## Reducing Wastage Capacity in OVSF Based CDMA Networks using Dynamic Rake Combiners

DAVINDER S. SAINI Department of Electronics and Communication Engineering Jaypee University of Information Technology Waknaghat, Distt. Solan.(H.P.) -173215 INDIA davinder.saini@juit.ac.in

*Abstract:*-Orthogonal variable spreading factor (OVSF) codes in CDMA networks are designed to handle quantized data rates. Handling non-quantized data rates in such networks leads to code capacity wastage if traditional single code assignment is used. Although, the use of multiple codes reduce this wastage capacity but the use of large number of fixed rake combiners per call increase cost and complexity of the system. I propose zero wastage designs in which the rake combiner's usage is made dynamic and the amount of rake combiners used depend upon the rate type, with more combiners given to the rate which deviates significantly from the quantized one. The average number of rakes per call is arbitrarily assumed, and if the rakes used for a particular call are less than the average (which happens for quantized or near quantized calls), the unused rakes can be used by future calls. The performance is significantly improved compared to the fixed rate systems. The amount of codes used on average is less than the codes required for existing multi code designs. In the reduced wastage capacity design, which is a special case of zero wastage designs, some wastage can be tolerated for simplicity and less equipment cost.

Key Words:- OVSF codes, CDMA, wastage capacity, code assignment, quantized and non-quantized rates.

## **1** Introduction

In CDMA based wireless networks, variable rates are handled by OVSF codes. The number of codes used per call depends upon the traffic type. Few years back, one code was used from the code tree for each call. The use of single code for new call may produce code blocking due to following reasons: 1) the scattering of vacant codes in the code tree; 2) quantized nature of rate handling capability of codes. The scattering of vacant codes creates external fragmentation problem [1], and can be avoided by the efficient single code assignment designs. The quantized nature of code capacity produces internal fragmentation [1], which cannot be eliminated by single code designs. For example, a new user with rate 17R (R is 7.5 kbps for WCDMA system), requires code with capacity 32R producing heavy wastage capacity 15R (32R-17R), which is 46% of 32R capacity code. To eliminate code capacity wastage, multiple codes are used for calls. A large number of multi code designs are available in literature, but none

of them eliminate code wastage completely. We propose multi code designs to eliminate code wastage capacity completely. These designs utilize variable number of rake combiners depending upon rate/traffic type.

OVSF codes are generated from binary tree structure given in [2]. Two different codes of the tree can be used only, if they don't have parent child relationship. This is due to the orthogonal property of the OVSF codes, which states that the two codes are orthogonal if, they does not appear in the same branch from root to a specific child. The maximum capacity of the code tree is equal to root code capacity. The OVSF codes suffer from the drawback of code blocking, which avoids utilization of full code tree capacity for new calls. To illustrate the code blocking, consider a 7 layer OVSF tree shown in Fig. 1, with busy codes  $C_{1,12}$ ,  $C_{1,56}$ ,  $C_{1,64}$ ,  $C_{2,2}$ ,  $C_{3,6}$  and  $C_{5,3}$ , which makes used code tree capacity equal to 25R. The code  $C_{ln}$  represents code in layer l with id  $n_l$ . The remaining capacity 64R-25R=39R is still available for

future calls. If a call with rate 16R arrives, and if the single code assignment facility is assumed, the call

selection among candidate codes. The design in [9] assigns new call to a vacant code whose neighbor is



Fig.1 A 7 layer OVSF code tree with maximum capacity of 64R and the used capacity of 25R

cannot be handled due to absence of 16R vacant code in the code tree, although the system has vacant capacity of amount 39R available. This is code blocking discussed earlier, and happens because the ancestors and descendants of a busy code are blocked and cannot be used by new calls.

A large number of single code and multi code assignment designs exist in literature. In left code assignment (LCA) [3], the new call is assigned a code from left side of the code tree. In crowded first assignment (CFA) [3], the optimum code lies in the most crowded area of the code tree. Though the CFA design is complex, but still this is most common single code design, as it reduces code blocking significantly. In fixed set partitioning (FSP) [4], the entire tree is partitioned into fixed portions and each portion is reserved for separate call rate class. Therefore, the number of partitions is equal to the number of classes. In dynamic code assignment (DCA) [5], the code reassignments are done in such a way that full code tree capacity is utilized. The optimum code is the one which require least code reassignments. The computationally efficient dynamic code assignment with call admission control (DCA-CAC) [6] reduces the complexity of traditional DCA further using two different ways: 1) Total resources are divided into number of mutually exclusive groups, with the numbers of groups equal to number of call arrival classes, 2) By deliberate rejection of those calls which may produce large code blocking for future higher rate calls. The fewer codes blocked (FCB) design presented in [7], selects a vacant code for assignment, which results in minimum number of parents blocked, that were not blocked previously. The recursive fewer codes blocked (RFCB) [8] design runs on top of FCB, with the additional benefit of resolving ties in amount of codes blocked, by ordered

handling the latest call in a particular layer, i.e., all the calls coming at almost similar times are grouped, so that this area become vacant in almost similar time. The multi code assignment schemes reduce rate wastage and code fragmentation. In multiple leaf code reservation scheme (MLCR) [10], the bandwidth reservation is done in advance for users, and the amount of codes reserved depends upon the user bandwidth. Further the bandwidth (code) reservation is temporary and codes are made unreserved after call admission process. The MLCR scheme provides fairness to the users with different bandwidth requirements. The multi code design in [11] finds the most suitable multi code combination required for a new call. The use of multiple codes results in more fragmented tree when the call is terminated, which leads to more code blocking of higher rates. The multirate multicode compact assignment (MMCA) [12] 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 design supports multirate real-time calls and keeps the code tree as flexible as possible while accepting new multirate calls. The multi code scheme [13] derives the optimal code under the constraints of allocated code amount and maximal resource wastage ratio. It gives superior performance using two and three codes in a multi code with a crowded-group-first strategy. The code utilization and blocking benefits are significant for a resource wastage ratio of 40%. The time based code assignment [14] explains the impact of remaining time for reducing code blocking. All above multi code designs try to use maximum rake combiners to due to the external fragmentation problem. The zero wastage designs proposed in our paper uses these rakes dynamically. The mobile station (MS) or base station (BS) requires more rake combiners than the existing systems, but the cost and complexity is not increased because the number of rake combiners utilized on average is not higher than the existing designs although the MS or BS can have more rake combiners available. Each call is handled by minimum rakes, and if a specific call requires fewer rakes than a specified number, the pending rakes (maximum rakes-used rakes) can be carried results are given in section 4, and the paper is concluded in section 5.

## 2 OVSF data rate types 2.1 Ouantized Rates

The rates in the form  $2^{l-1}R, 1 \le l \le L$  are called quantized rates, and can be handled by single code (rake) if the vacant code with capacity  $2^{l-1}R$  is

if (at least one vacant code with capacity 
$$2^{i+R}$$
 is available) "if 1"  
-Assign vacant code using CFA algorithm.  
else  
-Convert  $2^{l-1}R$  rate call into two equal rate fractions, namely,  $2^{l-2}R$  (say capacity  $P_1^2$ ), and  $2^{l-2}R$   
(capacity  $P_2^2$ ), where  $p_i^l$  denotes  $i^h$  capacity fraction in total  $j$  fractions.  
if  $(r \ge 2)$  "if  $2^n$   
if (at least 2 vacant codes with capacities  $P_1^2$  and  $P_2^2$  available) "if  $3^n$   
-Assign vacant codes for call portion  $P_1^2$  and  $P_2^2$  using CFA design.  
else  
-Divide fraction  $P_1^2$  into two equal fractions  $P_1^{2/2}$  and  $P_1^{2/2}$ . The three capacity portions, namely  
 $p_2^2, P_1^{2/2}$ , and  $P_1^{2/2}$  can be denoted by  $P_1^3$ ,  $1 \le i \le 3$ .  
if  $(r \ge 3)$  "if  $4^n$ "  
if (three vacant codes with capacities  $P_1^3$ ,  $1 \le i \le 3$  are available) "if  $5^n$   
-Assign vacant codes for call portion  $P_1^3$ ,  $1 \le i \le 3$  using CFA design.  
-Repeat the algorithm, till  $r$  capacity portions. In every step, split the fraction with highest  
capacity into two equal rate fractions. If  $r$  vacant capacity portions are not available, reject  
the call.  
endif 5  
endif 4  
endif 5  
endif 4  
Fig.2 Call handling with minimum rakes

forward to the next call which may not be handled otherwise due to insufficient rake combiners. For the worst case, all the rake combiners available at MS are utilized, and if the system is dominated by nonquantized rates (and specifically *rake consuming rates* which will be defined later), the wastage capacity can be completely eliminated.

The paper is organized as follows. Section 2 discusses various OVSF data rate types. Section 3 explains the proposed multi code designs. Simulation

available. The multi code assignment provides the two additional benefits while handling quantized rates: 1) Provide fairness to users with different call rates; 2) Increase the interference suppression. The higher spreading factor (*SF*) of a call increases the interference suppression capability especially due to burst errors. For a quantized call with rate  $2^{l-1}R$ , if the system is equipped with *r* rakes, there are three ways to handle the call.

#### 2.1.1 Call handling with minimum codes

The codes used for new call are least. The design is simple and cost effective, but it may produce wastage and blocking because the low rate codes are not better utilized. The call handling algorithm is given in Fig. 2.

## 2.2.2. Call handling with maximum codes

The algorithm is costly, but the blocking and wastage is significantly less than the design using minimum codes. The call handling algorithm is given in Fig. 3.

Due to maximum rakes use, the design provides least

```
- Find max(i), 1 \le i \le r, for which \sum_{i=1}^{i} 2^{l_j} = 2^l.
if (vacant codes for fractions 2^{l_j}, 1 \le j \le r available)
    -Use vacant codes with capacities 2^{l_j}, 1 \le j \le r as
      per CFA algorithm.
else
     -Reject call
endif
  Fig.3 Call handling with maximum rakes
```

The algorithm works in such a way that no single rate type is over served or under served. The design is illustrated in Fig. 4.

Convert rate  $2^{l-1}R$  into r fractions with where r-2 fractions are with capacity  $2^{l-i}R, 2 \le i \le r-1$  and two fractions with capacity  $2^{l-r} R$  each.

## 2.2 Non-quantized Rates

As mentioned earlier, the rates not in the form of  $2^{l-1}R$  are called non-quantized rates. For a new call kR,  $2^{m-2} \le k \le 2^{m-1}$ , define wastage capacity for single rake system as

$$WC_{k,1} = (2^{m-1} - k)/2^{m-1}$$
 (1)

In (1),  $WC_{k1}$  represents the fraction of single code capacity (with total capacity  $2^m R$ ) wasted in handling kR rate user. Some of the properties of wastage capacity are given as

For input rate kR, if wastage capacity is zero for rrake system, i.e.  $WC_{k,r}=0$ , then it is guaranteed that

$$WC_{k+2^{l} r+l} = 0, \ 1 \le l \le L-1$$
(2)

In general,

$$WC_{k+\sum_{l=1}^{L}2^{l},r+L_{l}} = 0$$
(3)

For input rate kR, if wastage capacity is non zero and can be expressed as

-For smallest integer  $m, m \leq r$ , find m fractions  $\sum_{i=1}^{m} 2^{l_i - 1} = k$ . Let  $r_i$  denotes minimum number of rakes to handle rate fraction  $2^{l_i-1}$ if  $\left(\sum_{i=1}^{m} r_{i} \leq r\right)$ -All the fractions can be handled by vacant codes (rakes), and the rate fractions are handled as per

discussion given in section 2.1.1.

-The spare rakes  $r - \sum_{i=1}^{m} r_i$  can be utilized if multi code is intended to use maximum codes

else

-Reject the call because sufficient rakes are not available endif

Fig. 4 Call Handling with optimum fairness

code scattering, and hence future availability of high rate codes is maximum.

#### 2.1.2 Call handling with optimum fairness

 $WC_{k,r} = p/q, 1 < r < L-1, 1 < k < 2^{L-1}$ , then  $WC_{k+2^{l},r+1} = p/(q+2^{l}), \ 2^{l} \ge k$ (4)Similarly for integer *i*, and  $1 \le l_i \le L - 1$ , И

$$VC_{k+\sum_{l=1}^{s}2^{l_{l}}, r+s} = p/[q+\sum_{l=1}^{s}2^{l_{l}}], \ \sum_{i=1}^{s}2^{l_{i}} \ge k$$
(5)

The wastage capacity in WCDMA networks for all possible data rates and rakes is illustrated in Table A.1 given in appendix A. The algorithm to handle non-

Considering *r* rakes in the system, if a call  $k_1R$  arrives, let the minimum possible rakes required are  $r_1$ . If  $r_1 \le r$ , the call  $k_1R$  can be handled with  $r_1$  rakes

-For smallest integer  $m, m \le r$ , find m fractions  $\sum_{i=1}^{m} 2^{l_i - 1} = k$ . Let  $r_i$  denotes minimum number of rakes to handle rate fraction  $2^{l_i - 1}$ .

if  $\left(\sum_{i=1}^{m} r_i \leq r\right)$ 

-All the fractions can be handled by vacant codes (rakes), and the rate fractions are handled as per discussion given in section 2.1.1.

-The spare rakes  $r - \sum_{i=1}^{m} r_i$  can be utilized if multi code is intended to use maximum codes

else

-Reject the call because sufficient rakes are not available endif

Fig. 5 Non- quantized rates handling

quantized rate is given in Fig. 5.

## 2.3 Rake Consuming Rates (RCR)

For r rake system, all quantized and nonquantized rates which cannot be handled by r codes (rakes) are called Rake Consuming Rates (RCR). More specifically, the quantized rate  $2^{l-1}R$  is called RCR if the system does not have: 1) one vacant code with capacity  $2^{l-1}R$  or; 2) two vacant codes with capacities  $2^{l-1}R$ , and  $2^{l-1}R$ , or continuing maximum r steps; 3) the system does not have 'r' rakes with capacities  $2^{l-2} R \cdot 2^{l-3} R \cdot \dots \cdot 2^{l-r} R \cdot 2^{l-r} R$ . Similarly, the non-quantized rate  $kR, k \neq 2^{l-1}$  is in the category of *RCR* if  $\sum_{i=1}^{r} 2^{l_i-1} \neq k$ , for all possible values of  $l_i \in [1, L]$ . All quantized and non-quantized rates which can be handled by r rake system are called non rake consuming rates (NRCR). Out of various rate categories, RCR rates are least dealt in literature. The RCR rates require additional rakes than r existing rakes. Our multi code design aims to reduce/nullify wastage capacity when the system is dominated by RCR rates.

# 3 Multi code designs

## 3.1 Zero Wastage Design I



Fig. 6 Flowchart for zero wastage design I

and the unused rakes  $r \cdot r_1$  are carried forward to next call  $k_2R$ . Therefore the maximum rakes available to handle second call  $k_2R$  are  $r_2^{\max} = r + (r \cdot r_1)$ . If this call requires  $r_2$  rakes, the call can be handled without wastage if  $r_2 \le r + (r - r_1)$ . Similarly, for  $i^{\text{th}}$  call  $k_iR$ , the maximum rakes which can be utilized are  $r_i^{\max} = i \times r - \sum_{i=1}^{i-1} r_i$  (6)

If the rakes used by  $i^{\text{th}}$  call are  $r_i$ , the rakes carried

is one, and is  $C_{2,1}$ , as shown in Fig. 7. The identifier <u>1</u> around code  $C_{2,1}$  signifies handling of 1<sup>st</sup> call, and the optimum codes  $C_{2,1}$  is selected according to CFA [3] design. Similarly, identifier <u>x</u><sub>y</sub> around a code represents that the code is handling *yR* rate fraction of call *x*. The balance rakes available after handling 1<sup>st</sup> call are 3-1= 2, and are carried forward to handle the 2<sup>nd</sup> call. The next call is 16*R*, which can use 5 (2 pending rakes and 3 regular rakes). Considering



Fig. 7 The updated code tree status for Fig. 1 using design I

forward to handle  $(i+1)^{\text{th}}$  call are  $r_i^{\text{max}} - r_i$ . Therefore,

minimum rakes usage again, the call is handled by 3 rakes (8+4+4), which are represented by  $2_{8}$ ,  $2_{4}$ , and

Arrival	Zero wast	tage desig	gn I	Zero wasta	ige design	П	Reduced wastage design				
Kate	Pending rakes (say 0 initially) + r		Rakes carried forward	Pending rakes (say 0 initially) + <i>r</i>	Rakes used	Rakes carried forward	Pending rakes (say 0 initially) + r	Rakes used	es Rakes d carried forward		
2R	3	1	2	3	1	2	3	1	2		
16 <i>R</i>	5	3	2	5	3	2	5	3	0		
15 <i>R</i>	5	5	0	5	5	0	3	Call	rejected		
1 <i>R</i>	3	1	2	3	1	2	3	1	2		
1 <i>R</i>	5	1	4	5	1	3	5	1	2		
1 <i>R</i>	7	1	6	6	1	3	5	1	2		
2R	9	1	8	6	1	3	5	1	2		
1R	11	1	10	6	1	3	5	1	2		

Table 1 Rakes usage for examples in Fig. 3 and Fig. 4 in zero wastage and reduced wastage designs

in this design the unused rakes of previous calls are utilized by current call. The design always produce zero wastage if r >>1. Even, for nominal value of r(say L/2), the algorithm produces zero wastage. The design is particularly useful if the system has large rake consuming rates. The flowchart of the design is shown in Fig. 6, illustrating the procedure to handle  $i^{th}$ new call. For illustration of zero wastage designs I, consider a 7 layer code tree shown in Fig. 1. Let the calls arrive in pattern 2R, 16R, 15R, R, R, R, 2R and R. The system is assumed to have 3 rakes. Starting with the first call 2R, the minimum rakes (codes) used  $\underline{2}_4$ . The relationship in pending rakes, rakes used and rakes carried forward for all calls is given in Table 1. For last two calls 2R and R, the rakes carried forward are 8 and 10, making total rakes available for these calls 11 and 14, which exceeds the maximum rakes required by any call which is L (L is 7 for the example assumed). The status of the code tree after handling all calls is shown in Fig. 7.

## 3.2 Zero Wastage Design II



Fig. 8 The updated code tree status for Fig. 1 using reduced wastage design

For an *L* layer code tree, the maximum rakes required to produce zero wastage are *L*-1. The design I keep on adding pending rakes even when the total rakes available go beyond *L*-1. In this design, the number of rakes carried forward can have two possible values: 1) if the pending rakes are  $i \times r - \sum_{j=1}^{i-1} r_j \ge L - m - 1$ , the rakes carried forward to  $(i+1)^{\text{th}}$  call are *L*-*m*-1; 2) if the pending rakes are  $i \times r - \sum_{j=1}^{i-1} r_j < L - m - 1$ , the rakes carried forward to the *i*<sup>th</sup> call is  $i \times r - \sum_{j=1}^{i-1} r_j$ .

Considering the call pattern similar to the one assumed for design I, in zero wastage design II, maximum L-1 (equal to 6) rakes are carried forward to handle next call. The procedure is identical to design till 5<sup>th</sup> call. The number of rakes carried forward cannot be more than 3 (making pending rakes for new call 6). Hence the pending rakes for all calls after 4<sup>th</sup> call are different compared to design I as shown in Table 1.

#### 3.3 Reduced Wastage Design

In this design, for a call kR, if minimum rakes required are  $r_1$ , the rakes carried forward to the second call are r- $r_1$ . If second call requires minimum  $r_2$  rakes, there are two possible value for amount of rakes carried forward: 1) if  $r_2 \le r$ , the rakes carried forward to third call are r- $r_2$ , i.e., all previous pending rakes are discarded; 2) if  $r_2 > r$ , the maximum rakes used are  $r + r_1$ ,  $r_1' \le r - r_1$ . The rakes carried forward are  $r - r_1 - r_1'$ . In general, for  $i^{th}$  call, if  $r_i \le r$  arrival the rakes carried forward are r- $r_i - r_{i-1}'$ . The design is simple and cost effective because it requires only previous call rakes information. For illustration of the reduced wastage design, consider the call arrival pattern similar to the one used



Fig. 9 Flowchart of the proposed multi code scheme

for design I and II, and 3 rakes availability in Fig. 1. The first call with rate 2R is handled with one rake as represented by <u>1</u> in Fig. 8, and the balance 2 rakes

will be carried further to handle the next call. The next call of rate 16R has a maximum of 5 rakes as discussed earlier. Using minimum rakes, the call is handled by 3 codes with rate 8R, 4R, 4R which are  $2_8$ ,  $2_4$  and  $2_4$ . As all the three rakes are consumed for 16R call, the number of rakes carried forward is zero. The next call of rate 15R requires 4 rakes, and hence is rejected. The next call of rate R now has its own 3 rake quota available. So the call will be handled by 1 rake, which is represented by 4. Similarly, all the remaining calls will be handled by codes represented by 5, 6, 7 and 8. The updated status of the tree is shown in Fig. 8.

## 3.4 Fair multi code design

The multi code design consists of the three steps: 1) identify the available codes in each layer of the tree; 2) the layers are arranged in descending order of number of vacant codes; 3) use the code which has  $2^{l_{i}-1} + 2^{l_{2}-1} = k$ , procedure stops, otherwise the procedure is repeated to maximum (*r*-1) times. After (*r*-1) steps, the fraction of rate *kR* handled is  $\sum_{i=1}^{r-1} 2^{l_{i}-1}$ . If we define,  $m=k-\sum_{i=1}^{r-1} 2^{l_{i}-1}$ , find  $min(j) | m \le 2^{j-1}$ . The rate fraction  $2^{j-1}R$  will be handled by  $r^{th}$  rake.

The flowchart for the fair multi code design is given in Fig. 9. For illustration of fair multi code design, consider OVSF code tree in Fig. 1 with N=[3,4,5,1,0,0,0], If a new call of 16*R* arrives and the system is equipped with 4 rakes, the combinations which can be used in the mentioned code tree are [8*R*,4*R*,4*R*] and [4*R*,4*R*,4*R*,4*R*] respectively. In both combinations, for each fraction, the algorithm recursively searches for a layer with maximum number of vacant codes. The value of *N* after utilizing combination I and II is [3,4,3,0,0,0,0] and [3,4,1,1,0,0,0] is preferred as it provides fair distribution of codes for future calls. Using CFA



Fig. 10 The updated code tree status for Fig. 1 using fair multi code design

maximum availability. If the call is not quantized, repeat above two steps until all rate fractions are handled.

For a new call kR, define vacant code vector N giving vacant codes in each layer, i.e.,  $N = [N_1, N_2, \dots, N_L]$ , where  $N_l$  is the number of vacant codes available in  $l^{th}$  layer. A vacant code in layer l is included in  $N_l$  if all its ancestors are blocked. Arrange coefficients  $N_l$ ,  $1 \le l \le L$  in descending order. Assuming that in the first attempt, the coefficient  $N_{l_1}$  is largest, the first rake handles rate fraction  $2^{l_1-1}R$  and decrement  $N_{l_1}$  by 1. The remaining capacity to be handled by (r-1) rakes is  $(2^{l-1} - 2^{l_1-1})R$ , and again all the coefficients are arranged in descending order. If  $N_{l_2}$  is largest in second attempt, the vacant code in  $l_2$  layer will handle rate fraction  $2^{l_2-1}R$ . If

design, the optimum codes used are  $C_{4,4}$ ,  $C_{3,2}$  and  $C_{3,4}$  respectively as shown in Fig. 10.

## **4** Numerical Results

For simulation, five classes of rates namely, R (7.5*kbps*), 2R, 4R, 8R and 16R are considered. The arrival rate for each of the  $l^{\text{th}}$  class (denoted by  $\lambda_l$ ) is assumed to vary between 1 and 4 calls per unit of time, and the call duration  $(1/\mu)$  is assumed to be 1 unit of time for all classes. If the traffic load for  $l^{th}$  class is defined as  $\lambda_l/\mu$ , the wastage capacity and other performances are compared for variable average traffic load  $\lambda/\mu$ , where,  $\lambda = \sum_{l=1}^{5} \lambda_l$ . If for a layer l,  $R_l^l$  denotes the rate of  $i^{th}$  new call (quantized or non-quantized), and  $M_l^l$  denotes the sum of capacities of

all used codes for this call, the wastage capacity can be rewritten as

$$W_i^{l} = (M_i^{l} - R_i^{l}) / R_i^{l}$$
<sup>(7)</sup>

where  $M_i^l = \sum_{j=1}^{r_i^l} C_{i,j}^l$ , and  $r_i^l$  is the number of rakes used, and the identifier  $C_{i,j}^l$  represents capacity of  $j^{th}$ code in the multi code  $M_i^l$ . Considering  $N_l$  calls for layer l, the total wastage capacity for L layer system is



Fig. 11 Comparison of wastage capacity for distributions, (*a*) *RCR* rates=70%, *NRCR* rates=30%, (*b*) *RCR* rates=30%, *NRCR* rates=70%.

given by  $W = \sum_{l=1}^{L} \sum_{i=1}^{N_l} W_i^{l} = \sum_{l=1}^{L} \sum_{i=1}^{N_l} (M_i^{l} - R_i^{l}) / R_i^{l}$ (8)

The first performance parameter considered is the code wastage capacity, and the comparison for the proposed reduced wastage design (represented by RWG) is done with multicode fragmentation (FRG) [1], multiple leaf code reservation (MLCR) [10], and multicode multirate compact assignment (MMCA) [12] designs discussed in section 1. The results are plotted in Fig. 11(a) and 11(b), in terms of rake consuming and non rake consuming rates only. Two distributions of *RCR* and *NRCR* rates are considered: 1) the probability of *RCR* and *NRCR* rates arrival is

![](_page_8_Figure_10.jpeg)

Fig. 12 Comparison of ratio of maximum blocking to average blocking for distribution, *RCR* rates=70%, *NRCR* rates=30%.

70% and 30%; 2) the probability of *RCR* and *NRCR* rates arrival is 30% and 70%. In Fig. 11, the symbol *des-n*, represents *n* rakes availability in design *des*, e.g., FRG-3, represents the fragmentation design with 3 rakes availability. The wastage capacity results Fig. 11(*a*) and 11(*b*) shows that the wastage capacity in the proposed design is significantly less than the existing alternatives. Further, comparing Fig. 11(*a*) and 11(*b*), the wastage capacity in the system dominated by *RCR* rates scenario is slightly more than the one with dominating *NRCR* rates scenario. Also, the wastage capacity reduces as the number of rakes in the system increases. The zero wastage design results are not plotted because they produce zero wastage.

The fairness in handling various rate users is also plotted in Fig. 12 for proposed fair multi code design (FMCD) with above mentioned schemes when the system has three rakes. The parameter maximum\_blocking/average\_blocking is used for fairness comparisons, and for perfect fair system, this parameter should have unit value. The results are plotted for the rate distribution *RCR*=70%, and *NRCR*=30%, which clearly shows that the proposed FMCD design is, by far, the most fair design.

## 5. Conclusion

The use of multiple codes in OVSF based CDMA always gives better results in handling non-quantized rates. Traditional multi code designs use either minimum rakes or maximum rakes for new rates. The use of more codes increases cost and complexity. The proposed multi code design uses the balanced rakes of quantized or closely quantized calls to handle nonquantized calls designated as rake consuming rates in the paper. Only few extra channels need to be carried forward to new calls. The wastage capacity is drastically reduced giving reduction in number of calls rejected. The increase in complexity in this zero blocking design can be reduced by suboptimal design with little compromise in increase of wastage capacity. Work can be done to make rakes carried forward adaptive to the call arrival distribution.

## **References:**

- 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 Communications*, Vol. 4, Jul. 2005, pp. 1516-1526,
- [2] F. Adachi, M. Sawahashi, and K. Okawa, Treestructured generation of orthogonal spreading codes with different lengths for forward link of DS-CDMA mobile radio, *Electronic Letters*, Vol. 33, No. 1, Jan. 1997, pp. 27–28.
- [3] 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, 2002, pp. 293–302.
- [4] J. S. Park and D. C. Lee, Enhanced fixed and dynamic code assignment policies for OVSF-CDMA systems, *Proc. ICWN*, Jun. 2003, pp. 620–625.
- [5] T. Minn and K. Y. Siu, Dynamic assignment of orthogonal variable spreading factor codes in W-CDMA, *IEEE Journal on Selected Areas in Communications*, Vol. 18, No. 8, Aug. 1998, pp. 1429–1440.
- [6] J. S. Park, L. Huang, and C. 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. 54, No. 1, Jan. 2008, pp. 286–296.

- [7] A. Rouskas and D. Skoutas, OVSF code assignment and reassignment at the forward link of W-CDMA 3G systems, in *Proc. IEEE PIMRC*, Sept.2002, pp.2404-2408.
- [8] A. N. Rouskas and D. N. Skoutas, Management of channelization codes at the forward link of WCDMA, *IEEE Commun. Lett.*, vol. 9, Aug.2005, pp. 679-681.
- [9] V. Balyan and D. S. Saini, Call Elapsed Time and reduction in code blocking for WCDMA Networks, in *Proc. Softcom* 2009, Croatia, Sept. 2009, pp. 1-5.
- [10] F. A. Cruz-Perez, J. L. Vazquez-Avila, A. Seguin-Jimenez, and L. Ortigoza-Guerrero, Call Admission and Code Allocation Strategies for WCDMA Systems With Multirate Traffic, *IEEE Journal on Selected Areas on Communications*, Vol.24, Jan. 2006, pp. 26-35.
- [11] D. S. Saini and M. Upadhyay, Multiple rake combiners and performance improvement in WCDMA systems, *IEEE Transactions on Vehicular Technology*, Vol. 58, No. 7, Sep. 2009, pp. 3361-3370.
- [12] 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, Nov. 2005, pp. 2114-2124.
- [13] M. X. Chen, Efficient integration OVSF code management architecture in UMTS, *Elsevier Computer Communications*, Vol. 31, No. 9, pp. 3103–3112, Sep. 2008.
- [14] 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, Nov./Dec. 2005, pp. 604-615.
- [15] C. M. Chao, OVSF Code Assignment Strategies with Minimal Fragmentations for WCDMA Systems, *Computer Networks*, Vol. 52, No. 12, Aug. 2008, pp. 2331-2343.

Appendix A

8									S							
Rako	N=1	2	3	4	5	6	7		3ak	1	2	3	4	5	6	7
Rate (Rbps)									Rate (Rbps)							
1	0	0	0	0	0	0	0		65	63/128	0	0	0	0	0	0
2	0	Ő	Ő	Ő	0	0	0		66	62/128	Ő	0	Ő	0	0	0
3	1/4	0	0	0	0	0	0		67	61/128	1/68	0	0	0	0	0
4	0	0	0	0	0	0	0		68	60/128	0	0	0	0	0	0
5	3/8	0	0	0	0	0	0	_	69	59/128	3/72	0	0	0	0	0
7	1/8	1/8	0	0	0	0	0		70	57/128	1/72	1/72	0	0	0	0
8	0	0	0	0	0	0	0		72	56/128	0	0	Ő	0	Õ	0
9	7/16	0	0	0	0	0	0		73	55/128	7/80	0	0	0	0	0
10	6/16	0	0	0	0	0	0		74	54/128	6/80	0	0	0	0	0
11	5/16	1/12	0	0	0	0	0	_	75	53/128	5/80	1/76	0	0	0	0
13	3/16	3/16	0	0	0	0	0	_	70	51/128	3/80	3/80	0	0	0	0
14	2/16	2/16	0	0	0	0	0		78	50/128	2/80	2/80	0	0	0	0
15	1/16	1/16	1/16	0	0	0	0		79	49/128	1/80	1/80	1/80	0	0	0
16	0	0	0	0	0	0	0		80	48/128	0	0	0	0	0	0
17	15/32	0	0	0	0	0	0		81	47/128	15/96	0	0	0	0	0
18	14/32	1/20	0	0	0	0	0	_	82	46/128	12/06	1/84	0	0	0	0
20	12/32	0	0	0	0	0	0	_	84	43/128	12/96	0	0	0	0	0
20	11/32	3/24	0	0	0	0	0		85	43/128	11/96	3/88	0	0	0	0
22	10/32	2/24	0	0	0	0	0		86	42/128	10/96	2/88	0	0	0	0
23	9/32	1/24	1/24	0	0	0	0		87	41/128	9/96	1/88	1/88	0	0	0
24	8/32	0	0	0	0	0	0		88	40/128	8/96	0	0	0	0	0
25	6/22	6/22	0	0	0	0	0	_	89	39/128	6/96	6/96	0	0	0	0
20	5/32	5/32	1/28	0	0	0	0	_	90	37/128	5/96	5/96	1/92	0	0	0
28	4/32	4/32	0	0	0	0	0		92	36/128	4/96	4/96	0	0	0	0
29	3/32	3/32	3/32	0	0	0	0		93	35/128	3/96	3/96	3/96	0	0	0
30	2/32	2/32	2/32	0	0	0	0		94	34/128	2/96	2/96	2/96	0	0	0
31	1/32	1/32	1/32	1/32	0	0	0		95	33/128	1/96	1/96	1/96	1/96	0	0
32	0	0	0	0	0	0	0	_	96	32/128	0	0	0	0	0	0
34	30/64	0	0	0	0	0	0	_	97	30/128	30/128	0	0	00	0	0
35	29/64	1/36	0	0	0	0	0		99	29/128	29/128	1/100	0	0	0	0
36	28/64	0	0	0	0	0	0		100	28/128	28/128	0	0	0	0	0
37	27/64	3/40	0	0	0	0	0		101	27/128	27/128	3/104	0	0	0	0
38	26/64	2/40	0	0	0	0	0		102	26/128	26/128	2/104	0	0	0	0
39	25/64	1/40	1/40	0	0	0	0	_	103	25/128	25/128	1/104	1/104	0	0	0
40	23/64	7/48	0	0	0	0	0	_	104	23/128	23/128	7/112	0	0	0	0
42	22/64	6/48	0	0	0	0	0		105	22/128	22/128	6/112	0	0	0	0
43	21/64	5/48	1/44	0	0	0	0		107	21/128	21/128	5/112	1/108	0	0	0
44	20/64	4/48	0	0	0	0	0		108	20/128	20/128	4/112	0	0	0	0
45	19/64	3/48	3/48	0	0	0	0		109	19/128	19/128	3/112	3/112	0	0	0
40	10/04	2/48	2/48	1/48	0	0	0		110	10/128	10/128	1/112	1/112	1/112	0	0
48	16/64	0	0	0	0	0	0		112	16/128	16/128	0	0	0	0	0
49	15/64	15/64	0	Ő	Ő	Ő	Ő		113	15/128	15/128	15/128	Ő	Ŏ	0	Ő
50	14/64	14/64	0	0	0	0	0		114	14/128	14/128	14/128	0	0	0	0
51	13/64	13/64	1/52	0	0	0	0		115	13/128	13/128	13/128	1/116	0	0	0
52	12/64	12/64	0	0	0	0	0		116	12/128	12/128	12/128	0	0	0	0
55	11/64	11/64	3/30	0	0	0	0		11/	10/128	10/128	10/128	2/120	0	0	0
55	9/64	9/64	1/56	1/56	0	0	0		119	9/128	9/128	9/128	1/120	1/120	0	0
56	8/64	8/64	0	0	0	0	0		120	8/128	8/128	8/128	0	0	0	0
57	7/64	7/64	7/64	0	0	0	0		121	7/128	7/128	7/128	7/128	0	0	0
58	6/64	6/64	6/64	0	0	0	0		122	6/128	6/128	6/128	6/128	0	0	0
59	5/64	5/64	5/64	1/60	0	0	0		123	5/128	5/128	5/128	5/128	1/124	0	0
61	4/04	4/04	4/04 3/6/	3/6/	0	0	0		124	4/128	4/128	4/128	4/128	3/129	0	0
62	2/64	2/64	2/64	2/64	0	0	0		125	2/128	2/128	2/128	2/128	2/128	0	0
63	1/64	1/64	1/64	1/64	1/64	0	0		127	1/128	1/128	1/128	1/128	1/128	1/128	0
64	0	0	0	0	0	0	0		128	0	0	0	0	0	0	0

# Table A.1 Relationship between code wastage capacity and rakes for various user rates in WCDMA

![](_page_11_Picture_2.jpeg)

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 PhD 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.