Wireless Pers Commun (2014) 75:465–482
DOI 10.1007/s11277-013-1372-9

Top Down Code Search to Locate An Optimum Code and Reduction in Code Blocking for CDMA Networks

Davinder S. Saini · Vipin Balyan

Published online: 22 September 2013

Abstract In this paper, a top down code search scheme is proposed that identify an optimum OVSF code for assignment at the base station of CDMA wireless networks. An optimum vacant code is the one whose usage produces least code blocking compared to other eligible codes. This scheme provides least code blocking compared to existing schemes without reassignments. In addition, the codes searched during locating the optimum code are significantly less than other existing schemes. The call establishment delay which is a significant factor for real time applications is directly proportional to the number of searches and should be low. The design is explained for single code, and extended to multi code assignment to improve code blocking. The multi code assignment is done using four ways. The first and second multi code schemes uses minimum and maximum rakes for a fixed rate system. The third scheme called scattered multi code scheme divide the incoming call into rate fractions equal to number of rakes available in the system, and each rate fraction is handled in a similar way in which the new call is handled in single code scheme. The rate fractions may be scattered or grouped in the code tree. The fourth multi code scheme, namely grouped multi code scheme allocates codes to all the fractions as close as possible. This maximizes future higher rate vacant codes availability by leaving a complete sub tree vacant when call using multi code ends.

Keywords OVSF code · Code blocking · Spreading factor (SF) · Code assignment and reassignment

D. S. Saini
Department of Electronics and Communication Engineering, Jaypee University of Information Technology, Waknaghat, Solan 173234, HP, India
e-mail: davinder.saini@juit.ac.in

V. Balyan (✉)
Department of Electronics and Communication Engineering, Jaypee Institute of Information Technology, Noida 201001, India
e-mail: vipin.balyan@rediffmail.com

1 Introduction

CDMA based wireless networks are superior to its older counterparts in handling multimedia rates. These rates are differentiated by QoS parameters [1,2] like delay, jitter, bandwidth and reliability. For real time applications like voice, and video conferencing, the call establishment delay is a significant QoS parameter and must be small. The amount of codes searched prior to call assignment decides the call establishment delay in OVSF based networks. The time required for single code search depends usually upon parameters like, frequency of operation, processor clock, type of traffic etc. Orthogonal variable spreading factor (OVSF) codes in 3G and beyond systems differentiate users by providing spreading factors (SFs) inversely proportional to their rates. OVSF codes are the limited resources in the downlink shared channel. The relationship between user rates, SF and channel transmission rates is given in Table 1 for WCDMA systems. For WCDMA, the SF varies from 4 to 512 in the downlink, and for mathematical simplicity, SF is normalized from 1 to 128 in this paper. The OVSF codes are generated using a binary tree structure [3]. Two significant constraints of OVSF codes for handling real time calls are *code blocking* and *number of codes searched* prior to assignment. When a code is assigned, its ancestors and descendants are blocked from assignment. Due to random arrival and departure time of calls, the used codes are scattered in the code tree causing reduction in spectral efficiency.

The code blocking [4] is illustrated in Fig. 1, where a 7 layer code tree is considered. A code C_{l,n_l}, $1 \leq l \leq L$ represents a code in layer l with branch number n_l, codes can be vacant, busy or blocked as shown in Fig. 1. The maximum capacity of the code tree is $64R$, (R is 7.5 kbps for WCDMA downlink) out of which $33R$ capacity is in use (due to 9 assigned codes in layer 1, 4 assigned codes in layer 2, 2 assigned codes in layer 3 and 1 assigned code in layer 4 with capacity $R, 2R, 4R$ and $8R$ respectively). The remaining capacity $31R$ is

Table 1 Relationship between user rate, SF and transmission rate for WCDMA Downlink

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

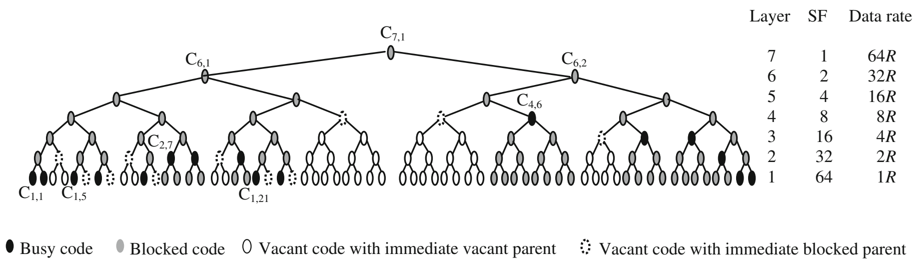

Fig. 1 A seven layer OVSF code tree

scattered along the code tree, and if a new call of rate $16R$ arrives, it will be blocked although the tree has $33R$ free capacity. Further, the new call should use the vacant code from location that produces minimum code blocking.

Many good schemes like crowded first assignment (CFA) [5], recursive fewer codes blocked (RFCB) [6] are proposed to reduce code blocking but they are not suitable for real time calls because the amount of codes searched are high. The code searches comparison for few single code schemes is given in Saini et al. [7]. The proposed scheme is optimum in the sense that it provides least blocking probability with significantly less code searches compared to popular CFA and RFCB schemes. Also, the top down (TD) scheme can be combined with multi code [8] and dynamic code assignment [9] for performance improvement. The DCA is always best as it provides zero code blocking but large reassignment overhead make it impractical for bandwidth constraint systems like 3G and beyond communication systems.

The OVSF code assignment problem is investigated deeply in last few years, and salient features of few noble schemes are given below. In left code assignment (LCA) [5], the code assignment is done from the left of the tree. In fixed set partitioning (FSP) [10], a code tree is divided into sub trees according to the traffic distribution. Both LCA and FSP schemes suffer from higher code blocking, and are used commonly for low traffic load. The adaptive code assignment (ADA) [11] makes the code tree division adaptive to arrival call distribution. In CFA, the new call is assigned to the most crowded portion in such a way that some part of the tree remains free for future high rate calls. CFA has two categories namely crowded first space and crowded first code. The performance evaluation parameter in crowded first space is capacity, and in crowded first code, it is number of codes. Both these schemes suffer from large code searches. The DCA scheme improves the spectral efficiency with the help of code reassignments. The code blocking can be made as small as zero but the cost and complexity factor increases due to reassignments. The fast dynamic code assignment (FDCA) [12] reduces the number of code reassignments without causing degradation in the spectral efficiency of system. The computationally efficient dynamic code assignment with call admission control (DCA-CAC) [13] reduces complexity of traditional DCA [8] in two different ways: (a) total resources are divided into a number of mutually exclusive groups, with numbers of groups equal to number of call arrival classes; (b) by deliberate rejection of those calls which may produce large code blocking for future higher rate calls. The fewer codes blocked (FCB) scheme in Rouskas and Skoutas [14] select that vacant code which blocks least new parents who were free earlier. The RFCB design [6] work on top of FCB. It resolves tie by ordered selection among candidate codes. The call elapsed time [15] approach assigns new call to a vacant code whose neighbor is busy for longest duration of time.

While all above mentioned schemes uses single code for incoming calls, the multi code assignment schemes use multiple codes to reduce rate wastage and code fragmentation. The multi code scheme in Saini and Upadhyay [16] finds the most suitable multi code combination. The multirate multicode compact assignment (MMCA) [17] scheme uses the concept of *compact index* to accommodate QoS differentiated mobile terminals. It does not perform code rearrangement and supports mobile terminals with different multi code transmission capabilities. The MMCA supports multi rate real time calls and keeps the code tree as flexible as possible in accepting new multi rate calls. The multi code scheme [18] identifies the optimal code with constraints of allocated code amount and maximal resource wastage ratio. The time based code assignment in Cheng and Hsieh [19] explains the impact of remaining time in reducing code blocking. The calls with similar remaining time are allocated to the same sub tree. The non blocking OVSF (NOVSF) codes given in Vadde and Qam [20] minimize the code blocking to zero. The code usage time is converted into multiple time slots and a single

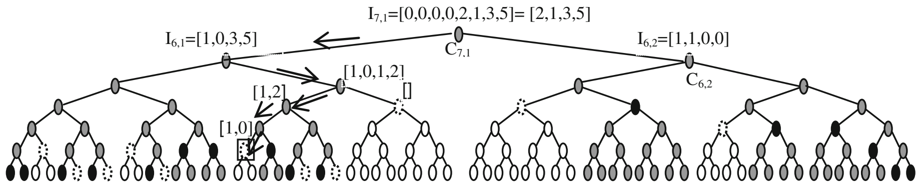

Fig. 2 Illustration of single code top down scheme

layer is sufficient to handle calls with variable rate requirements. The cost and complexity of the NOVSF codes is high. The rotated OVSF codes proposed in Chen et al. [21] reduces the code blocking significantly compared to other methods.

In this paper, for every new call, the optimum code search starts from the root of the code tree. The maximum number of searches is significantly reduced compared to most of the existing schemes. The algorithm does not depend upon the density of busy codes in the code tree, which increase its utility for heavy traffic load conditions. The single code scheme is extended to multi code with four variants. The first two methods have the flexibility to use minimum or maximum rakes. In the scattered scheme, new call is divided into many rate fractions, and each fraction is assigned optimum code similar to like single code scheme. While the individual rate fractions assignment is independent in scattered scheme, the assignment is dependent to each other in grouped scheme. The emphasis is given to assign codes as close as possible, and hence making their completion times identical.

The rest of the paper is organized as follows. Section 2 discusses the proposed single code TD scheme along with its multi code and dynamic code assignment extensions. Simulation environment and results are given in Sect. 3, and finally, the paper is concluded in Sect. 4. Also, the number of code searches required before assignment are derived for various schemes and placed in “Appendix”.

2 Top Down Scheme

2.1 Pure Top Down Scheme

Consider an L layers OVSF code tree with code C_{l,n_l} representing a code in layer l with branch number n_l, where, $1 \leq l \leq L$ and $1 \leq n_l \leq 2^{L-l}$. The total codes in a layer l are 2^{l-1}. For a new rate $2^{l-1}R$ call, the top down scheme identifies the optimum code in layer l, say $C_{l,n_{l\,opt}}$ whose assignment leads to least code blocking. For a code C_{l,n_l}, define code index I^l_{l,n_l} is given by

$$I_{l,n_l} = \left[I^{l-1}_{l,n_l}, I^{l-2}_{l,n_l} \ldots I^{1}_{l,n_l} \right] \tag{1}$$

In (1), I^i_{l,n_l} is the number of vacant codes in layer i below C_{l,n_l} whose immediate parents are blocked. For example in Fig. 2, for a code $C_{6,1}$, the value of $I^2_{6,1}$ is 3 as there are three vacant codes available under code $C_{6,1}$, namely, $C_{2,2}$, $C_{2,5}$, and $C_{2,9}$ respectively.

For a code C_{l,n_l} with index $I_{l,n_l} = \left[I^{l-1}_{l,n_l}, I^{l-2}_{l,n_l} \ldots I^{1}_{l,n_l} \right]$, the total vacant children under C_{l,n_l} are

$$N_{l,n_l} = \sum_{l'=1}^{l-1} I^{l'}_{l,n_l} \tag{2}$$

Also, the capacity available in the vacant children of code C_{l,n_l} is given by

$$P_{l,n_l} = \sum_{i=1}^{l-1} I_{l,n_l}^{i} \times 2^{i-1} R \tag{3}$$

The code index I_{l,n_l} is updated periodically and in addition, at the arrival and completion of new call. For a code C_{l,n_l}, the sibling is C_{l,n_l+1}, if n_l is odd, and C_{l,n_l-1}, if n_l is even. Let the sibling of code C_{l,n_l} is represented by C_{l,n_l^s}. At the arrival of a rate $2^{l-1}R$ call, the optimum vacant code in layer l is required. If we represent sibling of $C_{l_1,n_{l_1}}, l \leq l_1 \leq (L-1)$ by $C_{l_1,n_{l_2}}$, the code $C_{l_1,n_{l_1}}$ is in the path from root to optimum code (say C_{l,n_l-opt}) if, (a) $\sum_{i=1}^{l_1-1} I_{l_1,n_{l_1}}^{i} \leq \sum_{i=1}^{l_1-1} I_{l_1,n_{l_2}}^{i}$, and (b) $I_{l_1,n_{l_1}}^{l}$ is non zero, or $I_{l_1,n_{l_1}}^{l}$ is zero and $I_{l_1,n_{l_1}}^{l_1'} \neq 0, l \leq l_1' \leq l_1$. If both (a) and (b) fails and $I_{l_1,n_{l_2}}^{l_1'}$ is non zero for $l \leq l_1' \leq l_1$, the code $C_{l_1,n_{l_2}}$ will be in the path from root to the optimum code. The procedure is repeated upto layer l, if at least one vacant code of layer l available under all the optimum codes. Once all the codes $C_{l_1,n_{l_{1opt}}}, l \leq l_1 \leq (L-1)$ are identified, the optimum code in layer l is assigned to the call. The indices for these identified optimum codes, $C_{l_1,n_{l_{1opt}}}, l \leq l_1 \leq (L-1)$ are updated as follows, (i) If $I_{l,n_{l_{opt}}}^{l}$ is non zero, $i.e.$, the optimum vacant code exists in layer l (say$C_{l,n_{l_{opt}}}$) with all its parents blocked. This vacant code is assigned to the new call and the coefficients $I_{l_1,n_{l_{1opt}}}^{l}, l \leq l_1 \leq (L-1)$ are decremented by one, (ii) If $I_{l_1,n_{l_1-opt}}^{l}$ is zero (there is no vacant code in layer l directly) but $I_{l_1,n_{l_1-opt}}^{l_1'}, l+1 \leq l_1' \leq L-1$ is non zero, the optimum code in layer l is the leftmost child of $C_{l_1,n_{l_{1opt}}}$ in layer l, $i.e.C_{l,2^{l_1-l}\times(n_{l_1}-1)+1}$ code (say$C_{l,n_{l_{opt}}}$). The indices incremented are, (i) $I_{l_1',\lceil n_{l_1}-opt/2^{l_1'-l_1}\rceil}^{l_1-1}$ to $I_{l_1',\lceil n_{l_1}-opt/2^{l_1'-l_1}\rceil}^{l}, l \leq l_1' \leq L$ and (ii) $I_{l_1,n_{l_{1opt}}}^{l_1-1}$ to $I_{l_1,n_{l_{1opt}}}^{l}$. Also, the coefficients decremented are $I_{l_1',\lceil n_{l_1}-opt/2^{l_1'-l_1}\rceil}^{l_1}, l \leq l_1' \leq L$.

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

1. Generate new call with rate $2^{l-1}R, 1 \leq l \leq L$
2. *If current used capacity in the tree* $+ 2^{l-1}R \leq$ *total tree capacity.*
   - 2.1 Identify the optimum codes $C_{l_1,n_{l_{1opt}}}, l \leq l_1 \leq L-1$.
   - 2.2 Assign code $C_{l,n_{l_{opt}}}$ to the new call. Do code assignment and blocking.

     Update code indices $I_{l_1,n_{l_1}}^{l''}$, $C_{l_1,n_{l_{1opt}}}, l \leq l_1 \leq L-1, l_1 \leq l'' \leq (L-1)$.
   - 2.3 Go to 1.

   Else
   - 2.1 Reject call.
   - 2.2 Go to step 1.

   End

Table 2 Finding optimum $2R$ code in Fig. 1

Layer number	Candidate codes	1st candidate code parameters	2nd candidate code parameters	Optimum code selection
6	*$C_{6,1}$ and $C_{6,2}$	Code $C_{6,1}$ $I_{6,1}$= [1,0,3,5] $N_{6,1} = 9$ $P_{6,1} = 19R$	Code $C_{6,2}$ $I_{6,2}$ = [1,1,0,0] $N_{6,2} = 2$ $P_{6,2} = 12R$	$N_{6,2} < N_{6,1}$, $C_{6,2}$ is optimum but $I^2_{6,2} = 0$, so $C_{6,1}$ is optimum code
5	$C_{5,1}$ and *$C_{5,2}$	Code $C_{5,1}$ $I_{5,1}$= [2,3] $N_{5,1}$= 5 $P_{5,1} = 7R$	Code $C_{5,2}$ $I_{5,2}$ = [1,0,1,2] $N_{5,2} = 4$ $P_{5,2}$= $12R$	$N_{5,1} > N_{5,2}$, $C_{5,2}$ is optimum code
	*$C_{4,3}$ and $C_{4,4}$	Code $C_{4,3}$ $I_{4,3}$= [1,2] $N_{4,3}$= 3 $P_{4,3} = 4R$	Code $C_{4,4}$ $I_{4,4}$= [] No vacant children code	$I_{4,4}$= [], so $C_{4,3}$ is optimum code
3	*$C_{3,5}$ and $C_{3,6}$	Code $C_{3,5}$ $I_{3,5}$= [1,0] $N_{3,5}$= 1 $P_{4,3} = 2R$	Code $C_{3,6}$ $I_{3,6}$= [2] $N_{3,6}$= 2 $P_{3,6} = 2R$	$N_{3,5} < N_{3,6}$, $C_{3,5}$ is optimum code
2	*$C_{2,9}$	Only one code $C_{2,9}$ of rate $2R$vacant and is optimum code		

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

The scheme is simple and is useful for medium to high traffic load conditions. For low traffic load conditions, traditional LCA [5] and FSP [10] schemes can be used to provide simplicity and fewer code searches. To illustrate algorithm of TD scheme consider the 7 layer code tree status as shown in Fig. 1, if a new call with rate $2R$ arrives, the procedure of optimum path and code selection is shown in Fig. 2 and tabulated in Table 2. For simplicity, the code index is shown with only m coefficients when there is no vacant code in layer $m+1$ to L. For example, the code index of $C_{6,1}$ is $I_{6,1} = [1, 0, 3, 5]$ instead of [0, 0, 0, 1, 0, 3, 5]. The algorithm starts optimum code identification procedure at layer 6. There are two candidate codes namely, $C_{6,1}$ and $C_{6,2}$, with code indices $I_{6,1} = [1, 0, 3, 5]$ and $I_{6,2} = [1, 1, 0, 0]$ respectively. The value of $N_{6,1}$, $N_{6,2}$, $N_{6,1}$ and $P_{6,2}$ are 9, 2, 19 (1*8+3*2+5*1), and 12(1*8+1*4) respectively. If we are allowed to use the vacant codes whose immediate parents are free, then $C_{6,2}$ is optimum, but in this example, such code usage is not allowed and only those vacant codes can be used whose immediate parents are blocked. Therefore $C_{6,1}$ is the optimum code in layer 6. The procedure is repeated till layer 2 and the optimum code to be used for $2R$ call is $C_{2,9}$. The optimum path is shown by arrows in Fig. 2.

2.2 Multi Code Enhancement

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

2.2.1 Multi Code Assignment Scheme With Minimum Rakes

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

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

$$W_1 = kR - 2^{l_1-1}R \tag{4}$$

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

$$W_t = kR - \sum_{i=1}^{t} 2^{l_1-1}R \tag{5}$$

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

For illustration of the minimum rake scheme, if a call with rate $13R$ arrives with 4 rakes provision, the minimum possible fractions are $8R$, $4R$ and R respectively. For each of the fraction, the optimum code is identified like single code TD scheme.

2.2.2 Multi Code Assignment With Maximum Rakes (Minimum Scattering)

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

Lower rate splitting first The first part of the algorithm is to find minimum number of rake combiners required to handle new call according to Eqs. (4) and (5). Let $t|t < m$ steps leads to zero wastage capacity, $i.e. W_t = kR - \sum_{i=1}^{t} 2^{l_i-1}R$, and therefore the minimum rakes required are t. Let $2^{l_i^t-1}$, $t \leq m$, $i \leq t$ represents i^{th} rate fraction of total t fractions. If initially rate fraction vector is represented by $\overline{R} = [2^{l_1^t-1}, 2^{l_2^t-1}, \ldots 2^{l_{j_1}^t-1}, ..2^{l_t^t-1}]$, the algorithm identifies the rate fraction j_1 so that $2^{l_i^t-1}$ is smallest for $i = j_1$. The rate fraction $2^{l_i^t-1}$ is broken into two rate sub fractions of amount $2^{l_{j_1}^t-1}/2$ and $2^{l_{j_1}^t-1}/2$. The new rate fraction vector becomes

$$\begin{aligned}\overline{R} &= [2^{l_1^t-1}, 2^{l_2^t-1}, \ldots, 2^{l_{j_1}^t-1}/2, 2^{l_{j_1}^t-1}/2, ..2^{l_t^t-1}] \\ &= [2^{l_1^{t+1}-1}, 2^{l_2^{t+1}-1}, \ldots, 2^{l_{j_1}^{t+1}}, 2^{l_{j_1+1}^{t+1}-1}/2, ..2^{l_{t+1}^{t+1}-1}]\end{aligned} \tag{6}$$

The result in (6) can be extended to identify m_1 fractions and the optimum rate fraction vector becomes

$$\overline{R} = [2^{l_1^m-1}, 2^{l_2^m-1}, .., 2^{l_{m_1}^m-1}] \tag{7}$$

In (7), $m_1 = m$ if $k \geq m$, and $m_1 < m$ if $k < m$. All the coefficients of the rate fraction vector are handled by different rakes. For illustration, if $13R$ call arrives, the minimum fractions are $8R$, $4R$, and R. The lowest fraction that can be broken further is $4R$. This is broken into two fractions of rate R each. Therefore the 4 fractions are $8R$, $2R$, $2R$, and Rrespectively.

Again the code assignment for each fraction is done similar to the single code TD scheme.

Higher rate splitting first Considering the definition of $\overline{R}$, the algorithm identify the rate fraction j_1 so that $2^{l_i^t-1}, 1 \leq i \leq t$ is largest for $i = j_1$. The procedure can be repeated maximum j_1 times and the rate fraction vector is $\overline{R} = [2^{l_1^m-1}, 2^{l_2^m-1}, .., 2^{l_{m1}^m-1}]$ as given in Eq. (7). All the coefficients of the rate fraction vector are handled by different rakes.

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

2.2.3 Scattered Multi Code Scheme

The incoming rate $2^{l-1}R$ is divided into maximum m_1, $m_1 = \min(k, m)$ fractions 2^{l_i-1}, $1 \leq i \leq m_1$, $1 \leq l_i \leq l$ such that $\sum_{i=1}^{m_1} 2^{j_i} = 2^{l-1}$. The division is performed either by lower rate splitting first or higher rate splitting first depending upon the requirement. For each fraction 2^{j_i}, find the optimum code as discussed in Sect. 2.1. For illustration, let $13R$ call rate arrives with minimum fractions $8R$, $4R$, and R arrives. If the system has 4 rakes, the highest fraction that is broken into two fractions of rate $4R$ each. Therefore the 4 fractions are $4R$, $4R$, $4R$, and R respectively. The code assignment for each fraction is done similar to the single code TD scheme.

The algorithm of scattered multi code scheme is described below.

1. Enter number of rakes *m.*

2. Generate new call.

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

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

For $1 \leq i \leq m_1$

3.1.1 Choose the optimum code from layer l_i+1.

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

3.1.3 Update code indices.

End

3.2 Go to step 2.

Else

3.1 Reject call.

3.2 Go to step 1.

End

2.2.4 Grouped Multi Code Scheme

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

If at least one vacant code is available in layer l, identify optimum code $C_{l,n_{l_{opt}}}$ in layer l. Assign leftmost child of $C_{l,n_{l_{opt}}}$ $(i.e.\text{code}C_{l_1,2n_{l_{opt}}-1})$ to rate fraction $2^{l_1-1}R$.If code

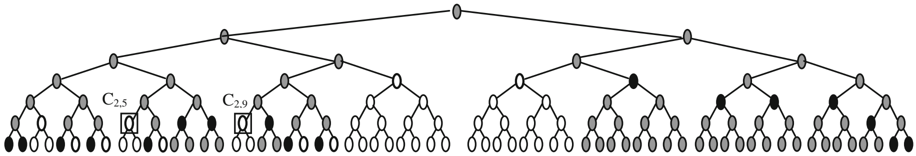

Fig. 3 Illustration of multi code top down scheme using minimum rakes for call handling

$C_{l_1,2n_{l_{opt}}-1}$ is denoted by $C_{l,n_{l_{opt}}}$, the vacant code used for 2[nd] fraction is $C_{l_2,2^{l_1-l_2}n_{l_{1opt}}-2^{l_1-l_2}+1}$. The result can be generalized for ith fraction and the code used is $C_{l_i,2^{l_{i-1}-l_i}n_{l_{1(i-1)opt}}-2^{l_{i-1}-l_i}+1}$.

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

Let a $4R$ call arrives with code tree status in Fig. 3 and the system is equipped with three rakes. Using minimum rake multi code scheme, the rate fractions are $2R$ and $2R$. The optimum codes are $C_{2,9}$ and $C_{2,5}$ respectively as shown in Fig. 3 with the help of Table 3.

Consider a partial code tree shown in Fig. 4. If a call $4R$ arrives and the system is equipped with 3 rakes, the three rate fractions become $2R$, R, and R respectively. The optimum code search identification starts from the root, and let the code $C_{4,8}$ is optimum for these rate fractions. The grouped code scheme uses codes $C_{2,30}$, $C_{1,61}$, and $C_{1,62}$ for the rate fractions $2R$, R, and R respectively, as the fractions are placed closest to each other. This is illustrated in Fig. 4b. If on the other hand, the scattered approach is used, the codes $C_{2,30}$, $C_{1,58}$, and $C_{1,64}$ are used for rate fractions $2R$, R, and R respectively as shown in Fig. 4c.

2.3 Dynamic Code Assignment Enhancement

For a new call with rate $2^{l-1}R$, the reassignments are required if $I_{l,n_l}^{l_1} = 0, l_1 \geq l$ and $\sum_{l_1=1}^{l-1} I_{l,n_l}^{l_1} \times 2^{l_1-1} \geq 2^{l-1}$. In DCA, the definition of optimum code is different. The code $C_{l_1,n_{l_1}}$ is in the path from root to optimum code if $I_{l_1,n_{l_1}}^{\cdot} 1 \leq l_1 \leq (L-1)$ is maximum, *i.e.* the number of vacant children in layer 1 to l-1 for code $C_{l_1,n_{l_1}}$ is highest. The algorithm does not intend to find the crowded part rather it finds a blocked code with capacity $2^{l-1}R$, which has least number of busy children. After selecting the best blocked code, all the calls handled by its busy children have to be shifted to appropriate (crowded) locations using TD scheme. When all calls are shifted the code becomes available for $2^{l-1}R$ call. The DCA algorithm alongwith TD scheme works as follows.

Table 3 Finding optimum codes using minimum rakes for a call of rate $4R$ code in Fig. 3

Layer number	Candidate codes	Minimum rakes				
		1st $2R$			2nd $2R$	
		1st candidate code parameters	2nd candidate code parameters	Candidate codes	1st candidate code parameters	2nd candidate code parameters
6	$*C_{6,1}$ and $C_{6,2}$	Code $C_{6,1}$	Code $C_{6,2}$	$*C_{6,1}$ and $C_{6,2}$	Code $C_{6,1}$	Code $C_{6,2}$
		$I_{6,1}$ = [1,0,3,5]	$I_{6,2}$ = [1,0,0,0]		$I_{6,1}$ = [1,0,2,5]	$I_{6,2}$ = [1,0,0,0]
		$N_{6,1} = 9$	$N_{6,2} = 2$		$N_{6,1} = 8$	$N_{6,2} = 2$
		$P_{6,1} = 19R$	$P_{6,2} = 12R$		$P_{6,1} = 17R$	$P_{6,2} = 12R$
5	$C_{5,1}$ and $*C_{5,2}$	Code $C_{5,1}$	Code $C_{5,2}$	$*C_{5,1}$ and $C_{5,2}$	Code $C_{5,1}$	Code $C_{5,2}$
		$I_{5,1}$ = [2,3]	$I_{5,2}$ = [1,0,1,2]		$I_{5,1}$ = [2,3]	$I_{5,2}$ = []
		$N_{5,1} = 5$	$N_{5,2} = 4$		$N_{5,1} = 5$	No vacant children code
		$P_{5,1} = 7R$	$P_{5,2} = 12R$		$P_{5,1} = 7R$	
4	$*C_{4,3}$ and $C_{4,4}$	Code $C_{4,3}$	Code $C_{4,4}$	$C_{4,1}$ and $*C_{4,2}$	Code $C_{4,1}$	Code $C_{4,2}$
		$I_{4,3}$ = [1,2]	$I_{4,4}$ = []		$I_{4,3}$ = [1,2]	$I_{4,4}$ = [1,1]
		$N_{4,3} = 3$	No vacant children code		$N_{4,3} = 3$	$N_{4,3} = 2$
		$P_{4,3} = 4R$			$P_{4,3} = 4R$	$P_{4,3} = 3R$
3	$*C_{3,5}$ and $C_{3,6}$	Code $C_{3,5}$	Code $C_{3,6}$	$*C_{3,3}$ and $C_{3,4}$	Code $C_{3,3}$	Code $C_{3,4}$
		$I_{3,5}$ = [1,0]	$I_{3,6}$ = [2]		$I_{3,5}$ = [1,1]	$I_{3,6}$ = []
		$N_{4,3} = 1$	$N_{3,6} = 2$		$N_{4,3} = 1$	No vacant children code
		$P_{4,3} = 2R$	$P_{3,6} = 2R$		$P_{4,3} = 2R$	
2	$*C_{2,9}$	$C_{2,9}$ is optimum code and procedure stops		$*C_{2,5}$	$C_{2,5}$ is optimum code and procedure stops	

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

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

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

2.1 Choose the optimum code in layer l_1, for which $I_{l_1,n_{l_1}}, 1 \leq l_1 \leq (L-1)$ is maximum.

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

2.2.1 $l_1 = l_1 - 1$.

2.2.2 *if* $l_1 == l$

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

else

- Go to step 2.1.

end

2.2.3 Go to step 2.1.

else

2.2.1 Sibling code is the optimum code

2.2.2 Go to step 2.2.

end

else

2.1 reject call.

2.2 go to step 1.

End

3 Results

The codes searched and blocking probability performances of the pure TD and its hybrid schemes are compared with existing good schemes. For simulation, five classes of users are considered with rates R, $2R$, $4R$, $8R$, and $16R$respectively. For ith class, the arrival rate is represented by. Call duration $1/\mu_i$ is exponentially distributed with mean value of 1 units of time. If we define $\rho_i = \lambda_i/\mu_i$ as traffic load of the ith class users, then for 5 class system the average arrival rate and average traffic load is $\lambda = \sum_{i=1}^{5} \lambda_i$ and $\rho =$

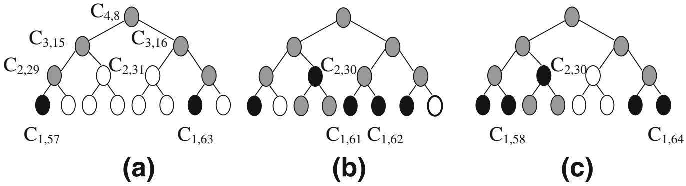

Fig. 4 Code tree status in **a** for illustration of **b** grouped **c** scattered assignment scheme

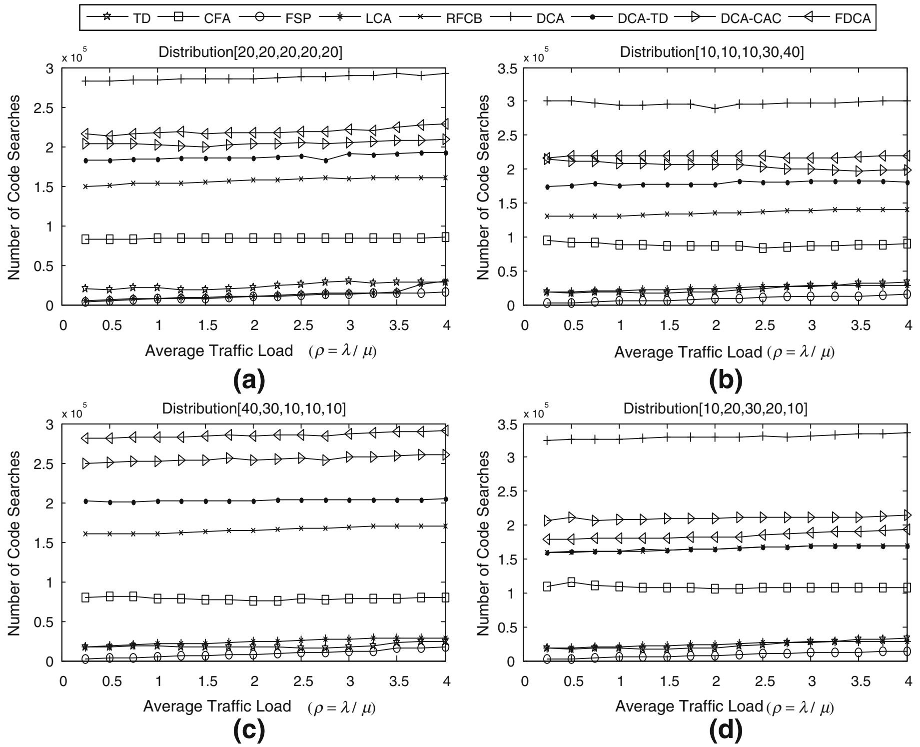

Fig. 5 Comparison of number of code searches in single code schemes for distribution: **a** [20,20,20,20,20], **b** [10,10,10,30,40], **c** [40,30,10,10,10], **d** [10,30,20,30,10]. *legend is above plots

$\sum_{i=1}^{5} \lambda_i/\mu_i$ respectively. The average arrival rate (or average traffic load as $1/\mu_i$ is 1) is assumed to be Poisson distributed with mean value varying from 0 to 4 calls per unit of time. In our simulation we consider call duration of all the calls equal *i.e.* $1/\mu = 1/\mu_i = 1$. Therefore, the average traffic load is $\rho = 1/\mu \times \sum_{i=1}^{5} \lambda_i = \lambda/\mu$. The maximum capacity of the tree is $128R$ (R is 7.5 kbps). Simulation is done for 5,000 users and result is average of 25 simulations. Define $[p_1, p_2, p_3, p_4, p_5]$ as capacity distribution matrix, where $p_i, i \in [1, 5]$ is the percentage fraction of the total tree capacity used by the ith class users. As the traffic load includes five different rates, where R, $2R$ are low rate (assumed) real time calls, and higher rates $4R$, $8R$, $16R$ can be considered as non real time calls. The number of code searches is another performance parameter included in this paper. Two distribution scenarios are analyzed, (i) [20,20,20,20,20], uniform distribution, (ii) [40,30,10,10,10], low rate calls dominating. The proposed TD code selection scheme is compared with CFA, FSP, leftmost code assignment (LCA), recursive fewer code blocked (RFCB), dynamic code assignment (DCA), computationally efficient dynamic code assignment with call admission control (DCA-CAC) and fast dynamic code assignment (FDCA) schemes discussed earlier.

3.1 Single Code Assignment Performance

The results of the number of code searches and code blocking probability are plotted in Figs. 5 and 6 respectively. The number of codes searched before assignment are smaller in the TD scheme as shown in Fig. 5 compare to all other popular approaches except FSP and LCA which

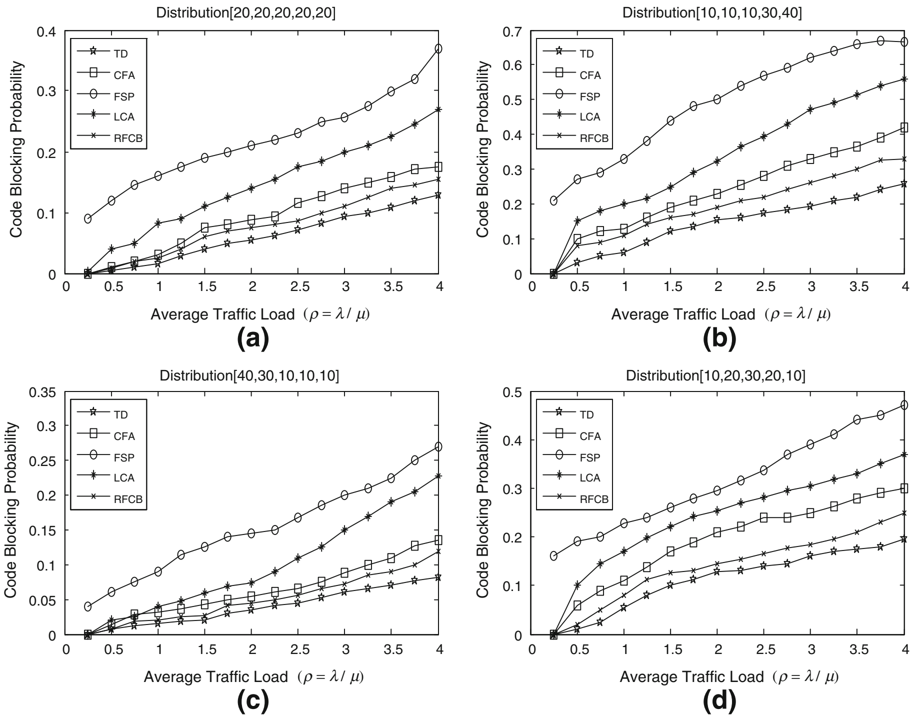

Fig. 6 Comparison of code blocking probability in single code schemes for distribution: **a** [20,20,20,20,20], **b** [10,10,10,30,40], **c** [40,30,10,10,10], **d** [10,30,20,30,10]

suffers from high code blocking probability. The number of codes searched in TD scheme is comparable to LCA and FSP. Since, both FSP and LCA produce large code blocking, they are outdated. The TD scheme selects an optimum code on each layer and selects the code with minimum number of vacant codes under it which leads to least blocking of vacant codes, when children of these codes is assigned to a call. Hence, TD scheme can be used in the pure form or in the integrated form with other novel single code and multi code methods to improve their performance. The comparison of TD scheme with CFA and DCA is given in "Appendix" which shows that, it requires lesser number of code searches as compared to CFA and DCA. The enhanced DCA using TD requires significantly fewer code searches as compare to traditional DCA, DCA-CAC and FDCA. All these schemes take advantage of zero code blocking probability of DCA. However, these schemes differ in throughput performance. The uniform distribution scenario is used to compare their throughput, DCA-TD provides highest throughput as shown in Fig. 7. The result in Fig. 6 shows that the TD scheme has significantly less code blocking for all four distributions. The code blocking probability is not plotted for DCA and its enhanced schemes as they always leads to zero code blocking.

3.2 Multi Code Assignment Performance

The performance improvements in multi code enhancements namely multi code top down scattered (MC-TDS) and multi code TD grouped (MC-TDG) schemes are compared with multi code scheme equipped with LCA and CFA. Again, two versions of multi code scheme

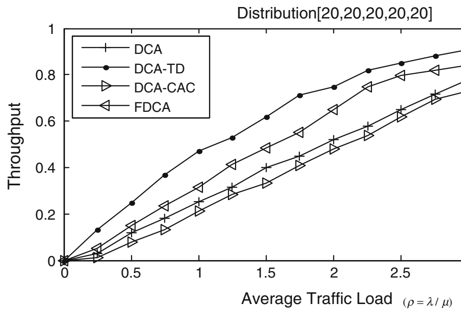

Fig. 7 Throughput comparison of in single code DCA and DCA enhanced schemes in uniform distribution scenario

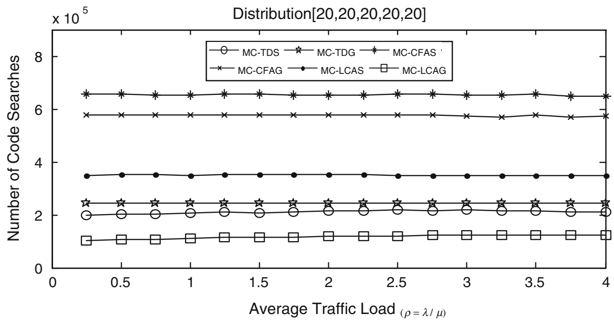

Fig. 8 Comparison of code searches in multi code schemes for uniform distribution

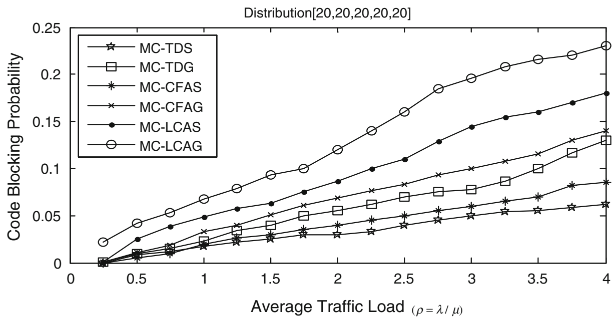

Fig. 9 Comparison of code blocking probability in multi code schemes for uniform distribution

are analyzed for both LCA and CFA, (i) Multi code left code assignment with scattered approach (MC-LCAS), (ii) Multi code left code assignment with grouped approach (MC-LCAG), (iii) Multi code crowded first assignment with scattered approach (MC-CFAS), (iv) Multi code crowded first assignment with grouped approach (MC-CFAG). The comparison is done for uniform distribution only. The remaining distributions gives similar performance. The MC-TDG scheme requires least number of code searches as compare to all other approaches as shown in Fig. 8. If CS_x denotes the code searches for scheme x, the various schemes can be arranged as

$$CS_{MC-TDG} < CS_{MC-LCAG} < CS_{MC-TDS} < CS_{MC-LCAS} < CS_{MC-CFAG} < CS_{MC-CFAG} \tag{8}$$

Also, MC-TDS scheme produce least blocking compared to other schemes as shown in Fig. 9. If BP_x denotes the blocking probability for scheme x, the various schemes can be arranged as

$$BP_{MC-TDS} < BP_{MC-CFAS} < BP_{MC-TDG} < BP_{MC-CFAG} < BP_{MC-LCAS} < CS_{MC-LCAG} \tag{9}$$

Therefore, the TD scheme is very useful for all traffic conditions and is better alternative for CDMA wireless networks using OVSF codes compared to existing schemes.

4 Conclusion

Real time calls cannot tolerate large call establishment delay therefore should be assigned using scheme that has less call establishment delay. In OVSF based networks, the choice of code assignment has significant impact on call establishment delay. The TD code assignment scheme proposed in this paper reduces this delay without compromising performance degradation in terms of code blocking. The TD scheme can also be integrated with dynamic code and multi code assignment schemes. The combination of dynamic code assignment and TD scheme provides best results for both code blocking and number of code searches (call establishment delay). Also, the requirement of large reassignment overhead encourages the use of multi code scheme with TD integration. The code index updation required for TD scheme can be made adaptive to call arrival rates for improvement in future.

Appendix: Code Searches

Pure Top Down Code Searches

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

$$N_{TD} = 2(L-l) + 1 \tag{10}$$

CFA Code Searches

For a call of rate $2^{l-1}R$, the maximum number of code searched to find vacant code in layer l is 2^{L-l}. If there are z vacant codes in layer l, for each code, C_{l,y_i}, $1 \leq i \leq z$, $1 \leq x_i \leq 2^{l-1}$, CFA [3] scheme finds number of busy codes under immediate parent of each C_{l,y_i}. The total number of code searches for parents of each vacant code in layer l+1 are $z \times 2^{l+1}$. If a unique immediate parent code (say$C_{l+1,\lceil y_i/2 \rceil}$) with maximum number of busy codes exists, new call will be assigned to its children and code searching stops. Otherwise, let z_1 number of parent codes in layer $l+1$ that leads to tie for maximum number of busy children. The number of code searches for layer $l+2$ are $z_1 \times 2^{l+2}$. If a unique result does not exist the procedure is repeated till layer L giving maximum code searches. The total number of code searches for CFA becomes

$$N_{CFA} = 2^{(L-l)} + z \times 2^{l+1} + z_1 \times 2^{l+2} + \ldots. + z_{L-2} \times 2^{l+L-1} \tag{11}$$

Top Down and Dynamic Code Assignment (DCA)

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

Conventional DCA

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

$$NC^1_{DCA} = 2^{L-l} \tag{12}$$

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

$$NC^2_{DCA} = k_1 \times (2 + 4 + \cdots .2^{l-1}). \tag{13}$$

Total number of codes searched in DCA becomes

$$NC'_{DCA} = NC^1_{DCA} + NC^2_{DCA} = 2^{L-l} + k_1 \times \sum_{i=1}^{l-1} 2^i \tag{14}$$

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

$$N''_{DCA} = \sum_{p=1}^{l-1} a_p \times 2^{L-p} \tag{15}$$

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

So, total number searches becomes

$$N_{DCA} = \sum_{l=1}^{L} \lambda_l \times [NC'_{DCA} + NC''_{DCA}] \tag{16}$$

$$N_{DCA} = \sum_{l=1}^{L} \lambda_l \times [NC'_{DCA} + NC''_{DCA}] \tag{17}$$

DCA Top Down

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

$$N_{TD-DCA} = 2(L-l) + 1 \tag{18}$$

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

$$N_{TD-DCA_2} = \sum_{l'=1}^{l-1} 2^{l'} \tag{19}$$

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

$$N_{TD-DCA_3} = \sum_{l'=1}^{l-1} p_{l'} \times (2 \times (L - l') + 1) \tag{20}$$

The total code searches in TD DCA are

$$N_{TD-DCA} = N_{TD-DCA_1} + N_{TD-DCA_2} + N_{TD-DCA_3} \tag{21}$$

$$N_{TD-DCA} = 2(L - l) + 1 + \sum_{l'=1}^{l-1} (2^{l'} + p_{l'} \times (2 \times (L - l') + 1)) \tag{22}$$

References

1. Karakayali, M. K., Yates, R. D., & Razoumov, L. V. (2006). Downlink throughput maximization in CDMA wireless networks. *IEEE Transaction on Wireless Communication*, *12*, 3492–3500.
2. Fossa, C. E., Nathaniel Jr., & Davis, J. (2002). A dynamic code assignment algorithm for Quality of Service in 3G wireless networks. In *Proceedings of the IEEE WC* (Vol. 1, pp. 1–6).
3. Adachi, F., Sawahashi, M., & Suda, H. (1997). Tree structured generation of orthogonal spreading codes with different lengths for forward link of DS-CDMA mobile. *Electronic Letter*, *33*, 27–28.
4. Chao, C. M., Tseng, Y. C., & Wang, L. C. (2005). Reducing internal and external fragmentations of OVSF codes in WCDMA systems with multiple codes. *IEEE Transaction on Wireless Communication*, *4*, 1516–1526.
5. Tseng, Y. C., Chao, C. M., & Wu, S. L. (2001). Code placement and replacement strategies for wideband CDMA OVSF code tree management. In *Proceedings of the IEEE GLOBECOM* (Vol. 1, pp. 562–566).
6. Rouskas, A. N., & Skoutas, D. N. (2005). Management of channelization codes at the forward link of WCDMA. *IEEE Communication Letter*, *9*, 679–681.
7. Saini, D. S., Kanwar, S., Parikh, P.,& Kumar, U. (2009). Vacant code searching and selection of optimum code assignment in WCDMA wireless network. In *Proceedings of the IEEE N GMAST* (Vol. 3, pp. 224–229).
8. Ferng, H. W., Chin, H. L., Shiung, D., & Chen, Y. T. (2005). An OVSF code tree partition policy for WCDMA based on the multi-code approach. In *Proceedings of the IEEE VTC* (Vol. 62, pp. 1212–1216).
9. Minn, T., & Siu, K. Y. (2000). Dynamic assignment of orthogonal variable spreading factor codes in WCDMA. *IEEE Journal in Selected Areas of Communication*, *18*, 1429–1440.
10. Park, J. S. & Lee, D. C. (2003). Enhanced fixed and dynamic code assignment policies for OVSF-CDMA systems. In *Proceedings of the IEEE ICWN* (pp. 620–625).
11. Saini, D. S., & Bhoosan, S. V. (2006). Adaptive assignment scheme for OVSF codes in WCDMA. In *Proceedings of the IEEE ICWMC*, p. 65.
12. Wan, C. S., Shih, W. K., & Chang, R. C. (2003). Fast dynamic code assignment in next generation wireless access network. *Computer Communication*, *26*, 1634–1643.
13. Park, J. S., Huang, L., & Kuo, C. C. J. (2009). Computationally efficient dynamic code assignment schemes with call admission control (DCA-CAC) for OVSF-CDMA systems. *IEEE Transaction on Vehicular Technology*, *57*, 286–296.
14. Rouskas, A., & Skoutas, D. (2002). OVSF code assignment and reassignment at the forward link of W-CDMA 3G systems. In *Proceedings of the IEEE PIMRC* (Vol. 13, pp. 2404–240).
15. Balyan, V. & Saini, D. S. (2009). Call elapsed time and reduction in code blocking for WCDMA networks. In *Proceedings of the IEEE SoftCOM* (Vol. 17, pp. 141–145).
16. Saini, D. S., & Upadhyay, M. (2009). Multiple rake combiners and performance improvement in WCDMA systems. *IEEE Transaction Vehicular Technology.*, *58*, 3361–3370.
17. Yang, Y., & Yum, T. S. P. (2005). Multicode multirate compact assignment of OVSF codes for QoS differentiated terminals. *IEEE Transaction Vehicular Technology*, *54*, 2114–2124.
18. Chen, M. X. (2008). Efficient integration OVSF code management architecture in UMTS. *Computer Communication*, *31*, 3103–3112.

19. Cheng, S. T., & Hsieh, M. T. (2005). Design and analysis of time-based code allocation schemes in W-CDMA systems. *IEEE Transaction Mobile Computing*, *4*, 3103–3112.
20. Vadde, K., & Qam, H. (2004). A code assignment algorithm for nonblocking OVSF codes in WCDMA. *Springer Telecommunication Systems*, *25*(3), 417–431.
21. Chen, Y. S., Lin, T. H., & Chien, J. Y. (2005). A fast code assignment strategy for a WCDMA rotated-OVSF tree with code-locality property. *Springer Telecommunication Systems*, *29*(3), 199–218.
22. Balyan, V., & Saini, D. S. (2013). *Vacant codes grouping and fast OVSF code assignment scheme for WCDMA networks*. Springer Journal of Telecommunication System, *52*(4), 1719–1729.

Author Biographies

Davinder S. Saini was born in Nalagarh, India in January 1976. He received B.E degree in Electronics and Telecommunication engineering from College of Engineering Osmanabad, India in 1998. He received M.Tech degree in communication systems from Indian Institute of Technology (IIT) Roorkee, India in 2001. He received Ph.D. degree in electronics and communication from Jaypee University of Information Technology Waknaghat, India in 2008. He is with Jaypee University of Information Technology Waknaghat since June 2002, and he is currently working as an Associate Professor in Electronics and Communication department. His research areas include Channelization (OVSF) codes and optimization in WCDMA, routing algorithms and security issues in MANETs.

Vipin Balyan received B.E degree in Electronics & Communication with honors from U.P. Technical University, Lucknow, Uttar Pradesh, India in 2003 and the M.Tech degree in Electronics & Networking from LaTrobe University, Bundoora, Melbourne, Australia in 2006. He has been with RKGIT, Ghaziabad affiliated to U.P. Technical University, Lucknow, Uttar Pradesh, India as a Lecturer for 2 years. He was associated with Jaypee University, Wakhnaghat, India as a Senior Lecturer. He is currently working as a Senior Lecturer in Jaypee Institute of Information Technology, Noida, India and pursuing his Ph.D in **Efficient Single Code Assignment in OVSF based WCDMA Wireless Networks.**