes which are assigned priority number ($G_P$) are

$$G_P = G_T - G_B - C_{p,l} \tag{3.3}$$

As given in Equation (3.3), the relatives of the code $C_{p,l}$ having capacity less than $2^{m-1}R$ are candidates for higher priority. The priority numbers are valid only for codes in layers 1 to $l$. If the entire children codes under the code $C_{m,n}$ are not vacant prior to new call, the group of codes which are assigned priority number ($G_P$) are given by

$$G_P = G_T - G_B - C_{p,l} \tag{3.4}$$

where $G_O$ is the sum of all the occupied (busy) codes under $C_{m,n}$. The procedure is repeated for every new call. So, the assignment scheme chooses the vacant code for non priority calls whose busy brother(s) has the latest arrival. This is done to provide priority class layer with most number of vacant codes. Therefore OVSF code tree scattering is intentionally increased at the beginning of code assignment for better handling of priority class users. This is exactly what was not done in previous compact code assignment schemes. We divide the codes in OVSF code tree in three groups.

- Non priority class codes in layer 1 to $m$-1.
- Priority class codes in layer $m$.
- Non priority class codes in layer $m$+1 to $L$.

The code priority in above three categories has the following properties for code assignment and code vacation.

*a. Non priority class codes in layer* 1 *to* $m$-1

When a code is assigned to the new call, the vacant codes which are the children of $m^{th}$ layer parent of the assigned code are assigned priority higher than the highest current value. When the existing busy code becomes vacant, all the relative codes and blocked codes in layers 1 to $m$-1 are given priority equal to the highest priority value under the $m^{th}$ layer parent.

*b. Codes in priority class in layer m*

When a code is assigned to the new call, none of the codes are given priority. When the codes are vacated, the codes in layer 1 to $m$-1 which are the children of assigned code are given priority less than the lowest priority in layer 1 to $m$-1. No priority is given to the codes in layer $m$ to L.

*c. Non-priority class codes in layer* $m$+1 *to L*

When a code is assigned in this group, priority number in any layer is not affected. When the codes are vacated, the codes in layer 1 to $m$ which are the children of vacated code are given priority less than the least priority number.

Define $a_l$, $t_{arr}$, $\lambda$ (equal to $1/t_{arr}$) as number of busy codes in layer $l$, inter arrival time, arrival rate of users. The priority number is refreshed after every $t_r$ (threshold time) units of time, where $t_r > kt_{arr}$, $1<k<N$, $N<(a_0+2a_1+4a_2+..128a_7)$. The priority number is also refreshed after every call completion and call arrival.

The code assignment scheme is illustrated in Figure 3.1 for an OVSF-CDMA system with maximum capacity of 32$R$. The codes in layer 4 (corresponding to 8$R$ users) are given highest priority. Initially the code tree is assumed to be completely unused. Let a new call with rate 4$R$ is allocated a code $C_{3,1}$. The code assignment, code blocking and code reservation for future calls is shown in Figure 3.1(a). All the relatives encircled are given priority number 1. For second 2$R$ call, the code is searched which do not have any priority number assigned. Starting from left code $C_{1,21}$ is used and all its relatives are

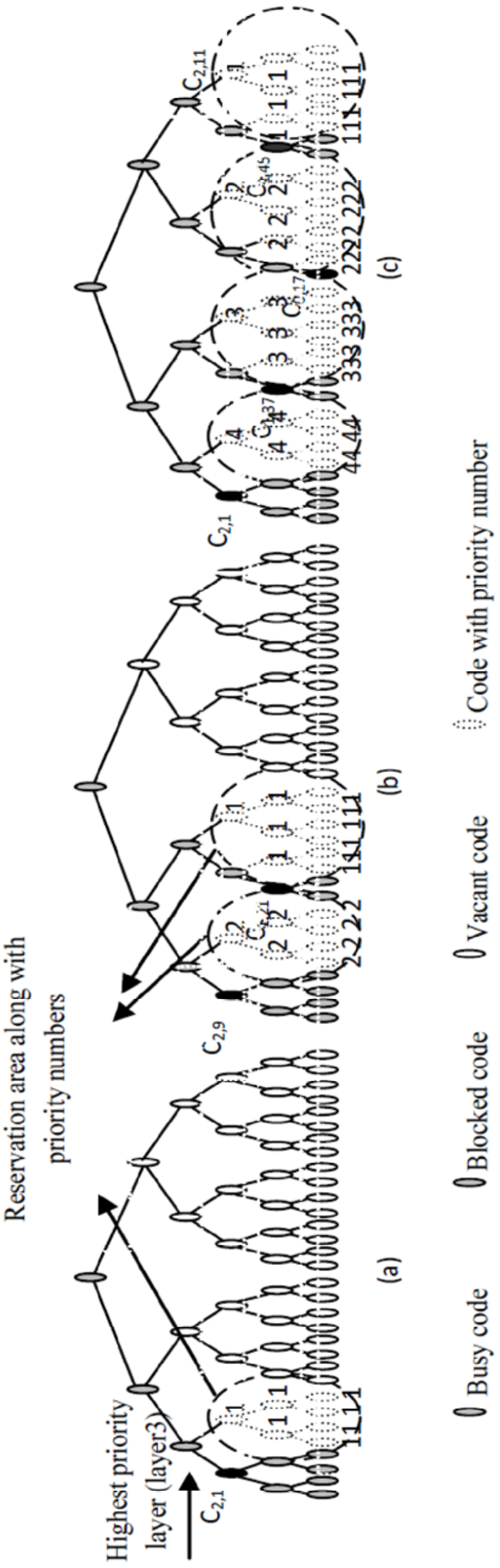


**Figure 3.1:** Illustration of code reservation scheme

given priority number 1 (as shown in Figure 3.1(a)). The priority number of codes in the layer of each relative which already has priority number is incremented by 1. In the example considered such codes are $C_{3,2}$, $C_{2,3}$, $C_{2,4}$, $C_{1,5}$, $C_{1,6}$, $C_{1,7}$, and $C_{1,8}$. Their new priority number becomes 2 (as represented in Figure 3.1(b). The codes $C_{3,1}$, $C_{2,5}$, $C_{1,17}$ and $C_{2,11}$ are codes used for new calls. For next calls $R$ and $2R$ using codes $C_{1,17}$ and $C_{2,13}$, the priority number assignment is shown in Figure 3.1(c). Now considering status in Figure 3.1(c), if a new $4R$ call arrives, the highest priority code $C_{3,8}$ will be used.

In case, the priorities need to be set for more than one class (say $c$ classes), the $L$ layer code tree is divided into $c$ groups. Each group belongs to one of the $c$ priority classes. The algorithm for the CRA scheme is shown below.

```
1. Enter arrival rate, service time, data rates, number of groups, priority class layer(s)
2. Generate new call with rate kR, $k=2^l$ and $l \in [1,8]$
3. If ($C_{used}$+rate of incoming call) < $C_{max}$
     • Do code allocation, code blocking and assignment of priority numbers according
       to the Equations (3.1-3.4)
     • Change the priority of codes in the code tree
     • Go to step 2
   Else
     • Discard call

   End
4. if (elapsed time<threshold time)
     • Refresh the priority number
5. Go to step2
```

### 3.3 Code assignment using group leaders (GL)

In group leader approach, we divide the leaves of the code tree into $2^{N-q-1}$ groups, where $q \in [0, q_{\max}]$ and $q_{\max} < N$. There are $2^{N-q-1}$ group leaders in layer $q$. For a group leader $C_{l,n}$, define *used capacity* as sum of capacities of all the children of $C_{l,n}$. The capacity of each group and the group leader is $2^q$. For a code $C_{l,n}$, the group leader is $C_{q\max,\left\lceil \frac{n}{2^{q\max - l}} \right\rceil}$. For a code $C_{q\max,n}$, the codes in the group are given by

$$C = C_{q,2^{q\max-q+1}n-2^{q\max-q+1}n+1}, \ldots, C_{q,2^{q\max-q+1}n} \quad 1 < q < q_{\max} \tag{3.5}$$

Lesser is the value of $q$, more is the number of groups making code tree compact for assignment of low data rates. The division is performed to make the code assignment most compact. The algorithm for assignment scheme is given below.

```
1.  Enter arrival rate, service time, data rates, number of groups
2. Generate new call
3. If ($C_{used}$+rate of incoming call) < $C_{max}$
    •   List all the vacant codes
    •   For all the vacant codes, find the used capacities of group leaders. The vacant
        code whose group leader has highest used capacity is the candidate for handling
        new call. If two or more vacant code leaders has same amount of used capacity,
        anyone can be used for code assignment
  Else
    •   Discard call

  End
```

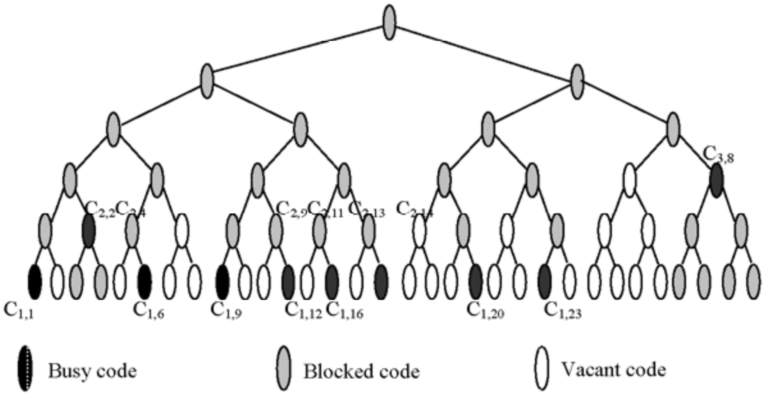


**Figure 3.2:** Code tree with capacity of 32 units

*4. Go to step2*

The above procedure gives reduction in external fragmentation of the remaining capacity making the code assignment compact. The code assignment in layers above group leader's layer (e.g. rates corresponding to layer 5, 6 and 7 in 8 code group example) can be done like LCA and RA schemes.

For illustration consider code tree in Figure 3.2 where layer 4 codes are can be treated as group leaders. If a new user with requirement of layer 1 code arrives, there are a large number of vacant code options. The vacant code $C_{2,4}$ has a capacity of $4R$ (sum of capacities of busy codes $C_{1,1}$, $C_{2,2}$ and $C_{1,6}$) under its group leader $C_{4,1}$. Similarly vacant codes $C_{2,9}$, $C_{2,11}$ has a capacity of $2R$The group leader capacity of $C_{2,4}$, $C_{2,3}$ and $C_{2,14}$ is maximum and any of the two can be assigned to the incoming call.

## 3.4 Code assignment using aggregate capacity under ancestors of eligible vacant codes

It is the most compact assignment scheme. The idea is to choose a vacant code such that the availability of higher layer codes after code assignment is maximized. For a code $C_{l,n}$, define *used capacity* as sum of capacities of all the children of $C_{l,n}$. Consider that a new call with the requirement of vacant code in layer $l$ arrives. The algorithm of code assignment is divided into following steps.

1. List all $n$ vacant codes $VC_{x,a_i}$ where $i \in [1, n_x], 1 \le p_i \le 2^{L-x-1}$ in layer $x$. For each vacant code, calculate the used capacities of first parent $P^{x+1}VC_{x,a_i}, i \in [1, n_{x+1}]$ in layer $x$+1 where $\lceil n_x / 2 \rceil \le n_{x+1} \le n_x$. If there exists a unique parent $P^{x+1}VC_{x,a_i}$ with the maximum used capacity, the code $VC_{x,a_i}$ is the candidate to handle new call. If no unique parent with maximum used capacity exists in layer $x$+1, calculate the *used capacities* of all the parents $P^{x+2}VC_{x,a_i}, i \in [1, n_{x+2}]$ in layer $x$+2 where $\lceil n_{x+1} / 2 \rceil \le n_{x+2} \le n_{x+1}$. Repeat the procedure for parents $P^{j}VC_{x,a_i}, i \in [1, n_j]$ where $| n_{j-1} / 2 | \le n_j \le n_{j-1}$ for all $j \in [x+1, 7]$ till unique parent exists.

2. If no unique parent exists with maximum used capacity in step 1, list all the parents in the layer $x$+1 given by $P^{x+1}VC_{x,a_i}, i \in [1, n'_{x+1}], n'_{x+1} \le n_{x+1}$, with same maximum used capacity from the set $P^{x+1}VC_{x,a_i}, i \in [1, n_{x+1}]$. For all such parents, list number of busy children in layer 1 to $x$denoted by $N^{x+1}VC_{x,a_i}, i \in [1, n'_{x+1}]$. The parent $P^{x+1}VC_{x,a_i}$ with the maximum number of busy children is the candidate to handle new call. This results in assigning most scattered portion of the code tree to handle new call giving highest number of vacant codes for future high rate calls. If all the parents $P^{x+1}VC_{x,a_i}, i \in [1, n'_{x+1}]$have same number of busy children, go to layer $x$+2 and so on till we reach layer 8.
3. If both above steps leads to more than one parent, for all the parents in layer $x$+1, $P^{x+1}VC_{x,a_i}, i \in [1, n'_{x+1}]$find the parent with most number of busy codes in layer 1. If unique parent exists, it is the candidate to handle new call. Otherwise check the parents $P^{j}VC_{x,a_i}, j \in [x+2, 8]$and $i \in [1, n'_j]$till unique parent exists.
4. If still no unique result exists, check the parents satisfying $P^{j}VC_{x,a_i}, j \in [x+1, 8]$and $i \in [1, n'_j]$ for most number of busy codes. If unique parent exists, the code $VC_{x,a_i}$is the candidate for handling new call. If all the three steps do not give unique result, list the vacant codes from the set $VC_{x,a_i}, i \in [1, n]$ whose ancestors have busy codes with least average elapsed time. It uses the fact that the code with least elapsed time will be vacated in the end. This increases the availability of higher layer codes for future calls.

For a code $C_{a,b}$, let the parent exist in layer $x$ with branch number $y$ denoted by $C_{x,y}$. Define scattering index $S(l,n)$

$$S(l,n) = \sum_{k=1}^{l-1} \sum_{l=1}^{m_k} VC_{k,l} \qquad (3.6)$$

Scattering index is a measure of number of busy codes under the parent $C_{l,n}$ who is parent of $C_{a,b}$

The algorithm of the code assignment scheme is given below.

```
1. Enter arrival rate, service time, data rates, number of groups
2. Generate new call
3. If ($C_{used}$+rate of incoming call) < $C_{max}$
      • List all vacant codes $VC_{x,a_i}, i \in [1,n]$
      • Find the suitable codes using the steps 1 to 4
      • Do code assignment and blocking
   Else
         Discard call
   End
4. Go to step 2
```

To illustrate the ancestor cost assignment scheme, consider tree in Figure 3.2 with new $2R$ rate arrival. There are five vacant code options. While checking the first parent of vacant codes, capacity tie occurs for codes $C_{2,4}$, $C_{2,9}$ and $C_{2,11}$ (with first parent $C_{3,2}$, $C_{3,5}$ and $C_{3,6}$). If first parent is used for optimization, check the elapsed time of busy codes under $C_{3,2}$, $C_{3,5}$ and $C_{3,6}$ i.e. codes $C_{1,6}$, $C_{1,20}$ and $C_{1,23}$. If the elapsed time of $C_{1,6}$, $C_{1,20}$ and $C_{1,23}$ is 4, 2, 3 units of time, code $C_{2,9}$ is picked for new call because probability of its parent remain blocked for longer time is highest (due to least elapsed time of $C_{1,20}$). If second parent of the candidate vacant codes are used for optimization, capacity tie occurs for parents of candidate codes $C_{2,4}$, $C_{2,13}$ and $C_{2,14}$. The vacant code $C_{2,4}$ is used for new call as its second parent has more number of busy leaves.

### 3.5 Grouping adjacent vacant codes

The scattered vacant codes at a particular time are known to BS and UE. Mostly the scattered codes in the levels are those vacant codes which lie between the busy nodes and are less in number. The probability of finding more consecutive vacant codes is always less than the probability of one or few vacant codes. The vacant codes which appear in groups are not assigned to the new call because there vacant parents can be used for users with higher rates. Therefore code assignment is done to use the vacant codes which do not appear in groups.

**Table 3.1:** Illustration of code selection using adjacent vacant code grouping scheme

(a) Group $S(0, b, n)$, $R$ rate call

| Adjacencies | Vacant codes | Neighbor/ Elapsed time | Code id |
|---|---|---|---|
| $b$=0, Single vacant code (no adjacent vacant code) | $C_{1,2}$ | $C_{1,1}/15$ $C_{1,6}/15$ $C_{1,9}/10$ | $S(1,0,1)$ |
| | $C_{1,5}$ | $C_{1,12}/10$ $C_{1,14}/7$ | $S(1,0,2)$ |
| | $C_{1,10}$ | $C_{1,16}/2$ | $S(1,0,3)$ |
| | $C_{1,11}$ | $C_{1,20}/5$ | $S(1,0,4)$ |
| | $C_{1,13}$ | $C_{1,23}/5$ | $S(1,0,5)$ |
| | $C_{1,15}$ | | $S(1,0,6)$ |
| | $C_{1,19}$ | | $S(1,0,7)$ |
| | $C_{1,24}$ | | $S(1,0,8)$ |
| $b$=1,Two consecutive vacant codes | $C_{1,7}$ | $C_{1,6}/15$ | $S(1,1,1)$ |
| | $C_{1,8}$ | $C_{1,6}/15$ | $S(1,1,2)$ |
| | $C_{1,17}$ | $C_{1,20}/5$ | $S(1,1,3)$ |
| | $C_{1,18}$ | $C_{1,20}/5$ | $S(1,1,4)$ |
| | $C_{1,21}$ | $C_{1,23}/5$ | $S(1,1,5)$ |
| | $C_{1,22}$ | $C_{1,23}/5$ | $S(1,1,6)$ |
| $b$=3,Four consecutive vacant codes | $C_{1,25}$ | $C_{1,29}/10$ $C_{1,29}/10$ | $S(1,3,1)$ |
| | $C_{1,26}$ | $C_{1,29}/10$ $C_{1,29}/10$ | $S(1,3,2)$ |
| | $C_{1,27}$ | | $S(1,3,3)$ |
| | $C_{1,28}$ | | $S(1,3,4)$ |

(b) Group $S(1, b, n)$, $2R$ rate call

| Adjacencies | Vacant codes | Neighbor/ Elapsed time | Code id |
|---|---|---|---|
| *b=0, Single vacant code* | $C_{2,4}$ | $C_{2,3}/15$ | $S(2,0,1)$ |
| | $C_{2,9}$ | $C_{2,10}/5$ | $S(2,0,2)$ |
| | $C_{2,11}$ | $C_{2,12}/5$ | $S(2,0,3)$ |
| *b=1, Two consecutive vacant codes* | $C_{2,13}$ | $C_{2,15}/10$ | $S(2,1,4)$ |
| | $C_{2,14}$ | $C_{2,15}/10$ | $S(2,1,5)$ |

Define $S(l,b,n)$ as a group of codes in layer $l$ accommodating all $b, 1 \le b \le 2^{8-l}$ adjacent vacant codes. Also $n$ gives the vacant code number with $b$ adjacent vacant codes. When system finds a vacant code, it checks for the consecutive single vacant code in its vicinity on either side of the vacant code. After finding single adjacent vacant code, the adjacent codes in group of three ($2^2$-1), seven ($2^3$-1) and so on up to $2^{7-l}$-1 are checked. Index $S(i,b,n)$ is a measure of scattering in the OVSF code tree. Lesser is the value of $b$, more is the vacant code scattering and new call pick the code with least adjacent vacant codes. For a code $C_{l,n}$, the adjacent vacant codes are

$$C_{l,2[((n+1)/2)-1]+1} \ldots C_{l,2[(n+1)/2]}, \text{ for one simultaneous adjacency} \tag{3.7}$$

$$C_{l,4[((n+3)/4)-1]+1} \ldots C_{l,4[(n+3)/4]}, \text{ for three simultaneous adjacencies} \tag{3.8}$$

The result can be generalized as

$$C_{l,N[((n+N-1)/N)-1]+1} \ldots C_{l,N[(n+N-1)/N]}, \text{ for other } N=1, 3, 7, 15 \text{ etc} \tag{3.9}$$

Basically, the code assignment design is divided into two steps

- Group all vacant codes according to adjacent vacant neighbors. Pick the group code with the minimum adjacent vacant codes. If tie occurs go to step 2.
- From all the candidate codes in group found in step 1, pick the vacant code with least elapsed time of busy neighbors. If tie occurs for elapsed times, any code with same least elapsed time is selected.

Let us deploy the above technique for allotment of codes to incoming rate $R$ and $2R$ in Figure 3.2. The busy codes are marked with dark shade while blocked codes are gray in color. No color indicated vacant codes. The codes are demarked with there designation and the elapsed duration in units. The elapsed duration for blocked codes is derived based on the highest elapsed duration of the busy codes under it or of its ancestors.

Searching algorithm is employed for finding the codes with various adjacencies i.e. for various values of $b$ and is grouped under group $S(l,b,n)$ where $l$ is level, $b$ are number of adjacencies and $n$ is the total no of codes with same adjacencies. The algorithm of the code assignment scheme is given below.

```
1. Enter arrival rate, service time, data rates, number of groups
2. Generate new call
3. If ($C_{used}$+rate of incoming call) < $C_{max}$
       • Arrange all the vacant candidate codes into groups S(l,b,n) , l∈[0,7]. Pick S(l,j,n), j=min(b)
       • Pick the vacant code with least elapsed time of busy neighbors as explained in section 3
       • Go to step 2
   Else
       • Discard call
   End
4. Go to step 2
```

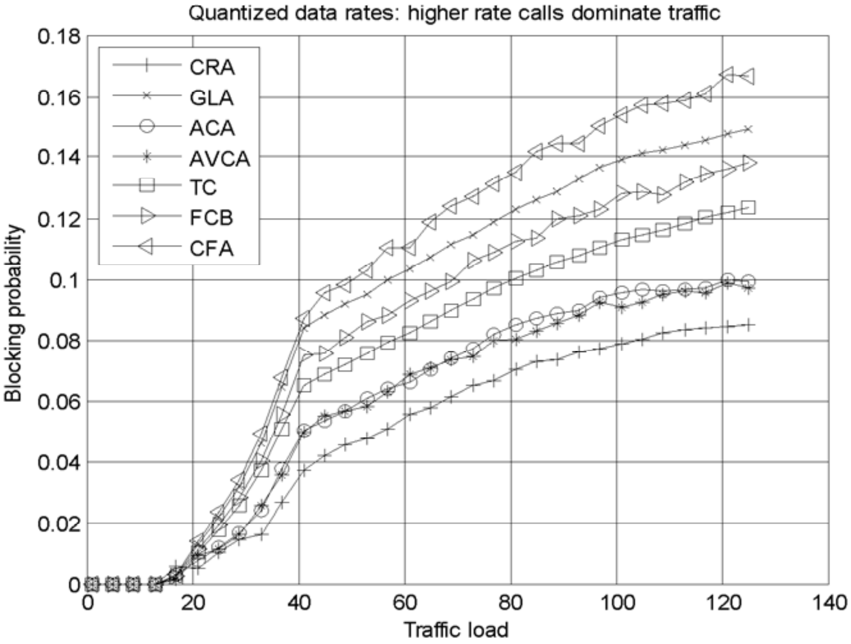
For incoming rate $R$ searching algorithm depicts that there are eight codes possible with adjacent zero vacant codes given by $C_{1,2}$, $C_{1,5}$, $C_{1,10}$, $C_{1,11}$, $C_{1,13}$, $C_{1,15}$, $C_{1,19}$ and $C_{1,24}$. From within these eight codes the code which has lowest elapsed time of its neighbor is selected. The elapsed times for busy codes at new arrival are illustrated in Table 3.1(b). The neighbor for any code is the code which comes under same parent as we go up the branch in OVSF tree. The neighbor who is adjacent but under different parent node is not considered for comparison of the elapsed time. Neighbor/elapsed time(in time units) for $C_{1,2}$, $C_{1,5}$, $C_{1,10}$, $C_{1,11}$, $C_{1,13}$, $C_{1,15}$, $C_{1,19}$ and $C_{1,24}$ are $C_{1,1}$/15, $C_{1,6}$/15, $C_{1,9}$/10, $C_{1,12}$/10, $C_{1,14}$/7, $C_{1,16}$/2, $C_{1,20}$/5 and $C_{1,23}$/5 respectively. These eight entries are designated in group $S(l,b,n)$ as $S(2,0,1)$, $S(2,0,2)$, and so on to $S(2,0,8)$. Since $S(2,0,6)$ have the having least elapsed time so this code is selected for accommodating new $R$ rate call.

Similarly for same scenario and incoming rate of $2R$, the searching algorithm finds vacant codes $C_{2,4}$, $C_{2,9}$ and $C_{2,2}$ as codes with least adjacency $b$=0. Neighbors/elapsed time for vacant codes for $2R$ call are $C_{3,3}$/15, $C_{3,10}$/5 and $C_{3,4}$/5 units respectively. These are designated in group $S(i,b,n)$ as $S(2,0,1)$, $S(2,0,2)$, $S(2,0,3)$. The entries $S(2,0,2)$, $S(2,0,3)$ equally probable and hence any of them is selected.

## 3.6 Simulation and Results

### *3.6.1 Input parameters*

- Call arrival process is assumed to be Poisson distributed with average value, $\lambda$ =1-128 calls/time.



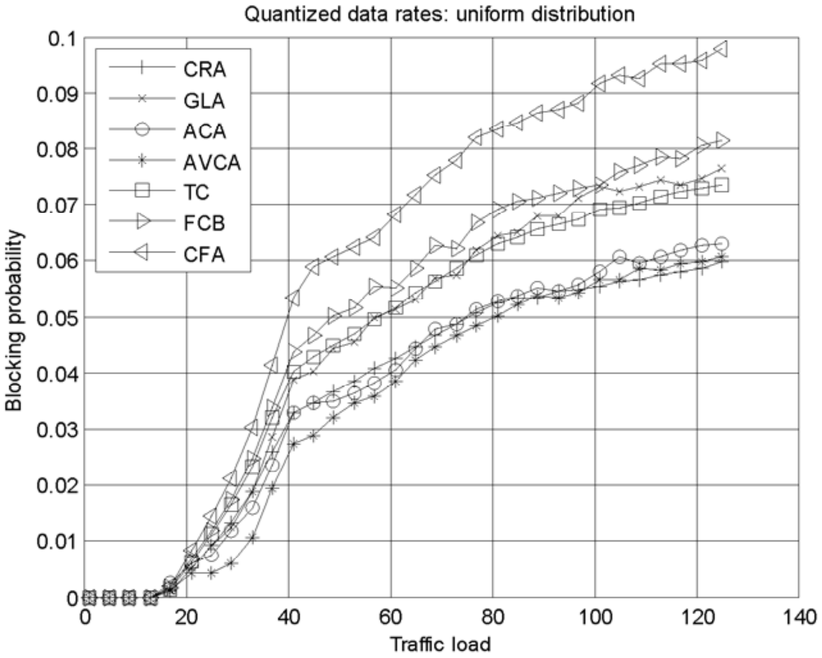
CRA- Code Reservation Assignment(ours), GLA-Group Leader Assignment(ours), ACA- Ancestors Cost Assignment(ours), AVCA-Adjacent Vacant Codes Assignment(ours), TC- Time based Code Scheme, FCB- Fewer Codes Blocked, CFA- Crowded First Assignment.

**Figure 3.3:** Quantized data rates with high rates dominating

- Service time is assumed to be negative exponential distributed with mean value $1/\mu$=1 units of time.
- Two categories of rates, quantized and non quantized are assumed. In quantized rates, there are eight classes of users with rates *R*, 2*R*, 4*R* ….128*R* (*R*=7.5kbps). For non quantized rates, there are 128 classes of users with rates *R*, 2*R*, 3*R* ….128*R*.
- The capacity of OVSF code tree is 128*R* with root in layer 8 (layer numbering starts from leaves).

### *3.6.2 Quantized data rates*

Consider that there are $G_k = 2^{k-1}, k = [1,2....8]$ servers in the $k^{\text{th}}$ layer corresponding to $G_k$ number of vacant codes. The total codes (servers) in the system assuming an eight set of classes are given by $G=\{G_1,G_2,G_3,G_4,G_5,G_6,G_7,G_8\}$. The maximum number of servers used to handle new call is equal to the number of rake combiners. Let $\lambda_k, \mu_k$ is the arrival rate, service rate of $k^{th}$ class of users. Traffic load for the $k^{\text{th}}$ class of users is given by $\rho_k = \lambda_k / \mu_k$. The code blocking for the $k^{\text{th}}$ class is defined by



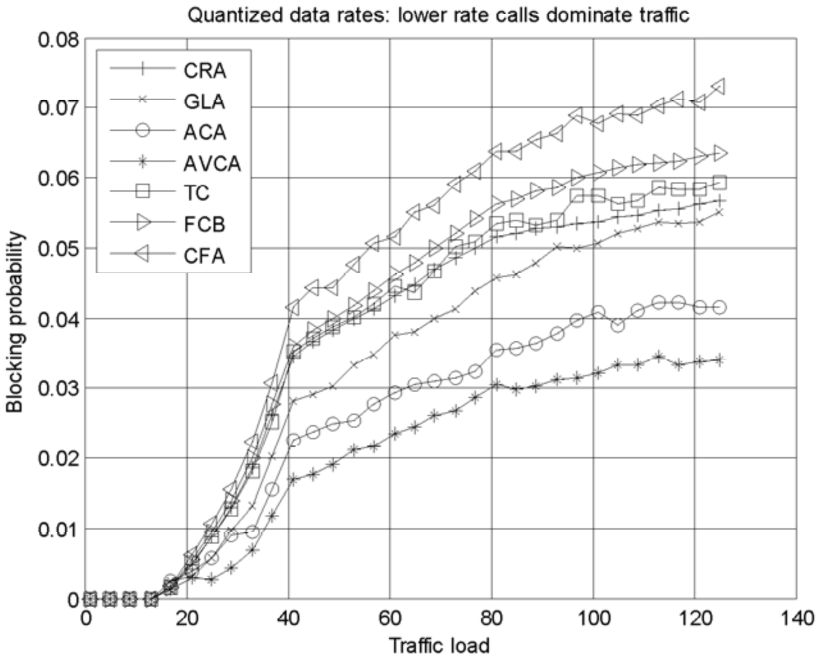
CRA- Code Reservation Assignment(ours), GLA-Group Leader Assignment(ours), ACA- Ancestors Cost Assignment(ours), AVCA-Adjacent Vacant Codes Assignment(ours), TC- Time based Code Scheme, FCB- Fewer Codes Blocked, CFA- Crowded First Assignment.

**Figure 3.4:** Quantized data rates with uniform distribution

$$P_{B_k} = \frac{\rho_k^{G_k} / G_k!}{\sum_{n=1}^{G_k} \rho_k^n / n!} \tag{3.10}$$

The average code blocking for 8 class system is

$$P_B = \sum_{k=1}^{8} \frac{\lambda_k}{\lambda} P_{B_k} \tag{3.11}$$



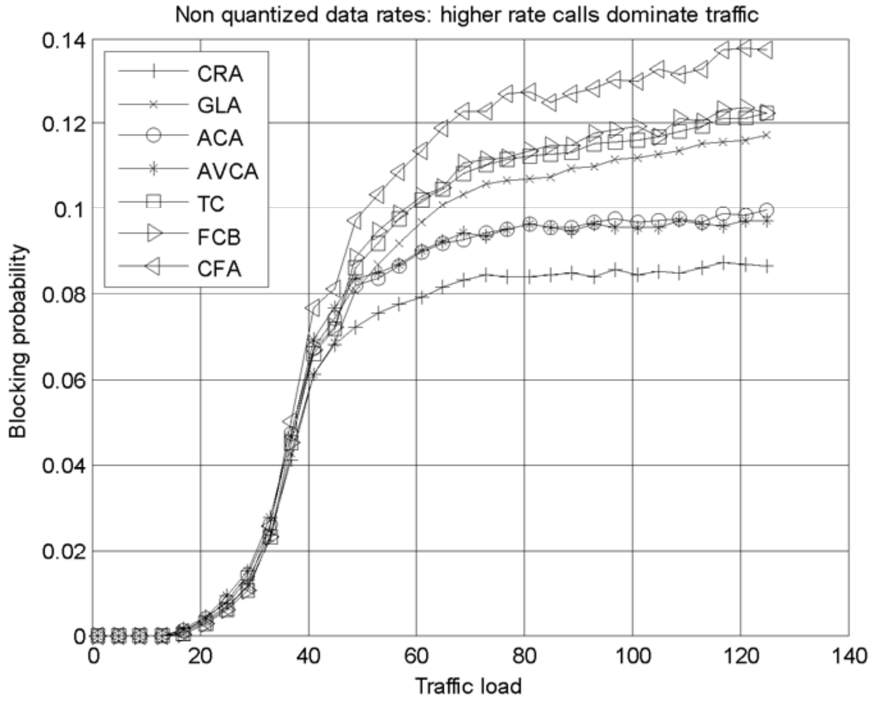
CRA- Code Reservation Assignment(ours), GLA-Group Leader Assignment(ours), ACA- Ancestors Cost Assignment(ours), AVCA-Adjacent Vacant Codes Assignment(ours), TC- Time based Code Scheme, FCB- Fewer Codes Blocked, CFA- Crowded First Assignment.

**Figure 3.5:** Quantized data rates with low rates dominating

We divide the eight classes of calls into two categories namely real time classes (especially for speech signals with rates $R$, $2R$, $4R$ and $8R$) and non real time calls (with rates $16R$, $32R$, $64R$ and $128R$). The video conferencing comes in higher rate (although it is real time category) class. Let the arrival distribution for the eight classes are given by $[p_1, p_2]$, where $p_1$ and $p_2$ is the probability of real time and non real time calls. Three arrival distributions are considered as given below

- [0.75,0.25] for low rate calls dominating the arrival process
- [0.5,0.5] for uniform distribution of eight classes
- [0.25,0.75] for high rate calls dominating the arrival process

The code blocking comparison of the proposed schemes is done with CFA [16], FCB



CRA- Code Reservation Assignment(ours), GLA-Group Leader Assignment(ours), ACA- Ancestors Cost Assignment(ours), AVCA-Adjacent Vacant Codes Assignment(ours), TC- Time based Code Scheme, FCB- Fewer Codes Blocked, CFA- Crowded First Assignment.

**Figure 3.6:** Non quantized data rates with high rates dominating

[20] and TC [18] schemes discussed earlier. The nomenclature for the proposed scheme is CRA (code reservation assignment), GLA (group leader assignment), ACA (ancestors cost assignment) and AVCA (adjacent vacant codes assignment). Result in Figure 3.3, 3.4 and 3.5 shows that the performance improvement using proposed compact assignment schemes.

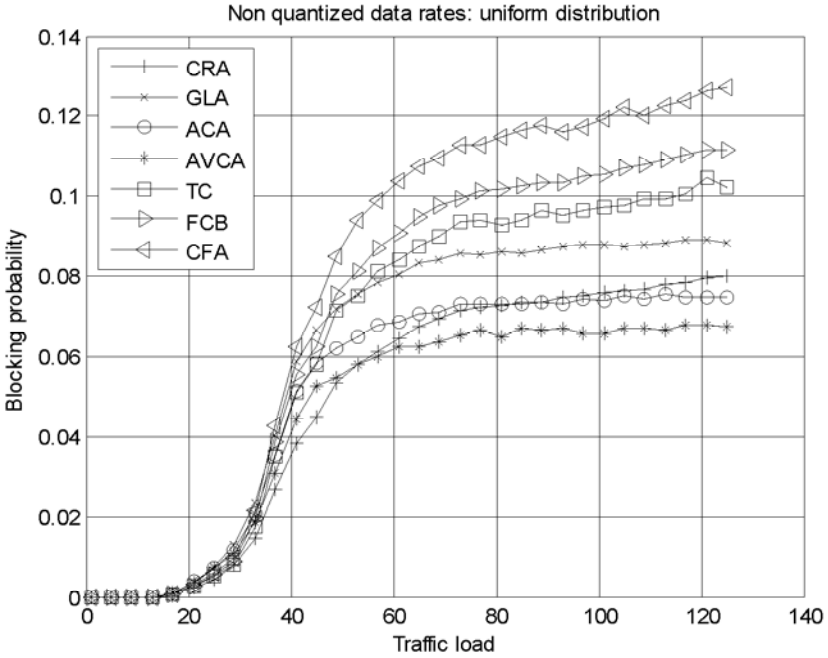
### *3.6.3 <u>Non quantized data rates</u>*

Consider there are 128 classes of users with rates $R$, $2R$,….$128R$. The arrival rate and service rate is $\lambda_k$ and $\mu_k$ , $k \in [1,128]$. The internal fragmentation adds to the code blocking due to code rate wastage with non quantized rates. The average code blocking is given by

$$P_B = \sum_{k=1}^{128} \frac{\lambda_k}{\lambda} P_{B_k} \tag{3.12}$$

The arrival distribution for the 128 classes is divided into two sets defined by [$p_1$, $p_2$,], where $p_1$ is the sum of probabilities of rate arrival $\lambda_i, i \in [1,2,.....15]$ assumed to be real time calls and $p_2$ is the sum of probabilities of rate arrival $\lambda_i, i \in [16,128]$ assumed to be non real time calls. Three distributions of rates are assumed

- [0.75,0.25] for low rate calls dominating the arrival process
- [0.5,0.5] for uniform distribution of four classes

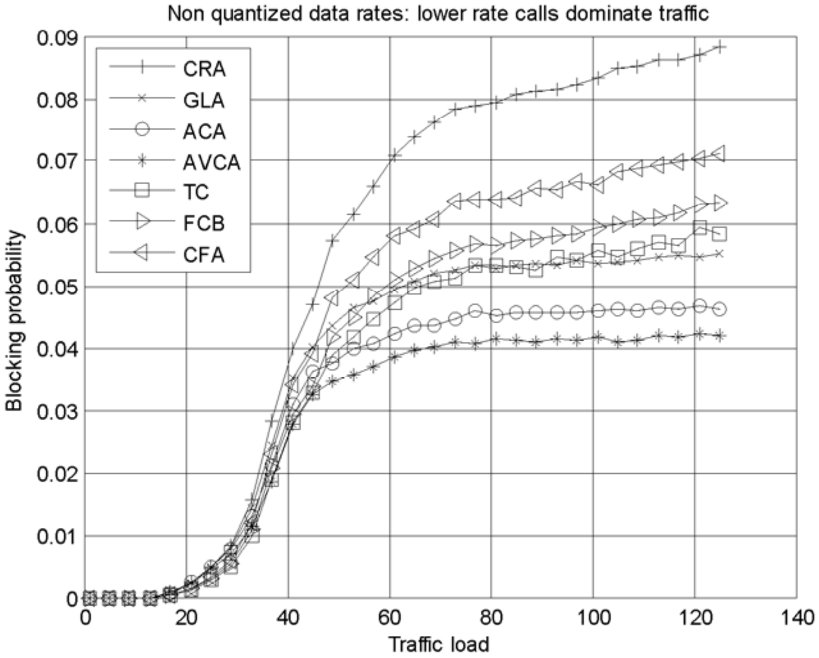


CRA- Code Reservation Assignment(ours), GLA-Group Leader Assignment(ours), ACA- Ancestors Cost Assignment(ours), AVCA-Adjacent Vacant Codes Assignment(ours), TC- Time based Code Scheme, FCB- Fewer Codes Blocked, CFA- Crowded First Assignment.

**Figure 3.7:** Non quantized data rates with uniform distribution

- [0.25,0.75] for high rate calls dominating the arrival process

Results in Figures 3.6, 3.7 and 3.8 shows that the performance improvement using proposed compact assignment schemes.



CRA- Code Reservation Assignment(ours)t, GLA-Group Leader Assignment(ours), ACA- Ancestors Cost Assignment(ours), AVCA-Adjacent Vacant Codes Assignment(ours), TC- Time based Code Scheme, FCB- Fewer Codes Blocked, CFA- Crowded First Assignment.

**Figure 3.8:** Non quantized data rates with low rates dominating.

## 3.4 Conclusion

OVSF codes are the limited resources at the downlink of 3G and beyond CDMA based mobile communication systems. The chapter discusses compact code assignment schemes for the efficient use of OVSF codes. The code blocking can be reduced by using code assignment scheme according to the arrival distribution of incoming calls. This also increases throughput and code utilization. The complexity of all the code assignment schemes is lower because single rake combiner is required at the BS and at the UE.

# CHAPTER 4

## FLEXIBLE ASSIGNMENTOF DATA CALLS

### 4.1 Introduction

Broadly speaking, the calls fall into two categories, (*a*) real time calls with fixed rate requirement, (*b*) data calls in which the call rate can be increased or decreased depending upon availability of resources. The call handling and resource utilization differs significantly depending upon whether the call flow involves single hop or multiple hops. Based upon this, in this chapter the single hop and multihop networks are treated differently.

### 4.2 Single hop networks

The single hop network is shown in Figure 4.1. If a new call with rate $2^l R$ arrives, the code assignment depends upon type of call (real time or data), code tree status and link bandwidth. For real time call with rate $2^l R$, if the link bandwidth $B \geq 2^l R$, the call is handled, otherwise the call is rejected.

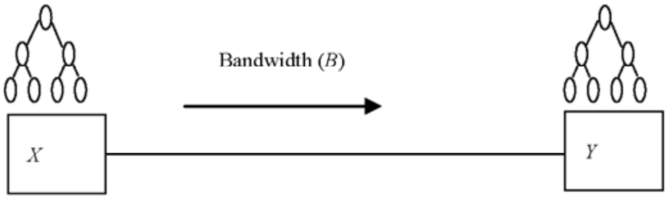


**Figure 4.1:** Single hop communication

If there are $M$ bandwidth variations $B_j, j \in [1, M]$ for one call and $G_j$ represents data transmitted in $j^{th}$ bandwidth usage, the average bandwidth of channel for call $2^l R$ is given by

$$B = \frac{\sum_{j=1}^{M} B_j}{M} \tag{4.1}$$

Also, the total transmitted data *'G'* considering all rate variations can be represented as

$$G = \sum_{j=1}^{M} G_j \tag{4.2}$$

For incoming data call with rate $2^l R$, if the maximum capacity of single vacant code is $2^{l'} R$, the channel transmission rate can have following variants, (*i*) if $2^{l'} R \geq 2^l R$, and $B \geq 2^{l'} R$, the transmission rate is $2^{l'} R$ with scaling factor $2^{l'-l}$, (*ii*) if $2^{l'} R \geq 2^l R$, and $2^l R \leq B \leq 2^{l'} R$, the transmission rate is $2^{\lfloor \log_2 B \rfloor} R$ with scaling factor $2^{\lfloor \log_2 B \rfloor - l}$ and the vacant code with capacity $2^{\lfloor \log_2 B \rfloor} R$ is used, (*iii*) if $2^{l'} R > 2^l R$ and $B < 2^l R$, the call is rejected because the channel has insufficient bandwidth to handle call, (*iv*) If $2^{l'} R < 2^l R$ , and $B \geq 2^l R$ , the call can be handled provided that the average traffic load is less than the maximum capacity of code tree.

If there are total $L$ call classes, and the average arrival rate and call duration of $l^{th}$ class is $\lambda_l$ and $1/\mu_l$ respectively, the traffic load for $l^{th}$ class is given by

$$\rho_l = \lambda_l / \mu_l \tag{4.3}$$

The average traffic load is

$$\rho = \sum_{i=1}^{L} (\lambda_l / \mu_l) \tag{4.4}$$

Let the maximum capacity of the code tree is $C_{max}$. If the system has zero code blocking, the new call with rate $2^l R$ can utilize low rate codes in layers 1 to $l$-1 if

$$\sum_{i=1}^{L} \rho_l + 2^l R / C_{\max} < 1 \tag{4.5}$$

If the system has average code blocking $P_B$, the code from lower layers can be utilized if

$$\sum_{i=1}^{L} \rho_l + 2^l R / C_{\max} + P_B < 1 \tag{4.6}$$

This makes the overall utilization of the code tree close to 100%. Let capacity threshold $T_l$ represents fraction of total capacity $C_{max}$ which prompts the use of codes in layers 1 to $l$-1. If total load $\rho$ and $P_B$ are known, ideal value of $T_l$ for zero blocking condition is given by

$$T_l = C_{\max} (1 - (\sum_{i=1}^{L} \rho_l + P_B)) \tag{4.7}$$

For blocking systems, $T_l$ can be defined as

$$T_l = C_{\max}(1-(\sum_{i=1}^{L}\rho_l + P_B)) - aR, aR << C_{\max} \tag{4.8}$$

If $aR$ is high, the utilization deviates significantly from 100% and the algorithm is simple as only few lower layer codes are used. If $aR$ is low, the utilization is close to 100% but the algorithm is complex. The new call is almost guaranteed to be accepted.

### 4.3 Multihop networks

The multihop wireless network is shown in Figure 4.2 for a single call flow with rate $2^l R$. Considering total N links in multihop network the details of a particular link (say link $i$) is shown in Figure4.3. Let $2^{l_i}R$ represents the capacity corresponding to maximum rate vacant code in the $i^{th}$ link and $B^i$ is the bandwidth of $i^{th}$ link. The rate scaling factor for $i^{th}$ link is $2^{l_i - l}$. For a real time call with rate $2^l R$, the call is handled only if $2^l R \le 2^{l_i} R$, and $2^l R \le B^i$. On the other hand, for data call with rate $2^l R$, the transmission rate for $i^{th}$ link

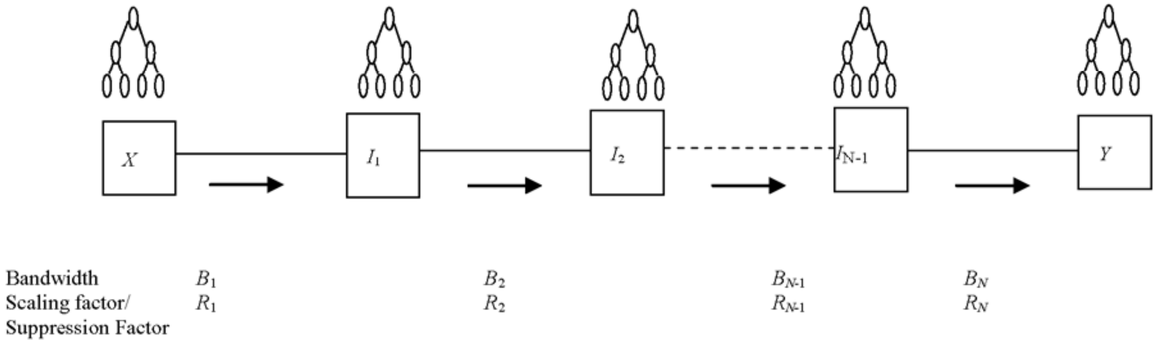


**Figure 4.2:** Multi hop communication

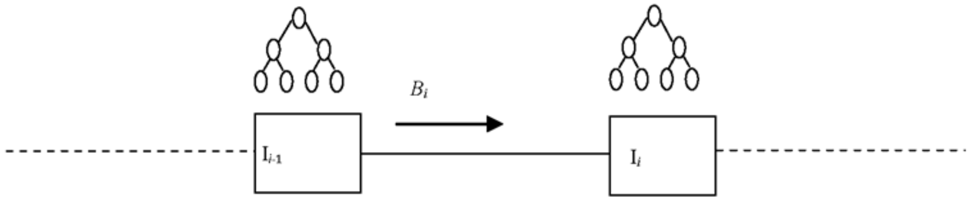


**Figure4.3:** $i^{th}$ multihop link

can be calculated using steps similar to one used for single hop networks replacing $B$ with $B^i, i \in [1, N]$ representing average bandwidth of $i^{th}$ hop/link and $2^{l'} R$ with $2^{l_i} R$ respectively. Additionally, for $i^{\text{th}}$ link, define. $R_i = \min \lfloor 2^{l_i R, B^i} \rfloor$. If $R_i = 2^{l_i} R$, the transmission rate is $2^{\lfloor \log_2 B^i \rfloor} R$. The overall transmission rate between $X$ and $Y$ is

$$R_O = \min[R_i], i \in [1, N] \tag{4.9}$$

In general, if the scaling/suppression of data call at $i^{\text{th}}$ link is denoted by $A_i$, for $1 \le i \le L-1$, following values of $A_i$ are permitted.

$$\mathrm{A_i} = \begin{bmatrix} 2^{\mathrm{i}}, \text{data rate increased due to excess resources} \\ 1 \\ 1/2^{\mathrm{i}}, \text{data rate reduced as instantaneous load is high but avg.load} \\ \qquad \text{is less than the maximum capacity of the code tree} \end{bmatrix}$$

(4.10)

For a new call with rate $2^l R$, for the $i^{\text{th}}$ link if there are $M_i$ bandwidth variations $B_{j_i}, j_i \in [1, M_i]$ due to shared channel, the average bandwidth of $i^{th}$ link is given by

$$B^i = \frac{\sum_{j_i=1}^{M_i} B^i_{j_i}}{M_i};\ 1 \le i \le N \tag{4.11}$$

If for $i^{th}$ link, $G_{j_i}, j_i \in [1, M_i]$ represents amount of data transmitted for $j_i^{\text{th}}$ bandwidth variation, the total transmitted data $G^i$ at the $i^{th}$ link is given by

$$G^i = \sum_{j_i=1}^{M_i} B^i_{j_i};\ 1 \le i \le N \tag{4.12}$$

The propagation time for $i^{\text{th}}$ link with length $d_i$, is defined as $t_{p_i} = d_i / v$ where $v$ is the speed of light. If the total data size is $D$ bits, the initial call duration for the single hop system is

$$t_d = D / 2^l R \tag{4.13}$$

Ignoring the propagation time (which is generally very small compared to transmission time), the new call duration is

$$t'_d = t_d \times (2^l R / 2^{\lfloor \log_2 B \rfloor}) \tag{4.14}$$

For $N$ link multihop communication, the new call duration (ignoring propagation time and data storage time) is

$$t'_d = (t_d \times 2^l R / N) \times \left( \frac{1}{2^{\lfloor \log_2 B^1 \rfloor}} + \frac{1}{2^{\lfloor \log_2 B^2 \rfloor}} + \ldots.. \frac{1}{2^{\lfloor \log_2 B^N \rfloor}} \right) \tag{4.15}$$

## 4.4 Proposed Design

Consider an $L$ layer OVSF code tree. If $C_{l,n}$ represents a code in layer $l$ with branch number $n$, the code tree capacity assigned to a new call $2^l R$ can have two variants,(a) Guaranteed capacity: minimum vacant code capacity required to handle real time call (and is equal to $2^l R$). If the code tree does not have guaranteed capacity, the call is rejected. (b) Available capacity: capacity of the vacant code closest to the root code (the root code exists in layer $L$). For data call $2^l R$ (in layer $l$-1), the available capacity can take values from $2^{l'} R (0 \le l' < l - 1)$ for rate suppression and $l \le l' < L - 1$ (for rate scaling), and if the call duration at call arrival is $t$, the updated call duration due to rate variation flexibility is given by

$$(2^l / 2^{l'}) \times t \tag{4.16}$$

For a code $C_{l,n}$, let code index $I_{l,n}$ defines the status of the code. The code indices can take values as follows,

*a.* $I_{l,n} = 0$, if code $C_{l,n}$ is free

*b.* $I_{l,n} = 1$, if code $C_{l,n}$ is blocked

*c.* $I_{l,n} = 2$, if code $C_{l,n}$ is used for a call with guaranteed capacity $2^l R$ and available capacity $2^l R$

*d.* $I_{l,n} = 2 + (l' - l)$, if the code $C_{l,n}$ is used for call with guaranteed capacity $2^l R$ and available capacity $2^{l'} R, l' > l$, and (*v*) $I_{l,n} = (l' - l)$ (negative number), if the vacant code is used from layer $l', l' < l$, and it happens when the code tree does not have vacant code in layer $l$ due to high value of the instantaneous traffic load but the average traffic load allows the use of vacant code in layer $l'$. For a new $2^l R$ call (data or real time), the flexible data rate algorithm works as follows.

*Case 1: Enough capacity available with highest capacity vacant code present in layer.*
$l' \mid l' = \max(l''), l'' \in [l, L]$

If the call is of data type, the vacant code in layer $l'$, say $C_{l',n_{l'}}$ is assigned. The call rate is intentionally increased by a factor $2^{l'-l}$ and subsequently the call duration is decreased by a factor $2^{l'-l}$. If there are multiple options for vacant code, the code with least code index value $I_{l',n_{l'}}$ is the candidate code to handle the call. For real time call the vacant code with capacity $2^l R$ is assigned.

*Case 2: Enough capacity is not available*

*Data call arrival:* Check the code indices for all busy codes starting from layer $l$+1. If for a layer $l', l' > l$, there is at least one busy code with code index greater than $2 + (l' - l)$ then the call can be handled by reducing the rate of previous ongoing call by shifting this call to lower layer code. More specifically, if there are $N_{l'}$ number of such busy codes in layer $l'$, pick the code $C_{l',n_{l'}}$ with largest code index value $I_{l',n_{l'}}$. Considering this code index value $I$, the code is currently handling a call with rate $2^{l'-I-2} R$, and reducing this $2^{l'-I}$ times the reduced capacity can be utilized by incoming call $2^l R$. The code used to handle new $2^l R$ call is $C_{l, 2^{l'-l} \times n_{l'} - 2^{l'-l} + 1}$ and the code $C_{l'-I-2, 2^{l'-l} \times n_{l'} - 2^{l'-l} + 1}$ for call $2^{l'-I-2} R$. Therefore the new call is handled applying code shifts (reassignments).

*Real time call arrival:* For the identified code $C_{l',n_{l'}}$ with largest code index value $I_{l',n_{l'}}$, use one child with capacity $2^l R$ (in layer $l-1$) for incoming call and one of the other children with capacity $2^l R$ for previous ongoing call. The capacity $2^{l'-1} - 2^{l+1} R$ can be used for future calls.

*Case 3: Case 1 and 2 fails*

*Data call arrival:* If both of the above cases do not provide a vacant code, reassignments are required but the reassignments can be applied only if the total available capacity (sum of capacities of vacant codes in all layers) is more than the capacity required by incoming call. The algorithm identifies layer $l' | l' = \max(l''), l'' \in [2, l]$ for which at least one busy code with code index greater than 2 exists. For this layer list all the busy codes along with their code indices. Pick the code $C_{l',n_{l'}}$ with highest code index *I*. This busy code was initially handling an ongoing call with guaranteed rate $2^{l'-(I-2)}R$. The algorithm reduces the rate of this ongoing call by $2^{I-2}$ times. The new code used to handle this call is $C_{l'-I-2, 2^{I-2} \times n_{l'-I-2} - 2^{I-2} + 1}$. If *t* is the total call duration had the call being handled fully by code $C_{l',n_{l'}}$ and $t_1$ is the call elapsed time at the arrival of new call $2^l R$, the new call duration for ongoing call is given by

$$t' = t_1 + (t - t_1) \times 2^{I-2} \tag{4.17}$$

If we take $t$-$t_1$=$t_2$, Equation (4.17) can be rewritten as

$$t' = t_1 + t_2 \times 2^{I-2} \tag{4.18}$$

In general, if there are total *N* rate transitions for ongoing call, the total call duration can be expressed as

$$\begin{aligned} t' &= t_1 + t_2 \times 2^{I_1-2} + t_3 \times 2^{I_2-2} + \ldots\ldots t_N \times 2^{I_{N-1}-2} \\ &= t_1 + \textstyle\sum_{i=1}^{N-1} (t_{i+1} \times 2^{I_i-2}) \end{aligned} \tag{4.19}$$

Also the average rate of the ongoing call becomes

$$\begin{aligned} R' &= \left[t_1 \times 2^{l_i'} R + t_2 \times 2^{l_i'-(I_1-2)} R + t_3 \times 2^{l_i'-(I_2-2)} R \ldots\ldots\ldots + t_N \times 2^{l_i'-(I_{N-1}-2)} R\right] / t \\ &= \left[t_1 \times 2^{l_i'} R + \textstyle\sum_{i=1}^{N-1} (t_{i+1} \times 2^{l_i'-(I_i-2)}) R\right] / t \end{aligned} \tag{4.20}$$

*Real time call arrival:* The call is handled as per *Case* 3 data call handling algorithm.

*Case 4: Case 1 to 3 fails*

*Data call arrival:* The call can be handled using vacant codes in layer 1 to *l*-1 if the average traffic load of the system is less than the maximum capacity of code tree. When an ongoing call is completed, the vacant code with the highest rate is identified and the codes which are currently handling the suppressed rate shift the call to the identified vacant code when the identified code has capacity more than the currently used code. The aim is to make code tree capacity utilization close to 100%. The utilization in *Case* 3 can be further increased if we use code reassignments till utilization becomes 100%. Typically there may be many such reassignments to make available capacity more than $2^{l}R$. If the suppressed call is currently handled by code $C_{l'-I-2,2^{I-2}\times n_{l'-I-2}-2^{I-2}+1}$, the algorithm shifts the call assigned to this code to the appropriate location within the code tree in a layer between $l'$ to $l'-(I-2)$ provided the new code is not $C_{l',n_{l'}}$ or its child.

*Real time call arrival:* The call is rejected.

While *Case* 3 requires one reassignment per call, *Case* 4 requires large number of reassignments (although greater utilization) even more than traditional DCA. In addition, the code indices are updated at call completion. If any ongoing call occupied by code $C_{l,n}$ is completed, the code indices of the sibling (or children of sibling) has to be updated.

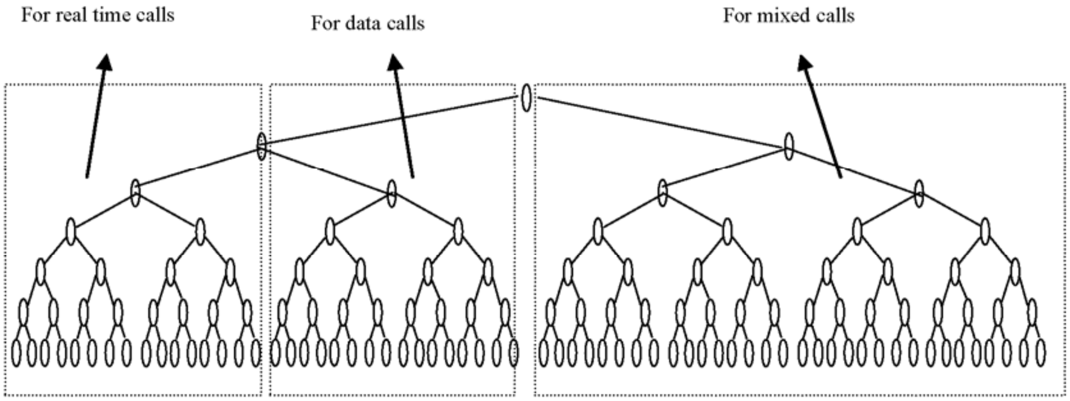


**Figure 4.4:** Code tree division according to call types.

The design discussed so far does not divide code tree according to the type of traffic. Although this increases throughput but complexity and cost of the design may be inappropriate. In another approach, the code tree can have different portions for real time calls, data calls and mixed (a connection where rate may be real or data type) calls. The code tree utilization may not be 100% although the design is simple and cost effective. The mechanism divide call rates into three categories (*i*) pure real time rate (*ii*) data call (*iii*) mixed rate as illustrated in Figure 4.4. Consequently, the code tree is also divided into three portions to treat these three categories differently. The data portion of the code tree can always be fully utilized as for pure data calls description given earlier. The second portion of the code tree may provide wastage and the only way to optimize utilization is to use full dynamic code assignment (DCA) [4] design. But too many reassignments are not preferred for real time calls. The third region is the most complex as far as code allocation is concerned. Let $N_r$, $N_d$ and $N_m$ represents total real time, data and mixed classes. Also, $N_r^i, i \in [1, N_r]$, $N_d^i, i \in [1, N_d]$ and $N_m^i, i \in [1, N_m]$ represents the $i^{\text{th}}$ call in each category. A particular wireless network may have one, two or all three categories of rates. If $C_{\max}^j$ denotes the maximum capacity reserved for $i^{\text{th}}$ pure real time rate, the portion of total code tree capacity used for real time calls is given by

$$C_{\max}^r = \sum_{i=1}^{N_r} C_{\max}^{r,i} \tag{4.21}$$

Similarly if $d$ and $m$ represents data and mixed calls, we have

$$C_{\max}^d = \sum_{i=1}^{N_d} C_{\max}^{d,i} \tag{4.22}$$

$$C_{\max}^m = \sum_{i=1}^{N_m} C_{\max}^{m,i} \tag{4.23}$$

Handling mixed rates require fairness among data and real time rates so that none of the two is overserved or underserved.

The division of code tree into various regions is made proportional to the amount of calls. The division changes frequently if arrival distribution changes rapidly.

*Fairness issue*

Due to the use of data calls for real time calls, some specific layer calls may be over used or under used. Also for real time calls, some specific layer may be overserved, which is unfair for other classes. To make the assignment perfectly good, the fairness is required for following calls.

1. Fairness among real time and data calls.
2. Fairness within real time and data calls.
3. Class (layer) fairness.

Let $F_l$ denotes the fairness of code tree capacity which must be available for rate $2^l R$ calls (real time or data)

The value of $F_l$ depends upon the $l^{th}$ class average arrival rate $\lambda_l$. Ideally, $F_l$ is defined as

$$F_l = (\lambda_l / \sum_{i=1}^{L} \lambda_l) \times C_{\max} \tag{4.24}$$

Practically the fairness index $F_l'$ can be chosen as

$$F_l' = aF_l, 0 \le a \le 1 \tag{4.25}$$

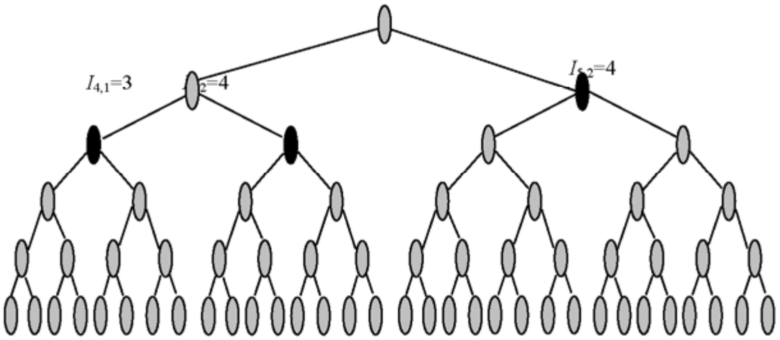


**Figure 4.5(a):** A 6 layer OVSF code tree

The code assignments and reassignments are explained in Figure 4.5 using a six layer code tree. In Figure 4.5(a), the tree is fully utilized by three ongoing calls occupying codes $C_{5,2}$, $C_{4,1}$ and $C_{4,2}$ with code indices $I_{5,2}$=4, $I_{4,1}$=3, and $I_{4,2}$=4 respectively. If a new call of rate 8$R$ arrives, there is no vacant code available, and the algorithm searches for busy code in layer $l \ge \log_2(8) = 3$ with highest code index value greater than 2. The code $I_{5,2}$ in the highest layer has an index equal to 4, and the ongoing call $C_{5,2}$ is shifted to $C_{4,3}$ making $I_{4,3}$ =3 and $C_{4,4}$ vacant. The call 8$R$ can now be handled by $C_{4,3}$ (with $I_{4,3}$=2) as shown in Figure4.5(b). If the current status of tree is as in Figure 4.6(a), and a new call of 16$R$ arrives, there is no vacant code with capacity $\ge$16$R$. The call handled by $I_{5,2}$ will be handled by $I_{4,3}$. If the instantaneous traffic load is high with average traffic load less then 8$R$, the call can be handled. The call currently using code $C_{5,2}$ is shifted to $C_{4,3}$. The call

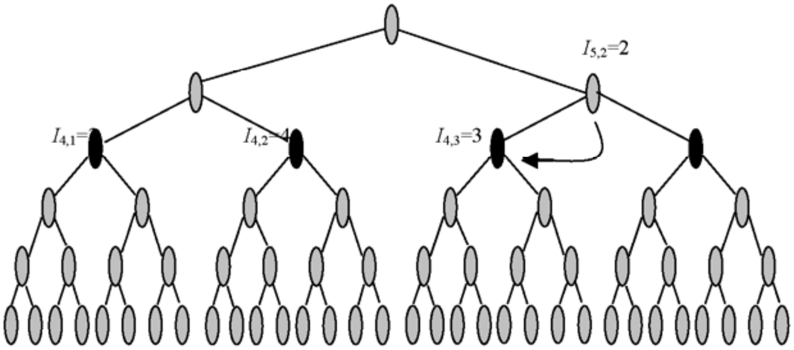


**Figure 4.5(b):** 8*R* call handling

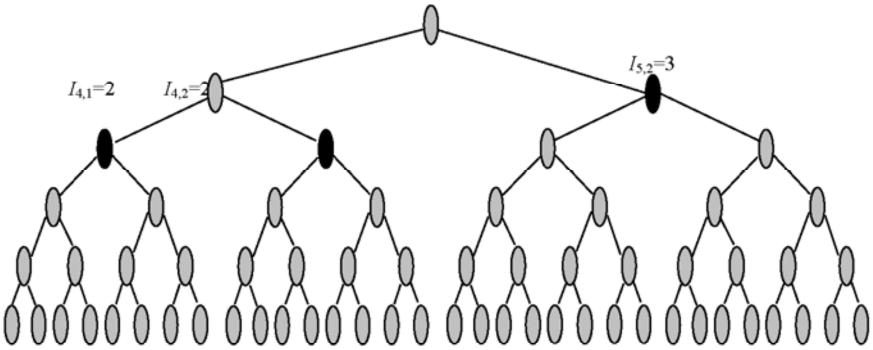


**Figure 4.6(a):** A 6 layer OVSF code tree

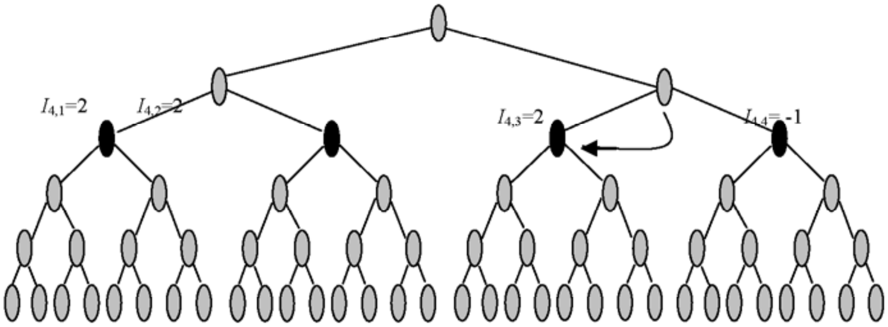


**Figure 4.6(b):** 16*R* call handling

rate is intentionally suppressed to 8*R*. The code $C_{4,4}$ can now be utilized for the incoming 16*R* call as shown in Figure 4.6(b).

## 4.5 Simulation and Results

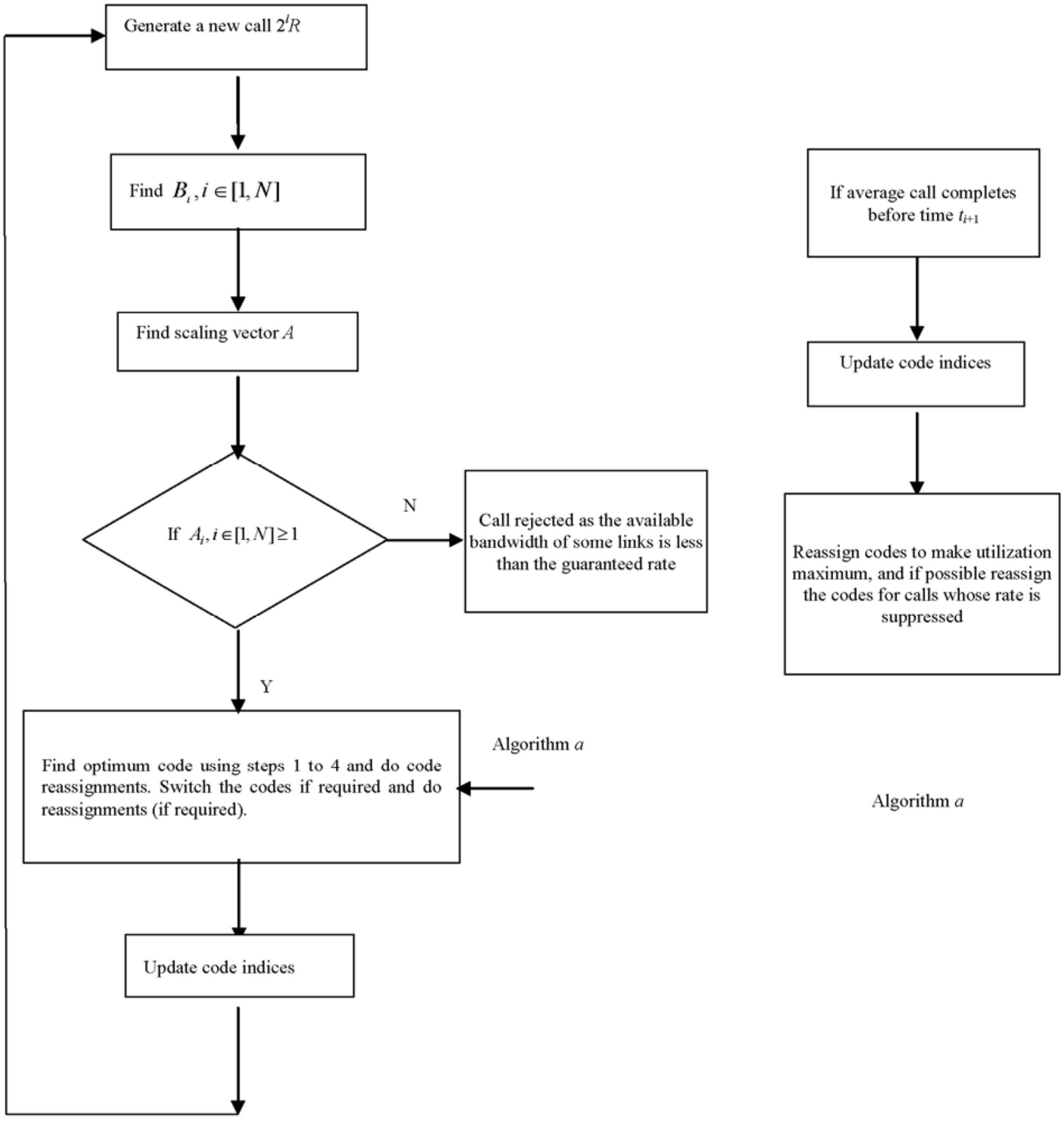


**Figure 4.7:** Flow chart of simulation model for real time calls

The flowchart of simulation model for real time calls is given in Figure 4.7, and for data calls in Figure 4.8. The simulation parameters used to compare proposed design with other existing designs are listed as follows.

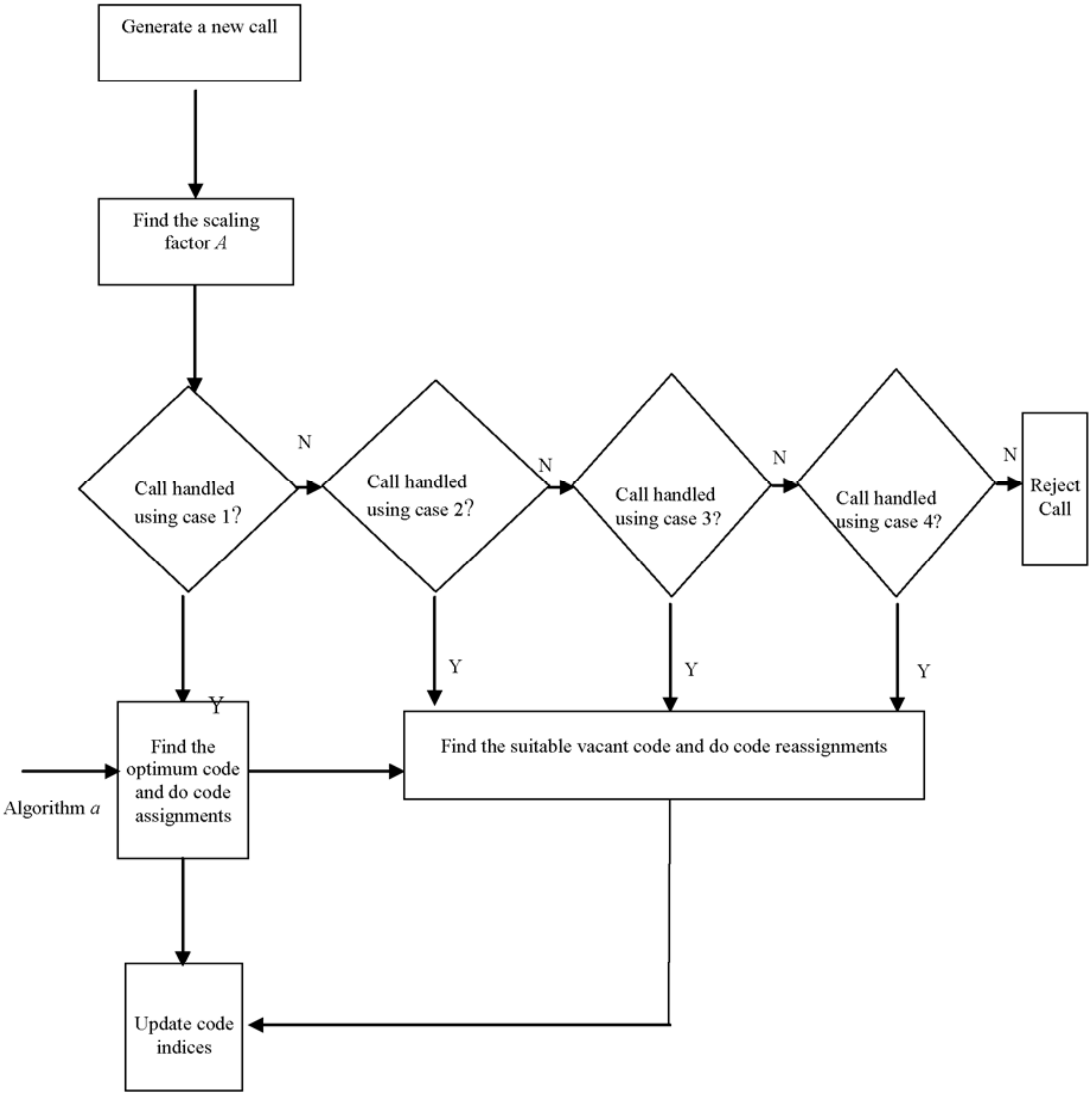
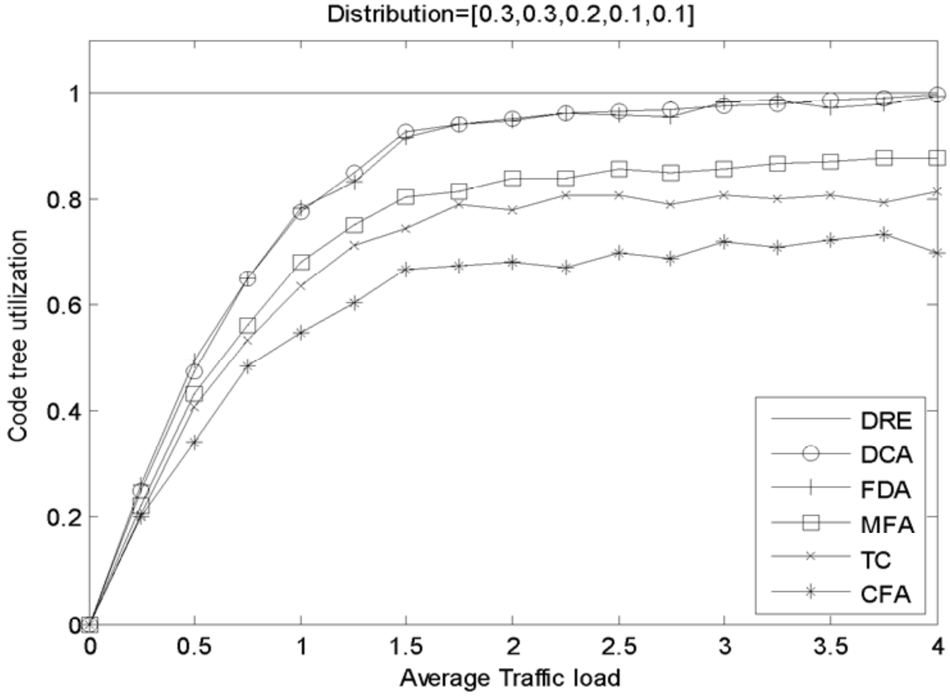


**Figure 4.8:** Flow chart of simulation model for data calls

- There are 5 classes of calls with rates *R*, 2*R*, 4*R*, 8*R*, 16*R*.

- The maximum capacity of the tree is 128*R*, with *R* equal to 7.5 kbps, which require 8 layers in the code tree. The 128*R* code tree capacity is divided into three portions one each for real time, data and mixed rates with capacity 32*R*, 32*R* and 64*R* respectively.
- Arrival rate λ is Poisson distributed with mean value 0-4 calls/minute.
- Call duration is exponentially distributed with mean value $1/\mu$ of 3 minutes.

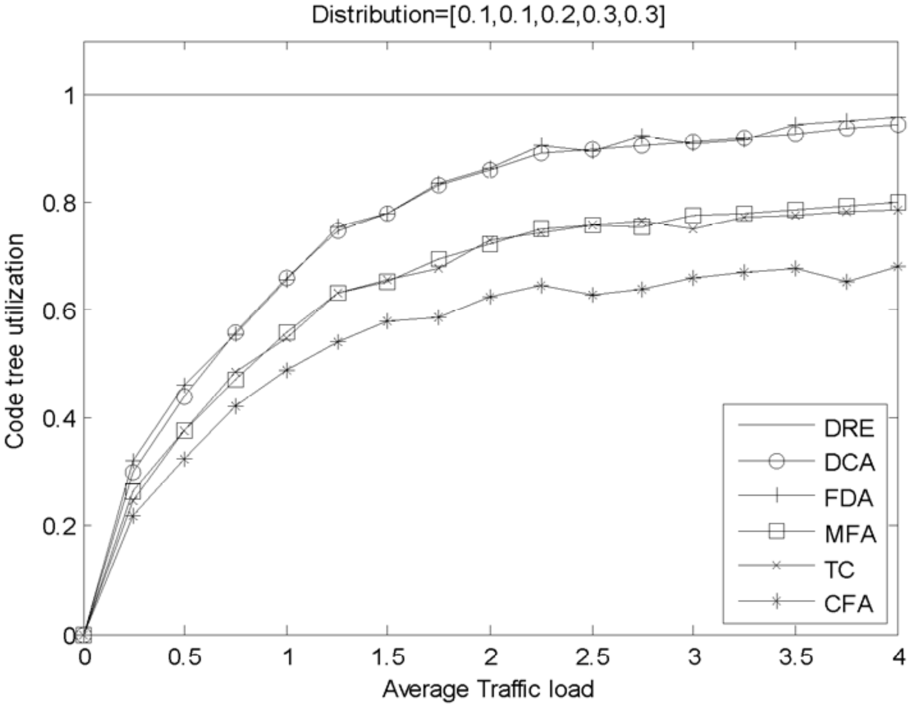


DRE- Data Rate Expansion (ours), DCA- Dynamic Code Assignment, FDA- Fast Dynamic Assignment, MFA- Maximally Flexible Assignment, TC- Time Based Code Scheme, CFA- Crowded First Assignment

**Figure 4.9:** Comparison of code tree utilization for [0.3,0.3,0.2,0.1,0.1] distributions

- Simulation is done for 10000 calls and result is average of 10 simulations.

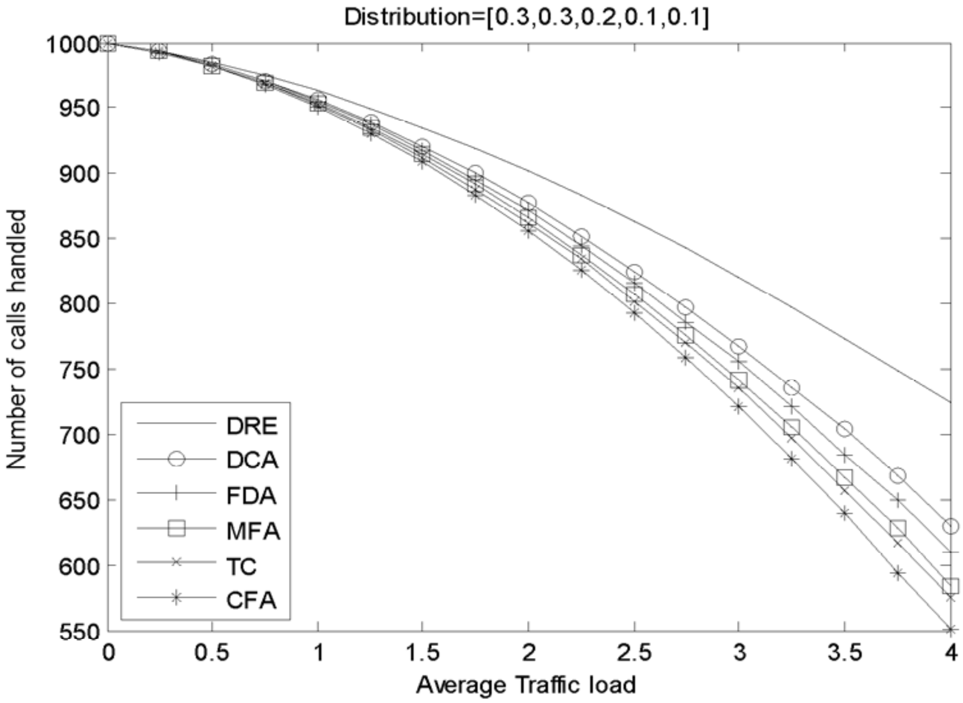
Let $\lambda_i, i \in [1,5]$ is the arrival rate and $\mu_i$ is service rate for $i^{th}$ class calls. Define $\rho_i = \lambda_i / \mu_i$ as traffic load of the $i^{th}$ class calls. The average arrival rate and average traffic load becomes $\lambda = \sum_{i=1}^{5} \lambda_i$ and $\rho = \sum_{i=1}^{5} (\lambda_i / \mu_i)$ respectively. In our simulation, we consider average call duration of all the classes equal, i.e. $1/\mu = 1/\mu_i$. Define



DRE- Data Rate Expansion (ours), DCA- Dynamic Code Assignment, FDA- Fast Dynamic Assignment, MFA- Maximally Flexible Assignment, TC- Time Based Code Scheme, CFA- Crowded First Assignment

**Figure 4.10:** Comparison of code tree utilization for [0.1,0.1,0.2,0.3,0.3] distributions

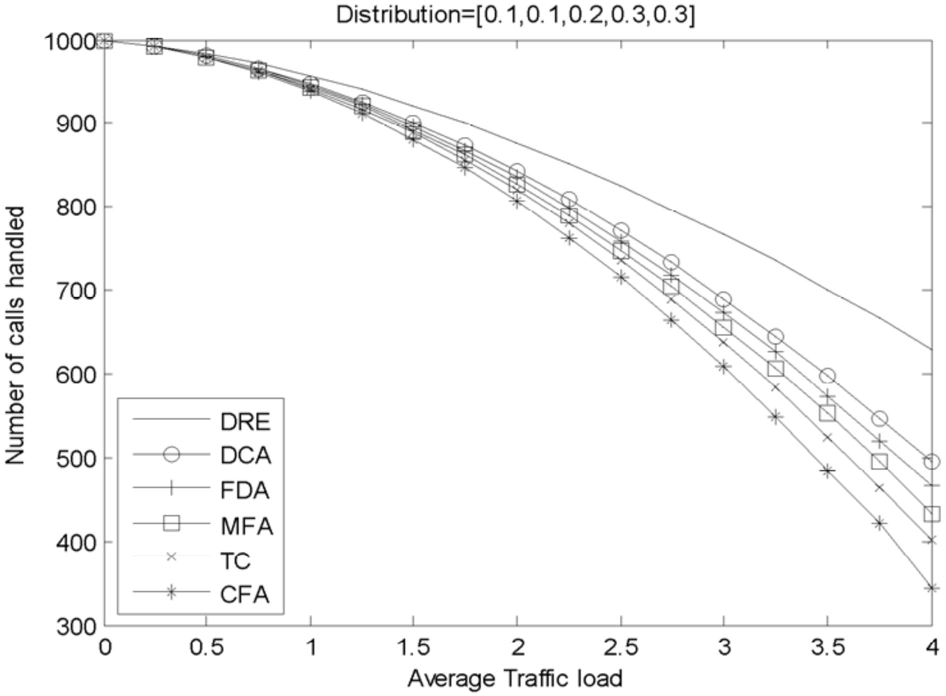
$[p_1,p_2,p_3,p_4,p_5]$ as probability distribution vector where $p_i$ is the arrival probability for $i^{th}$ class calls on average. Two distribution scenarios are investigated, (*a*) [0.3,0.3,0.2,0.1,0.1] where lower rate calls dominate, (*b*) [0.1,0.1,0.2,0.3,0.3] where higher rate calls dominate. The performance of the proposed data rate expansion (DRE) design is compared with the crowded first assignment CFA[16], dynamic code assignment (DCA) [12], fast dynamic assignment FDA [16], time code design TC[18], and maximally flexible assignment (MFA) [19] designs. The code tree utilization comparison is shown in Figure 4.9 and 4.10 for both distributions. The code tree utilization in the proposed design is always 1 because there is flexibility of rate expansion or suppression for data calls. The code tree utilization in dynamic assignment designs DCA and FDA is also close to 1 at higher loads as the code reassignments can make the assignment as compact as possible. The use of DCA requires significant reassignment overhead, which limits their use. The utilization for other designs deviates significantly from 1. The MFA, TC, and CFA design suffers in utilization because there is no



DRE- Data Rate Expansion (ours), DCA- Dynamic Code Assignment, FDA- Fast Dynamic Assignment, MFA- Maximally Flexible Assignment, TC- Time Based Code Scheme, CFA- Crowded First Assignment

**Figure 4.11:** Number of calls handled for [0.3,0.3,0.2,0.1,0.1] distributions

differentiation in real time and non real time calls. Further CFA has least code tree utilization compared to all other methods. The tree utilization further reduces for distribution favoring higher rate calls. This is because the low rate codes are scattered in the code tree, and the parents of these codes in higher layers are blocked. The total number of calls handled by various designs is also compared for two distributions in Figure 4.11 and 4.12. The total calls are assumed to be 10000. As expected, the proposed data rate expansion (DRE) design handles highest number of calls for both distributions. The calls handled in MFA, TC and CFA designs are significantly less as there is no provision for rate scaling. Also, the vacant capacity is scattered in the code tree and the calls are rejected due to the problem of external fragmentation. The number of calls rejected further reduces when the higher rate calls are dominating. This is the reason why the calls handled in Figure4.12) are lesser than the calls handled in Figure 4.11. The calls handled in DCA can approach DRE if the reassignments are done frequently and in addition at the arrival and departure of the call.



DRE- Data Rate Expansion (ours), DCA- Dynamic Code Assignment, FDA- Fast Dynamic Assignment, MFA- Maximally Flexible Assignment, TC- Time Based Code Scheme, CFA- Crowded First Assignment

**Figure 4.12:** Number of calls handled for [0.1,0.1,0.2,0.3,0.3]distributions

## 4.6 Conclusion

The occurrence of scattered vacant codes in the code tree due to random call arrival and departure, increase future high rate calls blocking. The compact single code and multi code assignment schemes discussed use the most scattered vacant code(s) reducing code/call blocking. The multi code design gives the added benefit of handling non quantized rates and reducing the code wastage capacity. The frequent call arrival and completion requires regular update for the most scattered vacant group, which may increase the computation/decision time for new calls.

In contrast to real time calls, data calls can be better handled because there is a flexibility to increase or decrease their rates according to available codes in the OVSF code tree. In this work, a code assignment design with rate variation flexibility is investigated. Further, the rate of the non real time call can be increased only if the call is stored for sufficient time in the buffer available at BS and MS. The code tree utilization is maximized as capacity of all the codes is fully utilized in the code tree. The rate of data call can be increased or decreased depending upon available codes, instantaneous and average traffic load. Both single hop networks and multiple hop networks can be benefited. In the proposed scheme, the size of buffers increases as the design stores most of the non real time calls before allocating codes. Hence, the complexity and cost factor increases, and work can be done to limit the complexity while compromising some utilization. Also, the fairness issue among various data call classes and for mixed system (containing both real time and data calls) can be investigated.

# CHAPTER 5

# SOME OTHER NOVEL CODE ASSIGNMENT SCHEMES

## 5.1 Introduction

This chapter describes few multi code schemes for different requirements. The multi code scheme uses multiple codes to handle each call. In the first multi code design, two different scenarios are considered, first, the rake limited scenario and second, the scattering limited scenario. In the latter two further categories are right code splitting and left code splitting. Second multi code design does code assignment in two ways, first, using higher rate codes and second, using lower rate codes.

## 5.2Multi code Design I

### *5.2.1Rake limited scenario*

Assume *m* rake OVSF WCDMA system for a new user arrival with rate $kR, 1 \le k \le 128$. The procedure for finding the minimum rake required to handle rate *kR* is given below. Find $\max(k_1) \mid kR - k_1R \ge 0$ where $k_1 = 2^{n_1}, 0 \le n_1 \le 6$. If $kR - k_1R = 0$, a single rake is sufficient to handle new incoming call, otherwise define wastage capacity for a single rake system given by

$$W_1 = kR - k_1R \tag{5.1}$$

Therefore for non zero wastage capacity in single rake system, find $\max(k_2) \mid kR - k_1R - k_2R \ge 0$ where $k_2 = 2^{n_2}, 0 \le n_2 \le 5$. The result can be extended to maximum of *m* steps for *m* rake system. After $t \mid t \le m$ number of steps, wastage capacity is

$$W_t = kR - k_1R - k_2R \ldots k_tR \quad = kR - \sum_{i=1}^{t} k_iR \tag{5.2}$$

For *m*=8, there is no wastage capacity but the severe complexity in the BS and UE requires m less than 8. As seen from Equation 5.2, the wastage capacity is the limitation for non quantized rates $(kR \neq 2^n)$ and it reduces as we increase number of rakes.

### **5.2.2*Scattering limited scenario***

The efficient resource allocation is supreme requirement, so all the *m* rakes should be used to handle new call (if possible) irrespective of the minimum number of rakes required to handle new call. In OVSF based systems, the resource allocation can be made efficient if code scattering is least. Code scattering is due to the lower rate calls arrival and random arrival and departure time of calls. The aim of minimum code scattering design is to break incoming rate into fractions such that future availability of high rate codes is highest. The incoming rate is divided into fractions such that all the rakes available are utilized. The minimum scattering design can have two categories.

#### *A. Right code first splitting*

The first part of the algorithm is to find minimum number of rake combiners required to handle new call according to Equations 5.1 and 5.2. Let $t \mid t \leq m$, number of steps leads to zero wastage capacity, i.e. $W_t = kR - \sum_{i=1}^{t} k_i R = 0$ .If $k_{t-1}R \geq 2R$, replace $k_{t-1}R$ by $k_{t-1+j}R$ and $k_{t+j}R$ where $k_{t-1+j} = k_{t+j} = (k_{t-1})/2$. If $(t+j) \leq m-1$ and if $k_{t+j}R \geq 2$ , replace $k_{t+j}R$ by $k_{t+j}$ and $k_{t+1+j}$ where $k_{t+1+j} = k_{t+j} = (k_{t+j})/2$, otherwise if $(t+j) \leq m-1$ and if $k_{t+j}R \leq 1$ , find $k_{t+j-c}$ where $1 \leq c \leq t$ such that $k_{t+j-c} \geq 2$. Then replace $k_{t+j-c}R$ by $k_{t+j}R$ and $k_{t+1+j}R$ where $k_{t+1+j} = k_{t+j} = (k_{t+j-c})/2$. The procedure is repeated until *t+j* becomes equal to *m*. The wastage capacity is given by $kR - \sum_{i=0}^{m-1} k_i R$.

#### *B. Left code first splitting*

The first part of the algorithm is to find minimum number of rake combiners required to handle new call according to Equations 5.1 and 5.2. Let $t \mid t \leq m$, number of steps leads to zero wastage capacity, i.e. $W_t = kR - \sum_{i=1}^{t} k_i R = 0$ .Then in this case, if $k_1 R \geq 2$ , replace $k_1 R$ by $k_1^{'} R$ and $k_t R$ where $k_t = k_1^{'} = (k_1)/2$. If $t \leq m-1$continue this until $t$ becomes

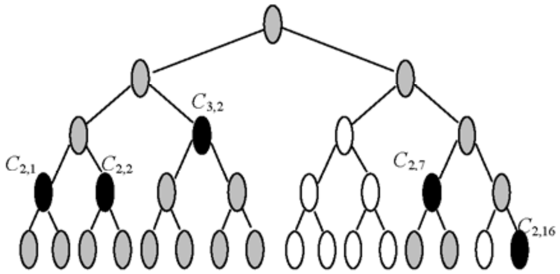


**Figure 5.1:** Illustration of multi code assignment scheme using left code first splitting

equal to *m*. The wastage capacity is given by $kR - \sum_{i=1}^{m-1} k_i R$.

To illustrate the multi code assignment schemes, consider an OVSF code tree with five layers. For a new call with rate 6*R*, the multi code schemes works as follows. In multi code utilizing minimum rakes, the 6*R* call can be split into two rakes with rates of 4*R* and 2*R*. If leftmost first assignment is considered, the vacant 4*R* and 2*R* codes from the left of the code tree can be used and the new tree status is given in Figure 5.1. If multi code assignment with fixed number of rakes (say 3) is employed, we have two options to handle new call depending upon whether left splitting is used or right splitting is used. For left split first category, the 6*R* call can be represented by 4*R*+2*R* and further by 2*R*+2*R*+2*R* (using all 3 rakes). For right split first category, the 6*R* call can be represented by 4*R*+2*R* and further by 4*R*+*R*+*R*. After handling 6*R* call, the status of the code tree in shown in Figure 5.2.

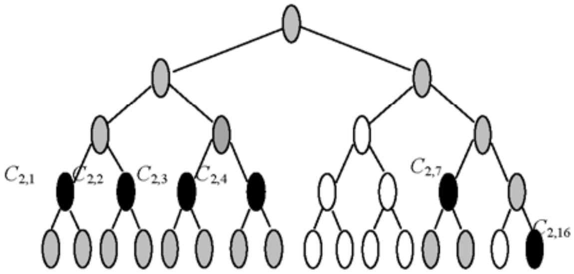
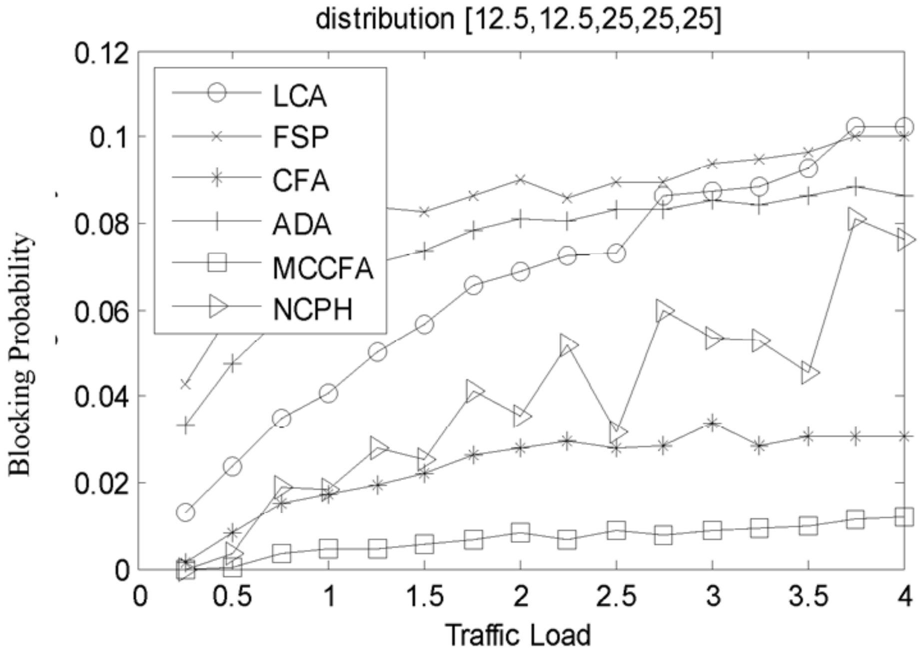


**Figure 5.2:** Illustration of multi code assignment scheme using right code first splitting

### *5.2.3 Simulation and Results*



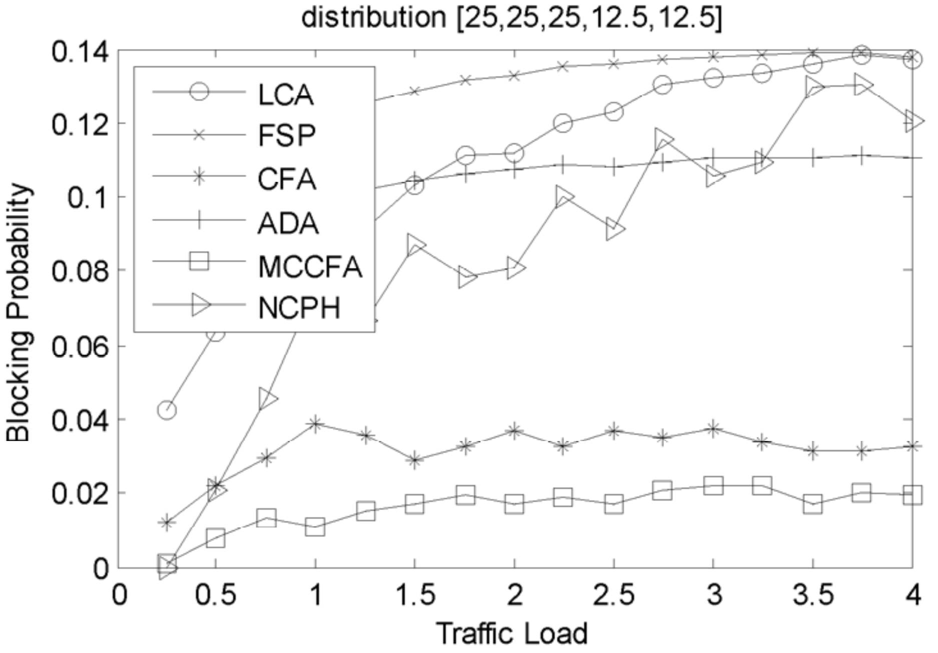
LCA- Leftmost First Code Assignment, FSP- Fixed Set Portioning Scheme, CFA- Crowded First Assignment ADA- Adaptive Code Assignment, MCCFA- Multi Code with Crowded First Assignment (ours), NCPH- Next Code Precedence High

**Figure 5.3:** Code blocking vs traffic load for rate distribution [12.5,12.5,25,25,25]

*A. Input Data*

- Call arrival process is Poisson with mean arrival rate, $\lambda$ =0-4 calls/ unit time.
- Call duration is exponentially distributed with a mean value, $1/\mu$ =3 units of time.
- Possible OVSF code rates are *R, 2R, 4R*, 8*R* and 16*R.*
- For Fixed set partitioning scheme, the tree capacity given to *R*, 2*R*, 4*R*, 8*R* and 16*R* users is 32*R*, 16*R*, 16*R*, 32*R* and 32*R.* This is done because *R* to 16*R* rates do not divide the tree in equal portions.

- The number of rakes used in multi rake scheme is 3.



LCA- Leftmost First Code Assignment, FSP- Fixed Set Portioning Scheme, CFA- Crowded First Assignment ADA- Adaptive Code Assignment, MCCFA- Multi Code with Crowded First Assignment (ours), NCPH- Next Code Precedence High

**Figure 5.4:** Code blocking vs traffic load for rate distribution [25,25,25,12.5,12.5]

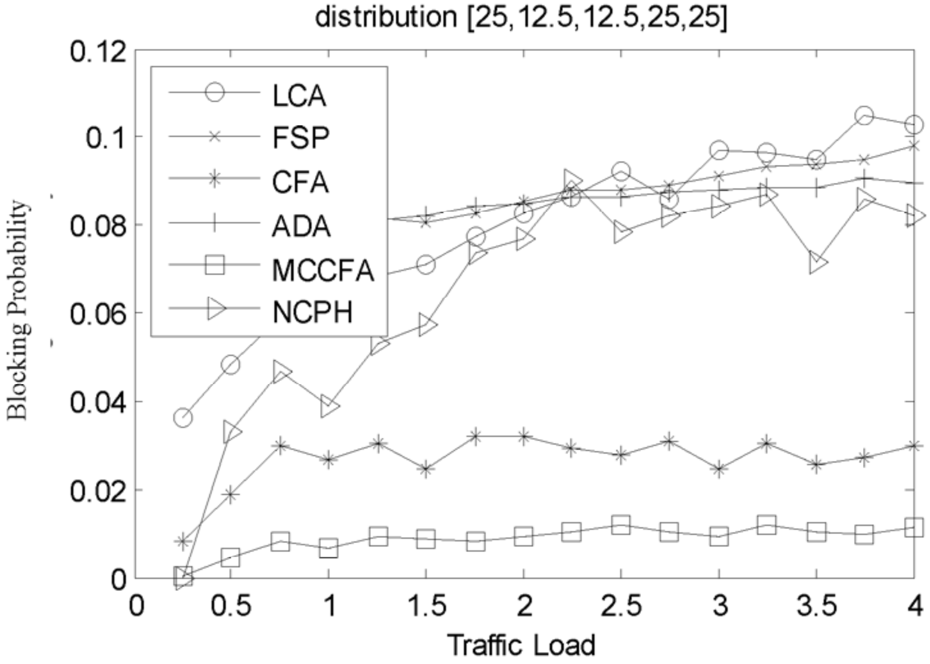
*B. Results*

We compare the code assignment and reassignment schemes for number of codes searched and blocking probability.

*Code blocking*

As assumed, we consider five different quantized rate arrival classes $\lambda \in \{\lambda_0, \lambda_1, \lambda_2, \lambda_3, \lambda_4\}$. The service time is $1/\mu$ for all traffic classes. The code blocking comparison is done for different call rate distributions. Let the rate distribution is given by [$p_0$,$p_1$,$p_2$,$p_3$,$p_4$], where $p_i$ is the probability of user with rate $2^i R$. The distributions considered are

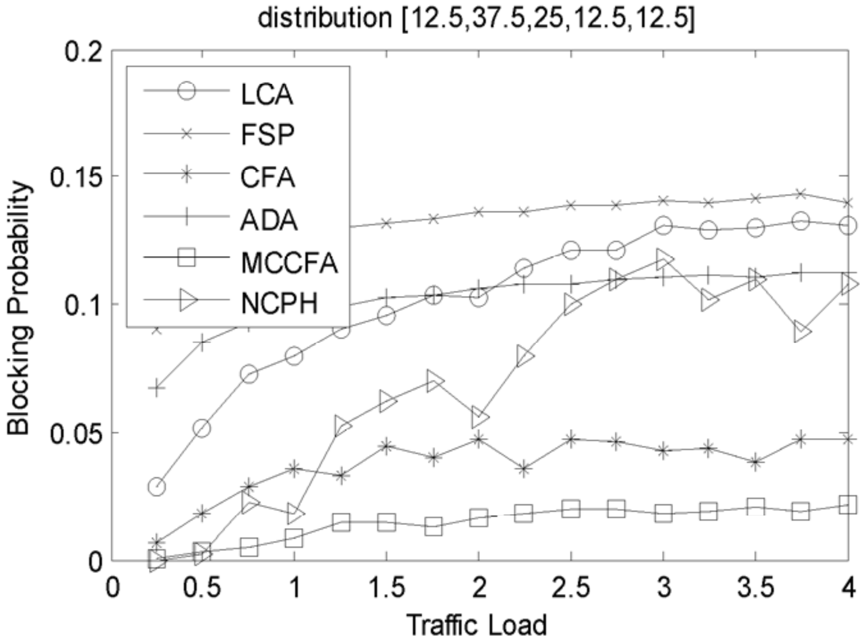
- [12.5, 12.5, 25, 25, 25], i.e. the percentage probability for rate $R$, $2R$, $4R$, $8R$, $16R$ is 12.5, 12.5, 25, 25, 25. High rates dominate in this distribution.
- [25, 25, 25, 12.5, 12.5], i.e. low rates dominate traffic.
- [25, 12.5, 12.5,25, 25] and
- [12.5, 37.5, 25, 12.5, 12.5]



LCA- Leftmost First Code Assignment, FSP- Fixed Set Portioning Scheme, CFA- Crowded First Assignment ADA- Adaptive Code Assignment, MCCFA- Multi Code with Crowded First Assignment (ours), NCPH- Next Code Precedence High

**Figure 5.5:** Code blocking vs traffic load for rate distribution [25, 12.5,12.5,25,25]

The blocking probability for the code group $k \in \{0,1....4\}$ is given as



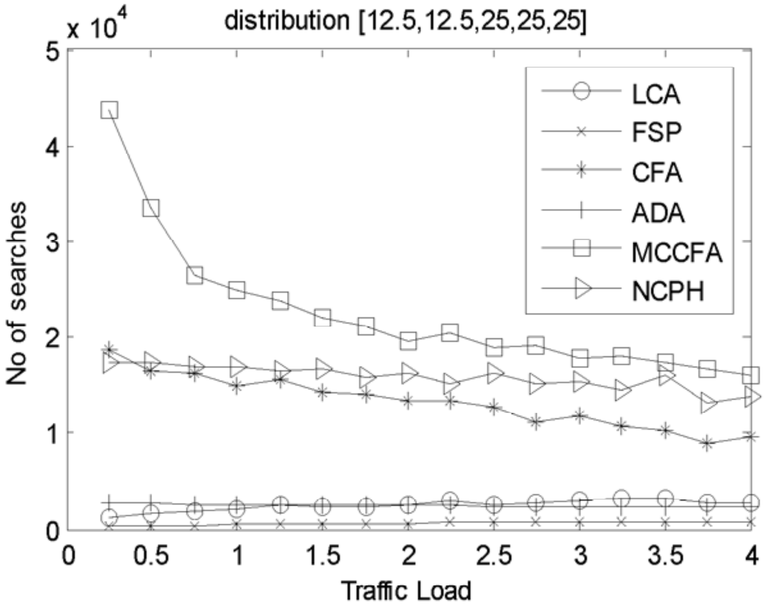
LCA- Leftmost Code Assignment, FSP- Fixed Set Portioning Scheme, CFA- Crowded First Assignment, ADA- Adaptive Code Assignment, MCCFA- Multi Code with Crowded First Assignment (ours), NCPH- Next Code Precedence High

**Figure 5.6:** Code blocking vs traffic load for rate distribution [12.5, 3 7.5, 25, 12.5, 12.5]

$$P_k = \frac{\rho_k^{G_k} / G_k!}{\sum_{n=1}^{G_k} \rho_k^n / n!} \qquad (5.3)$$

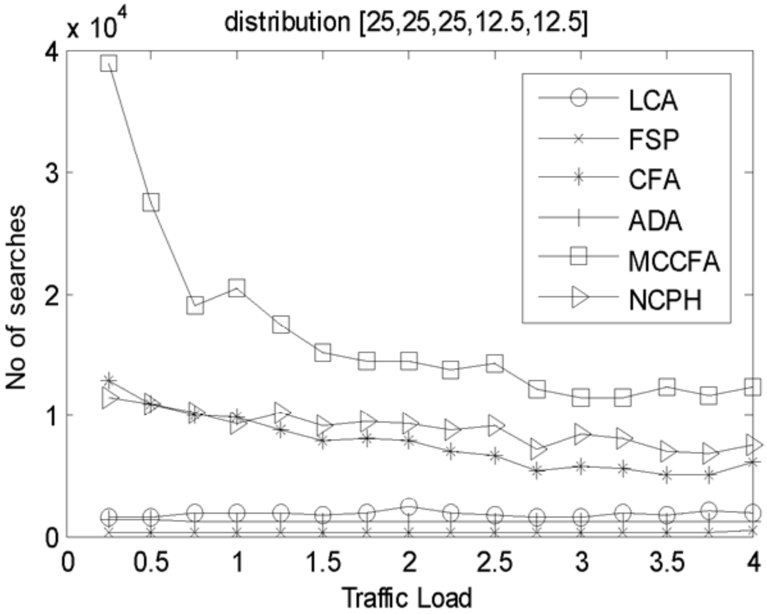
The code blocking comparison for various schemes is shown in Figure 5.3, 5.4, 5.5 and 5.6 respectively. If $B_x$ represents the code blocking in scheme $x$, the blocking in all the schemes is given by $B_{FSP}>B_{ADA}>B_{LCA}>B_{CFA}>B_{DCA}>B_{MCCFA}$. It is a good practice to use CFA, DCA or MCCFA for better results in terms of blocking. We can use hybrid of these three schemes with FSP to incorporate decision time benefit.

### *Code Searches*



LCA- Leftmost Code Assignment, FSP- Fixed Set Portioning Scheme, CFA- Crowded First Assignment
ADA- Adaptive Code Assignment, MCCFA- Multi Code with Crowded First Assignment (ours), NCPH- Next Code Precedence High

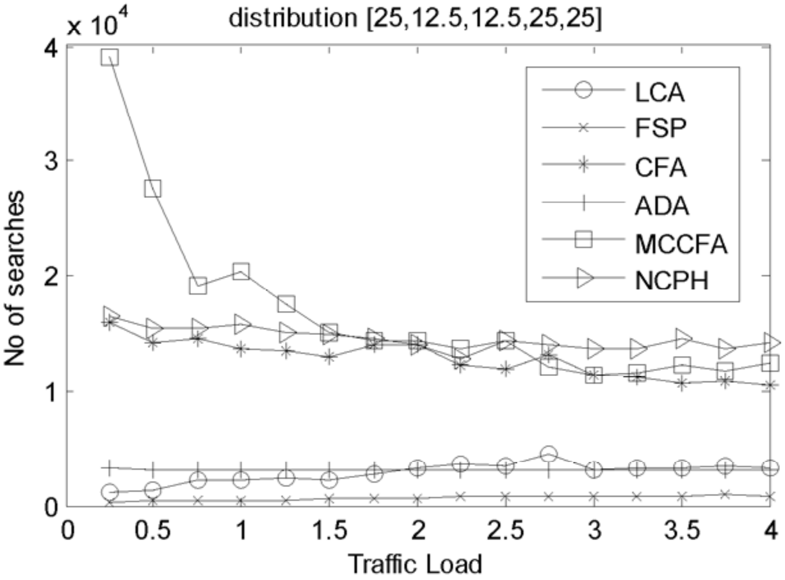
**Figure 5.7:** Number of code searches vs traffic load for rate distribution [12.5, 12.5, 25, 25, 25]



LCA- Leftmost Code Assignment, FSP- Fixed Set Portioning Scheme, CFA- Crowded First Assignment, ADA- Adaptive Code Assignment, MCCFA- Multi Code with Crowded First Assignment (ours), NCPH- Next Code Precedence High

**Figure 5.8:** Number of code searches vs traffic load for rate distribution [25, 25, 25, 12.5, 12.5]

The total number of codes in the system is, $G_0 + G_1 + G_2 + G_3 + G_4$, where $G_x$ is the total

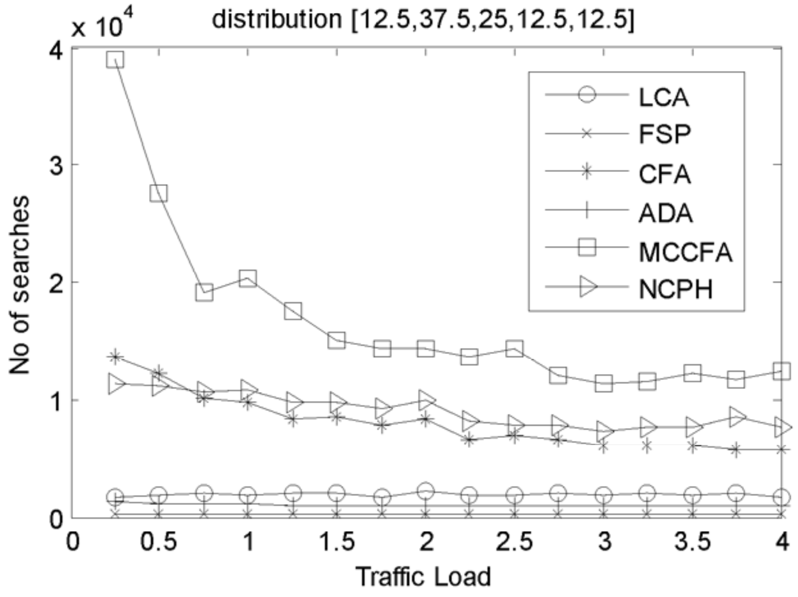


LCA- Leftmost Code Assignment, FSP- Fixed Set Portioning Scheme, CFA- Crowded First Assignment, ADA- Adaptive Code Assignment, MCCFA- Multi Code with Crowded First Assignment (ours), NCPH- Next Code Precedence High

**Figure 5.9:** Number of code searches vs traffic load for rate distribution [25, 12.5, 12.5, 25, 25]

number of codes corresponding to class $x$in the system. As discussed earlier, the calculation of number of code searched is measure of speed of the code assignment scheme. We consider same rate distribution as given in code blocking section. In adaptive code assignment (ADA) scheme, the code reservation is done in advance for the codes of lower layers. This requires less number of codes to be searched before getting suitable vacant code. In next code precedence high (NCPH) scheme, the code next to the assigned code as well as all of its ancestors and descendants are given a two dimensional precedence number (n,y), where ‘n’ is the layer number and ‘y’ is the vacant code priority number. The precedence number is to be used by next incoming calls. Also the code blocking is smaller than the most of assignment schemes which do not incorporate reassignment facility. The code searches results are given in Figure 5.7, 5.8, 5.9 and 5.10

for leftmost first code assignment scheme (LCA), fixed set portioning scheme (FSP), crowded first assignment (CFA), adaptive code assignment (ADA) [67] scheme, next code precedence high (NCPH) scheme and multi code with crowded first assignment (MCCFA) , which is the proposed scheme. If $N_x$ denotes number of codes searched in scheme $x$, the various code assignment schemes searches can be listed as $N_{FSP}<N_{LCA}<N_{ADA}<N_{CFA}<N_{NCPH}<N_{MCCFA}$.



LCA- Leftmost Code Assignment, FSP- Fixed Set Portioning Scheme ,CFA- Crowded First Assignment, ADA- Adaptive Code Assignment, MCCFA- Multi Code with Crowded First Assignment (ours), NCPH- Next Code Precedence High

**Figure 5.10:** Number of code searches vs traffic load for rate distribution [12.5, 37.5, 25, 12.5, 12.5]

## 5.3 Multi code Design II

The multi code design consists of two steps:

a. Division of incoming rate into quantized fractions depending upon the number of rakes available

b. Assign code to each of the fraction using LCA or CFA. In most of our discussion we use CFA scheme as it gives better results.

The division of the incoming rate into fractions can have number of options .For *k* rake multi code system with input rate *rR*, there are three multi code options.

### 5.3.1.*Code assignment using higher rate codes (Algorithm 1)*

Assume $r = 2^n$. The vacant code with rate $2^n$ is searched using CFA scheme and is given to incoming call if at least one is available. If code is not available, use two vacant codes with capacity $2^{n-1}R$. If two vacant codes are not available, number of codes with the capacity $2^{n-2}R$ each is used and is given by

$$a_{n-2} = (r - a_{n-1} \times 2^{n-1}) / 2^{n-1} \tag{5.4}$$

where $a_i$ is the number of $2^i$ capacity codes required to handle complete or partial incoming call. Also, if $a_{n-2}$ codes are vacant the call can be handled if $a_{n-1} + a_{n-2} \leq k$.

Equation 5.4 can be generalized for finding the number of $2^{n-i}R$ codes required given by

$$a_{n-i} = (r - \sum_{j=1}^{i-1} a_{n-j} \times 2^{n-j}) / 2^{n-j-1} \tag{5.5}$$

The call can be handled if $\sum_{i=1}^{k-1} a_{n-i} \leq k$ .

In general the multi code used to handle *rR* can be represented by $C = [a_L, a_{L-1}, a_{L-2}, ......a_2, a_1]$.

The coefficient $a_i$ has following properties.

- for all $i > n, a_i = 0$
- for all $i < n, a_i = 0$ or $2^m, \max(m) = j2^m$ for layer *n-j*

### 5.3.2. *Code assignment using lower rate codes (Algorithm 2)*

For integers, $i_1$, *k* and *r*, the new rate *r* can be handled by $i_1$ leaves if $i_1 \leq k$ and $r \leq i_1$. If the condition is not met, codes from layer 1 and 2 are searched. For integers $i_2$, check for $2i_2 + i_1 = r \mid i_1 + i_2 \leq k$, the procedure is generalized for $j^{\text{th}}$ layer and is given by

$$\sum_{j=1}^{l} 2^{j-1} \times i_j = r \mid \sum_{j=1}^{l} i_j \leq k \tag{5.6}$$

The algorithm use more number of lower layer codes and utilize all $k$ rates to handle the call.

### *5.3.3 Weighted code assignment(Algorithm 3)*

In this design the selection of the multi code is done in such a way that the codes from most of the layers are used for single call. The scheme does not favor lower rate or higher rate codes and hence provide fairness to code assignment scheme. The algorithm of the

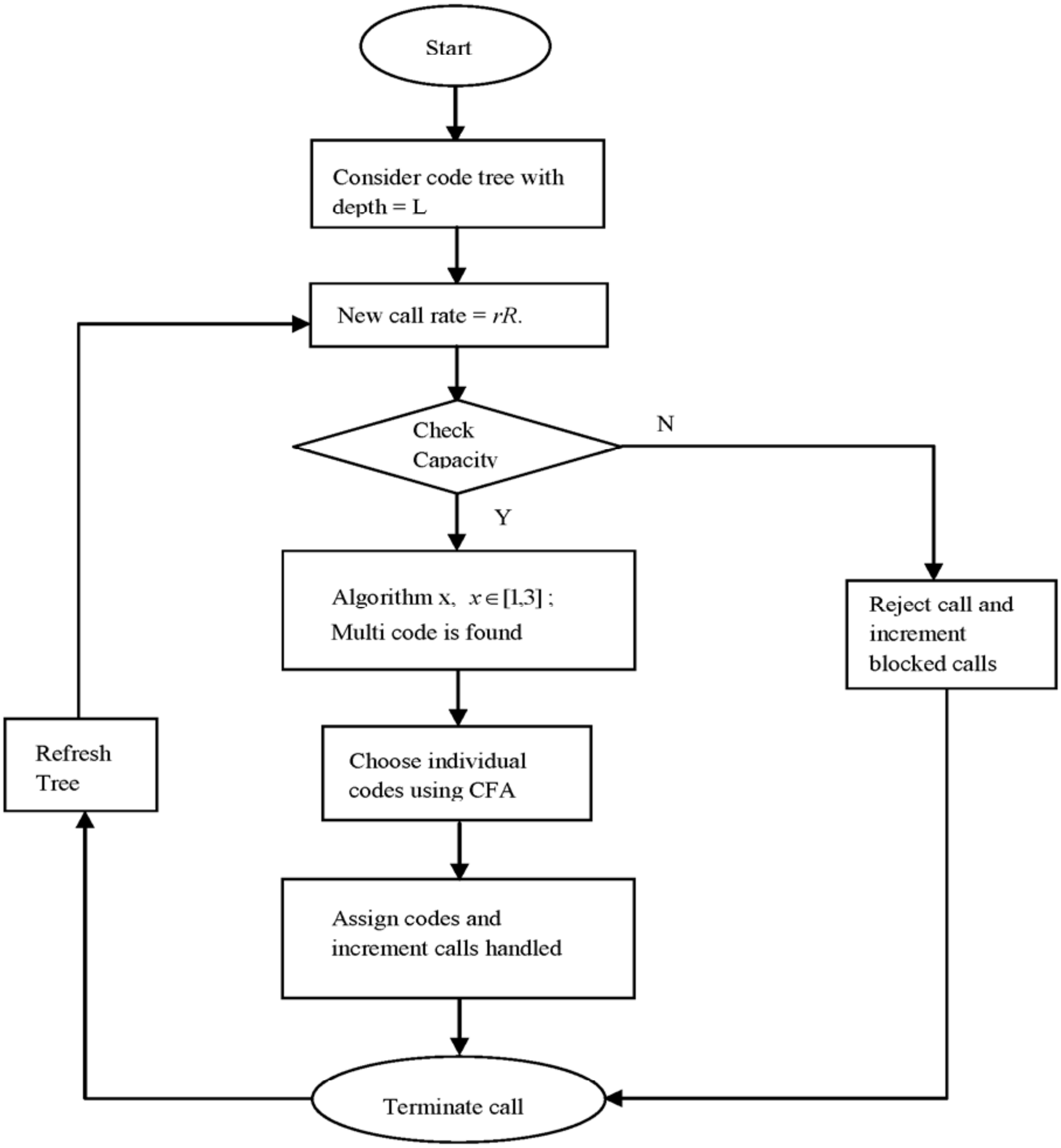


**Figure 5.11:** Flowchart for the proposed multi code scheme

code assignment is as follows.

Define a multi code $C = [i_1, i_2, i_3, ......i_{L-1}, i_L$ , where $i_j$ is the weight (number) of layer $j$ codes used for incoming call. Find $n \mid 2^n = r$. If $n < k-1$, the first multi code choice of $i_j$ is

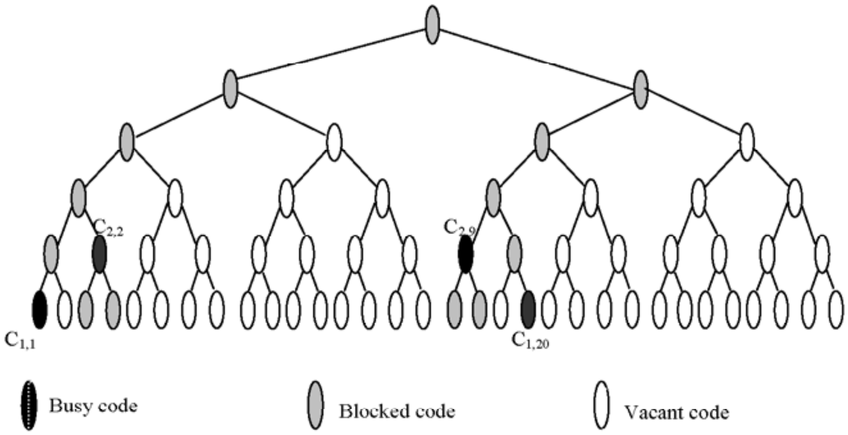


**Figure 5.12:** 6 layer OVSF code tree

given by

$$i_j = \begin{cases} 2; j = 1 \\ 1; 2 \le j \le n\text{-}1 \\ 0; j > n \end{cases} \tag{5.7}$$

If the vacant codes calculated in equation 5.6 are not available, the second multi code

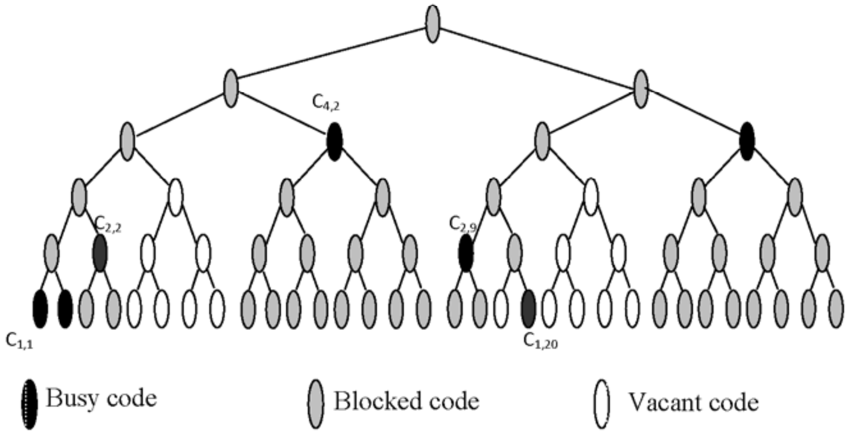


**Figure 5.13:** Handling 16R call using Algorithm 1

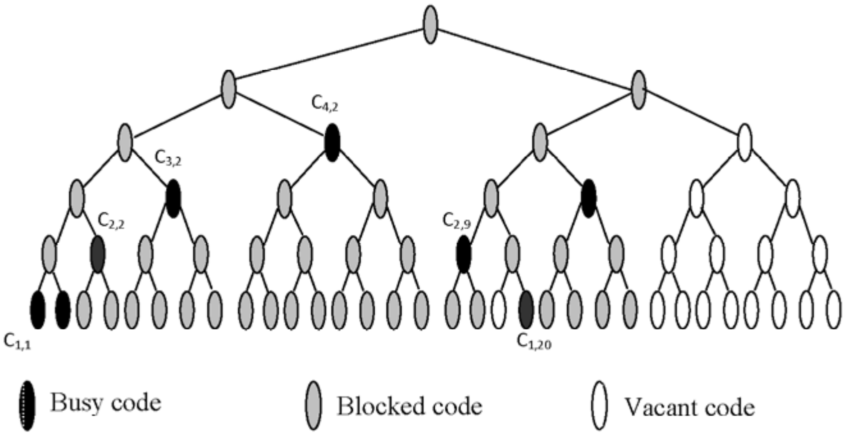


**Figure 5.14:** Handling 16R call using Algorithm 2

scheme is $i_1$=2, $i_j$=1 for $2 \le j \le n-1$ and $i_j = 0$ for $n-1 \le j < L$. The procedure is extended till multi code is found. In general the multi code after *k* steps consists of

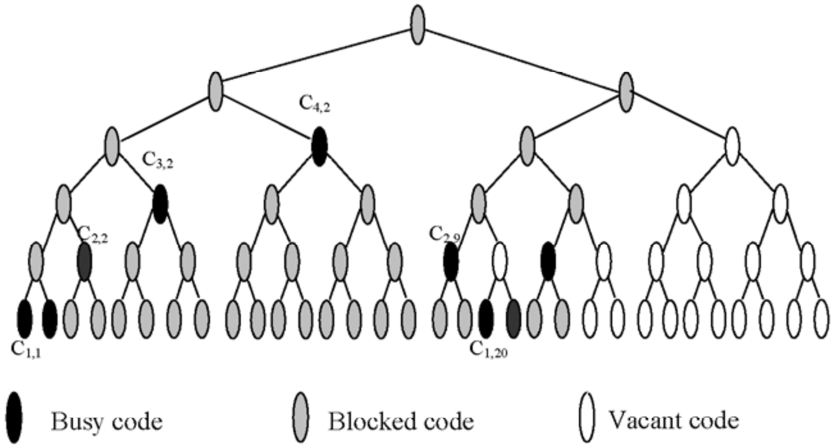


**Figure 5.15:** Handling 16R call using Algorithm 3

coefficients $i_j$ is given by

$$i_j = \begin{cases} 2;\, j = 1 \\ 1;\, 2 \le j \le n\text{-}k \\ 0;\, j > n - k \end{cases} \tag{5.8}$$

If $n \geq k$ . If n<k-1, the first multi code choice of *ij* is

$$i_j = \begin{cases} 0;\ j < n-k+1 \\ 1;\ n-1 \leq j \leq n\text{-}k+2 \\ 2;\ j = n-k+1 \end{cases} \tag{5.9}$$

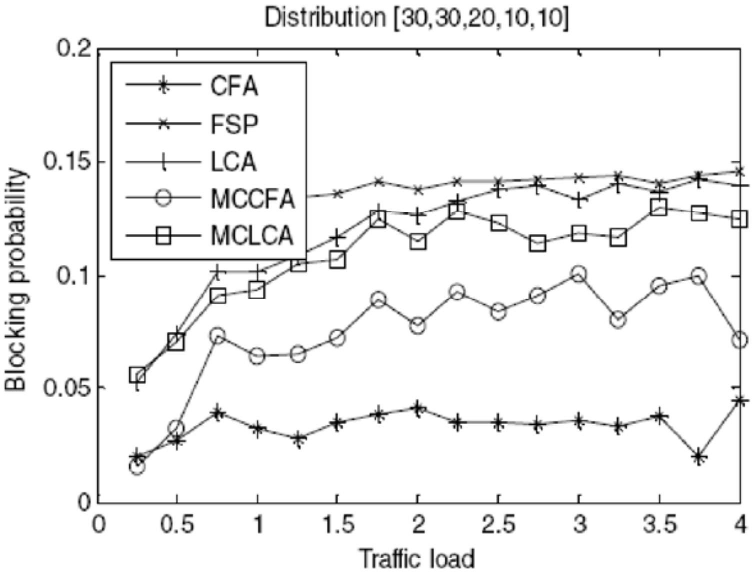
If the vacant codes calculated are not available, call is rejected. The individual code in each of the multi code is chosen according to crowded first code schemes. The flowchart of the multi code scheme is illustrated in Figure 5.11.

The multi code scheme is illustrated using a 6 layer code tree as shown in Figure 5.12. Figure 5.13, 5.14 and 5.15 illustrate the multi code scheme using algorithm 1, 2 and 3 respectively.

### *5.3.4 Simulation and Results*

*A. Input Data*

- Call arrival process is Poisson with mean arrival rate λ =1-4 calls/time units.



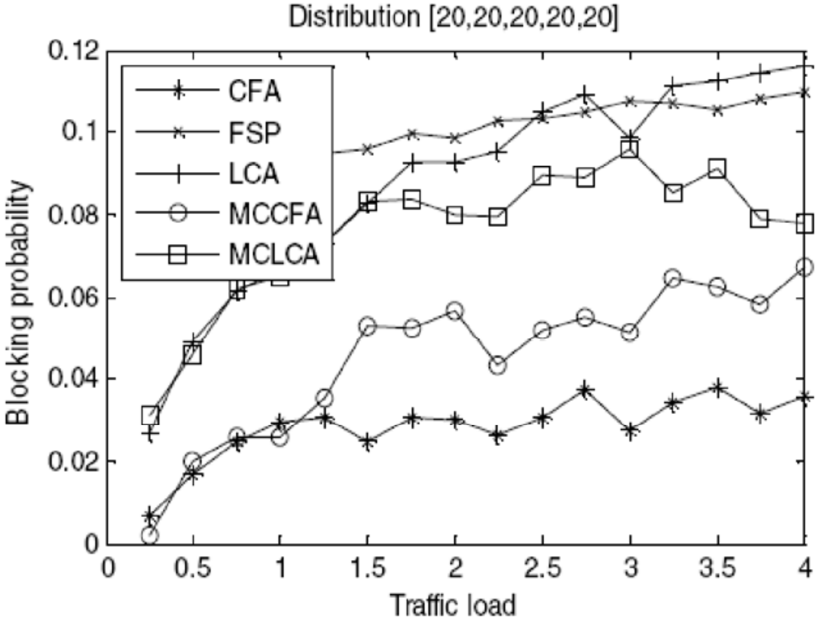
CFA- Crowded First Assignment, FSP-Fixed Set Partitioning, LCA- Left Code Assignment, MCCFA-Multi code Assignment with CFA (ours), MCLCA- Multi code Assignment with LCA (ours)

**Figure 5.16:** Blocking probability vs traffic load for [30, 30, 30, 10, 10]

- Call duration is exponentially distributed with a mean value $1/\mu=3$ time unit.
- Possible arrival rates of users 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.
- For multi code schemes, number of rakes considered are 3.

*B. Results*

The input calls are divided into five classes $\lambda_i, i=[1,5]$, which are the coefficients of arrival rate vector $\lambda$, i.e. $\lambda=[\lambda_1,\lambda_2,\lambda_3,\lambda_4,\lambda_5]$. The number of servers is $G_k=2^{k-1}, k=[1,2,3,4,5]$ in the $k^{\text{th}}$ layer corresponding to $G_k$ number of vacant codes. The formulae for the number of servers vary according to the arrival distribution. The total codes (servers) in the system for five set of classes are given by set G = {$G_1$, $G_2$, $G_3$, $G_4$, $G_5$}. The maximum number of servers used to handle new call is equal to the number of rake combiners. Traffic load for the $k^{\text{th}}$ class of users is given by $\rho_k=\lambda_k/\mu$. The code



CFA- Crowded First Assignment, FSP-Fixed Set Partitioning, LCA- Left Code Assignment, MCCFA-Multi code Assignment with CFA (ours), MCLCA- Multi code Assignment with LCA (ours)

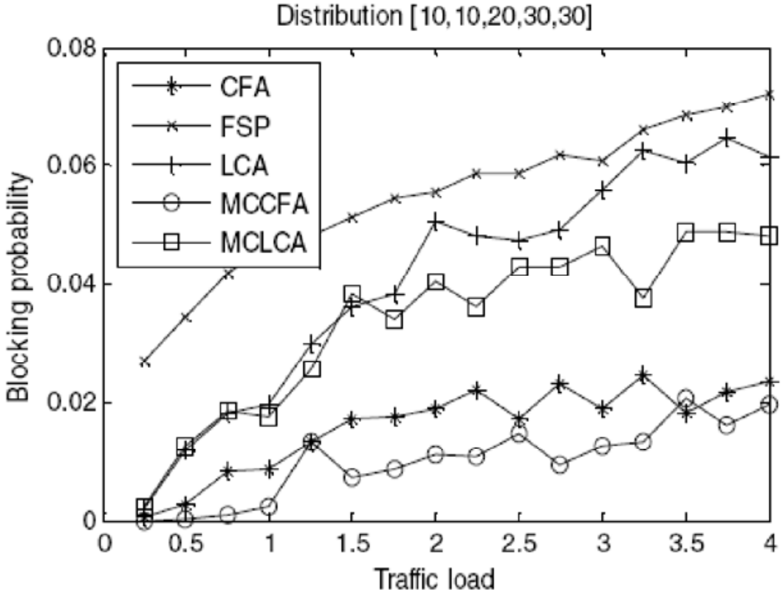
**Figure 5.17:** Blocking probability vs traffic load for [20, 20, 20, 20, 20]

blocking for the $k^{\text{th}}$ class is defined by

$$P_{B_k} = \frac{\rho_k^{G_k} / G_k!}{\sum_{n=1}^{G_k} \rho_k^n / n!} \tag{5.10}$$

The average code blocking for five class system is

$$P_B = \sum_{k=1}^{5} \frac{\lambda_k}{\lambda} P_{B_k} \tag{5.11}$$



CFA- Crowded First Assignment, FSP-Fixed Set Partitioning, LCA- Left Code Assignment, MCCFA-Multi code Assignment with CFA (ours), MCLCA- Multi code Assignment with LCA (ours)

**Figure 5.18:** Blocking probability vs traffic load for [10, 10, 20, 30, 30]

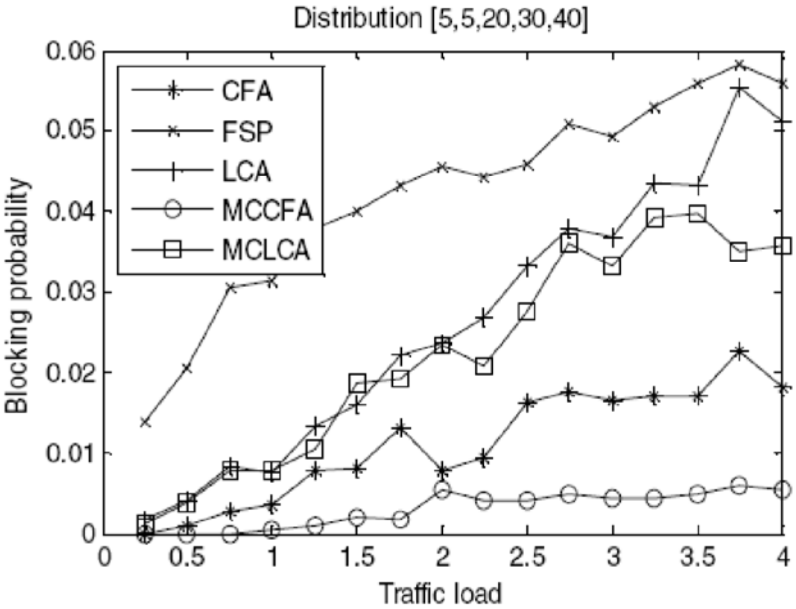
The percentage code blocking is $P_B \times 100$. Let the arrival distribution for the five classes is given by [$p_1$, $p_2$, $p_3$, $p_4$, $p_5$], where $p_i, i = [1,5]$ is the percentage probability of $i^{\text{th}}$ class of users with rate $2^{5-i}R$. The simulation is done for 10000 new calls and result is the average of 10 simulations. Four arrival distributions considered are given below

- [30, 30, 20, 10, 10], i.e. the percentage probability for rate 16*R*, 8*R*, 4*R*, 2*R*, *R* is 30, 30, 20, 10, 10. High rates dominate in this distribution.
- [20, 20, 20, 20, 20], i.e. the uniform capacity distribution.

- [10, 10, 20, 30, 30] and
- [5, 5, 20, 30, 40], i.e. low rates dominate.

Two configurations of the multi code design are considered (a) multi code LCA (MCLCA) (b) multi code CFA (MCCFA). The scheme is compared with LCA, FSP and CFA schemes discussed in Chapter 1.

Result in Figure 5.16 and 5.17 show that for high rates dominating scenario and uniform distribution, traditional CFA outperforms all code assignment schemes. In the low rates domination scenario as shown in Figure 5.18 and 5.19, which is more practical as there are large low rate real time calls, the multi code assignment with CFA (MCCFA) gives best results.



CFA- Crowded First Assignment, FSP-Fixed Set Partitioning, LCA- Left Code Assignment, MCCFA-Multi code Assignment with CFA (ours), MCLCA- Multi code Assignment with LCA (ours)

**Figure 5.19:** Blocking probability vs traffic load for [5, 5, 20, 30, 40]

## 5.4. Fair code assignment

### 5.4.1 *Fair single code design*

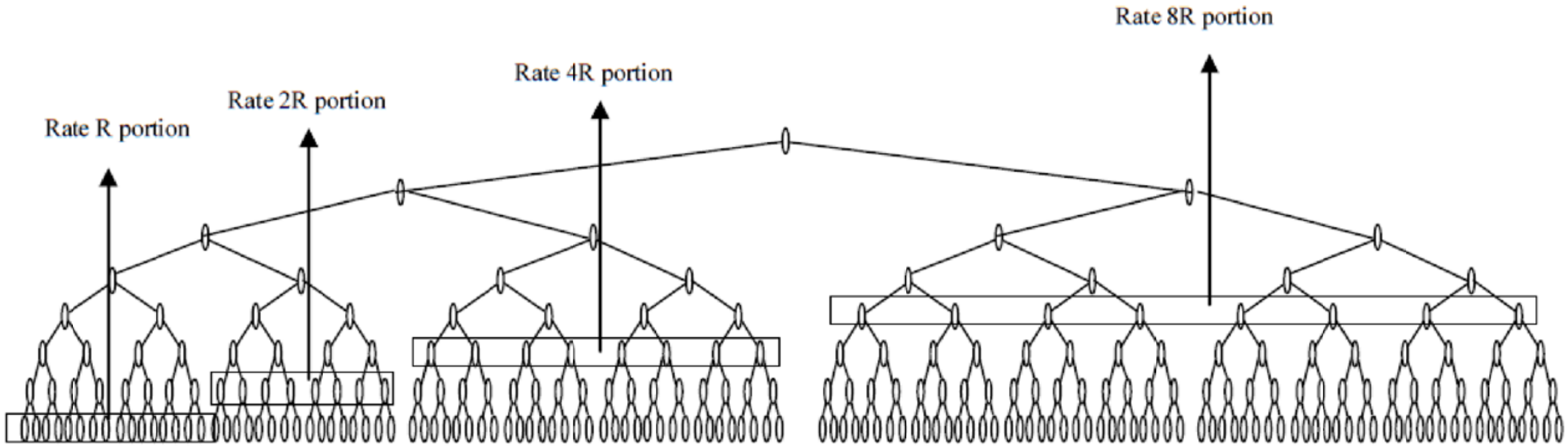


**Figure 5.20:** Division of code tree into 4 portions for 4 class system

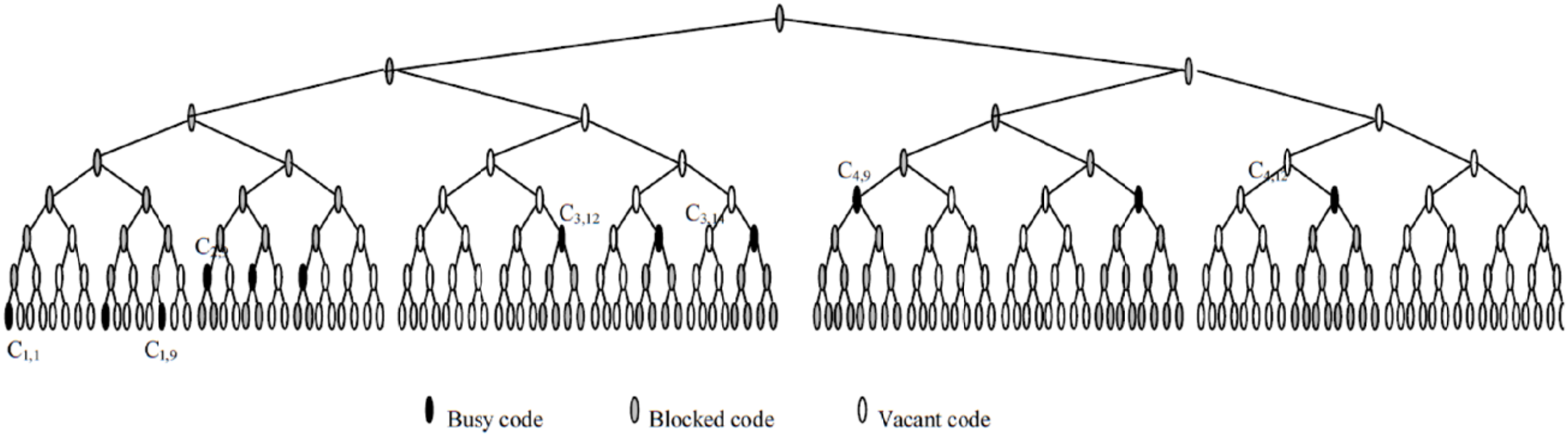


**Figure 5.21:** Code tree prior to new call arrival

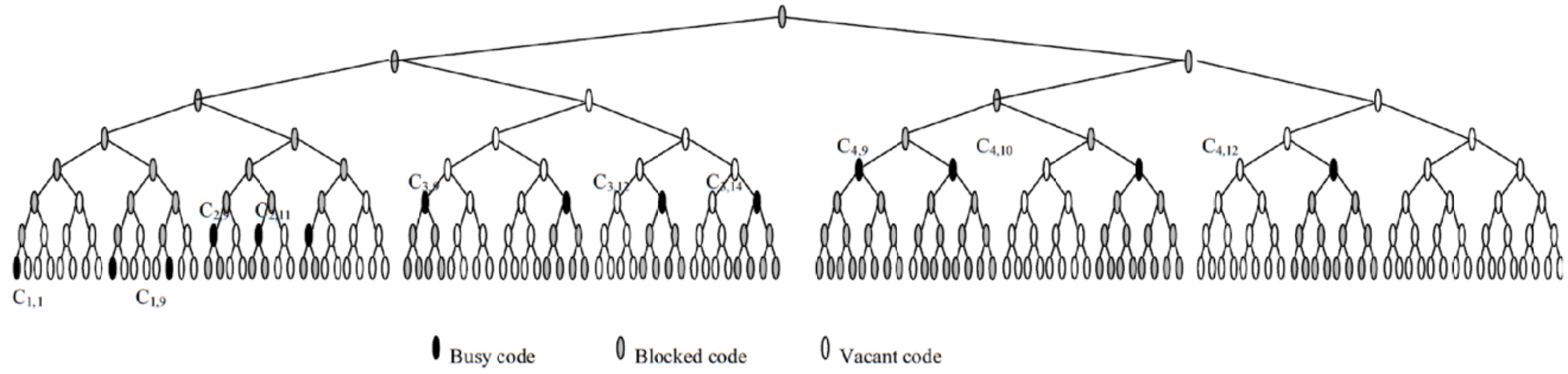


**Figure 5.22:** Code tree after handling 8*R* and 4*R* call arrival

Consider an L layer OVSF system. Let $\lambda_i$ is the arrival rate of the $i^{th}$ class user. We divide the code tree into $n$ different regions. The capacity of the $i^{th}$ region (denoted by $C_i$) is given by

$$C_i = \frac{\lambda_i \times C_{\max}}{\sum_{i=1}^{n} \lambda_l} \tag{5.12}$$

The vacant code is searched in layer $l$. If a new call with rate $2^l R$ arrives, it can be handled using following methods. The aim is to divide the code tree into portions proportional to arrival rates of different classes. The design requires less number of codes searches compared to existing single code scheme. Also the design is better than FSP scheme in the sense that the portioning is proportional to the arrival distribution. For illustration of single code design, we divide the code tree into 4 different regions with capacity allotted to rate $R$, $2R$, $4R$ and $8R$ users is $16R$ (16 vacant codes), $16R$ (8 vacant codes), $32R$ (8 vacant codes) and $64R$ (8 vacant codes) respectively as shown in Figure 5.20. Assuming current status of code tree shown in Figure 5.21, if a new call with rate $8R$ arrives then it search the code in rate $8R$ portion and assign a code $C_{4,10}$ of $8R$ as shown in Figure 5.22. For new call with rate $4R$, the code in rate $4R$ portion is identified and hence code $C_{3,9}$is assigned to new $4R$ call.

### *5.4.2. Fair multi code design*

Consider OVSF system with m rakes. The multi code design consists of the following three steps:

a. Listing out the number of available codes in each layer of the tree.
b. Arranging the rates in descending rate according to their availability.
c. Use the available rate which has maximum availability and according to received call rate, then again repeat step (2) and (3) until received call rate satisfies.

Let $N_l$ is the number of codes available in $l^{th}$ layer, then a vector $N$ is formed such that $N$= [$N1$, $N2$, $N3$, $N4$, $N5$, $N6$, $N7$, $N8$]. Arrange all $N_i$, $1 \leq i \leq 8$ coefficients in descending

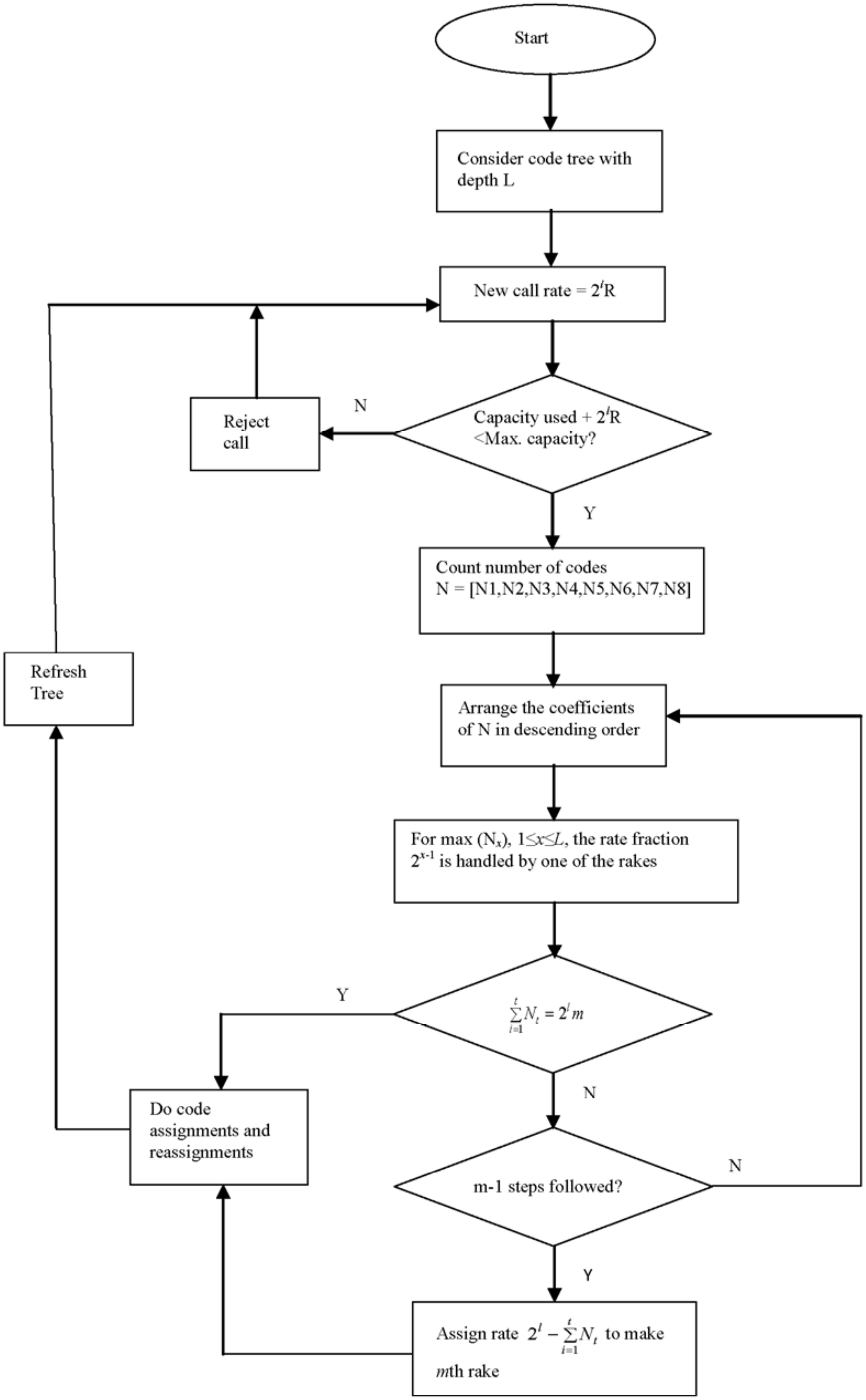


**Figure 5.23:** Flow chart for proposed multi code scheme

order. Consider that in the first attempt the coefficient $N_{x_1}$ is largest. The first rake will handle the rate fraction $2^{x_1}R$. The remaining capacity to be handled by (m-1) rakes is $(2^l - 2^{x_1})R$. Again all $N_i$ coefficients are arranged in descending order. If $N_{x_2}$ is largest in second attempt the second rate will handle rate fraction $2^{x_2}R$. If $x_1 + x_2 = 2^l$, procedure stops otherwise procedure is repeated maximum (m-1) times. After (m-1) steps, the

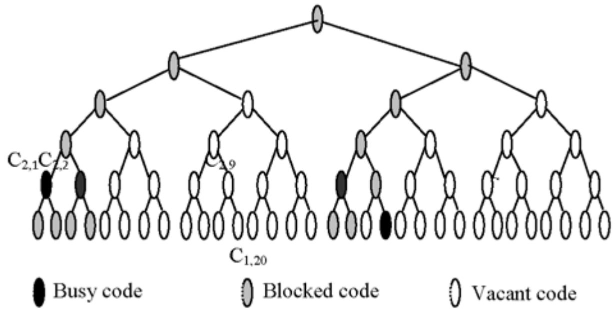


**Figure 5.24:** OVSF Code tree with six layers

fraction of $2^l R$ rate handled will be $\sum_{i=1}^{m-1} 2^{x_i}$. Define $r = 2^l - \sum_{i=1}^{m-1} 2^{x_i}$. Find $\min(k) \mid r \le 2^k$. The rate fraction $r$ will be handled by the $m^{\text{th}}$ rake with capacity $2^k R$.

The flowchart for the multi code assignment scheme is given in Figure 5.23. To illustrate multi code design, consider a six layer tree with the status as shown in Figure 5.24. If a new call with rate 16*R* arrives and system is equipped with four rakes, the algorithm search for the layers with most number of vacant codes. This is repeated for maximum of 4 (equal to the number of rakes) steps. The codes with rates 8*R*, 4*R* and 4*R* will be used for the new call for optimum fairness. The selection of codes is indicated Figure 5.25.

### 5.4.3. Simulation and Results

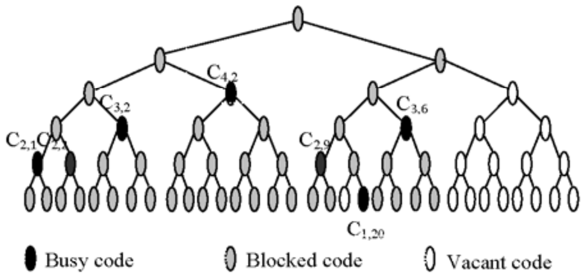
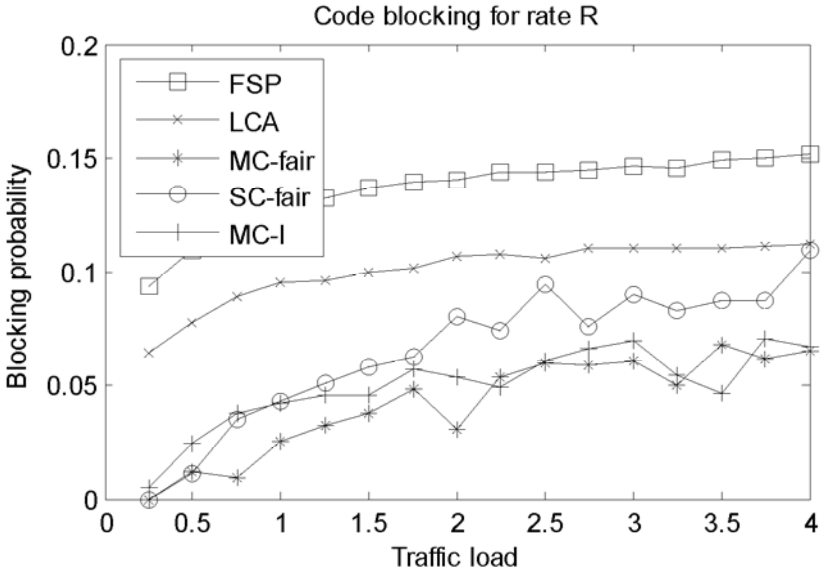


**Figure 5.25:** OVSF code tree for illustration of proposed scheme

### *A. Input Data*

- Call arrival process is Poisson with mean arrival rate, λ=1-4 calls/ unit time.
- Call duration is exponentially distributed with a mean value, $1/\mu$ =3 units of time.
- Possible OVSF code rates are *R, 2R, 4R*, 8*R* and 16*R*.
- For Fixed set partitioning scheme, the tree capacity given to *R*, 2*R*, 4*R*, 8*R* and 16*R* users is 32*R*, 16*R*, 16*R*, 32*R* and 32*R*. This is done because *R* to 16*R* rates do not divide the tree in equal portions.
- The number of rakes used in multi rake scheme is 3.



FSP- Fixed Set Partitioning, LCA- Left Code Assignment, MC-fair-Multi code fair(ours)
SC-fair- Single Code fair(ours), MC-1- Multi code scheme

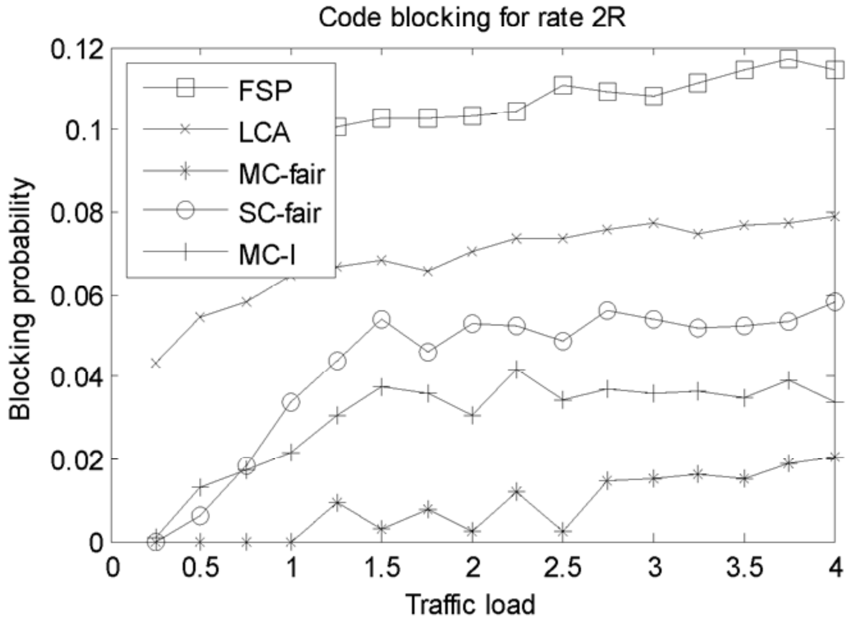
**Figure 5.26:** Blocking probability vs traffic load for R rate users

### *B. Results*

We compare the code assignment and reassignment schemes for number of codes searched and blocking probability.

*Blocking probability*

As assumed, we consider five different quantized rate arrival classes $\lambda \in \{\lambda_0, \lambda_1, \lambda_2, \lambda_3, \lambda_4\}$. The total number of codes in the system is $G_0 + G_1 + G_2 + G_3 + G_4$, 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 blocking probability for the code group $k \in \{0,1....4\}$ is given as

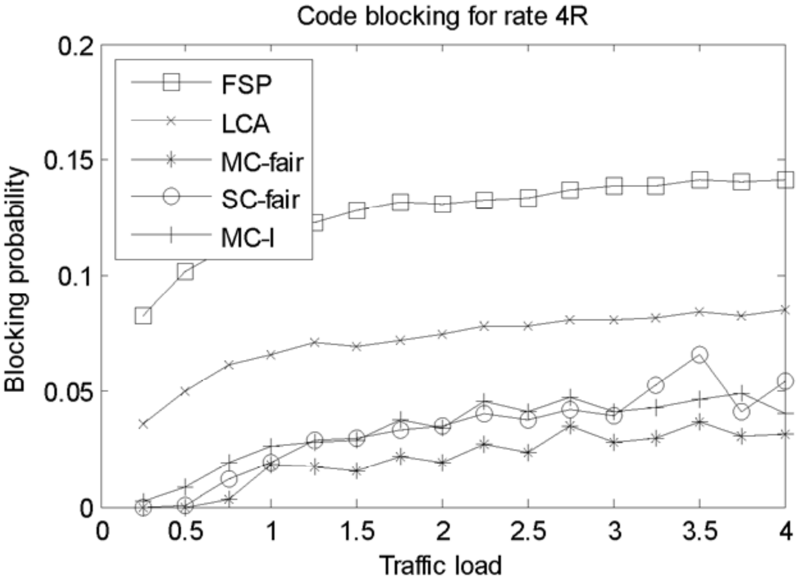


FSP- Fixed Set Partitioning, LCA- Left Code Assignment, MC-fair-Multi code fair(ours)
SC-fair- Single Code fair(ours), MC-1- Multi code scheme

**Figure 5.27:** Blocking probability vs traffic load for 2*R* rate users

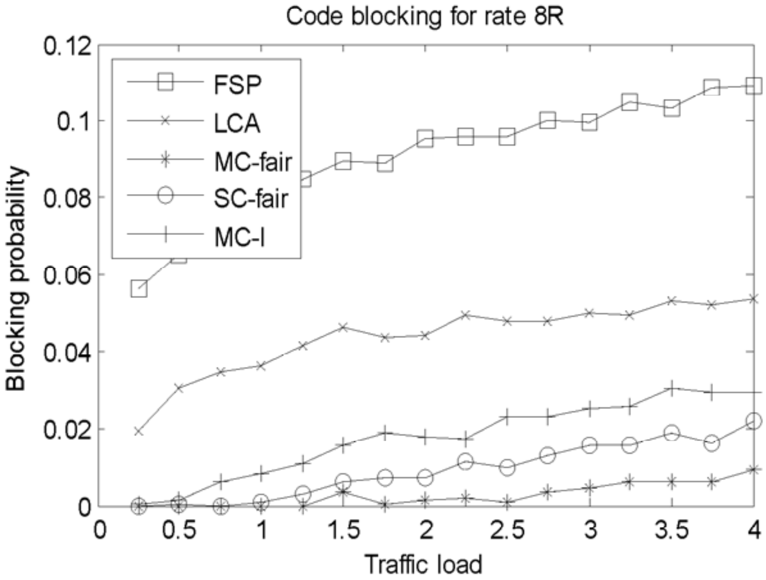
$$P_k = \frac{\rho_k^{G_k} / G_k!}{\sum_{n=1}^{G_k} \rho_k^n / n!} \tag{5.13}$$

The rate distribution in the system can be considered as $[p_1, p_2, p_3, p_4]$, where $p_i$ is the probability of arrival rate $2^{i-1}$R. For our simulation we assumed distribution equal to [0.25, 0.25, 0.25. 0.25]. We compare the code blocking of the proposed single code fair



FSP- Fixed Set Partitioning, LCA- Left Code Assignment, MC-fair-Multi code fair(ours)
SC-fair- Single Code fair(ours), MC-1- Multi code scheme

**Figure 5.28:** Blocking probability vs traffic load for 4*R* rate users



FSP- Fixed Set Partitioning, LCA- Left Code Assignment, MC-fair-Multi code fair(ours)
SC-fair- Single Code fair(ours), MC-1- Multi code scheme

**Figure 5.29:** Blocking probability vs traffic load for 8*R* rate users

(SC-fair), proposed multi code fair (MC-fair) scheme with the existing leftmost code assignment (LCA), fixed set partitioning (FSP) and multi code (MC-I) [17] schemes. The code blocking is compared for rate *R*, 2*R*, 4*R* and 8*R*. The simulation is performed for 2000 users and the result is the average of 10 simulations. The code locking comparison for various schemes is shown in Figure 5.26, 5.27, 5.28 and 5.29 respectively. If $B_x$ represents the code blocking in scheme x, the blocking in all the schemes is given by $B_{FSP}>B_{LCA}>B_{SC\text{-}fair}>B_{MC\text{-}I}>B_{MC\text{-}fair}$. Therefore fair single code scheme is superior to LCA and FSP with the added benefit of fairness. Also the fair multi code scheme is superior to its counterpart MC-I scheme discussed in [17].

## 5.5Hybrid code assignment

Consider an $L$ ($L$=8 in WCDMA) layer OVSF system. Let $C_{l,n}$ represent code in layer $l$ with branch number $n$, $0 \le n \le 2^{7-l}$. Let $\lambda_i$, $\mu_i$ is the arrival rate and service rate for the $i^{\text{th}}$

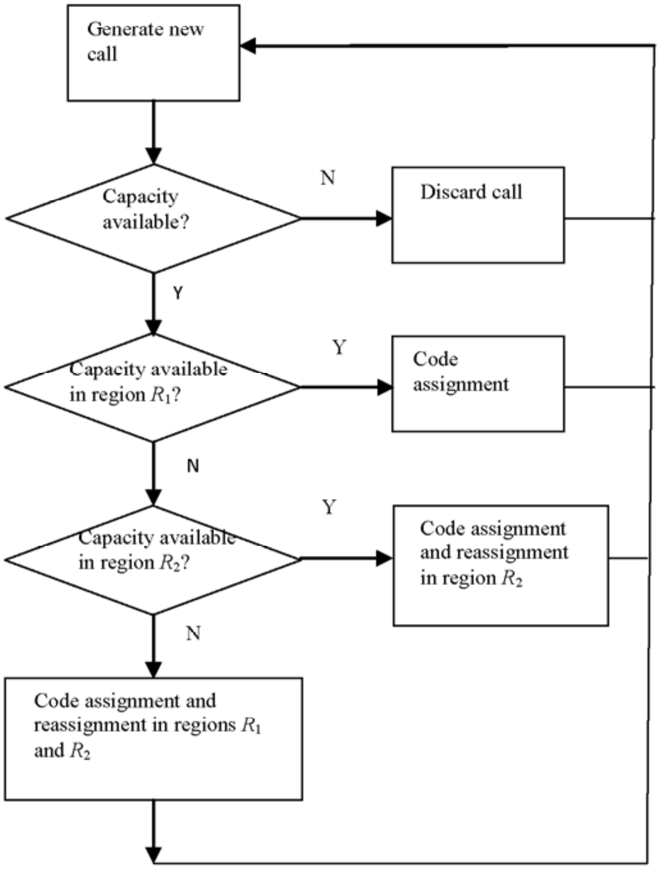


**Figure 5.30:** Flowchart for hybrid RDA and DCA design

user. Define $\rho_i = \lambda_i / \mu_i$ as traffic load for $i^{\text{th}}$ class user. In the proposed methods, we divide the full OVSF code tree in two regions. One region (region $R_1$) always provides codes according to traditional leftmost code assignment or crowded first assignment and the other region (region $R_2$) provides codes according to dynamic code assignment or multi code assignment. Region $R_1$ provides simplicity to the code assignment mechanism reducing decision time (number of code searched). Pure (LCA/CFA) usage may lead to code blocking which can be reduced by region $R_2$(DCA/MC) usage. The capacity in $R_1$ and $R_2$ region depends on the traffic conditions. Initially capacity of both regions is assumed as $2^{L-1}R$. The aim is to identify the optimum capacity of region $R_2$ such that cost and complexity (reassignments in DCA or number of codes in multi code) is least. If $C_{max}$ is the maximum capacity of OVSF code tree, the selection of capacity in region $R_2$ is

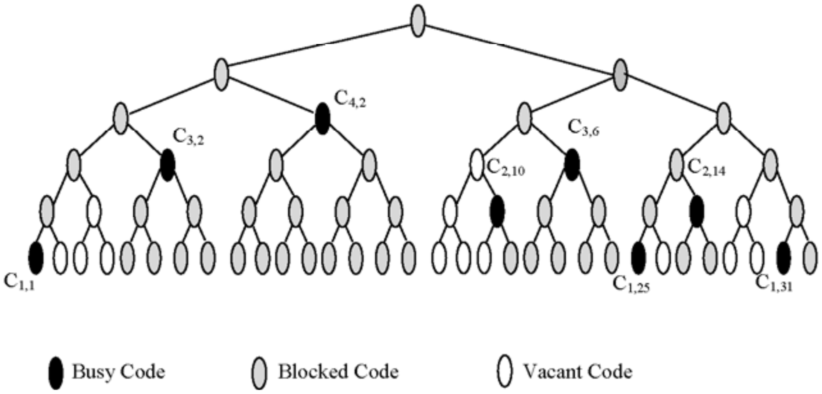


**Figure 5.31:** OVSF code tree with six layers

done as follows. For system with $n$ different user classes, if $\sum_{i=1}^{n} \rho_i \le 1$, i.e. traffic load is less than 1, only region $R_1$ is used. If $\sum_{i=1}^{n} \rho_i > 1$, the capacity of region $R_2$is $C_1 = C_{\max} \times (1 - 1 \big/ \sum_{i=1}^{n} \rho_i)$. Therefore the capacity of region $R_2$ is $C_{max}$-$C_1$. This is essential for a system with highly dynamic traffic conditions. The flowchart of the proposed design is shown in Figure 5.30.

### *5.5.1 Region 1: LCA/CFA and Region 2: DCA*

If a new call $2lR$ arrives, the vacant code in the $l^{th}$ layer is checked in region $R1$. If code is available, the call is handled and algorithm stops. If vacant code is not available in the region $R_1$and the remaining capacity of the regions is more then the capacity of incoming call, the code availability is checked in the $l^{th}$ layer of the DCA region (region $R2$). If vacant code is available, call is handled and algorithm stops. Otherwise, do code reassignment in region $R_2$ and generate the vacant code in layer $l$ using different cost functions used in [17]. If reassignments are not possible in region $R_2$, the region $R_1$ can also be utilized. This is generally the case when traffic is high where we can increase the $R_2$ region capacity to $kR$, where $k > 2^{L-1}$ and region $R_1$ capacity becomes $(2^L - k)R$. The design is illustrated with 6 layer OVSF code tree shown in Figure 5.31. The $32R$ code

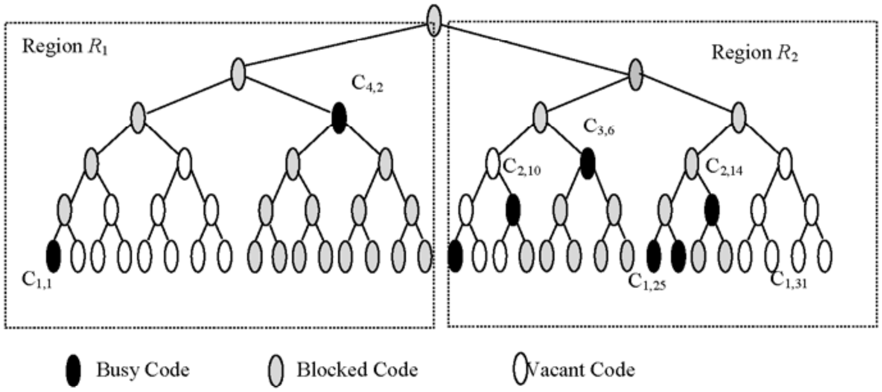


**Figure 5.32:** Usage of DCA in region $R_2$

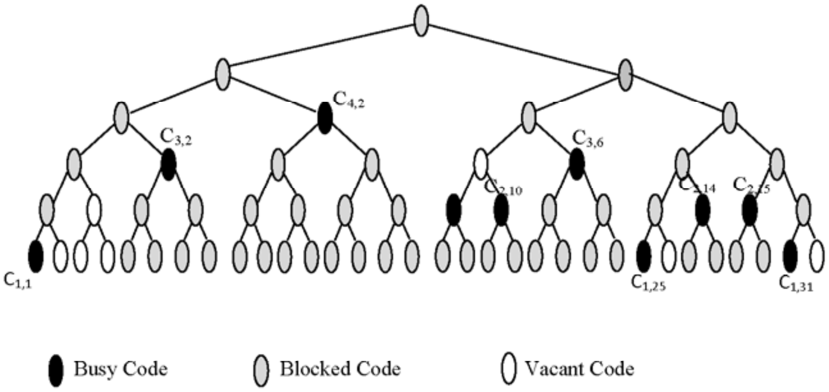


**Figure 5.33:** Usage of MC in Region $R_2$ (4R)

tree is divided into two regions with capacity 16*R* each. If a call of rate 4*R* arrives, first region $R_1$is checked. As we can see from the Figure 5.32, in region $R_1$ call of 4*R* cannot be handled. Then region $R_2$ is checked. This region also cannot handle 4*R* call because of blocking. So reassignment is done. We reassign $C_{1,31}$ to $C_{1,17}$ which is available. And thus $C_{3,8}$ can be assigned to handle 4*R* call as shown in Figure 5.33.

### *5.5.2.* ***Region 1: LCA/CFA and Region 2: MC***

As earlier, first go to region *R*1. Assume the system is equipped with *r* rakes. In this design if the vacant code is not available and the incoming call rate is within the maximum capacity of two region, the vacant code checked in the region *R*2 and the best multi code is used according to traditional multi code [17] schemes. If the vacant multi code is not available in region *R*2, region *R*1 can also be included for vacant multi code search. Here also the choice of capacities of two regions depends upon traffic conditions. To illustrate the design, consider the 6 layer tree shown in Figure 5.32. Let the system is equipped with 3 rakes. If a call of rate 4*R* arrives. The same procedure is repeated first by checking the RDA region. In this region, 4*R* call cannot be handled. Then in multi code region code availability is checked. In this region this call cannot be handles as 4*R*. So we break the 4*R* call into 2*R*+2*R*, which can be handled by $C_{2,9}$ and $C_{2,15}$ as shown in Figure 5.33. Now let us consider that a call of rate 6*R* arrives. As earlier, the region $R_1$ is checked first. But it cannot handle 6*R*, so multi code region is checked. We break the 6*R*

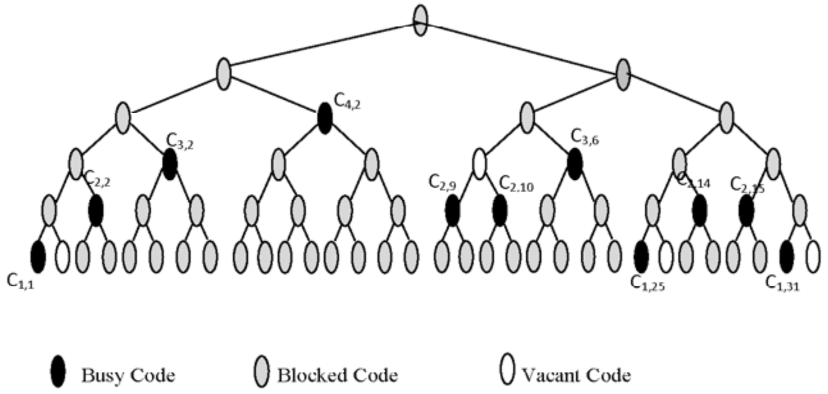


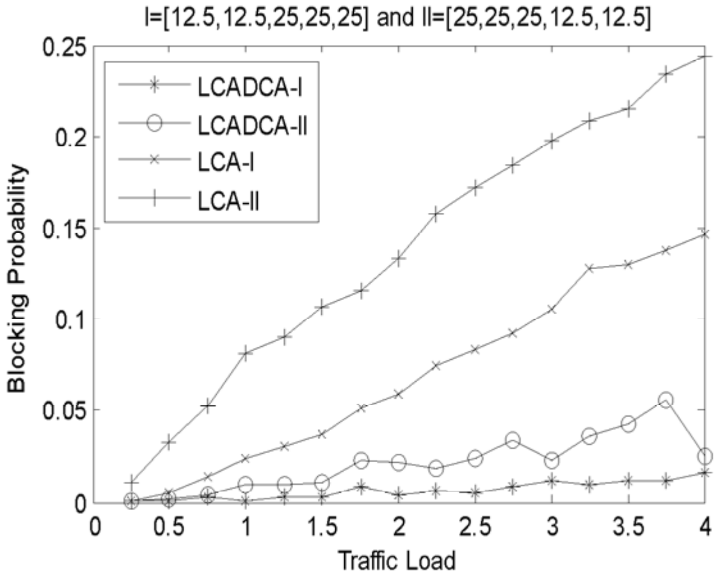
**Figure 5.34:** Usage of MC in Region $R_1$ and $R_2$ (6R)

call into 3 parts of $2R$ each and then combine regions $R_1$ and $R_2$ to handle this $6R$ call. Thus we assign the call to $C_{2,2}$, $C_{2,9}$ and $C_{2,15}$ and the code tree status is given in Figure 5.34.

### ***5.5.3 Simulation and Results***

*A. Input Data*

- Call arrival process is Poisson with mean arrival rate, $\lambda$ = 1-4 calls/ unit time.
- Call duration is exponentially distributed with a mean value, $1/\mu$ =3 units of time.
- Possible OVSF code rates are $R$, $2R$, $4R$, $8R$ and $16R$.
- The total capacity of the code tree is 128R which is divided into two regions, R1 and R2. Both regions are assumed to have capacity 64R. Region 1 is for LCA and region R2 is for DCA/MC.

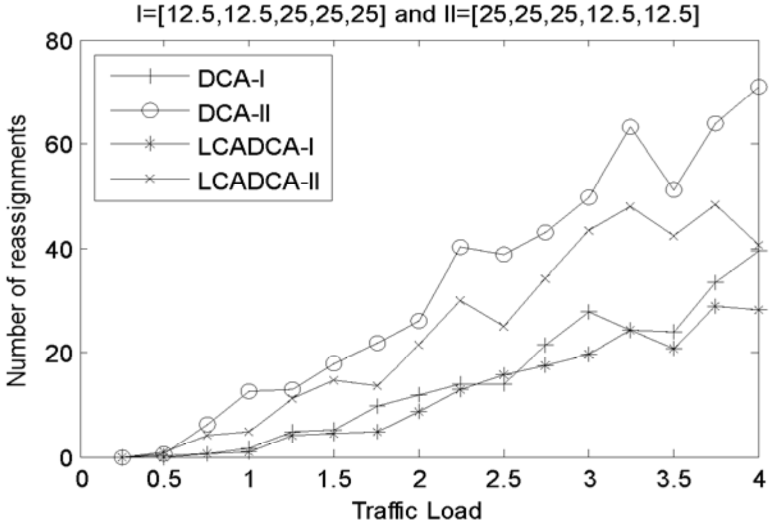


LCA-Left Code Assignment LCA-DCA-LCA plus DCA(ours)

**Figure 5.35:** Comparison of blocking probability for LCA and LCADCA

- The number of rakes used in multi rake scheme is 3.

- The simulation is performed for 1000 calls and the result is the average of 10 runs.



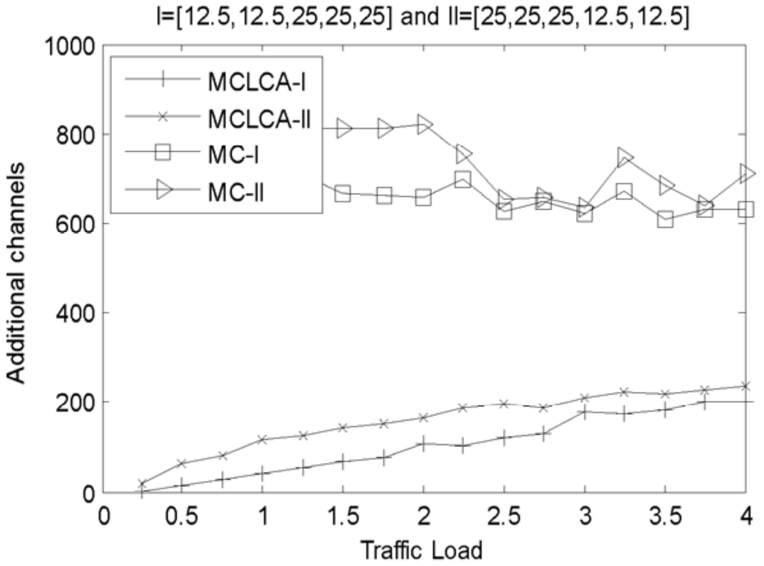
DCA-Dynamic Code Assignment LCADCA-LCA plus DCA(ours)

**Figure 5.36:** Comparison of number of code assignments vs traffic load for LCA and LCADCA

*B. Results*

As mentioned, we consider five different quantized rate arrival classes $\lambda \in \{\lambda_0, \lambda_1, \lambda_2, \lambda_3, \lambda_4\}\}$. The total number of codes in the system is, $G_0$+ $G_1$+ $G_2$+ $G_3$+ $G_4$ 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. Simulation was performed for leftmost code assignment in region $R_1$ and DCA/MC in region $R_2$. For CFA scheme in region $R1$, performance looks similar except code blocking performance is better. Therefore we compare the results only with LCA scheme. The performance parameters evaluated for LCA plus DCA combinations (denoted by LCADCA) are code blocking and number of reassignments. The complexity of the DCA scheme always depends upon the selection of minimum cost branch and number of reassignments. Figure 5.35 compares the traditional

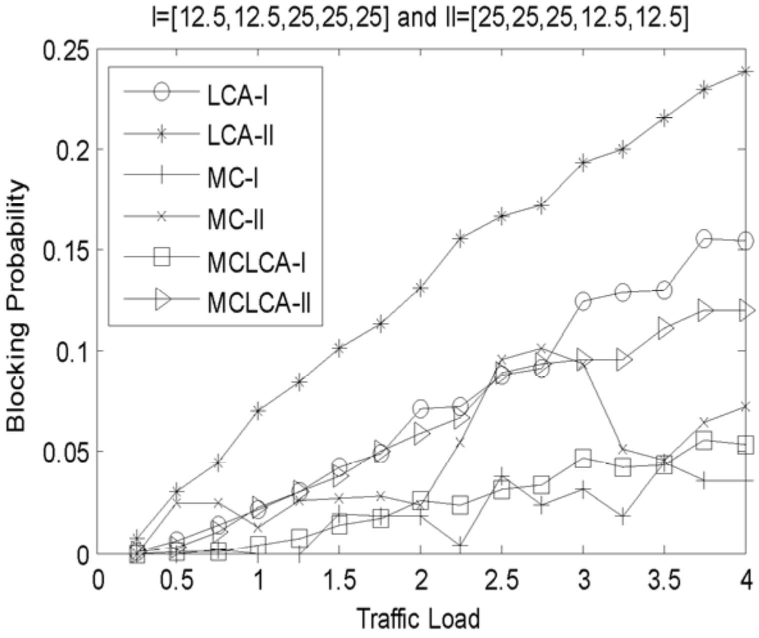
DCA with the proposed LCA plus DCA (region *R*1 is LCA and region *R*2 is DCA) for



MC-LCA-Multi code LCA(ours) MC-Multi code

**Figure 5.37:** Comparison of blocking probability for MC and MCLCA

two new call distributions (12.5,12.5,25,25,25) and (25,25,25,12.5,12.5), where ($p_1$,$p_2$,$p_3$,$p_4$,$p_5$) represents rate distribution for five classes (16*R*, 8*R*, 4*R*, 2*R*, *R*) of users. The code blocking in the proposed design is better than LCA and worse than pure DCA (which is zero and not plotted) but as shown in Figure 5.36, the reassignments in LCADCA scheme are significantly less than the traditional DCA which leads to lower complexity and cost in the proposed hybrid scheme. For hybrid LCA and MC scheme (denoted by MCLCA) the performance parameters considered are code blocking and additional number of channels (rakes) compared to the LCA scheme. Higher is number of the channels used for user (s), higher is the complexity. The LCA scheme is single code scheme and requires number of channels equal to total accepted calls. In MC scheme, the number of channels (rakes) used may differs as more than one channel can be used to handle calls. Pure MC scheme may use too much channels which may lead to unfair system. We compare the MC-LCA hybrid scheme with LCA and pure MC scheme for code blocking. The additional channels required are simulated for hybrid scheme and pure MC scheme. As for hybrid DCA, we compare the traditional MC scheme with the proposed LCA plus MC scheme (region $R1$ is LCA and region $R2$ is MC) for two different new call distributions (12.5,12.5,25,25,25) and (25,25,25,12.5,12.5) in Figure 5.37. The results shows that though the code blocking performance suffers due to hybrid scheme but the number of additional channels (plotted in Figure 5.38) required in the hybrid scheme is significantly less than MC scheme. The saving in number of channels is even more for system with more rakes.



MC-LCA-Multi code LCA(ours) LCA- Left Code Assignment

**Figure 5.38:** Comparison of additional channels required for LCA, MC and MCLCA

## 5.6 Conclusion

In this chapter, different code assignment and reassignment schemes are discussed so that the maximum calls are handled and the code tree is efficiently utilized. Single code and multicode fair designs are proposed for reduction in code blocking per class basis. Both the schemes are adaptive to the arrival rate distribution. The cost and complexity of the mentioned schemes can be more than schemes which do not consider fairness. Also two cost effective and comparatively less costly designs for WCDMA networks are proposed. The code blocking may increase in the proposed designs but the limitation is not too severe in low to medium traffic condition.

# CHAPTER 6

# CONCLUSION AND FUTURE SCOPE

OVSF codes are limited resources in 3G and beyond WCDMA wireless networks. They are used as channelization codes in the forward link of WCDMA. These codes are represented in form of tree where each code is orthogonal to another code if one is neither the parent nor the child of another. OVSF codes suffer from code blocking, due to which a new call cannot be handled even if the tree has enough capacity to handle that call. The main aim of all the schemes proposed in thesis is the efficient utilization of code tree so that maximum number of calls can be handled. This is done by choosing either a single code or multi code to handle the new call. Also type of call (data call or real time call) is another factor to decide how the call is to be handled. Scattered codes are utilized in the first approach, where the aim of the algorithm is to make the crowded area more crowded so that a certain portion of the tree remains free for the new calls. Another approach considered the reservation of codes by giving them priority number. The new call is handled by the codes having the highest priority number. Group leader assignment scheme divides the tree into leaves with group leaders.

In flexible assignment we have considered single hop and multi hop networks separately and accordingly divided the code tree in terms of available capacity and guaranteed capacity. Code index is calculated for each code and then the new code is assigned according to the value of code index. In this design, we have divided the code tree portion wise to handle the different types of calls, i.e., real time calls, data calls and mixed calls so that whenever a new call comes, the tree assigns a code from the portion to which it belongs.

In the penultimate chapter we have considered miscellaneous multicode schemes. Throughout the thesis, only two performance parameters have been considered, namely, code blocking and number of code searches. In future, work can be done to optimize the performance depending upon various other parameters like interference and frequency selective nature of the channel. Also the work can be extended in two dimensional OVSF codes where the second dimension appears due to OFDM usage.

# References

[1] E. Dahlman, B. Gudmunson, M. Nilsson, and J. Skold, “UMTS/IMT-2000 Based on Wideband CDMA,”*IEEE Communications Magazine*, vol. 36, no. 9, pp. 70-80, September 1998.

[2] E. Dahlman, P. Beming, J. Knutsson, F. Ovesjo, M. Persson, and C. Roobol, “WCDMA—The Radio Interface for Future Mobile Multimedia Communications,” *IEEE Transactions on Vehicular Technology*, vol. 47, no. 4, pp. 1105-1118, November 1998.

[3] F. Adachi, M. Sawahashi, and H.Suda, “Wideband DS-CDMA for next generation mobile communication Systems,” *IEEE Communication Magazine*, vol. 36, no. 9 pp. 56-69, September 1998.

[4] T. Ojanperä and R. Prasad, “An Overview of Air Interface Multiple Access for IMT-2000/UMTS,” *IEEE Communications Magazine*, vol. 36, no. 9, pp. 82-95, September 1998.

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

[6] C. M. Chao, Y. C. Tseng and L. C. Wang, “Reducing Internal and External Fragmentation of OVSF Codes in WCDMA Systems with Multiple Codes,” *IEEE Transactions on Wireless Communication*, vol. 4, no. 4, pp. 1516-1526, July 2005.

[7] C. M Chao, Y. C Tseng, L. C Wang, “Reducing internal and external fragmentations of OVSF codes in WCD MA systems with multiple codes,” *Proceedings of IEEE WCNC2003* vol. 1, pp.693 - 698 , March 2003.

[8] Y. C. Tseng, C. M. Chao, and S. L. Wu, “Code placement and replacement strategies for wideband CDMA OVSF code tree management,” *Proceedings of Global Telecommunication Conference, 2001, GLOBECOM '01*, vol. 1, pp.562-566, 2001.

[9] J. S. Park and D. C. Lee, “Enhanced fixed and dynamic code assignment policies for OVSF-CDMA systems,” *Proceedings of ICWN 2003*, Las Vegas, pp. 620-625, June 2003.

[10] A. Richardson, “WCDMA Design Handbook” Cambridge University Press, 2005.

[11] "*Radio Interface Protocol Architecture*" 3rd Generation Technical Specification 25.301 (Release 1999).

[12] T. Minn and K. Y. Siu, "Dynamic Assignment of orthogonal variable spreading factor codes in W-CDMA,"*IEEE Journal Selected Areas in Communication*, vol. 18, no.8, pp. 1429-1440, Aug. 2000.

[13] C. E. Fossa and N. J. Davis, "Dynamic code assignment improves channel utilization for bursty traffic in third-generation wireless networks," *Proceedings of IEEE International Conference Communications, ICC'02*, vol. 5, pp. 3061–3065, 2002.

[14] J. Xu, X. Shen, and J. W. Mark, "Dynamic Bandwidth Allocation with fair scheduling for WCDMA systems," *IEEE Wireless Communications*, vol. 9, no. 2, pp. 26-32, April 2002.

[15] D. N. Skoutas, A.N. Rouskas, "A Dynamic Traffic Scheduling Algorithm for the downlink shared channel in 3G WCDMA," *Proceedings of IEEE International Conference on Communications*, vol. 5, pp. 2975 - 2979, 20-24 June 2004.

[16]Y. C. Tseng and C. M. Chao, "Code placement and replacement strategies for wideband CDMA OVSF code tree management," *IEEE Transactions on Mobile Computing* vol. 1, no.4, pp. 293-302, October/December 2002.

[17] Y. Yang, T. S. P. Yum, "Maximally flexible assignment of orthogonal variable spreading factor codes for multirate traffic" *IEEE Transactions on Wireless Communication* vol. 3, no.3 pp. 781-792, May 2004.

[18] S. T. Cheng and M. T. Hsieh, "Design and analysis of time-based code allocation schemes in W-CDMA systems," *IEEE Transactions on Mobile Computing*, vol. 4, no. 6, 604-615, November-December 2005.

[19] Y. Yang and T. S. P. Yum, "Multicode multirate compact assignment of OVSF codes for QoS differentiated terminals," *IEEE Transactions on Vehicular Technology*, vol. 54, no. 6, pp. 2114-2124, November 2005

[20] J. S. Park, L. Huang, and C. J. Kuo, "Computationally efficient dynamic code assignment schemes with call admission control (DCA-CAC) for OVSF-CDMA systems," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 1, pp.286-296, January 2008.

[21] A. N. Rouskas and D. N. Skoutas, “Management of channelization codes at the forward link of WCDMA” *IEEE Communications Letters*, vol. 9, no. 8, pp. 679-681, August 2005.

[22] C. S. Wan, W. K. Shih and R. C. Chang, “Fast dynamic code assignment in next generation wireless access networks,” *Elsevier Journal Computer Communications* vol. 26, no. 14, pp.1634-1643, September 2003.

[23] A. C. Kam, T. Minn, K. Y. Siu, “Supporting rate guarantee and fair access for bursty data traffic in WCDMA,” *IEEE Journal on Selected Areas in Communications*, vol.19, no. 11, pp. 2121-2130. November 2001.

[24] A.N. Rouskas and D.N. Skoutas, “OVSF code assignment and reassignment at the forward link of W-CDMA 3G systems,” *Proceedings of IEEE Personal Indoor and Mobile Radio Communications, PIMRC*, vol. 5, pp. 2404-2408, 2002.

[25] Y. R Tsai, L. C. Lin, “Quality-Based OVSF code assignment and reassignment strategies for WCDMA systems,” *IEEE Transactions on Vehicular Technology*, vol. 58, no.2, pp. 1027-1031, February 2009.

[26] A.C. Kam, T. Minn, K.Y. Siu, “Reconstruction methods of tree structure of orthogonal spreading codes for DS-CDMA”, *IEICE Transactions on Fundamentals of Electronics, Communication and Computer Sciences*, pp. 2078-2084, November 2000.

[27] A. N. Rouskas and D.N. Skoutas, “Comparison of code reservation schemes at the forward link in WCDMA,” *Proceedings of 4th IEEE International Workshop on Mobile and Wireless Communications Networks 2002*, Greece, pp. 191–195, September 2002.

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

[29] L. H. Yen and M. C. Tsou, “An OVSF code assignment scheme utilizing multiple rake combiners for W-CDMA,” *Proceedings of IEEE International Conference Communications ICC’03*, vol. 5, no. 5, Anchorage, AK, pp. 3312–3316, May 2003.

[30] S. Malik and D. Zeghlache, “Improving Throughput and Fairness on the downlink shared Channel in UMTS WCDMA Networks,” *European Wireless 2002*, Florence, Italy, February 2002.

[31] H. Qam and N. Challa, “Supporting rate and soft delay guarantees for data traffic in the downlink shared channel of WCDMA,” *Proceedings of International Conference on Wireless Networks (ICWN) 2002*, Las Vegas, USA, June 2002.

[32] M. D. Amico, M. L. Merani, and F. Maffioli, “*Efficient Algorithms for the Assignment of OVSF Codes in Wideband CDMA*,” *Proceedings of IEEE ICC02*, vol. 5, pp. 3055 - 3060, April- May 2002.

[33] H.W. Ferng, H. L. Chin, D. Shiung and Y.T. Chen, “An OVSF code tree partition policy for WCDMA systems based on the multi-code approach,” *Proceedings of VTC 2005*, vol. 2, pp.1212 – 1216, 25-28 September 2005.

[34] J. S. Park and D. C. Lee, “On static and dynamic code assignment policies in the OVSF code tree for CDMA networks,” *Proceedings of IEEE Military Communications Conference (MILCOM’02)*, vol. 2, Anaheim, CA, pp. 785–789, October 2002.

[35] C.M. Chao, Y.C. Tseng, L.C Wang, “Dynamic bandwidth allocation for multimedia traffic with rate guarantee and fair access in WCDMA Systems,”*IEEE Transactions on Mobile Computing*, vol.4, no.5, pp.420-429, September/October 2005.

[36] J. Xu, *et al*., “Efficient channel utilization for real-time video in OVSF CDMA systems with QoS assurance,” *IEEE Transactions on Wireless Communication*, vol. 5, no. 6, pp. 1382-1391, June 2006.

[37] S. Guo, J. Peng and J. Wu, “A reassignment code allocation scheme for multirate services on OVSF CDMA cellular networks,” *International Conference on Wireless Communications, Networking and Mobile Computing, WiCom 2007*, pp. 965-969, September 21-25, 2007.

[38] C. Perez, *et al.*, “Call admission and code allocation strategies for WCDMA systems with multirate traffic,” *IEEE Journal on Selected Areas in Communications*, vol.24, no. 1, pp.26-35, January 2006.

[39] R. Assarut, K. Kawanishi, U. Yamamoto, Y. Onozato, and M. Matsushita, “Region division assignment of orthogonal variable spreading-factor codes in W-CDMA,” *Proceedings of IEEE Vehicular Technology Conference, 2001*, vol. 3, Atlantic City, NJ, pp. 1884-1888, Fall 2001.

[40] Y. S. Chen, T. L. Lin, “Code Placement and Replacement Schemes for WCDMA Rotated-OVSF Code Tree Management,” *IEEE Transactions on Mobile Computing*, vol. 05, no. 3, pp. 224-239, March 2006.

[41] Y. S Chen, H. C. Chang, “Multi-code placement and replacement schemes for W-CDMA rotated-OVSF code tree,” *IEEE 6th Circuits and Systems Symposium* vol. 1, pp.345 - 348, 2004.

[42]H. Çam, “Nonblocking OVSF Codes for 3G Wireless and Beyond Systems,” *Proceedings of 3G Wireless'2002 & WAS'2002. International Conference on Third Generation Wireless and Beyond*, May 28-31, San Francisco, CA, USA, pp. 148-153, 2002.

[43] H. Çam, “Nonblocking OVSF Codes and Enhancing Network Capacity for 3G Wireless and beyond Systems,” *Special Issue of Computer Communications on 3G Wireless and Beyond for Computer Communications*, vol. 26, no. 17, pp. 1907-1917, November 2003.

[44] H. Cam, “Non-blocking OVSF codes and enhancing network capacity for 3G wireless and beyond system,” *Elsevier Journal of Computer Communication*, vol. 26, no. 17, pp. 1907-1917, November 2003.

[45] Y. Sekine, *et al.*, “Hybrid OVSF code assignment scheme in W-CDMA,” *Proceedings of IEEE Pacific Rim Conference Communications, Computers and Signal Processing (PACRIM’03)*, vol. 1, Victoria, BC, Canada, pp. 384–387, August 2003.

[46] J. S. Park, L. Huang, D. C. Lee, and C.C. J. Kuo, “Optimal code assignment and call admission control for OVSF-CDMA systems constrained by blocking probabilities,” *IEEE GLOBECOM 2004*, Dallas, vol. 5, pp. 3290-3294 December 2004.

[47] Y. Yang and T. S. P. Yum, “Non rearrangeable compact assignment of orthogonal variable-spreading-factor codes for multi-rate traffic,” *IEEE Vehicular Technology Conference (VTC)*, *Atlantic City, NJ*, vol. 2, pp. 938–942, 2001.

[48] W. T. Chen, H. C. Hsiao, and Y. P. Wu, “A novel code assignment scheme for W-CDMA systems,” *Proceedings of IEEE Vehicular Technology Conference*, vol. 2, no. 2, pp.1182-1186, 2001.

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

[50] L. L. Yang, “Performance of multiantenna multicarrier direct sequence direct-sequence code division multiple access using orthogonal variable spreading factor codes- assisted space-time spreading in time selective fading channels,” *IET Communications*, vol. 2, no. 5, pp.708-719, May 2008.

[51] M. E. Gonzalez, “Dynamic code allocation scheme for multirate services on OVSF CDMA networks based on intracell interference reduction,” *IEEE 64th Vehicular Technology Conference, 2006, VTC-Fall, 2006*, pp. 1-6.

[52] S. M. A. El-atty, “Efficient connection admission control based-code searching algorithm in 3G/mesh networks,” *IEEE GCC Conference and Exhibition*, 2011, pp. 9-12, February 2011.

[53] S. Rahimi, “Selective technique in code assignment strategies for wideband CDMA OVSF code tree management,” *IEEE Annual Conference on Wireless and Microwave Technology, WAMICON ’06*, pp. 1-4, 4-5 Dec 2006.

[54] M. K. Karakayali, “Downlink Throughput maximization in CDMA wireless networks,” *IEEE Transactions on Wireless Communications*, vol. 5, no.12, pp.3492-3500, December 2006.

[55] C. C. Yeh, P. L. Wang, “Dynamic bandwidth allocation and scheduling for OVSF-WCDMA systems,” *International Conference on Communications and Mobile Computing (CMC) 2010*, vol. 3, pp. 475-479.

[56] P.Lin, C.H. Gan and C.C. Hsu, “OVSF code channel assignment with dynamic code set and buffering adjustment for UMTS,” *IEEE Transactions on Vehicular Technology*, vol. 54, no. 2, pp. 591-602, March 2005.

[57] L. C. Ping *et. al*, “A multiple-rate, multicarrier direct-sequence CDMA system with 2-D OVSF codes in fading channels,” *IEEE Transactions on Wireless Communications*, vol. 6, no. 6, pp. 2162-2171, June 2007.

[58] D. I. Lyu, “Spreading code selection for multiple chip rate DS-CDMA systems,” *IEEE Transactions on Vehicular Technology*, vol. 54, no. 3, pp. 1215-1218, May 2005.

[59] M. A. S. Al-Adwany and M.B. Ayoub, “Performance evaluation of non blocking OVSF codes in WCDMA systems,” *IEEE Middle East Conference on Antennas and Propagation (MECAP)*, 2010, pp. 1-6, October 20-22, 2010.

[60] J. D. Chimeh, S.M. Mousavinejad and A. Mahmoodi, "An algorithm for restraining the OVSF codes shortage in HSDPA," *International Conference on New Technologies, Mobility and Security (NTMS)*, 2011, pp. 1-4, February 7-10, 2011.

[61] H. Yu *et. al*., "A new multipath interference mitigation technique for high-speed packet transmission in WCDMA downlink," *IEEE Signal Processing Letters*, vol. 12, no. 9, pp. 601-604, September 2005.

[62] K. J. Lin, Y. C. Tseng, "Adaptive selection combining for soft handover in OVSF W-CDMA systems", *IEEE Communication Letters*, vol. 8, no. 11, pp. 656-658, November 2004.

[63] I. Mortoyo, *et. al*, "Chaos codes vs. orthogonal codes for CDMA," IEEE 11th International Symposium on Spread Spectrum Techniques and Applications (ISITA), 2010, pp. 189-193, 17-20 October 2010.

[64] S. Kasapovic, *et.al*, "Comparison ways of assignment of codes in UMTS networks", *Proceedings of the 35th International Convention, MIPRO 2012*, pp. 552-556, 21-25 May 2012.

[65] D. Kumar, and C. Chellappan, "System Modeling and Analysis for an Efficient Resource Allocation in MC-CDMA Cellular Wireless Networks to Support Internet Traffic", *Journal of Internet Technology*, vol. 11, no. 3, pp. 427-436, 2010.

[66] W. Ouyang, C. W. Yu, M. Liu, and Y. Chang, "Graph model for optimal OVSF code placement strategies", *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 9, no. 3 pp. 133 – 141, 2012.

[67] Davinder S.Saini and S.V.Bhooshan, "Adaptive assignment scheme for OVSF codes in WCDMA**",** *Proceedings of IEEE ICWMC'06*, Bucharest, July 2006.

# Author's Publications

## Journals

1. Davinder S Saini and Neeru Sharma , "Reduction in code blocking using scattered vacant codes for OVSF based WCDMA networks", *IET Communications*, vol. 7, no. 1, pp. 40-48, January 2013.

2. Davinder S Saini and Neeru Sharma, "OVSF Code Assignment Schemes At the Forward Link of WCDMA Systems, *Actapress International Journal of Computers and Applications, 2010,* vol. 32, no.3, ,pp. 253-260.

3. Neeru Sharma and Davinder Saini, "Code scattering and reduction in OVSF code blocking for 3G and beyond mobile communication systems", *WSEAS Transactions on Communications*, vol. 11, no. 2, pp. 91-101, February 2012.

4. Davinder S Saini and Neeru Sharma, "Performance Improvement In OVSF Based CDMA Networks Using Flexible Assignment of Data Calls", *International Journal of Computer Networks and Communications (IJCNC )*vol.3, no.6, pp. 197-212, November 2011.

## Conferences

1. Davinder S Saini and Neeru Sharma, "An Efficient Multi Code Design for Code Blocking Reduction in 3G Wireless Networks*"*, *in proc. of IEEE Sarnoff*, pp.1-5, April 12-14, NJ, USA, 2010. (IEEE explore) Digital Object Identifier: 10.1109/SARNOF.2010.5469756

2. Davinder S Saini, Neeru Sharma "Fair Single Code and Multi Code Assignment Schemes for OVSF Based Networks*"*, *IEEE Latincom 2009*, pp.1-5, Sept. 2009, Columbia (IEEE explore).

3. Neeru Sharma, Davinder S Saini,"Hybrid Single Code and Multi CodeDesign to Reduce Complexity in OVSF Codes Based Wireless Networks*"*, *IEEE Latincom* 2009, pp. 1-5, Sept. 2009, Columbia. (IEEE explore).