CHAPTER 3
OPTIMUM VACANT CODE IDENTIFICATION

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

<table>
<thead>
<tr>
<th>User Rate (kbps)</th>
<th>SF</th>
<th>Transmission Rate (Mcps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.5</td>
<td>512</td>
<td>3.84</td>
</tr>
<tr>
<td>15</td>
<td>256</td>
<td>3.84</td>
</tr>
<tr>
<td>30</td>
<td>178</td>
<td>3.84</td>
</tr>
<tr>
<td>60</td>
<td>64</td>
<td>3.84</td>
</tr>
<tr>
<td>120</td>
<td>32</td>
<td>3.84</td>
</tr>
<tr>
<td>240</td>
<td>16</td>
<td>3.84</td>
</tr>
<tr>
<td>480</td>
<td>8</td>
<td>3.84</td>
</tr>
<tr>
<td>960</td>
<td>4</td>
<td>3.84</td>
</tr>
</tbody>
</table>

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

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

3.1 TOP DOWN SCHEME
3.1.1 SING LE CODE TOP DOWN SCHEME

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

<table>
<thead>
<tr>
<th>Layer Number</th>
<th>Candidate Codes</th>
<th>1st Candidate Code Parameters</th>
<th>2nd Candidate Code Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>$^*C_{5,1}$ and $C_{6,2}$</td>
<td>$C_{6,1}$, $l_{6,1}= [1,0,3,5]$ $N_{6,1} = 9$ $P_{6,1} = 19R$</td>
<td>$C_{6,2}$, $l_{6,2} = [1,1,0,0]$ $N_{6,2} = 2$ $P_{6,2} = 12R$</td>
</tr>
<tr>
<td>5</td>
<td>$C_{5,1}$ and $^*C_{5,2}$</td>
<td>$C_{5,1}$, $l_{5,1} = [2,3]$ $N_{5,1} = 5$ $P_{5,1} = 7R$</td>
<td>$C_{5,2}$, $l_{5,2} = [1,0,1,2]$ $N_{5,2} = 4$ $P_{5,2} = 12R$</td>
</tr>
<tr>
<td>4</td>
<td>$^*C_{3,5}$ and $C_{4,4}$</td>
<td>$C_{4,3}$, $l_{4,3} = [1,2]$ $N_{4,3} = 3$ $P_{4,3} = 4R$</td>
<td>$C_{4,4}$, $l_{4,4} = []$ No vacant children code</td>
</tr>
<tr>
<td>3</td>
<td>$^*C_{3,5}$ and $C_{3,6}$</td>
<td>$C_{3,5}$, $l_{3,5} = [1,0]$ $N_{3,5} = 1$ $P_{3,5} = 2R$</td>
<td>$C_{3,6}$, $l_{3,6} = [2]$ $N_{3,6} = 2$ $P_{3,6} = 2R$</td>
</tr>
</tbody>
</table>

| 2            | $^*C_{2,9}$ | $C_{2,9}$ is optimum code and procedure stops |

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

$C_{l,n_{l_{opt}}}$ which leads to least code blocking. For a code $C_{l,n_l}$, define index $I_{l,n_l} = [I_{l,n_l}^1, I_{l,n_l}^2, \ldots, I_{l,n_l}^L]$, where the coefficient $I_{l,n_l}^i, 1 \leq i \leq L$ has following properties:
Figure 3.1: A seven layer CVSF code tree.
Figure 3.2: Illustration of single code top down scheme.
- \( I_{l,n_i}^{p}, 1 \leq p \leq L \) represents total vacant children under code \( C_{l,n_i} \) in layer \( l' \).

- \( I_{l,n_i}^{p}, l+1 \leq p \leq L \), represents the status of parent of \( C_{l,n_i} \) in layer \( l' \), with \( I_{l,n_i}^{p} = 0 \) representing blocked/busy parent.

- \( I_{l,n_i}^{1} \) is the status of the code \( C_{l,n_i} \) itself (0 for busy/blocked code and 1 if \( C_{l,n_i} \) is vacant with the condition that all the parents are blocked).

For a code \( C_{l,n_i} \) with index \( I_{l,n_i} = [I_{l,n_i}^{1}, I_{l,n_i}^{1-1}, ..., I_{l,n_i}^{1}] \), the total vacant children under \( C_{l,n_i} \) are

\[
N_{l,n_i} = \sum_{p=1}^{l-1} I_{l,n_i}^{p} \tag{3.1}
\]

Also, the vacant capacity in the children of \( C_{l,n_i} \) is

\[
P_{l,n_i} = \sum_{p=1}^{l-1} I_{l,n_i}^{p} \times 2^{p-1} R \tag{3.2}
\]

If for code \( C_{l,n_i} \), the value of coefficient \( I_{l,n_i}^{p}, 1 \leq p \leq L-1 \) is \( p \), due to \( p \) vacant children of \( C_{l,n_i} \) in layer \( l' \), the indices \( I_{l,n_i}^{p-1}, ..., I_{l,n_i}^{1} \) do not include the children of these \( p \) vacant codes. In other words, a vacant code whose parent is also vacant is not counted in the definition of code indices. The code index \( I_{l,n_i} \) is updated periodically and in addition, at the arrival and completion of new call.

For a code \( C_{l,n_i} \), the sibling is \( C_{l,n_i+1} \), if \( n_i \) is odd, and \( C_{l,n_i-1} \), if \( n_i \) is even. At the arrival of call with rate \( 2^{l-1} R \), the optimum vacant code in layer \( l \) is required. If we represent sibling of \( C_{l,n_i}, 1 \leq l \leq L-1 \) by \( C_{l,n_{i+1}} \), the code \( C_{l,n_{i+1}} \) is in the path from root to optimum code if, (a) \( I_{l,n_{i+1}}^{l} \) is non zero and \( \sum_{i=1}^{l-1} I_{l,n_{i+1}}^{i} = \sum_{i=1}^{l-1} I_{l,n_{i}}^{i} \) and, or (b) \( I_{l,n_{i+1}}^{l} \) is zero but \( I_{l,n_{i+1}}^{l}, l \leq l' \leq L \) is non zero and, \( \sum_{i=1}^{l-1} I_{l,n_{i+1}}^{i} = \sum_{i=1}^{l-1} I_{l,n_{i}}^{i} \). If both (a) and (b) fails and \( I_{l,n_{i+1}}^{l}, l \leq l' \leq L \) is non zero, the code \( C_{l,n_{i+1}} \) will be in the path from root to optimum code. The procedure is repeated till layer \( l \) if for all identified optimum codes at least one vacant code is available in layer \( l \). If on the other hand for an optimum code in layer \( l' \), \( l < l' < L-1 \), there is no vacant child in layer \( l \), the procedure stops at layer, and the optimum code in layer \( l \) is the leftmost child of the optimum code in layer.
Once all the codes $C_{l,n_{opt}}$, $1 < l < L - 1$ are identified, the optimum code in layer $l$ is assigned to the call. The code indices for codes $C_{l,n_{opt}}$, $1 \leq l \leq L - 1$, are updates as follows,

(i) If $I_{l,n_{opt}}^l$ is non-zero, i.e., the optimum vacant code exists in layer $l$ (say $C_{l,n_{opt}}$) with all its parents blocked. This vacant code is assigned to the new call and the coefficients $I_{l,n_{opt}}^l$, $1 \leq l \leq L - 1$ are decremented by one, (ii) If $I_{l,n_{opt}}^l$ is zero (there is no vacant code in layer $l$ directly) but $I_{l,n_{opt}}^{l+1}$, $1 \leq l \leq L - 1$, is non-zero, the optimum code in layer $l$ is the leftmost child of $C_{l,n_{opt}}$ in layer $l$, i.e., $C_{l,2^{l-1}(n-1)+1}$ code (say $C_{l,n_{opt}}$). The coefficients incremented are, (i)

$I_{l,n_{opt}}^{l-1}$ to $I_{l,n_{opt}}^l$, $2^{l-1}$, $1 \leq l \leq L$ and (iii) $I_{l,n_{opt}}^{l-1}$ to $I_{l,n_{opt}}^l$. Also, the coefficients decremented are $I_{l,n_{opt}}^{l+1}$, $2^{l-1}$, $1 \leq l \leq L$.

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

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

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

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

   2.2 Assign code $C_{l,n_{opt}}$ to the new call. Do code assignment and blocking. Update code indices $I_{l,n_{opt}}^n$, $C_{l,n_{opt}}$, $1 \leq l \leq L - 1, l \leq n \leq L - 1$.

   2.3 Go to 1.

Else

   2.1 Reject call.

   2.2 Go to step 1.

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

![Diagram](image)

Figure 3.3: Illustration of multi code topdown scheme (a) scattered approach (b) grouped approach.
3.1.2 MULTI CODE ENHANCEMENT

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

A. MINIMUM RAKES USAGE

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

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

\[
W_1 = kR - 2^{h-1} R
\]  

(3.3)

For non zero \( W_1 \), find \( \max(l_2), 1 \leq l_2 \leq L-2 \) for which condition \( kR - 2^{h-1} R - 2^{h-1} R \geq 0 \) 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

\[
W_t = kR - \sum_{i=1}^{t} 2^{h-1} R
\]  

(3.4)

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

B. MAXIMUM RAKES USAGE

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

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

\[
\overline{R} = [2^{h-1}, 2^{h-1}, ..., 2^{h-1}/2, 2^{h-1}/2, ..., 2^{h-1}] = [2^{h-1}, 2^{h-1}, ..., 2^{h-1}, 2^{h-1}, ..., 2^{h-1}] / 2, ..., 2^{h-1}]
\]

(3.5)

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

\[
\overline{R} = [2^{m-1}, 2^{m-1}, ..., 2^{m-1}]
\]

(3.6)

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

HIGHER RATE SPLITTING FIRST

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

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

C. SCATTERED MULTI CODE SCHEME

The incoming rate \( 2^{l-1} R \) is divided into maximum \( m_1 \), \( m_1 = \min(k,m) \) fractions \( 2^{l-1}, 1 \leq i \leq m_1, 1 \leq l_i \leq l \) 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 $2^i$, find the optimum code as discussed in section 3.1.

The algorithm of scattered multi code scheme is described below.

1. Enter number of rakes $m$.
2. Generate new call.
3. If current used capacity in the tree + $2^{l-1} R \leq$ total tree capacity.
   3.1 Divide $2^{l-1} R$ into $m_1$ rate fractions, $2^{l-1}, 1 \leq i \leq m_1, 1 \leq l \leq l$.
      For $1 \leq i \leq m_1$
      3.1.1 Choose the optimum code from layer $l + 1$.
      3.1.2 Assign this code to rate fraction $2^i$.
      3.1.3 Update code indices.
      End
   3.2 Go to step 2.
Else
   3.1 Reject call.
   3.2 Go to step 1.
End

D. GROUPED MULTI CODE SCHEME

For new call $kR$, find $m_1$ fractions $\sum_{i=1}^{m_1} 2^i = k$. 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 $C_{l, n_{i_{opt}}}$ in layer $l$.

Assign leftmost child of $C_{l, n_{i_{opt}}}$ (i.e. code $C_{l_1, 2n_{i_{opt}} - 1}$) to rate fraction $2^{l-1} R$. If code $C_{l_1, 2n_{i_{opt}} - 1}$ is denoted by $C_{l_1, n_{i_{opt}}}$, the vacant code used for $2^{nd}$ fraction is $C_{l_2, 2^{l-2} n_{i_{opt}} - 2^{l-2} + 1}$. The result can be generalized for $i^{th}$ fraction and the code used is $C_{l, 2^{l-1} - i, n_{i_{opt}} - 2^{l-1} - i + 1}$.  

56
If there is no vacant code in layer \( l \) and at least one vacant code is available in layer \( l-1 \), identify \( C_{h,n_{\text{opt}}} \) and assign this code to the first rate fraction \( 2^{l-1} R \). The vacant area is checked for the remaining rate \( k_1 = (k - 2^{l-1}) R \), \textit{i.e.}, the vacant code is checked in layer \( l \), \( 2^l - k_1 \geq 0 \), \textit{for min}(l'). Again starting from layer \( l' \), the \( m_1 \)-1 rate fractions except 1st rate fractions are assigned codes adjacent to each other. The algorithm can be repeated for third fraction if there is no vacant code in layer \( l-1 \). The code selection may not be optimum but the future multi code use becomes simple.

For illustration of grouped multi code scheme, let a \( 4R \) call arrives with code tree status in Figure 3.1 and the system is equipped with three rakes. Considering maximum rakes use, the three rate fractions are \( 2R \), \( R \), \( R \). The optimum codes using scattered approach are \( C_{2,9} \) (for rate \( 2R \)), \( C_{1,7} \) and \( C_{1,5} \) (for rate \( R \)) as shown in Figure 3.3(a). If grouped multi code scheme is used, the optimum codes are \( C_{2,13} \) (for rate \( 2R \)), \( C_{1,27} \) and \( C_{1,28} \) (for rate \( R \)) as shown in Figure 3.3(b). The scheme requires fewer number of codes searches as compared to most popular UCA[22] scheme as discussed in Appendix B.

### 3.1.3 DYNAMIC CODE ASSIGNMENT ENHANCEMENT

For a new call with rate \( 2^{l-1} R \), the reassignments are required if \( I^l_{l,n_l} = 0, l' \geq l \) and \( \sum_{i=1}^{l-1} I^l_{l_i,n_i} \times 2^{l_i-1} \geq 2^{l-1} \). In DCA [25], the definition of optimum code is different.

The code \( C_{l,n_l} \) is in the path from root to optimum code if \( I^l_{l,n_l} = 0, 1 \leq l' \leq l-1 \) is maximum, \textit{i.e.}, the number of vacant children in layer \( 1 \) to \( l-1 \) for code \( C_{l,n_l} \) is highest. The algorithm does not intend to find the crowded part rather it finds a blocked code with capacity \( 2^{l-1} R \), which has least number of busy children. After selecting the best blocked code, all the calls handled by its busy children have to be shifted to appropriate (crowded) locations using top down scheme.

When all calls are shifted the code becomes available for \( 2^{l-1} R \) call. The DCA algorithm along with top down scheme works as follows

1. Generate new call of rate \( 2^{l-1} R \).
2. \textit{If current used capacity in the tree} + \( 2^{l-1} R \leq \text{total tree capacity} \).
Figure 3.4: Comparison of number of code searches in single code schemes for distribution: (a) [20,20,20,20,20] (b) [40,30,10,10,10]
Figure 3.5: Comparison of number of code searches in single code schemes for distribution: (a) [10,10,10,30,40], (b) [10,30,20,30,10].
Figure 3.6: Comparison of Code Blocking Probability in single code schemes for distribution: (a) [20,20,20,20,20] (b) [40,30,10,10,10]
Figure 3.7: Comparison of Code Blocking Probability in single code schemes for distribution: (a) [10,10,10,30,40] (b) [10,30,20,30,10]
2.1 Choose the optimum code in layer \( I^n \), for which \( I^l_{p_{mn}}, 1 \leq l^n \leq l \) is maximum.

2.2 If (the optimum code has at least one blocked child in layer \( I \))

2.2.1 \( I^n = I^n - 1 \).

2.2.2 If \( I^n = I \):

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

else

- Go to step 2.1.

end

2.2.3 Go to step 2.1.

else

2.2.1 Sibling code is the optimum code.

2.2.2 Go to step 2.2.

end

else

2.1 Reject call.

2.2 Go to step 1.

End

3.2 Results

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

**Figure 3.8:** Comparison of number of code searches in multi code schemes for uniform distribution, (ii) $[40,30,10,10,10]$, low rates call dominating. The proposed top down (TD) code selection scheme is compared with crowded first assignment $[22]$ (CFA), fixed set
partitioning [27] (FSP), leftmost code assignment [22] (LCA), recursive fewer code blocked (RFCB).

3.2.1 SINGLE CODE ASSIGNMENT

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

![Graph showing code blocking probability vs average traffic load]

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

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

3.2.2 MULTI CODE ASSIGNMENT

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

$$CS_{MC\_TDS} < CS_{MC\_LCAG} < CS_{MC\_TDS} < CS_{MC\_LCAS} < CS_{MC\_CFAG} < CS_{MC\_CFAG}$$ (3.7)

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

$$BP_{MC\_TDS} < BP_{MC\_CFAS} < BP_{MC\_TDS} < BP_{MC\_CFAG} < BP_{MC\_LCAS} < CS_{MC\_LCAG}$$ (3.8)

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

3.3 CONCLUSION

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